

69039583

LIBRARIES
MICHIGAN STATE UNIVERSITY
EAST LANSING, MICH 48824-1048

This is to certify that the
dissertation entitled

HIGH ORDER NUMERICAL METHODS FOR INVISCID
AND VISCOUS FLOWS ON UNSTRUCTURED GRIDS

presented by

YUZHUN SUN

has been accepted towards fulfillment
of the requirements for the

Ph.D degree in Mechanical Engineering



Major Professor's Signature

12/17/2004.

Date

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

<u>DATE DUE</u>	<u>DATE DUE</u>	<u>DATE DUE</u>

**HIGH ORDER NUMERICAL METHODS FOR INVISCID
AND VISCOUS FLOWS ON UNSTRUCTURED GRIDS**

By

Yuzhi Sun

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering

2004

ABSTRACT

HIGH ORDER NUMERICAL METHODS FOR INVISCID AND VISCOUS FLOWS ON UNSTRUCTURED GRIDS

By

Yuzhi Sun

The spectral volume (SV) method is a newly developed high-order, conservative, and efficient finite volume method for hyperbolic conservation laws on unstructured grids. It has been successfully demonstrated for scalar conservation laws and multi-dimensional Euler equations. In this study, the SV method is compared with another high-order method for hyperbolic conservation laws capable of handling unstructured grids named the discontinuous Galerkin (DG) method. Their overall performance in terms of the efficiency, accuracy and memory requirement is evaluated using the scalar conservation laws and the two-dimensional Euler equations. To measure their accuracy, problems with analytical solutions are used. Both methods are also used to solve problems with strong discontinuities to test their ability in discontinuity capturing. Both the DG and SV methods are capable of achieving the formal order of accuracy while the DG method has a lower error magnitude and takes more memory. They are also similar in efficiency. The SV method appears to have a higher resolution for discontinuities because the data limiting can be done at the sub-element level.

The SV method is also successfully extend to the Navier-Stokes equations. First, the SV method is extended to and tested for the diffusion equation. In this study, three

different formulations named Naïve SV, Local SV and Penalty SV for the diffusion equation are presented. The Naïve SV formulation yields an inconsistent and unstable scheme, while the other two formulations are consistent, convergent and stable. A Fourier type analysis is performed for all the formulations, and the analysis agrees well with the numerical results. Second, the Local SV method is chosen to be extended to solve the Navier-Stokes equations since it gives the optimum accuracy in solving the diffusion equation. The formulation of the Local SV method for the two-dimensional compressible Navier-Stokes equations is described. Accuracy studies are performed on the scalar convection-diffusion and the Navier-Stokes equations using problems with analytical solutions. It is shown that the designed order of accuracy is achieved for 1st, 2nd and 3rd order reconstructions. The solver is then used to solve other viscous laminar flow problems to demonstrate its capability.

Copyright by

YUZHONG SUN

2004

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my advisor, Professor Z.J. Wang, for his invaluable guidance throughout this study. He has taught me many things and this work would not have been possible without his patience and encouragement.

I would also like to thank Professors Chichia Chiu, Farhad Jaber, Tom Shih, and Mei Zhuang for serving on my committee and for reviewing the work and the dissertation.

I would like to give special thanks to my husband Dongsheng Wu, and my daughter Emily Wu, who have constantly supported and encouraged me throughout the study.

TABLE OF CONTENTS

LIST OF TABLES.....	viii
LIST OF FIGURES.....	xi
NOMENCLATURE.....	xiii
CHAPTER 1	
INTRODUCTION.....	1
1.1 Background of Computational Fluid Dynamics (CFD).....	1
1.2 Numerical Methods in CFD.....	2
1.3 Motivation and Objectives of This Study.....	5
1.4 Outline of the Dissertation.....	6
CHAPTER 2	
FRAMEWORK OF DG AND SV METHODS.....	8
2.1 DG Method.....	8
2.1.1 Space Discretization.....	9
2.1.2 Grids and Data Reconstructions.....	11
2.1.3 Time Integration.....	13
2.1.4 Monotonicity Limiter.....	13
2.2 SV Method.....	15
CHAPTER 3	
EVALUATION OF DG AND SV METHODS.....	19
3.1 Number of Operations and Memory Requirement.....	19
3.1.1 DG Method.....	20
3.1.1.1 Numbers of operations.....	20
3.1.1.2 Memory requirement.....	22
3.1.2 SV Method.....	23
3.1.2.1 Numbers of operations.....	23
3.1.2.2 Memory requirement.....	25
3.2 Accuracy and CPU Times.....	25
3.2.1 Scalar Conservation Laws.....	25
3.2.1.1 2D linear wave equation.....	25
3.2.1.2 2D Burger's equation.....	30
3.2.2 Euler Equations.....	33
3.2.2.1 Vortex propagation problem.....	33
3.2.2.2 Double Mach reflection problem.....	38
3.3 Conclusions.....	42

CHAPTER 4	
SV METHOD FOR THE DIFFUSION EQUATION.....	43
4.1 Three Formulations.....	43
4.1.1 Naïve SV Formulation.....	43
4.1.2 Local SV Formulation.....	46
4.1.3 Penalty SV Formulation.....	48
4.2 Fourier Analysis.....	49
4.2.1 Naïve SV Formulation.....	53
4.2.2 Local SV Formulation.....	56
4.2.3 Penalty SV Formulation.....	59
4.3 Conclusions.....	61
CHAPTER 5	
EXTENSION OF SV METHOD TO CONVECTION-DIFFUSION EQUATIONS....	63
5.1 Formulation for 1D Convection-Diffusion Equation.....	63
5.2 Formulation for 2D Linear Convection-Diffusion Equation.....	66
5.3 Accuracy Study.....	68
5.3.1 1D linear Convection-Diffusion Equation.....	68
5.3.2 1D Viscous Burger's Equation.....	69
5.3.3 1D Fully Nonlinear Case.....	69
5.3.4 2D Linear Convection-Diffusion Equation.....	71
5.4 Conclusions.....	71
CHAPTER 6	
EXTENSION OF SV METHOD TO THE NAVIER-STOKES EQUATIONS.....	72
6.1 Formulation for Navier-Stokes Equations.....	72
6.2 Numerical Experiments.....	76
6.2.1 The Couette Flow.....	76
6.2.2 Laminar Flow along a Flat Plate.....	81
6.2.3 Subsonic Flow over a Circular Cylinder.....	91
6.2.4 Laminar Subsonic Flow around the NACA0012 Airfoil.....	98
6.3 Conclusions.....	106
CHAPTER 7	
SUMMARY AND FUTURE WORK.....	108
APPENDIX A	
GAUSSIAN QUADRATURE FORMULAS.....	112
APPENDIX B	
PUBLICATIONS ASSOCIATED WITH THIS WORK.....	116
BIBLIOGRPHY.....	117
VITA.....	123

LIST OF TABLES

CHAPTER 3

Table 3.1	Number of operations for the DG method.....	22
Table 3.2	Number of operations for the SV method.....	24
Table 3.3	Errors and CPU times at $t = 1$ for a 2D linear wave equation using the DG method on the regular mesh.....	28
Table 3.4	Errors and CPU times at $t = 1$ for a 2D linear wave equation using the SV method on the regular mesh.....	28
Table 3.5	Errors and CPU times at $t = 1$ for a 2D linear equation using the DG method on the irregular mesh.....	29
Table 3.6	Errors and CPU times at $t = 1$ for a 2D linear wave equation using the SV method on the irregular mesh.....	29
Table 3.7	Errors and CPU time on the 2D Burger's equation at $t = 0.1$ using DG on the irregular mesh.....	31
Table 3.8	Errors and CPU time on the 2D Burger's equation at $t = 0.1$ using SV on the irregular mesh.....	31
Table 3.9	Errors and CPU time on the 2D Burger's equation at $t = 0.45$ using DG on the irregular mesh.....	32
Table 3.10	Errors and CPU time on the 2D Burger's equation at $t = 0.45$ using SV on the irregular mesh.....	32
Table 3.11	Errors and CPU times for the propagating vortex case at $t = 10$ using the DG method on the regular mesh.....	36
Table 3.12	Errors and CPU time for the propagating vortex case at $t = 10$ using the SV method on the regular mesh.....	36
Table 3.13	Errors and CPU times for propagating vortex case at $t = 10$ using the DG method on the irregular mesh.....	37
Table 3.14	Errors and CPU times for the propagating vortex case at $t = 10$ using the SV method on the irregular mesh.....	37

CHAPTER 4

Table 4.1	L_1 and L_∞ errors and orders of accuracy based on the local SV formulation for the diffusion equation.....	47
Table 4.2	L_1 and L_∞ errors and orders of accuracy based on the penalty SV formulation for the diffusion equation.....	49

CHAPTER 5

Table 5.1	L_1 and L_∞ errors and orders of accuracy on 1D linear convection-diffusion equation (at $t = 1.0$).....	68
Table 5.2	L_1 and L_∞ errors and orders of accuracy on 1D viscous Burger's equation (at $t = 1.0$).....	70
Table 5.3	L_1 and L_∞ errors and orders of accuracy on the 1D fully nonlinear convection-diffusion equation ($t = 1.0$).....	70
Table 5.4	L_1 and L_∞ errors and orders of accuracy on the 2D linear convection-diffusion equation ($t = 1.0$).....	71

CHAPTER 6

Table 6.1	L_1 and L_∞ errors and orders of accuracy on Couette flow (Density in Estimation 1).....	79
Table 6.2	L_1 and L_∞ errors and orders of accuracy on Couette flow (Temperature in Estimation 1).....	79
Table 6.3	L_1 and L_∞ errors and orders of accuracy on Couette flow (Density in Estimation 2).....	80
Table 6.4	L_1 and L_∞ errors and orders of accuracy on Couette flow (Temperature in Estimation 2).....	80
Table 6.5	Pressure part ($C_{d,p}$) and viscous Part ($C_{d,v}$) of the drag coefficient for the NACA0012 Airfoil.....	105
Table 6.6	Comparison of pressure part ($C_{d,p}$) and viscous part ($C_{d,v}$) of the drag coefficient for the NACA0012 airfoil computed by the present method with various structured and unstructured solvers.....	106

APENDIX

Table A.1	Christoffel weights W_i and roots $\xi_i, i = 1, 2, \dots, n$, for Legendre polynomials of degree l to 6.....	113
Table A.2	Weights and evaluation points for integration on triangles.....	115

LIST OF FIGURES

CHAPTER 2

Figure 2.1	Degrees of freedom for DG.....	12
Figure 2.2	Spectral volumes of various degrees.....	15
Figure 2.3	SV partitions used in numerical simulations.....	18

CHAPTER 3

Figure 3.1	Regular and irregular grids.....	27
Figure 3.2	Computational domain and boundaries for double Mach reflection problem.....	38
Figure 3.3	Density contours computed with second order DG scheme using a TVD limiter.....	40
Figure 3.4	Density contours computed with second order SV scheme using a TVD limiter.....	41

CHAPTER 4

Figure 4.1	1D linear spectral volumes.....	43
Figure 4.2	The numerical solutions versus the exact solution using the <i>linear reconstruction</i> based on the Naïve SV formulation for the diffusion equation.....	45
Figure 4.3	The numerical solutions versus the exact solution using the <i>quadratic reconstruction</i> based on the Naïve SV formulation for the diffusion equation.....	45

CHAPTER 5

Figure 5.1	One-dimensional spectral volumes.....	64
------------	---------------------------------------	----

CHAPTER 6

Figure 6.1	Computational grids for Couette flow case.....	77
------------	--	----

Figure 6.2	Convergence trend of numerical solution to the steady exact solution in Couette flow test.....	78
Figure 6.3	The computational domain for the flat plate case.....	81
Figure 6.4	Meshes for the flat plate case.....	83
Figure 6.5	u-velocity profiles for the flat plate case.....	84
Figure 6.6	Skin friction coefficients along a flat plate	87
Figure 6.7	Residual history of continuity equation with cubic reconstruction.....	90
Figure 6.8	Grid for the case of subsonic flow over a circular cylinder.....	92
Figure 6.9	Instantaneous Mach contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$	93
Figure 6.10	Instantaneous entropy contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$	94
Figure 6.11	Instantaneous vorticity contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$	94
Figure 6.12	Pressure histories for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$	95
Figure 6.13	Computational grid for the NACA0012 test case.....	98
Figure 6.14	Mach isolines around the NACA0012 airfoil ($Re = 5000, Ma = 0.5, \alpha = 0^\circ$)	100
Figure 6.15	Pressure coefficient distribution along the NACA0012 airfoil ($Re = 5000, Ma = 0.5, \alpha = 0^\circ$)	103
Figure 6.16	Skin friction coefficient distribution along the NACA0012 airfoil ($Re = 5000, Ma = 0.5, \alpha = 0^\circ$)	104
Figure 6.17	Residual history of continuity equation with cubic reconstruction for NACA0012 test case.....	105

NOMENCLATURE

Q	=	flow variable, or conservative flow vector
E	=	total energy
F	=	flux vector
f	=	flux in x direction
g	=	flux in y direction
ρ	=	density
u	=	velocity component in x direction
v	=	velocity component in y direction
p	=	pressure
T	=	absolute temperature
μ	=	dynamic viscosity coefficient
γ	=	ratio of specific heat
C_p	=	specific heat at constant pressure,
P_r	=	Prandtl number
Ω	=	a space domain
$\partial\Omega$	=	boundary of domain Ω
φ	=	a test function, or a scalar limiter
k	=	order of interpolation polynomial
n	=	dimension of the interpolation polynomial space in 2D
n	=	number of control volumes in a spectral volume
T	=	upper bound of time
t	=	time
Δt	=	time step
$\mathbf{r}$	=	position vector
∇	=	gradient operator
$C_{i,j}$	=	the j -th control volume in the i -th spectral volume
$V_{i,j}$	=	volume of the control volume $C_{i,j}$
K	=	number of faces in $C_{i,j}$
A_r	=	the r -th face in $C_{i,j}$
L_j	=	shape function in a spectral volume
λ_i	=	triangular coordinate
$\ \Gamma_\Pi\ $	=	Lebesgue constant
$\mathbf{n}$	=	the outward unit normal vector

CHAPTER 1

INTRODUCTION

1.1 Background of Computational Fluid Dynamics (CFD)

The physical aspects of transport phenomena in the macro-scale are governed by the Newton's laws of motion and the fundamental principles of mass, energy and species conservation. The final objective of most engineering investigations is to obtain a quantitative description of the physical problem by analytical, experimental or numerical methods.

By the turn of the twentieth century, the development of closed form analytical solutions for flow field problems had reached a highly mature stage and it was being realized that a large class of problems still remained which were not amenable to exact analytical solution methods. Experimental fluid dynamics has played an important role in validating and delineating the limits of the various approximations to the governing equations. The wind tunnel, as a piece of experiment equipment, provides an effective means of simulating real flows. Traditionally this has provided a cost-effective alternative to full-scale measurement. In the design of equipment that depends critically on the flow behavior, e.g. aircraft design, full-scale measurement, as part of the design process is economically unavailable. The steady improvement in the speed of computer and memory size since 1950s has led to the emergence of computational fluid dynamics (CFD) to study the characteristics of fluid dynamics using digital computers. CFD is significantly cheaper than wind tunnel testing and will become even more so in the future.

1.2 Numerical Methods in CFD

The success of CFD is really dependent on two factors, i.e. improvement on computer hardware and highly efficient computational algorithms. Therefore, there have been intensive efforts to develop highly efficient and accurate numerical algorithms to seek higher quality numerical solutions with less CPU time.

The basic issue of quality of numerical solutions in CFD simulation is fundamentally important: how accurate are the numerical simulations and how does one obtain the most accurate results given a fixed computational resource? These questions lie at the core of modern numerical methods that aim to control the error in the computed solution and to optimize the computational process.

Many methodologies have been developed to address this issue for the hyperbolic systems in the last three decades. One of the most successful algorithms is the Godunov method [24], which laid a solid foundation for the development of modern upwind methods [20,25-26,46,60-61]. For example, van Leer [60-61] extended the first-order Godunov method to second-order by using a piece-wise linear data reconstruction and a limiter to remove spurious numerical oscillations near steep gradients. In addition, for better efficiency the exact Riemann solver used in the Godunov method was replaced by approximate Riemann solvers or flux-splitting procedures, such as the flux-vector splitting [59] by Steger and Warming, the flux-difference splitting [52] by Roe, the smoother flux vector splitting [62] by Van Leer, the differentiable approximate Riemann solver [45] by Osher, and AUSM [40] by Liou, FUSS [63] by Wang, among many others.

One of the most popular schemes for obtaining solutions on unstructured meshes is the discontinuous Galerkin finite element (DG) method, which was introduced in the early 1970's for the numerical solution of first-order hyperbolic problems (see [6,13,15-18,23, 35,36,39,48-50]). Simultaneously, but independently, it was proposed as non-standard schemes for the numerical approximation of second-order elliptic equations [2, 68]. In recent years there has been renewed interest in the discontinuous Galerkin method due to its favorable properties, such as a high degree of locality, stability in the absence of streamline-diffusion stabilization for convection-dominated diffusion problem [29], and the flexibility of locally varying the polynomial degree in hp-version approximations, since no point wise continuity requirements are imposed at the element interfaces. Much attention has been paid to the analysis of the DG method applied to non-linear hyperbolic equations and hyperbolic systems [11,12,27], several other types of non-linear equations (including the Hamilton-Jacobi equation [30], and non-linear Schrödinger equation [37], and other non-linear problem [14]). Also, it was extended to the compressible Navier-Stokes equations [7].

An alternative to the finite element method is the finite-volume method, in which the governing equations are solved in integral form over the discrete volumes formed by the cells of a mesh. Description of various finite-volume methods on unstructured meshes are given by Barth and Jespersen [5], Whitaker, et al. [69], Jameson, et al. [32-34], and Maripolis and Jameson [42]. Barth [3] presents a detailed account of the implementation of finite volume schemes for the Euler and Navier-Stokes equations using efficient edge-

based data structures. Finite volume schemes generally solve for quantities averaged over cells of the actual mesh in the case of cell-centered schemes or over cells of a dual mesh in the case of vertex schemes. In any event, in order to evaluate the residual, a polynomial data distribution must be reconstructed from these averaged quantities. To achieve higher than second order accuracy, a higher order distribution must be constructed in each cell, requiring information from more distant neighbors. This was done by Barth and Frederickson [4] for quadratic reconstruction (and hence third order accuracy). Hu and Shu [31] further devised a fourth order scheme without expanding the third order stencil.

More recently, a high-order, conservative, yet efficient method named the spectral volume (SV) method was presented by Wang [64] for hyperbolic conservation law. The SV method is a finite volume method, in which the concept of a “spectral volume” is introduced to achieve high-order accuracy in an efficient manner similar to spectral element and multidomain spectral methods. Each spectral volume is further subdivided into control volumes, and cell-averaged data from these control volumes are used to reconstruct a high-order approximation in the spectral volume. Then Riemann solvers are used to compute the fluxes at spectral volume boundaries. Cell-average state variables in the control volumes are updated independently. Furthermore, total variation diminishing and total variation bounded limiters are introduced in the SV method to remove/reduce spurious oscillations near discontinuities. Unlike spectral element and multidomain spectral methods, the SV method can be applied to fully unstructured grids. A very desirable feature of the SV method is that the reconstruction is carried out analytically,

and the reconstruction stencil is always nonsingular, in contrast to the memory and CPU-intensive reconstruction in a high-order k -exact finite volume method.

1.3 Motivation and Objectives of This Study

The newly developed spectral (finite) volume method has been successfully demonstrated for hyperbolic conservation laws including non-linear systems on unstructured grids in a series of papers [65-67]. A framework has been established to easily solve non-linear time-dependent hyperbolic systems of conservation laws using explicit, non-linear Runge-Kutta time discretization [58] with approximate Riemann solvers and TVB (total variation bounded) non-linear limiters [54]. One objective of this study is to give a further numerical demonstration that the SV method is comparable to other high order methods and also possesses some unique properties. To do so, we evaluate the DG and SV methods on hyperbolic conservation laws, since the DG and SV methods seem to be the most efficient among the high-order methods on unstructured grids.

Ultimately, we wish to extend the SV method to the Navier-Stokes equations to perform large eddy simulation and direct numerical simulation of turbulence flow for problems with complex geometries. So, another objective of this study is to extend the SV method [64-67] to the Navier-Stokes equations.

A key in the extension is to properly discretize the second order viscous terms. In a second-order finite volume method, the solution gradients at an interface are computed

by averaging the gradients of the neighboring cells sharing the face, and were shown to be adequate. For higher-order elements, special care has to be taken in computing the solution gradients. For example, Cockburn and Shu developed the so-called local discontinuous Galerkin method to treat the second order viscous terms and proved stability and convergence with error estimates [19] motivated by the successful numerical experiments of Bassi and Rebay [7]. Baumann and Oden [8], Oden, Babuska and Baumann [44] introduced a different discontinuous Galerkin method for the discretization of the second order viscous terms. Riviere, Wheeler and Girault [51] analyzed three discontinuous Galerkin approximations for solving elliptic problems in two or three dimensions. More recently, Shu [57] summarized three different formulations of the discontinuous Galerkin method for the diffusion equation, and Zhang and Shu [72] performed a Fourier type analysis for these three formulations.

Motivated by the DG approach in handling the viscous term, three SV formulations for pure diffusion equations will be presented in this research, and one of them will be successfully applied to 1D and 2D scalar convection-diffusion equations, eventually to viscous flows governed by the Navier-Stokes equations. The spatial convergence rate of the SV method will be established for some scalar cases and Couette flow, and the designed order of accuracy will be studied.

1.4 Outline of the Dissertation

The dissertation is arranged as follows. We first review the framework of the discontinuous Galerkin (DG) spectral volume (SV) methods in Chapter 2. Then in

Chapter 3, we evaluate the DG and SV methods in terms of the number of operations, memory requirement, accuracy and CPU times for inviscid flows. In Chapter 4, we present three SV formulations for the diffusion equation. The extension of the SV method to the viscous flow is presented in Chapter 5 and Chapter 6. Finally, a summary of the present study and recommendations for further investigations are given in Chapter 7.

CHAPTER 2

FRAMEWORK OF DG AND SV METHODS

The DG method is a finite element method using discontinuous solution and test spaces (usually piecewise polynomials of suitable degree), which means that the state variables are not continuous across element boundaries. The fluxes through the element boundaries are then computed using an approximate Riemann solver, mimicking the successful Godunov finite volume method [24]. Due to the use of Riemann fluxes across element boundaries, the DG method is fully conservative at the element level. The SV method [64-67] is a finite volume method. For a given unstructured grid, each element (called a spectral volume) is further partitioned into structured sub elements named control volumes (CVs). Mean state-variables at the CVs inside a SV are employed to construct a high-order polynomial within the element or SV, which is then utilized to update the means at the CVs. The reconstruction problem can be solved analytically, and is identical for all simplexes. Therefore a high-order SV method is much more efficient than a high-order k-exact FV method, in which a reconstruction problem must be solved for each control volume. The SV method is fully conservative at the sub-cell control volume level. Both methods are reviewed next.

2.1 DG Method

Consider the following two-dimensional conservation laws

$$Q_t + \nabla \cdot F = 0, \quad \Omega \times (0, T) \quad (2.1)$$

equipped with proper initial and boundary conditions. In Eq. (2.1), $F = (f, g)$ is the flux vector. Multiplying Eq. (2.1) by a test function φ , integrating over the computational domain Ω , and performing integration by parts, we obtain the following weak statement of the problem

$$\int_{\Omega} \varphi Q_t dV + \oint_{\partial\Omega} \varphi F(Q) \cdot \mathbf{n} dS - \int_{\Omega} \nabla \varphi \cdot F(Q) dV = 0, \forall \varphi \quad (2.2)$$

Note that the integral in Eq. (2.2) is understood to be performed in a component-wise manner if Q is a column vector.

2.1.1 Space Discretization

Assume that the computational domain Ω is subdivided into N non-overlapping triangular elements $\{T_i\}$. By applying Eq. (2.2) to each element T_i , we can obtain the discrete analogue of Eq. (2.2) on the computational grid. Let the solution and test function be piece-wise polynomials in each element. Denote the polynomial basis as $\xi(\mathbf{r}) = \{\xi_1(\mathbf{r}), \dots, \xi_n(\mathbf{r})\}^T$. If the polynomial is of order k , the dimension of the polynomial space in 2D is $n = (k+1)(k+2)/2$. The solution and the test function on element T_i can be expressed as

$$Q_i(\mathbf{r}, t) = \sum_{j=1}^n Q_i^j(t) \xi_j(\mathbf{r}), \quad \varphi_h = \sum_{j=1}^n \varphi_h^j \xi_j(\mathbf{r}). \quad (2.3)$$

The expansion coefficients Q_i^j denote the degrees of freedom (DOFs) of the numerical solution on element T_i . Note that there is no global continuity requirement for Q_i , which is generally discontinuous across the element boundaries. Using the solution and test function, Eq. (2.2) on element T_i becomes

$$\frac{d}{dt} \int_{T_i} \varphi_h Q_i dV + \oint_{\partial T_i} \varphi_h \mathbf{F} \bullet \mathbf{n} dS - \int_{T_i} \nabla \varphi_h \bullet \mathbf{F} dV = 0. \quad (2.4)$$

Equation (2.4) must be satisfied for any test function φ_h . Since ξ_j is the basis function for φ_h , Eq. (2.4) is equivalent to the following system of n equations

$$\frac{d}{dt} \int_{T_i} \xi_j Q_i dV + \oint_{\partial T_i} \xi_j \mathbf{F} \bullet \mathbf{n} dS - \int_{T_i} \nabla \xi_j \bullet \mathbf{F} dV = 0, \quad 1 \leq j \leq n. \quad (2.5)$$

Because the approximate solution is discontinuous at the element boundaries, the interface flux is not uniquely defined. It is at this stage the Riemann flux used in the Godunov finite volume method [24] is borrowed. The interface flux function $\mathbf{F} \bullet \mathbf{n}$ is replaced by a Riemann flux $\hat{F}(Q^L, Q^R, \mathbf{n})$, where Q^L and Q^R are the state variables at the left and right side of the interface. In order to guarantee consistency and conservation, the Riemann flux must satisfy

$$\hat{F}(Q, Q, \mathbf{n}) = \mathbf{F}(Q) \bullet \mathbf{n}, \quad \hat{F}(Q^L, Q^R, \mathbf{n}) = -\hat{F}(Q^R, Q^L, -\mathbf{n}). \quad (2.6)$$

The surface and volume integrals in Eq. (2.5) can be computed with Gauss quadrature formulas of suitable orders of accuracy, which are given in Appendix A. Following the arguments given in [13], the surface integral must be exact for polynomials of degree $2k$, while the volume integral must be exact for polynomials of degree $2k-1$, i.e.,

$$\begin{aligned} \oint_{\partial T_i} \xi_j \mathbf{F} \bullet \mathbf{n} dS &= \sum_{r=1}^K \int_{A_r} \xi_j \mathbf{F} \bullet \mathbf{n} dS. \\ \int_{A_r} \xi_j \mathbf{F} \bullet \mathbf{n} dS &\approx \sum_{s=1}^{ns} w_{rs} \xi_j(\mathbf{r}_{rs}) \hat{F}(Q^L(\mathbf{r}_{rs}), Q^R(\mathbf{r}_{rs}), \mathbf{n}_r) A_r, \end{aligned}$$

$$\int_{T_i} \nabla \xi_j \cdot \mathbf{F} dV \approx \sum_{s=1}^{nv} w_s \nabla \xi_j(\mathbf{r}_s) \cdot \mathbf{F}(Q_i(\mathbf{r}_s)) V_i. \quad (2.7)$$

where K is the number of planar faces of T_i , ns is the number of quadrature points on a planar face for the surface integral, nv is the number of quadrature points in the element for the volume integral, w_{rs} and w_s are the Gauss quadrature weights, $\mathbf{r}_{rs}$ and $\mathbf{r}_s$ are the Gauss quadrature points. Let $U^i = \{Q_i^1, \dots, Q_i^n\}^T$ be the DOFs for element T_i , and W^i

denote the mass matrix $\left\{ \int_{T_i} \xi_j \xi_l dV \right\}$. Equation (2.5) can be further written as

$$\frac{dU^i}{dt} + (W^i)^{-1} \left(\oint_{\partial T_i} \xi \mathbf{F} \cdot \mathbf{n} dS - \int_{T_i} \nabla \xi \cdot \mathbf{F} dV \right) = 0. \quad (2.8)$$

By assembling together all the elemental contributions, a system of ordinary differential equations that govern the evolution of the discrete solution can be written as

$$\frac{dU}{dt} = R(U), \quad (2.9)$$

where U is the global vector of DOFs, and $R(U)$ is the global residual vector with the element vector being

$$R^i(U) = -(W^i)^{-1} \left(\oint_{\partial T_i} \xi \mathbf{F} \cdot \mathbf{n} dS - \int_{T_i} \nabla \xi \cdot \mathbf{F} dV \right). \quad (2.10)$$

2.1.2 Grids and Data Reconstructions

The degrees of freedom are chosen as the values at certain points in each element, which are shown in Figure 2.1.

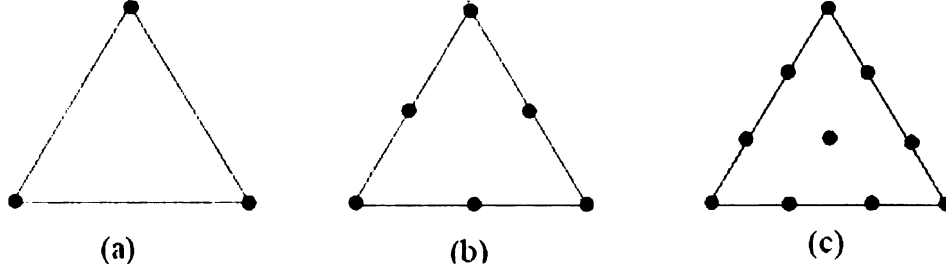


Figure 2.1 Degrees of freedom for DG,
(a) linear element, (b) quadratic element, (c) cubic element

The basis functions in terms of the triangular coordinates are given next.

Linear element:

$$\begin{aligned}\xi_1 &= \lambda_1, \\ \xi_2 &= \lambda_2, \\ \xi_3 &= 1 - \lambda_1 - \lambda_2.\end{aligned}$$

Quadratic element:

$$\begin{aligned}\xi_1 &= \lambda_1 \cdot (2\lambda_1 - 1), \\ \xi_2 &= \lambda_2 \cdot (2\lambda_2 - 1), \\ \xi_3 &= \lambda_3 \cdot (2\lambda_3 - 1), \\ \xi_4 &= 4\lambda_1 \cdot \lambda_2, \\ \xi_5 &= 4\lambda_2 \cdot \lambda_3, \\ \xi_6 &= 4\lambda_3 \cdot \lambda_1.\end{aligned}$$

Cubic element:

$$\begin{aligned}\xi_1 &= \lambda_1 \cdot (3\lambda_1 - 1) \cdot (3\lambda_1 - 2) / 2, \\ \xi_2 &= \lambda_2 \cdot (3\lambda_2 - 1) \cdot (3\lambda_2 - 2) / 2, \\ \xi_3 &= \lambda_3 \cdot (3\lambda_3 - 1) \cdot (3\lambda_3 - 2) / 2, \\ \xi_4 &= \lambda_1 \cdot \lambda_2 \cdot (3\lambda_1 - 1) \cdot 9 / 2, \\ \xi_5 &= \lambda_2 \cdot \lambda_3 \cdot (3\lambda_2 - 1) \cdot 9 / 2, \\ \xi_6 &= \lambda_3 \cdot \lambda_1 \cdot (3\lambda_3 - 1) \cdot 9 / 2,\end{aligned}$$

$$\xi_7 = \lambda_1 \cdot \lambda_2 \cdot (3\lambda_2 - 1) \cdot 9/2,$$

$$\xi_8 = \lambda_2 \cdot \lambda_3 \cdot (3\lambda_3 - 1) \cdot 9/2,$$

$$\xi_9 = \lambda_3 \cdot \lambda_1 \cdot (3\lambda_1 - 1) \cdot 9/2,$$

$$\xi_{10} = 27\lambda_1 \cdot \lambda_2 \cdot \lambda_3,$$

where $\lambda_j, j = 1, 2, 3$ are the triangular coordinates described in Appendix A,

$$\lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_j \geq 0.$$

2.1.3 Time Integration

An explicit multi-stage third-order TVD (total variation diminishing) Runge-Kutta scheme is employed for time integration [55]. The Runge-Kutta scheme can be expressed in the following form:

$$\begin{aligned} U^{(1)} &= U^n + \Delta t R(U^n); \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}[U^{(1)} + \Delta t R(U^{(1)})]; \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}[U^{(2)} + \Delta t R(U^{(2)})]. \end{aligned} \tag{2.11}$$

Where, U^n is the global vector of DOFs at time $t = t_n$, and U^{n+1} is the global vector of DOFs at time $t = t_{n+1} = t_n + \Delta t$.

2.1.4 Monotonicity Limiter

For the non-linear Euler equations, it is necessary to perform data limiting to maintain stability if the solution contains discontinuities. There are two possible ways of applying limiters in the system setting. One way is to apply a limiter to each characteristic variable.

The other is to apply a limiter to each of the conservative variables. In one dimension, the former has the nice property of naturally degenerating to the scalar case if the hyperbolic system is linear. In multiple dimensions, characteristic variables are defined in a particular direction, e.g. in the face normal direction. In a fully unstructured grid, there is no coordinate direction to define a characteristic variable. Therefore it is difficult to design characteristics-based limiters in multiple dimensions. In this research, we choose the component-wise approach for the limiter, which should also be much more efficient than the characteristic approach. To this end, we first establish the following numerical monotonicity criterion for each element

$$\bar{Q}_i^{\min} \leq Q_i(\mathbf{r}_s) \leq \bar{Q}_i^{\max}, \quad (2.12)$$

where $\bar{Q}_i^{\min}$ and $\bar{Q}_i^{\max}$ are the minimum and maximum cell-averaged solutions among all its neighboring elements sharing a face with T_i , and $Q_i(\mathbf{r}_s)$ is the solution at any of the quadrature points. If the inequality (2.12) is violated for any quadrature point, then it is assumed that the element is close to a discontinuity, and the solution in the element is forced locally linear, i.e.,

$$Q_i(\mathbf{r}) = \bar{Q}_i + \nabla Q_i \bullet (\mathbf{r} - \mathbf{r}_i), \quad \forall \mathbf{r} \in T_i, \quad (2.13)$$

where $\mathbf{r}_i$ is the position vector of the centroid of T_i . The magnitude of the solution gradient is maximized subject to the monotonicity condition given in the inequality (2.12). The original polynomial is used to compute an initial guess of the gradient at the element centroid, i.e.,

$$\nabla Q_i = \left(\frac{\partial Q_i}{\partial x}, \frac{\partial Q_i}{\partial y} \right) \Big|_{\mathbf{r}_i}.$$

This gradient may not satisfy the inequality (2.12). Therefore it is limited by multiplying a scalar limiter $\varphi \in [0, 1]$ so that the following solution satisfies the inequality (2.12)

$$Q_i(r) = \bar{Q}_i + \varphi \nabla Q_i \bullet (r - r_i). \quad (2.14)$$

The scalar limiter can be obtained by examining the numerical solutions at all the quadrature points [65].

2.2 SV Method

In the SV method, the element T_i is named a *spectral volume*, which is further partitioned into subcells named *control volumes* (CVs), indicated by $C_{i,j}$, as shown in Figure 2.2.

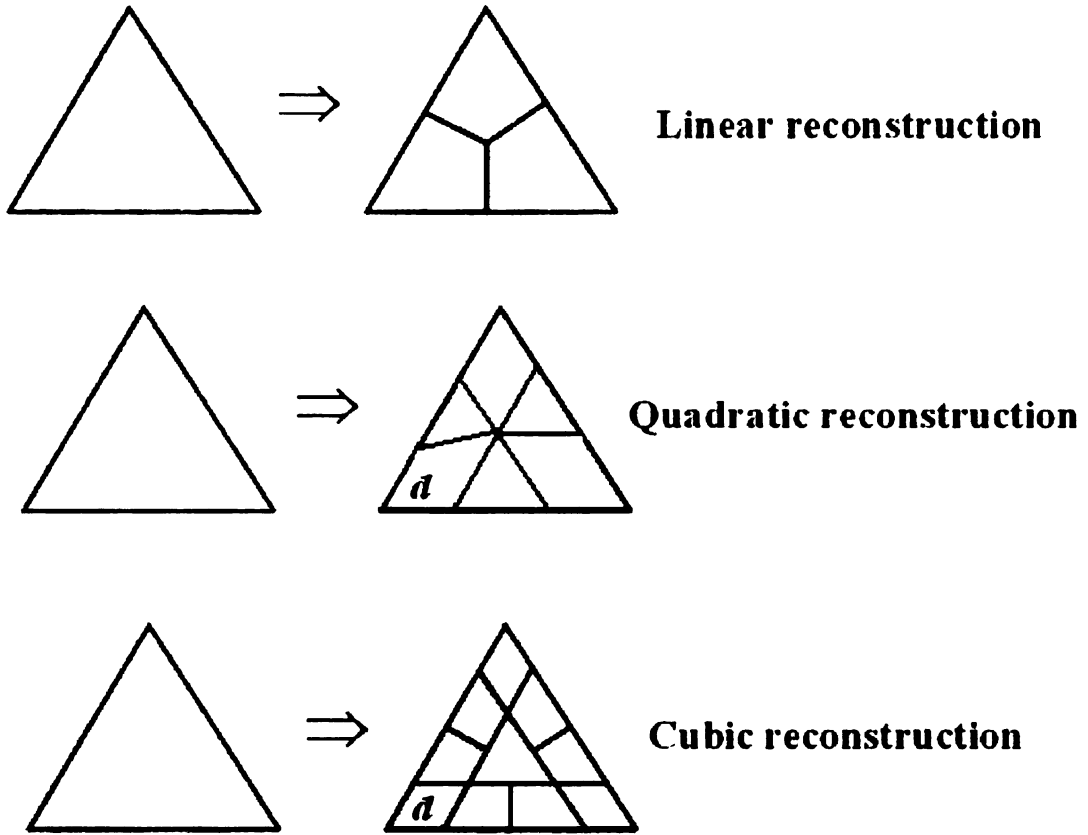


Figure 2.2 Spectral volumes of various degrees

To approximate the solution as a polynomial of degree k in two dimensions (2D), we need to partition the SV into $n = (k+1)(k+2)/2$ sub-cells. The degrees of freedom (DOFs) in a SV are the volume-averaged mean variables $\bar{Q}_{i,j}$ at the n CVs. There are numerous ways of partitioning a SV, and not every partition is admissible in the sense that the partition may not be capable of producing a degree k polynomial. Once n mean solutions in the CVs of an admissible SV are given, a unique polynomial reconstruction can be obtained from

$$p_i(\mathbf{r}) = \sum_{j=1}^n L_j(\mathbf{r}) \bar{Q}_{i,j}, \quad (2.15)$$

where $L_j(\mathbf{r})$ are also degree k polynomials satisfying

$$\int_{C_{i,j}} L_m(\mathbf{r}) dV = V_{i,j} \delta_{jm}, \quad (2.16)$$

and $V_{i,j}$ is the volume of $C_{i,j}$. This high-order polynomial reconstruction facilitates a high-order update for the mean solution of each CV. Integrating Eq. (2.1) in each CV, we obtain

$$\frac{d\bar{Q}_{i,j}}{dt} V_{i,j} + \sum_{r=1}^K \int_{A_r} (\mathbf{F} \cdot \mathbf{n}) dS = 0, \quad (2.17)$$

where K is the total number of faces in $C_{i,j}$. The flux integral in Eq. (2.17) is then replaced by a Gauss-quadrature formula (see Appendix A) that is exact for polynomials of degree k

$$\int_{A_r} (\mathbf{F} \cdot \mathbf{n}) dS \approx \sum_{s=1}^{ne} w_{rs} \mathbf{F}(Q(\mathbf{r}_{rs})) \cdot \mathbf{n}_r A_r, \quad (2.18)$$

where ne is the number of quadrature points on the r -th face, w_{rs} are the Gauss quadrature weights, $\mathbf{r}_{rs}$ are the Gauss quadrature points. Since the reconstructed polynomials are piece-wise continuous, the solution is discontinuous across the boundaries of a SV, although it is continuous across interior CV faces. The fluxes at the interior faces can be computed directly based on the reconstructed solutions at the quadrature points. The fluxes at the boundary faces of a SV are again computed using approximate Riemann solvers given the left and right reconstructed solutions. The Runge-Kutta scheme is again used for time integration.

The TVD limiter in the SV method [65] is very similar to the one described in the last section. The main difference is that the limiter is applied for the sub-cell averaged state variables, rather than for the averaged state variables of macro element, i.e., the SV. This is possible because of the inherent local resolution in the SV method. In order to make an objective comparison with the DG method, the limiters are implemented in a similar fashion.

Remark: In Wang's paper [65], the Lebesgue constant is employed to quantify the quality of the reconstructions. Following the paper [65], the Lebesgue constant $\|\Gamma_\Pi\|$ is expressed as

$$\|\Gamma_\Pi\| = \max_{r \in E} \sum_{j=1}^n L_j(r)$$

for a given partition of a triangle E . It was shown in paper [65] that the smaller the Lebesgue constant, the better the interpolation polynomial. In this work, we use good enough SV partitions so that the interpolation polynomial is convergent when the

computational grid is refined. Figure 2.3 shows the SV partitions used in our numerical simulations for linear, quadratic, and cubic reconstructions. The order of grid nodes, faces, control volumes and Gauss quadrature points (GQPs) are also shown in one spectral volume.

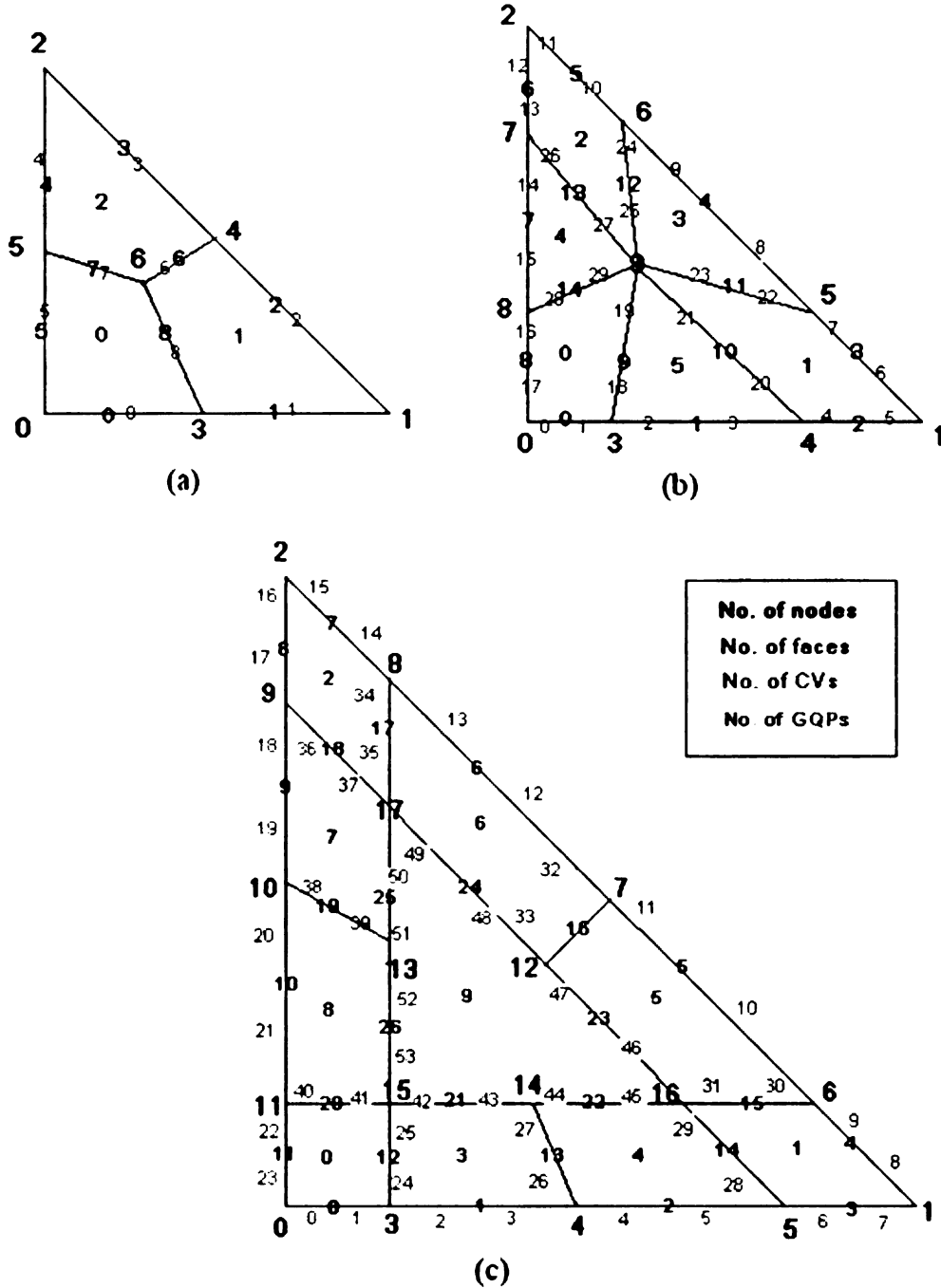


Figure 2.3 SV partitions used in numerical simulations
(a) linear SV, (b)quadratic SV,(c) cubic SV
(this figure presented by color)

CHAPTER 3

EVALUATION OF DG AND SV METHODS

3.1 Number of Operations and Memory Requirement

In order to provide a reasonable estimate of the number of operations for both methods, we need to specify the governing equation and the Riemann solver. Two equations are considered in this research. One is the 2D scalar linear conservation law in which Q is a scalar, and $f = aQ$ and $g = bQ$ with a and b being constants. The other equation is the 2D Euler equations. In both cases, the Rusanov flux [53] (also called local Lax-Friedrich's flux) is selected. The Rusanov flux for the scalar conservation laws takes the following form

$$\hat{F}(Q^L, Q^R, \mathbf{n}) = \begin{cases} Q^L(an_x + bn_y) & \text{if } (an_x + bn_y) > 0 \\ Q^R(an_x + bn_y) & \text{otherwise} \end{cases} \quad (3.1)$$

Since modern computers can execute multiplications as fast as additions, 1 operation is defined to be one multiplication or one addition. Internal functions such as *sqrt* is assumed to cost 10 operations. In addition, each *if* statement is also counted as 1 operation. In this case, this scalar Riemann solver costs $M_R = 5$ operations (3 operations to compute $an_x + bn_y$, one *if* statement, and another multiplication). The analytical flux takes M_a operations.

For the Euler equations, the Rusanov Riemann flux [53] takes the following form

$$\hat{F}(Q^L, Q^R, \mathbf{n}) = \frac{1}{2} \{ [F(Q^L) + F(Q^R)] \bullet \mathbf{n} - \alpha(Q^R - Q^L) \} \quad (3.2)$$

where $\alpha = |\bar{v}_n| + \bar{c}$, $\bar{v}_n$ is the average face normal velocity, and $\bar{c}$ the average speed of sound at the interface. Given the vector of conservative variables, it is estimated that an analytical flux evaluation costs $M_a = 24$ operations, and a Riemann flux takes $M_R = 85$ operations.

For simplicity we have not considered the cost of limiters in the number of operations. We do believe that the limiters in the SV method are more expensive to be implemented than those in the DG method because data limiting is carried out for each element in the DG method, but for each sub-cell (CV) in the SV method.

3.1.1 DG Method

We consider linear, quadratic and cubic elements, which are expected to yield second, third and fourth-order spatial accuracy respectively. The DOFs for these elements are shown in Figure 2.1. Over each element T_i , the residual vector can be written as

$$R^i(U) = -\left(W^i\right)^{-1} \begin{bmatrix} \int_{T_i} (\mathbf{F} \bullet \nabla \xi_1) dV - \oint_{\partial T_i} \hat{F} \xi_1 dS \\ \vdots \\ \int_{T_i} (\mathbf{F} \bullet \nabla \xi_n) dV - \oint_{\partial T_i} \hat{F} \xi_n dS \end{bmatrix}. \quad (3.3)$$

3.1.1.1 Numbers of operations

The total number of operations can be roughly divided into three main parts, corresponding to the cost for computing the state variables at all the Gauss quadrature

points (N_1), the number of operations to compute the fluxes (N_2), and the cost to multiply the mass matrix (N_3).

There are a total of $(nv+3*ns)$ quadrature points that are used for surface and volume integrals. We need n multiplications and $n-1$ additions to compute one state variable given the DOFs. Assume Q has nc component. Then the total number of operations to compute the solutions at all the quadrature points is then

$$N_1 = nc * (2 * n - 1) * (nv + 3 * ns). \quad (3.4)$$

Note that we have ignored the number of operations to compute the limiter for simplicity. To evaluate the volume integral, we need to compute the (analytical) fluxes at nv quadrature points relating to n shape functions, while $3*ns$ Riemann fluxes are necessary to evaluate the surface integral. However Riemann fluxes are shared between two neighboring elements. Therefore we need to halve the number of operations for the Riemann fluxes when evaluating the number of operations per element. We also need to include the number of operations to carry out the Gauss quadrature formula. Thus we obtain

$$N_2 = n * nv * M_a + 3 * ns * M_R / 2 + nc * n * (2 * nv - 1) + nc * n * (3 * ns - 1) \quad (3.5)$$

N_3 is simply the cost of a square matrix multiplying a vector, which is $2*n*n-n$ for one component. For nc components, we therefore have

$$N_3 = nc * (2 * n * n - n). \quad (3.6)$$

Note that $N_3 = 0$ if an orthogonal basis is used.

The total cost to compute the residual vector for a single element is then

$$N_T = N_1 + N_2 + N_3 \quad (3.7)$$

The numbers of operations for the DG schemes of second to fourth orders are listed in Table 3.1.

Table 3.1 Number of operations for the DG method

Equation	k	N	nv	Ns	N_T
Scalar Conservation Law	1	3	3	2	141
	2	6	6	3	512
	3	10	12	4	1496
Euler Equations	1	3	3	2	831
	2	6	6	3	2627
	3	10	12	4	7334

3.1.1.2 Memory requirement

The memory requirement for the DG method is estimated as the following:

- Two solutions; one at the current time step, and the other at the last time step.
- Residual.
- Volume, centroid coordinates, face area, and face unit normal.
- Coordinates of quadrature points (face, cell)
- Gradient of shape function on quadrature points (face, cell).
- Shape functions, and their gradients at the centroid of the elements.

The storage requirement is roughly 90 words per element for a second-order DG scheme, 221 words per element for a third-order DG scheme, and 512 words per element for a fourth-order DG scheme for the 2D Euler equations.

3.1.2 SV Method

The degrees of freedom in the SV method are the mean state variables at the sub-cells, i.e. control volumes. Over each spectral volume T_i , the residual can be expressed as

$$R^i(U) = \begin{bmatrix} - \oint \hat{F} dS \\ \partial C_{i,1} \\ \vdots \\ - \oint \hat{F} dS \\ \partial C_{i,n} \end{bmatrix}. \quad (3.8)$$

3.1.2.1 Numbers of operations

There are two kinds of faces in a spectral volume. The faces that lie on the SV boundaries are called *Riemann faces*, because the state variables are discontinuous across these faces. The other faces that lie inside a SV are named *continuous faces* because the state variables are continuous across these faces. Denote the total number of faces in a SV with nf , and the number of Riemann faces nr . Then the number of continuous faces is then $(nf - nr)$. Let the number of quadrature points on each face (edge) be ne . Then the number of operations to compute the state variables at all the quadrature points is $nc*nf*ne*(2n-1)$. In addition, a total of $(nf - nr)*ne$ analytical fluxes need to be computed while $nr*ne$ Riemann fluxes must be computed. Since the Riemann faces are shared between two neighboring SVs, the number of operations is again halved. We also include the number

of operations to carry out the Gauss quadrature formula $(2*ne-1)*nf*nc$. Since the mass matrix in the SV method is always the identity matrix, number of operations in the SV method can be written as

$$N_T = N_1 + N_2 + N_3$$

where

$$N_1 = nc * nf * ne * (2n - 1) \quad (3.9)$$

$$N_2 = (nf - nr) * ne * M_a + nr * ne * M_R / 2 + (2 * ne - 1) * nf * nc \quad (3.10)$$

$$N_3 = 0 \quad (3.11)$$

The numbers of operations for the SV schemes of second to fourth orders are listed in Table 3.2 for both the scalar and system conservation laws.

Table 3.2 Number of operations for the SV method

Equation	k	N	ne	nf	nr	N_T
Scalar Conservation Law	1	3	1	9	6	81
	2	6	2	15	9	468
	3	10	2	27	12	1287
Euler Equations	1	3	1	9	6	543
	2	6	2	15	9	2553
	3	10	2	27	12	6168

3.1.2.2 Memory requirement

The permanent memory requirement is estimated as follows:

- Two solutions, one at the current time step, and the other at the last time step;
- The residual, volumes and centroid coordinates for the CVs, the face unit normal vectors and areas for the sub-cell grid;
- Face to cell and face to node connectivity for the sub-cell grid;
- Coordinates of the sub-cell grid;
- A connectivity linking each quadrature point on a face to a point of the local standard SV to reconstruct the solution at the quadrature point.

The storage requirement for the 2D Euler equations is roughly 99 words per element for a second-order SV scheme, 194 words per element for a third-order SV scheme, and 361 words per element for a fourth-order SV scheme.

3.2 Accuracy and CPU Times

The following, we will do the comparison of DG and SV methods in terms of accuracy and CPU time on both 2D scalar conservation laws and 2D inviscid flows governed by Euler equations.

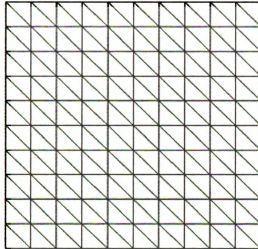
3.2.1 Scalar Conservation Laws

3.2.1.1 2D linear wave equation

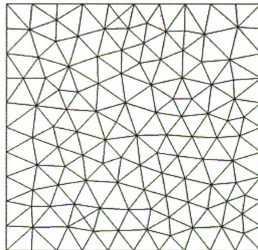
We first test the performance of both methods for the following linear scalar conservation law

$$u_t + u_x + u_y = 0; \quad 0 < x < 2, \quad 0 < y < 2$$

with $u(x, y, 0) = \sin(\pi(x + y))$, and periodic boundary condition. The numerical simulation was carried out until $t = 1$ on two different grids, one regular and one irregular as shown in Figure 3.1. The finer meshes are produced recursively from the coarser meshes by dividing each triangle into 4 smaller triangles. The third-order TVD Runge-Kutta time integration scheme was used with a sufficiently small time step that the errors are independent of the time step. The same time step was used in both the DG and SV methods although larger time steps are permitted in the SV method for stability. The errors are computed based on the cell-averaged state variable on the element or the SV. No limiters were employed in the simulations since the problem is smooth. Tables 3.3 and 3.4 present the errors and CPU times using both methods on the regular mesh, while Tables 3.5 and 3.6 display the errors and CPU times using both methods on the irregular mesh. Note that on the regular mesh, both methods achieved the expected numerical order of accuracy in both L_1 and L_∞ norms. However, the DG method consistently produced L_1 and L_∞ errors of smaller magnitude than the SV method. The SV method, on the other hand, is about 15% - 140% faster than the DG method depending on the order of accuracy of the scheme. On the irregular grid, the DG method is capable of achieving the expected order of accuracy in both L_1 and L_∞ norms. Although the SV method achieved the expected order of accuracy in L_1 norm, the third-order SV scheme showed a reduction of half an order in L_∞ norm. This may indicate the quality of the quadratic SV partition can be further improved. Again, the SV method is consistently faster than the DG method on the irregular grid.



(a) Regular (10x10x2)



(b) Irregular (10x10x2)

Figure 3.1 Regular and irregular grids

Table 3.3 Errors and CPU times at $t = 1$ for a 2D linear wave equation using the DG method on the regular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10x2	1.14e-02	---	2.43e-02	---	3.33e-01
	20x20x2	2.31e-03	2.30	5.83e-03	2.06	2.76e+00
	40x40x2	5.09e-04	2.19	1.42e-03	2.04	2.23e+01
	80x80x2	1.18e-04	2.11	3.49e-04	2.02	1.81e+02
	160x160x2	2.84e-05	2.06	8.65e-05	2.01	1.45e+03
3	10x10x2	3.45e-04	---	7.65e-04	---	7.590e-01
	20x20x2	4.27e-05	3.02	9.66e-05	2.99	6.37e+00
	40x40x2	5.32e-06	3.00	1.21e-05	3.00	5.09e+01
	80x80x2	6.65e-07	3.00	1.51e-06	3.00	4.27e+02
	160x160x2	8.31e-08	3.00	1.89e-07	3.00	3.37e+03
4	10x10x2	1.39e-05	---	2.43e-05	---	1.66e+00
	20x20x2	8.59e-07	4.02	1.52e-06	4.00	1.33e+01
	40x40x2	5.34e-08	4.01	9.54e-08	4.00	1.08e+02
	80x80x2	3.33e-09	4.00	5.97e-09	4.00	8.47e+02
	160x160x2	2.08e-10	4.00	3.73e-10	4.00	7.28e+03

Table 3.4 Errors and CPU times at $t = 1$ for a 2D linear wave equation using the SV method on the regular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10x2	4.02e-02	---	5.86e-02	---	1.21e-01
	20x20x2	1.06e-02	1.92	1.59e-02	1.88	9.47e-01
	40x40x2	2.71e-03	1.97	4.09e-03	1.96	8.81e+00
	80x80x2	6.83e-04	1.99	1.03e-03	1.99	8.39e+01
	160x160x2	1.71e-04	2.00	2.59e-04	1.99	6.05e+02
3	10x10x2	3.73e-03	---	5.21e-03	---	4.68e-01
	20x20x2	4.77e-04	2.97	7.12e-04	2.87	4.19e+00
	40x40x2	6.04e-05	2.98	9.05e-05	2.98	3.67e+01
	80x80x2	7.59e-06	2.99	1.14e-05	2.98	2.91e+02
	160x160x2	9.51e-07	3.00	1.43e-06	2.99	2.21e+03
4	10x10x2	5.90e-05	---	8.40e-05	---	1.32e+00
	20x20x2	3.73e-06	3.98	5.37e-06	3.97	1.35e+01
	40x40x2	2.35e-07	3.99	3.34e-07	4.01	8.61e+01
	80x80x2	1.48e-08	3.99	2.09e-08	4.00	6.90e+02
	160x160x2	9.24e-10	4.00	1.31e-09	4.00	5.72e+03

Table 3.5 Errors and CPU times at $t = 1$ for a 2D linear equation using the DG method on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	2.17e-02	---	6.05e-02	---	3.77e-01
	20x20	4.67e-03	2.22	1.64e-02	1.88	3.26e+00
	40x40	1.07e-03	2.12	4.17e-03	1.97	2.65e+01
	80x80	2.56e-04	2.06	1.05e-03	1.99	2.13e+02
	160x160	6.26e-05	2.03	2.62e-04	2.00	1.75e+03
3	10x10	7.34e-04	---	2.56e-03	---	9.12e-01
	20x20	8.72e-05	3.07	4.42e-04	2.53	7.50e+00
	40x40	1.07e-05	3.03	6.34e-05	2.80	6.00e+01
	80x80	1.33e-06	3.01	8.19e-06	2.95	4.8e+02
	160x160	1.66e-07	3.00	1.03e-06	2.99	4.29e+03
4	10x10	4.10e-05	---	1.80e-04	---	1.93e+00
	20x20	2.41e-06	4.09	1.30e-05	3.79	1.57e+01
	40x40	1.47e-07	4.04	8.54e-07	3.92	1.27e+02
	80x80	9.08e-09	4.01	5.72e-08	3.90	1.02e+03
	160x160	5.65e-10	4.01	3.72e-09	3.94	8.87e+03

Table 3.6 Errors and CPU times at $t = 1$ for a 2D linear wave equation using the SV method on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	6.71e-02	---	1.18e-01	---	1.38e-01
	20x20	1.83e-02	1.87	3.40e-02	1.80	1.30e+00
	40x40	4.71e-03	1.96	9.25e-03	1.88	1.17e+01
	80x80	1.19e-03	1.98	2.42e-03	1.94	8.61e+01
	160x160	3.00e-04	1.99	6.20e-04	1.96	8.38e+02
3	10x10	8.36e-03	---	1.68e-02	---	5.59e-01
	20x20	1.15e-03	2.86	2.95e-03	2.51	4.79e+00
	40x40	1.52e-04	2.92	5.28e-04	2.48	3.88e+01
	80x80	2.01e-05	2.91	1.31e-04	2.01	3.27e+02
	160x160	2.64e-06	2.93	2.85e-05	2.20	2.71e+03
4	10x10	2.28e-04	---	7.39e-04	---	1.53e+00
	20x20	1.37e-05	4.06	5.45e-05	3.76	1.35e+01
	40x40	8.50e-07	4.01	3.54e-06	3.94	1.03e+02
	80x80	5.33e-08	4.00	2.21e-07	4.00	7.98e+02
	160x160	3.35e-09	3.99	1.34e-08	4.04	7.31e+03

3.2.1.2 2D Burger's equation

Consider the two-dimensional nonlinear wave equation:

$$u_t + uu_x + uu_y = 0, \quad -1 < x < 1, \quad -1 < y < 1$$

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

We perform the simulation until $t = 0.1$ when the solution is still smooth. The errors and CPU times are documented in Tables 3.7 and 3.8. We also test the performance of TVB limiters [65], in which the simulation is performed until $t = 0.45$ when a shock wave appeared in the solution. The solution errors in the smooth region $[-0.2, 0.4] \times [-0.2, 0.4]$ are computed and presented in Tables 3.9 and 3.10.

Table 3.7 Errors and CPU time on the 2D Burger's equation at $t = 0.1$ using DG on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	1.06e-02	---	3.48e-02	---	3.74e-02
	20x20	2.75e-03	1.95	1.14e-02	1.61	3.22e-01
	40x40	6.82e-04	2.01	3.21e-03	1.83	2.65e+00
	80x80	1.70e-04	2.00	8.24e-04	1.96	2.18e+01
	160x160	4.24e-05	2.00	2.08e-04	1.98	1.74e+02
3	10x10	6.80e-04	---	3.17e-03	---	1.80e-01
	20x20	1.14e-04	2.57	8.32e-04	1.93	1.54e+00
	40x40	1.79e-05	2.68	1.62e-04	2.36	1.29e+01
	80x80	2.73e-06	2.71	3.45e-05	2.23	9.76e+01
	160x160	4.08e-07	2.74	5.89e-06	2.55	7.76e+02
4	10x10	6.01e-05	---	4.58e-04	---	2.91e-01
	20x20	3.68e-06	4.03	3.76e-05	3.61	2.34e+00
	40x40	2.34e-07	3.98	2.47e-06	3.93	1.93e+01
	80x80	1.61e-08	3.86	2.07e-07	3.58	1.53e+02
	160x160	1.20e-09	3.75	1.95e-08	3.41	1.21e+03

Table 3.8 Errors and CPU time on the 2D Burger's equation at $t = 0.1$ using SV on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	5.79e-03	---	1.76e-02	---	1.34e-02
	20x20	1.46e-03	1.99	4.91e-03	1.84	1.29e-01
	40x40	3.67e-04	1.99	1.41e-03	1.80	1.17e+00
	80x80	9.41e-05	1.96	5.60e-04	1.34	8.62e+00
	160x160	2.39e-05	1.97	2.68e-04	1.06	6.92e+01
3	10x10	6.28e-04	---	2.15e-03	---	1.08e-01
	20x20	1.18e-04	2.42	6.20e-04	1.80	9.80e-01
	40x40	1.91e-05	2.62	1.38e-04	2.17	7.99e+00
	80x80	3.02e-06	2.66	3.28e-05	2.07	6.83e+01
	160x160	4.64e-07	2.70	6.09e-06	2.43	5.09e+02
4	10x10	6.73e-05	---	4.43e-04	---	2.28e-01
	20x20	5.19e-06	3.70	5.61e-05	2.98	1.89e+00
	40x40	3.93e-07	3.72	4.46e-06	3.65	1.50e+01
	80x80	2.95e-08	3.74	3.10e-07	3.85	1.21e+02
	160x160	2.38e-09	3.63	3.13e-08	3.31	1.13e+03

Table 3.9 Errors and CPU time on the 2D Burger's equation at $t = 0.45$ using DG on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	1.78e-03	---	6.11e-03	---	4.22e-01
	20x20	1.75e-04	3.35	5.90e-04	3.37	3.37e+00
	40x40	3.29e-05	2.41	1.97e-04	1.58	2.73e+01
	80x80	6.84e-06	2.27	5.06e-05	1.96	2.26e+02
	160x160	1.53e-06	2.16	9.94e-06	2.35	1.82e+03
3	10x10	9.49e-04	---	4.85e-03	---	9.20e-01
	20x20	7.68e-05	3.63	4.62e-04	3.39	7.41e+00
	40x40	7.73e-06	3.31	5.11e-05	3.18	6.04e+01
	80x80	8.26e-07	3.23	9.77e-06	2.39	5.06e+02
	160x160	9.58e-08	3.11	2.12e-06	2.20	4.11e+03
4	10x10	7.18e-04	---	4.87e-03	---	1.91e+00
	20x20	8.80e-06	6.35	1.14e-04	5.42	1.55e+01
	40x40	2.63e-07	5.06	2.22e-05	2.36	1.26e+02
	80x80	1.50e-08	4.13	2.60e-08	9.73	1.20e+03
	160x160	3.63e-09	2.05	5.93e-09	2.14	9.88e+03

Table 3.10 Errors and CPU time on the 2D Burger's equation at $t = 0.45$ using SV on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	1.99e-03	---	3.80e-03	---	2.05e-01
	20x20	4.84e-04	2.04	1.53e-03	1.31	1.67e+00
	40x40	1.14e-04	2.09	3.92e-04	1.97	1.81e+01
	80x80	2.87e-05	1.99	1.29e-04	1.60	1.57e+02
	160x160	7.11e-06	2.01	3.31e-05	1.97	1.21e+03
3	10x10	1.05e-03	---	6.00e-03	---	6.73e-01
	20x20	8.03e-05	3.71	6.20e-04	3.27	5.85e+00
	40x40	7.30e-06	3.46	3.89e-05	3.99	5.82e+01
	80x80	8.68e-07	3.07	7.30e-06	2.41	4.53e+02
	160x160	1.11e-07	2.97	1.46e-06	2.32	3.61e+03
4	10x10	4.39e-04	---	3.25e-03	---	1.61e+00
	20x20	4.63e-06	6.57	7.14e-05	5.51	1.43e+01
	40x40	9.87e-08	5.55	1.39e-06	5.69	1.23e+02
	80x80	1.55e-08	2.67	3.56e-08	5.28	1.02e+03
	160x160	3.65e-09	2.09	7.14e-09	2.32	8.70e+03

3.2.2 Euler Equations

We consider two-dimensional inviscid flow, and the governing equations are described as follows,

$$\frac{\partial Q}{\partial t} + \nabla \bullet F = 0$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, f = \begin{bmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ u(E + p) \end{bmatrix}, g = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v v + p \\ v(E + p) \end{bmatrix},$$

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

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

with the ratio of specific heat γ being a constant. $F = [f, g]$ is the flux vector.

3.2.2.1 Vortex propagation problem

To compare the numerical accuracy by DG and SV methods, we test the vortex propagation problem since it has analytical solution. This is an idealized problem for the Euler equations in 2D, which was used by Shu [56].

The mean flow is $\{\rho, u, v, p\} = \{1, 1, 1, 1\}$. An isotropic vortex is then added to the mean flow, i.e., with perturbations in u , v , and temperature $T = p/\rho$, and no perturbation in entropy $S = p/\rho^\gamma$:

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

where $(\bar{x}, \bar{y}) = (x-5, y-5)$, $r^2 = \bar{x}^2 + \bar{y}^2$, and the vortex strength $\varepsilon = 5$. In the numerical simulation, the computational domain is taken to be $[0, 10] \times [0, 10]$, with characteristic inflow and outflow boundary conditions imposed on the boundaries.

It can be readily verified that the Euler equations with the above initial conditions admit an exact solution that moves with the speed $(1, 1)$ in the diagonal direction. Both the DG and SV methods were employed to simulate this problem. The numerical simulation was carried out until $t = 10$ on the two different grids shown in Figure 3.1.

The errors are computed based on the volume-averaged density on the element or the SV. Table 3.11 and Table 3.12 present the errors and recorded CPU times of both methods on the regular mesh, while Table 3.13 and Table 3.14 display the errors and CPU times on the irregular mesh. Note that the DG method is more consistent in achieving the expected order of accuracy than the SV method on both the regular and irregular grids. It is interesting to see that the second-order SV method produced smaller error than the second-order DG method for the Euler equations. The third and fourth-order DG schemes appear to have smaller error magnitude than the corresponding SV schemes.

Based on the CPU times for the regular mesh, we note for per time step per element that the DG method takes 43.47, 104.14, and 212.35 μs for linear, quadratic, and cubic elements respectively, while SV method spends 27.42, 99.88, and 237.69 μs for the 2nd, 3rd and 4th order schemes. The SV method is faster than the DG method at 2nd and 3rd order, but is slightly slower at 4th order.

Table 3.11 Errors and CPU times for the propagating vortex case at $t = 10$ using the DG method on the regular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10x2	7.74e-04	--	4.02e-03	--	8.91e+00
	20x20x2	1.05e-04	2.88	1.12e-03	1.84	7.40e+01
	40x40x2	1.52e-05	2.79	2.73e-04	2.04	6.12e+02
	80x80x2	2.39e-06	2.67	1.21e-04	1.17	4.70e+03
3	10x10x2	2.86e-04	--	2.22e-03	--	2.11e+01
	20x20x2	7.54e-05	1.92	9.98e-04	1.15	1.72e+02
	40x40x2	1.26e-05	2.58	1.68e-04	2.57	1.36e+03
	80x80x2	1.14e-06	3.47	2.60e-05	2.69	1.10e+04
4	10x10x2	1.20e-04	--	8.12e-04	--	4.45e+01
	20x20x2	6.60e-06	4.18	7.14e-05	3.51	3.65e+02
	40x40x2	1.47e-07	5.49	3.42e-06	4.38	2.89e+03
	80x80x2	3.90e-09	5.24	2.10e-07	4.03	2.31e+04

Table 3.12 Errors and CPU time for the propagating vortex case at $t = 10$ using the SV method on the regular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10x2	9.50e-04	--	6.19e-03	--	3.77e+00
	20x20x2	1.92e-04	2.31	1.61e-03	1.94	2.96e+01
	40x40x2	4.14e-05	2.21	8.65e-04	0.90	2.38e+02
	80x80x2	9.92e-06	2.06	2.96e-04	1.55	1.91e+03
3	10x10x2	9.42e-04	--	5.49e-03	--	2.19e+01
	20x20x2	9.20e-05	3.36	9.91e-04	2.47	1.68e+02
	40x40x2	9.84e-06	3.22	2.45e-04	2.02	1.30e+03
	80x80x2	1.11e-06	3.15	3.56e-05	2.78	1.02e+04
4	10x10x2	1.82e-04	--	1.20e-03	--	5.05e+01
	20x20x2	1.01e-05	4.17	9.03e-05	3.73	3.91e+02
	40x40x2	4.90e-07	4.37	1.01e-05	3.16	3.35e+03
	80x80x2	3.16e-08	3.95	5.81e-07	4.12	2.46e+04

Table 3.13 Errors and CPU times for propagating vortex case at $t = 10$ using the DG method on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	6.29e-04	--	3.32e-03	--	1.02e+01
	20x20	1.15e-04	2.45	9.97e-04	1.74	8.80e+01
	40x40	2.78e-05	2.05	3.05e-04	1.71	7.07e+02
	80x80	4.20e-06	2.73	8.93e-05	1.77	5.62e+03
3	10x10	1.27e-04	--	8.88e-04	--	2.41e+01
	20x20	1.83e-05	2.79	3.26e-04	1.45	2.05e+02
	40x40	1.93e-06	3.25	9.86e-05	1.73	1.60e+03
	80x80	1.52e-07	3.67	1.61e-05	2.61	1.29e+04
4	10x10	4.23e-05	--	3.69e-04	--	5.12e+01
	20x20	2.56e-06	4.05	6.17e-05	2.58	4.94e+02
	40x40	9.80e-08	4.71	3.08e-06	4.32	3.38e+03
	80x80	2.67e-09	5.20	2.12e-07	3.86	2.69e+04

Table 3.14 Errors and CPU times for the propagating vortex case at $t = 10$ using the SV method on the irregular mesh

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order	CPU (s)
2	10x10	1.04e-03	--	5.45e-03	--	4.38e+00
	20x20	2.54e-04	2.03	1.92e-03	1.51	3.42e+01
	40x40	8.98e-05	1.50	8.72e-04	1.14	3.16e+02
	80x80	2.34e-05	1.94	2.90e-04	1.59	2.22e+03
3	10x10	5.06e-04	--	3.47e-03	--	2.53e+01
	20x20	7.44e-05	2.77	7.25e-04	2.26	1.93e+02
	40x40	9.75e-06	2.93	1.63e-04	2.15	1.51e+03
	80x80	1.51e-06	2.69	3.37e-05	2.27	1.19e+04
4	10x10	1.09e-04	--	5.57e-04	--	5.71e+01
	20x20	7.36e-06	3.89	8.76e-05	2.67	4.45e+02
	40x40	3.84e-07	4.26	5.75e-06	3.93	3.51e+03
	80x80	2.22e-08	4.11	4.00e-07	3.85	2.81e+04

3.2.2.2 Double Mach reflection problem

This problem is also a standard test case [71] for high-resolution schemes, and has been studied extensively by many researchers. The computational domain for this problem is chosen to be $[0, 4] \times [0, 1]$, which is displayed in Figure 3.2. The reflecting wall lies at the bottom of the computational domain starting from $x=1/6$. Initially a right-moving Mach 10 shock is positioned at $x=1/6, y=0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the region from $x=0$ to $x=1/6$ and a solid wall boundary condition is used for the rest. For the top boundary of the computational domain, the solution is set to describe the exact motion of the Mach 10 shock. The left boundary is set at the exact post-shock condition, while the right boundary is set as an outflow boundary.

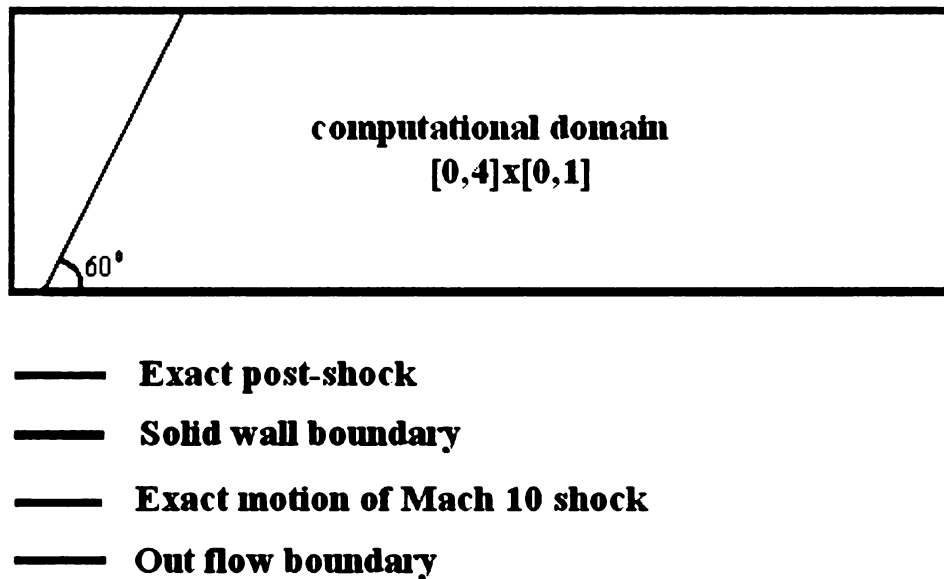
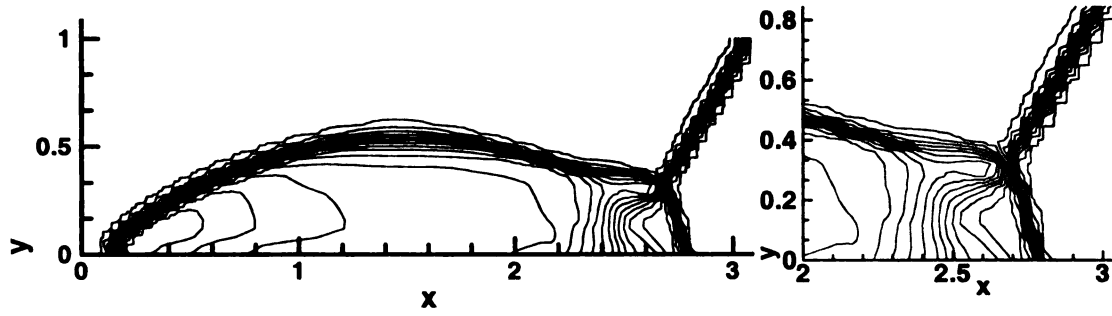


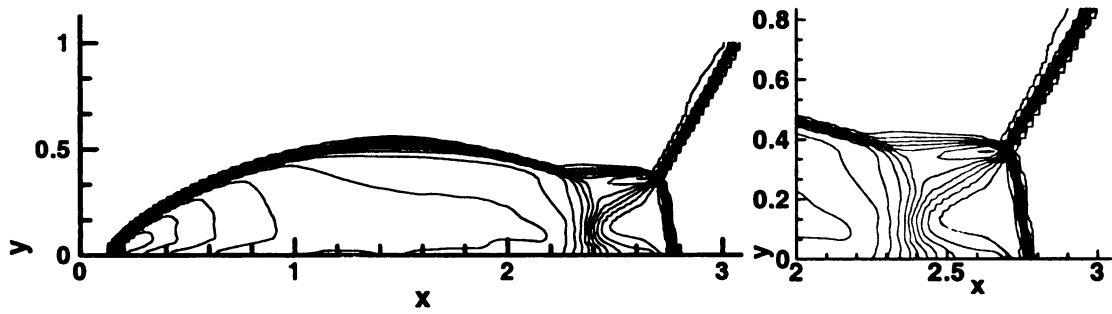
Figure 3.2 Computational domain and boundaries for double Mach reflection problem
(this figure presented by color)

The numerical simulation was carried out until $t = 0.2$. A mesh refinement study was carried out on three different grids. The grids are generated from regular Cartesian meshes by subdividing each Cartesian cell into two triangles. The coarse grid has $25 \times 100 \times 2$ triangles, the medium grid $50 \times 188 \times 2$ triangles, and the fine grid consists of $120 \times 480 \times 2$ triangles. The density contours with 30 equally spaced contour lines from $\rho = 1.528$ to $\rho = 20.863$ are shown in Figure 3.3 and Figure 3.4 for the second order DG scheme and SV scheme. Note that the “blown-up” region was also shown in those figures.

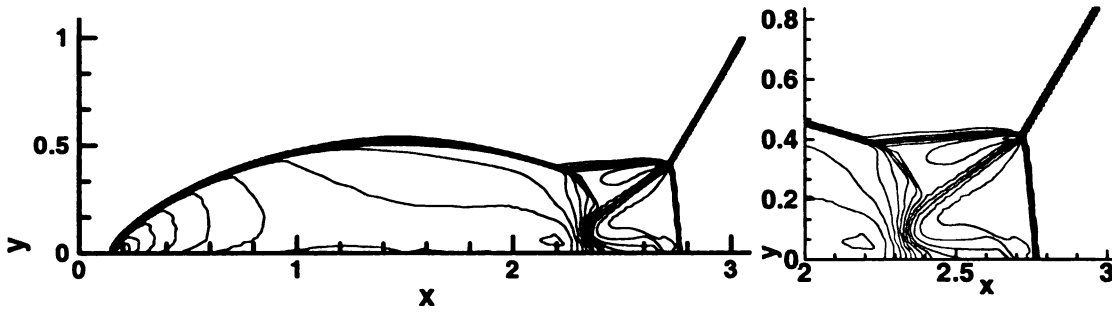
Note that the SV method has a higher resolution than the DG method for the shock, slip line and the other finer features near the triple point. The main reason is that the TVD limiter in the SV method is applied for the sub-cells, but the limiter in the DG method is applied for the elements (macro SVs).



(a) Coarse grid

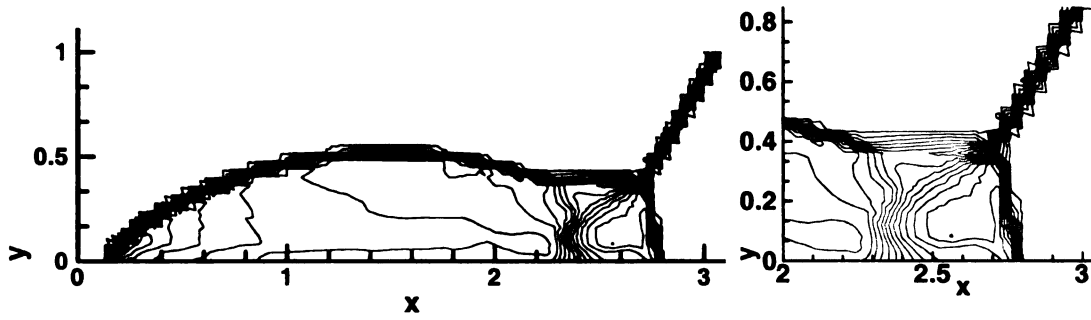


(b) Medium grid

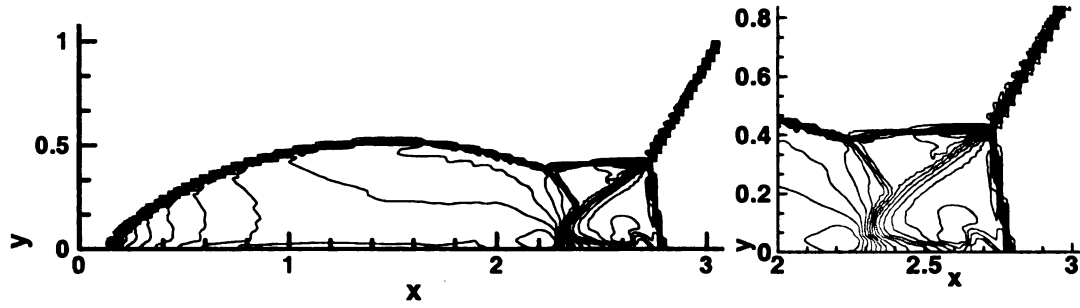


(c) Fine grid

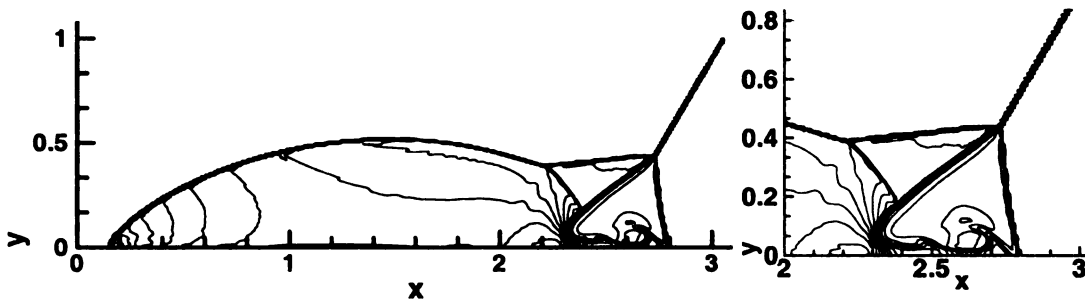
Figure 3.3 Density contours computed with second order DG scheme using a TVD limiter (30 equally spaced contour lines from $\rho = 1.528$ to $\rho = 20.863$).



(a) Coarse grid



(b) Medium grid



(c) Fine grid

Figure 3.4 Density contours computed with second order SV scheme using a TVD limiter (30 equally spaced contour lines from $\rho = 1.528$ to $\rho = 20.863$)

3.3 Conclusions

We have presented a comparison of the DG and SV methods for the 2D scalar conservation laws and Euler equations. Generally speaking, both DG and SV method have achieved the desired order of accuracy. The DG method has a lower error magnitude than the SV method while SV is faster than DG. In the scalar case, the SV schemes are consistently faster than the DG schemes of the same order of accuracy for each residual evaluation. For the Euler equations, the 2nd-order SV scheme is faster than the 2nd-order DG scheme. However, 3rd and 4th order SV schemes are quite similar to the corresponding DG schemes in terms of efficiency (<12 % in difference). It is also clear that the SV method has a higher resolution for discontinuities than the DG method because of the sub-cell average based data limiting. We also confirm that the SV method takes less memory and allows larger time steps than the DG method for both the 2D scalar conservation laws and Euler equations.

CHAPTER 4

SV METHOD FOR THE DIFFUSION EQUATION

As a first-step towards extending the SV method to the Navier-Stokes equations, the SV method is extended to and tested for the diffusion equation. In this chapter, we consider the following 1D diffusion equation

$$u_t = u_{xx}, \quad x \in [0, 2\pi] \quad (4.1)$$

with periodic boundary conditions and initial condition $u(x, 0) = \sin(x)$.

4.1 Three Formulations

4.1.1 Naïve SV Formulation

Directly following the basic formulation described in [64] for the one-dimensional hyperbolic conservation law, we integrate Eq. (4.1) in control volume $C_{i,j}$, which is a sub-cell of a spectral volume $S_i = [x_{i-1/2}, x_{i+1/2}]$ depicted in Figure 4.1, replace the flux by a

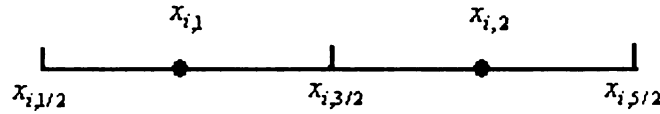


Figure 4.1 Linear spectral volume

numerical flux and obtain

$$\frac{d\bar{u}_{i,j}(t)}{dt} - \frac{1}{h_{i,j}} (\hat{u}_x|_{i,j+1/2} - \hat{u}_x|_{i,j-1/2}) = 0. \quad (4.2)$$

Since there is no convection term in the diffusion equation, the first derivative is “naturally” computed by taking a simple average of the derivatives from the two neighboring CVs, i.e.,

$$\hat{u}_x|_{i,j+1/2} = \frac{1}{2}((u_x)_{i,j+1/2}^+ + (u_x)_{i,j+1/2}^-). \quad (4.3)$$

For time integration, we employ the third order TVD Runge-Kutta method [19]. This formulation was used to compute a numerical solution for Eq. (4.1) at $t = 0.7$. Two different grids were used in the simulation. In Figures 4.2 and 4.3, the numerical solutions with 40 and 320 SVs are compared with the exact solution using linear and quadratic reconstructions. It seems this formulation leads to a seemingly converged, but wrong solution. Note that the numerical solutions have an $O(1)$ error, which does not decrease with grid refinement. The same phenomenon was reported by Zhang and Shu [72] for DG.

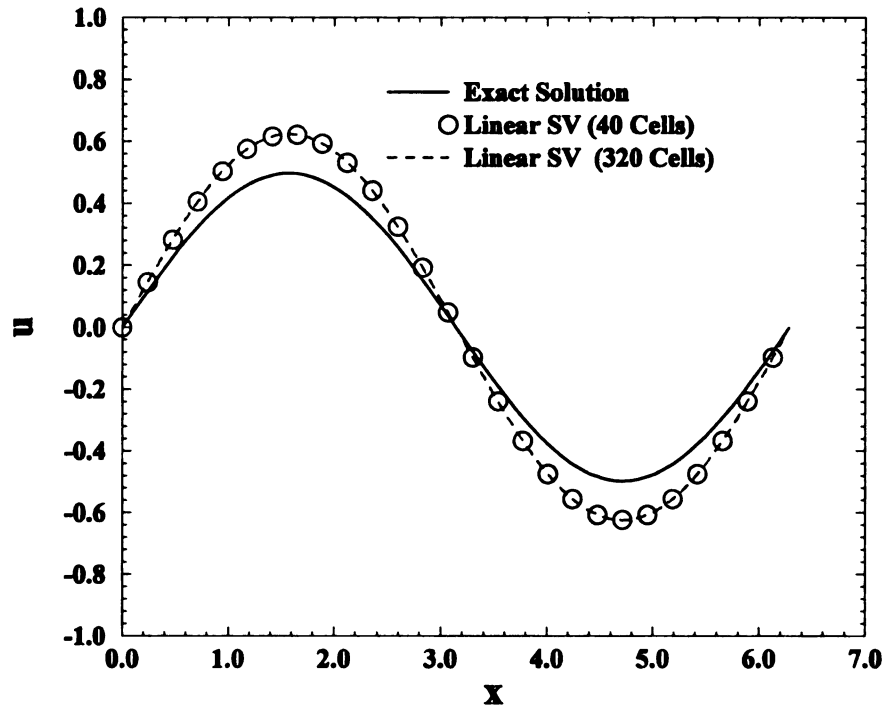


Figure 4.2 The numerical solutions versus the exact solution using the *linear reconstruction* based on the Naïve SV formulation for the diffusion equation

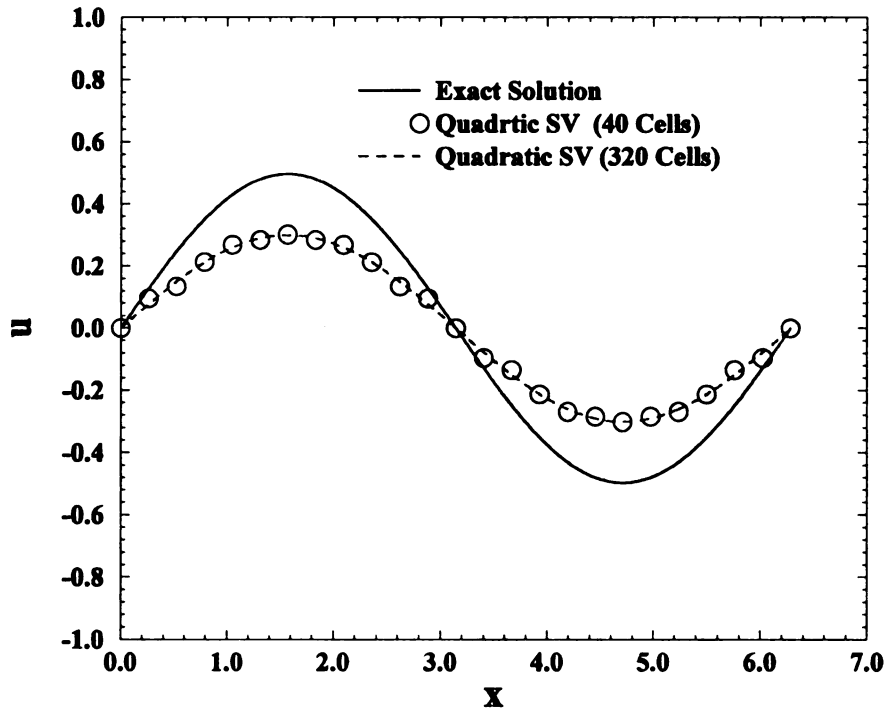


Figure 4.3 The numerical solutions versus the exact solution using the *quadratic reconstruction* based on the Naïve SV formulation for the diffusion equation

4.1.2 Local SV Formulation

The second formulation is obtained by mimicking the local discontinuous Galerkin method [19]. A new variable q is introduced, which is equal to u_x , the gradient of u .

Then the diffusion equation becomes the following system

$$\begin{cases} u_t - q_x = 0 \\ q - u_x = 0 \end{cases}. \quad (4.4)$$

The spectral volume method is then applied to this system directly. Integrating Eq. (4.4) over each control volume, we obtain

$$\begin{cases} \frac{d\bar{u}_{i,j}}{dt} - \frac{1}{h_{i,j}}(\hat{q}|_{i,j+1/2} - \hat{q}|_{i,j-1/2}) = 0 \\ \bar{q}_{i,j} - \frac{1}{h_{i,j}}(\hat{u}|_{i,j+1/2} - \hat{u}|_{i,j-1/2}) = 0. \end{cases} \quad (4.5)$$

The numerical fluxes are chosen as following [57]

$$\hat{u}|_{i,j+1/2} = u|_{i,j+1/2}^+ \quad (4.6)$$

$$\hat{q}|_{i,j+1/2} = q|_{i,j+1/2}^-, \quad (4.7)$$

i.e. we alternatively take the downwind value for u and upwind value for q (we could of course also take the opposite pattern). Let m be the degree of the reconstruction polynomial. Numerical solutions are computed at $t = 1.0$ for the three cases $m = 1$ (linear reconstruction), $m = 2$ (quadratic reconstruction) and $m = 3$ (cubic reconstruction). The L_1 and L_∞ errors and numerically observed orders of accuracy are presented in Table 4.1, from which we note that a $(m+1)^{th}$ order of accuracy is achieved for a degree m polynomial reconstruction.

Table 4.1 L_1 and L_∞ errors and orders of accuracy based on the local SV formulation for the diffusion equation

Order of accuracy	h	L_1 error	L_1 order	L_∞ error	L_∞ order
2 (Linear SV)	$2\pi/10$	2.35e-02	---	3.61e-02	---
	$2\pi/20$	6.00e-03	1.97	9.44e-03	1.94
	$2\pi/40$	1.51e-03	1.99	2.37e-03	1.99
	$2\pi/80$	3.78e-04	2.00	5.94e-04	2.00
	$2\pi/160$	9.45e-05	2.00	1.49e-04	2.00
	$2\pi/320$	2.36e-05	2.00	3.71e-05	2.01
3 (Quadratic SV)	$2\pi/10$	1.15e-03	---	1.78e-03	---
	$2\pi/20$	1.42e-04	3.02	2.22e-04	3.00
	$2\pi/40$	1.76e-05	3.01	2.77e-05	3.00
	$2\pi/80$	2.20e-06	3.00	3.46e-06	3.00
	$2\pi/160$	2.75e-07	3.00	4.32e-07	3.00
	$2\pi/320$	3.44e-08	3.00	5.40e-08	3.00
4 (Cubic SV)	$2\pi/10$	6.99e-05	---	1.07e-04	---
	$2\pi/20$	4.36e-06	4.00	6.82e-06	3.97
	$2\pi/40$	2.72e-07	4.00	4.27e-07	4.00
	$2\pi/80$	1.70e-08	4.00	2.66e-08	4.00
	$2\pi/160$	1.05e-09	4.02	1.65e-09	4.01
	$2\pi/320$	5.38e-11	4.29	8.45e-11	4.29

4.1.3 Penalty SV Formulation

In order to remedy the first formulation, Baumann and Oden [8], also Oden, Babuska, and Baumann [44], and Riviere, Wheeler and Girault [51] introduced a penalty term to the numerical flux in the DG implementation. However if the formulation of Baumann and Oden [8] is used directly in the SV method, the penalty term vanishes because the weighting function is piece-wise constant in the SV method. Therefore the Baumann and Oden formulation for the SV method is identical to the first formulation. Instead, a penalty-like term in the following form is added to the numerical flux for the SV method at the interface. The SV scheme then becomes

$$\frac{d\bar{u}_{i,j}(t)}{dt} - \frac{1}{h_{i,j}}(\hat{u}_x|_{i,j+1/2} - \hat{u}_x|_{i,j-1/2}) = 0 \quad (4.8)$$

$$\hat{u}_x|_{i,j+1/2} = \frac{1}{2}((u_x)^+_{i,j+1/2} + (u_x)^-_{i,j+1/2}) + \frac{\epsilon}{h_{i,j}}(u|_{i,j+1/2}^+ - u|_{i,j+1/2}^-), \quad (4.9)$$

where ϵ is a constant. A Fourier analysis is performed for this formulation in the case of $m = 1$, and it is found that ϵ must be one to preserve 2nd order accuracy. Furthermore, numerical simulations have showed that this formulation can achieve 2nd order accuracy for linear and quadratic reconstructions, and 4th order accuracy for cubic reconstructions. Table 4.2 shows the L_1 and L_∞ errors and the numerically observed orders of accuracy at $t = 1.0$.

Table 4.2 L_1 and L_∞ errors and orders of accuracy based on the penalty SV formulation for the diffusion equation

Order of accuracy	h	L_1 error	L_1 order	L_∞ error	L_∞ order
2 (Linear SV)	$2\pi/10$	6.05e-03	---	9.35e-03	---
	$2\pi/20$	1.51e-03	2.00	2.34e-03	2.00
	$2\pi/40$	3.78e-04	2.00	5.92e-04	1.98
	$2\pi/80$	9.46e-05	2.00	1.48e-04	2.00
	$2\pi/160$	2.36e-05	2.00	3.71e-05	2.00
	$2\pi/320$	5.91e-06	2.00	9.28e-06	2.00
3 (Quadratic SV)	$2\pi/10$	2.77e-03	---	4.28e-03	---
	$2\pi/20$	6.77e-04	2.03	1.05e-03	2.03
	$2\pi/40$	1.68e-04	2.01	2.63e-04	2.00
	$2\pi/80$	4.20e-05	2.00	6.60e-05	1.99
	$2\pi/160$	1.05e-05	2.00	1.65e-05	2.00
	$2\pi/320$	2.63e-06	2.00	4.13e-06	2.00
4 (Cubic SV)	$2\pi/10$	6.47e-05	---	1.00e-04	---
	$2\pi/20$	3.99e-06	4.02	6.16e-06	4.02
	$2\pi/40$	2.48e-07	4.01	3.88e-07	3.99
	$2\pi/80$	1.55e-08	4.00	2.43e-08	4.00
	$2\pi/160$	9.70e-10	4.00	1.52e-09	4.00
	$2\pi/320$	6.40e-11	3.92	1.01e-10	3.91

4.2 Fourier Analysis

In this analysis, we follow a technique described by Zhang and Shu [72], and focus on the linear reconstruction only. In this case, a SV is partitioned into two equal CVs, as shown in Figure 4.1, and CV-averaged mean solutions are $\bar{u}_{j,1}$ and $\bar{u}_{j,2}$. Assuming that the

mesh is uniform, we have $h = h_j = \frac{2\pi}{N}$, $h_{j,1} = h_{j,2} = \frac{h}{2}$. The linear reconstruction can be expressed as

$$p_j(x) = L_1(x)\bar{u}_{j,1} + L_2(x)\bar{u}_{j,2}, \quad (4.10)$$

with

$$L_1(x) = -\frac{2}{h}(x - x_{j,1/2} + x - x_{j,5/2}) + \frac{1}{h}(x - x_{j,1/2} + x - x_{j,3/2}), \quad (4.11)$$

$$L_2(x) = \frac{1}{h}(x - x_{j,1/2} + x - x_{j,3/2}). \quad (4.12)$$

The derivative of $p_j(x)$ is constant in S_j , i.e.,

$$p_j'(x) = -\frac{2}{h}\bar{u}_{j,1} + \frac{2}{h}\bar{u}_{j,2}. \quad (4.13)$$

All the three SV formulations can be cast in the following form

$$\frac{d}{dt} \begin{bmatrix} \bar{u}_{j,1} \\ \bar{u}_{j,2} \end{bmatrix} = A \cdot \begin{bmatrix} \bar{u}_{j-1,1} \\ \bar{u}_{j-1,2} \end{bmatrix} + B \cdot \begin{bmatrix} \bar{u}_{j,1} \\ \bar{u}_{j,2} \end{bmatrix} + C \cdot \begin{bmatrix} \bar{u}_{j+1,1} \\ \bar{u}_{j+1,2} \end{bmatrix}, \quad (4.14)$$

where A , B and C are constant matrices. We seek general solutions of the following form

$$u(x, t) = \hat{u}_k(t)e^{ikx},$$

where k is the index of modes ($k = 1, 2, \dots$) representing the wave number, and i represents the units of the imaginary number. Obviously, the analytical solution for (4.1)

is $u(x, t) = e^{ikx - k^2 t}$. In addition, we have

$$\begin{bmatrix} \bar{u}_{j,1}(t) \\ \bar{u}_{j,2}(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{h}\hat{u}_k(t) \int_{x_{j,1/2}}^{x_{j,3/2}} e^{ik\xi} d\xi \\ \frac{2}{h}\hat{u}_k(t) \int_{x_{j,3/2}}^{x_{j,5/2}} e^{ik\xi} d\xi \end{bmatrix}.$$

Assuming $\eta = \xi - x_{j,3/2}$, we obtain

$$\begin{bmatrix} x_{j,3/2} \\ \int_{x_{j,1/2}}^{x_{j,3/2}} e^{ik\xi} d\xi \\ x_{j,1/2} \\ x_{j,5/2} \\ \int_{x_{j,3/2}}^{x_{j,5/2}} e^{ik\xi} d\xi \\ x_{j,3/2} \end{bmatrix} = \begin{bmatrix} 0 \\ \int_{-h/2}^0 e^{ik(\eta+x_{j,3/2})} d\eta \\ -h/2 \\ h/2 \\ \int_0^{h/2} e^{ik(\eta+x_{j,3/2})} d\eta \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \int_{-h/2}^0 e^{ik\eta} d\eta \\ -h/2 \\ h/2 \\ \int_0^{h/2} e^{ik\eta} d\eta \\ 0 \end{bmatrix} e^{ikx_{j,3/2}}.$$

Therefore, the solutions we are looking for can be expressed as

$$\begin{bmatrix} \bar{u}_{j,1}(t) \\ \bar{u}_{j,2}(t) \end{bmatrix} = \begin{bmatrix} \hat{u}_{k,1}(t) \\ \hat{u}_{k,2}(t) \end{bmatrix} \cdot e^{ikx_{j,3/2}}, \quad (4.15)$$

where

$$\begin{bmatrix} \hat{u}_{k,1}(t) \\ \hat{u}_{k,2}(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{h} \cdot \hat{u}_k(t) \int_{-h/2}^0 e^{ik\eta} d\eta \\ \frac{2}{h} \cdot \hat{u}_k(t) \int_0^{h/2} e^{ik\eta} d\eta \end{bmatrix}.$$

The initial condition can be computed from

$$\begin{bmatrix} \bar{u}_{j,1}(0) \\ \bar{u}_{j,2}(0) \end{bmatrix} = \begin{bmatrix} \frac{2}{h} \cdot \int_{x_{j,1/2}}^{x_{j,3/2}} e^{ix} dx \\ \frac{2}{h} \cdot \int_{x_{j,3/2}}^{x_{j,5/2}} e^{ix} dx \end{bmatrix}, \quad (4.16)$$

by taking the imaginary part. Note that this analysis depends on the assumption of uniform mesh size and periodic boundary conditions. Substituting Eq. (4.15) into Eq. (4.14), we obtain the following evolution equation

$$\begin{bmatrix} \hat{u}_{k,1}'(t) \\ \hat{u}_{k,2}'(t) \end{bmatrix} = G(k, h) \cdot \begin{bmatrix} \hat{u}_{k,1}(t) \\ \hat{u}_{k,2}(t) \end{bmatrix}, \quad (4.17)$$

where the amplification matrix is given by

$$G(k, h) = e^{-ikh} \cdot A + B + e^{ikh} \cdot C. \quad (4.18)$$

The following equation can be used to find the initial condition for (4.17),

$$\begin{bmatrix} \hat{u}_{k,1}(0) \\ \hat{u}_{k,2}(0) \end{bmatrix} = \frac{2}{h} \cdot e^{-ikx_{j,3/2}} \cdot \begin{bmatrix} x_{j,3/2} \\ \int e^{ix} dx \\ x_{j,1/2} \\ x_{j,5/2} \\ \int e^{ix} dx \\ x_{j,3/2} \end{bmatrix}. \quad (4.19)$$

In particular for the low frequency mode $k = 1$, we have

$$\begin{bmatrix} \hat{u}_{1,1}(0) \\ \hat{u}_{1,2}(0) \end{bmatrix} = \frac{2}{h} \cdot \begin{bmatrix} 0 \\ \int e^{ix} dx \\ -h/2 \\ h/2 \\ \int e^{ix} dx \\ 0 \end{bmatrix} = \frac{2}{ih} \cdot \begin{bmatrix} 1 - e^{-ih/2} \\ e^{ih/2} - 1 \end{bmatrix}. \quad (4.20)$$

Let λ_1 and λ_2 be the two eigenvalues of the amplification matrix $G(k, h)$, V_1 and V_2 be the corresponding eigenvectors of $G(k, h)$, the general solution of Eq. (4.17) can be expressed as

$$\begin{bmatrix} \hat{u}_{k,1}(t) \\ \hat{u}_{k,2}(t) \end{bmatrix} = \alpha \cdot e^{\lambda_1 t} V_1 + \beta \cdot e^{\lambda_2 t} V_2. \quad (4.21)$$

By studying the properties of this general solution at the lowest mode ($k = 1$), we can obtain consistency and convergence results; by investigating the boundedness of the

general solution at the high modes (large k), we can establish the stability of the formulations.

4.2.1 Naïve SV Formulation

The naïve SV formulation can be expressed as

$$\begin{cases} \frac{d\bar{u}_{j,1}}{dt} = \frac{2}{h^2}\bar{u}_{j-1,1} - \frac{2}{h^2}\bar{u}_{j-1,2} - \frac{2}{h^2}\bar{u}_{j,1} + \frac{2}{h^2}\bar{u}_{j,2} \\ \frac{d\bar{u}_{j,2}}{dt} = \frac{2}{h^2}\bar{u}_{j,1} - \frac{2}{h^2}\bar{u}_{j,2} - \frac{2}{h^2}\bar{u}_{j+1,1} + \frac{2}{h^2}\bar{u}_{j+1,2} \end{cases}. \quad (4.22)$$

The coefficient matrices A , B , and C are

$$A = \begin{bmatrix} \frac{2}{h^2} & -\frac{2}{h^2} \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{2}{h^2} & \frac{2}{h^2} \\ \frac{2}{h^2} & -\frac{2}{h^2} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 \\ -\frac{2}{h^2} & \frac{2}{h^2} \end{bmatrix}, \quad (4.23)$$

and the amplification matrix $G(k, h)$ is given by

$$G(k, h) = \begin{bmatrix} \frac{2}{h^2}e^{-ikh} - \frac{2}{h^2} & -\frac{2}{h^2}e^{-ikh} + \frac{2}{h^2} \\ -\frac{2}{h^2}e^{ikh} + \frac{2}{h^2} & \frac{2}{h^2}e^{ikh} - \frac{2}{h^2} \end{bmatrix}. \quad (4.24)$$

The eigenvalues and eigenvectors of the amplification matrix $G(k, h)$ are

$$\lambda_1 = -\frac{4}{h^2}(1 - \cos(kh)), \quad \lambda_2 = 0, \quad (4.25)$$

$$V_1 = \begin{bmatrix} e^{-ikh} \\ 1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (4.26)$$

Note that $G(k, h)$ has a zero eigenvalue, which may cause a weak instability for this semi-discrete system.

We first study the lowest mode, i.e., $k = 1$. Applying the initial condition Eq. (4.20), the coefficients α and β can be computed as

$$\alpha = \frac{4(1 - \cos(h/2))}{ih(e^{-ih} - 1)}, \quad \beta = -\frac{2(1 - 2e^{-ih/2} + e^{-ih})}{ih(e^{-ih} - 1)}. \quad (4.27)$$

We therefore have the explicit solution of the SV scheme (4.22) with the initial condition (4.18). For example

$$\bar{u}_{j,1}(t) = (\alpha \cdot e^{\lambda_1 t} \cdot e^{-ih} + \beta \cdot e^{\lambda_2 t}) \cdot e^{ix_{j,3/2}} \quad (4.28)$$

Applying a Taylor expansion assuming small h , we obtain the imaginary part of $\bar{u}_{j,1}(t)$ to be

$$\text{Im}\{\bar{u}_{j,1}(t)\} = \left(\frac{1 + e^{-2t}}{2}\right) \cdot \sin(x_{j,1}) + O(h),$$

where $x_{j,1} = \frac{1}{2}(x_{j,1/2} + x_{j,3/2})$. The solution is about $0.6233\sin(x_{j,1})$ at $t = 0.7$,

which agrees very well with the numerical solution shown in Figure 4.2. We also clearly see that the scheme is not consistent, i.e. the numerical solution does not converge to the solution of the PDE (which equals to $\sin(x)e^{-t}$).

Next we study the stability of the Naïve SV formulation by considering the high modes (large k). When $\cos(kh) = 1$, the amplification matrix $G(k, h) = 0$. Therefore the solution to Eq. (4.17) remains to be the initial solution given by Eq. (4.21). When $\cos(kh) \neq 1$, the

amplification matrix $G(k, h)$ is diagonalizable with the matrix composed of the eigenvectors

$$R = \begin{bmatrix} e^{-khi} & 1 \\ 1 & 1 \end{bmatrix}, \quad (4.29)$$

which has an inverse when $\cos(kh) \neq 1$

$$R^{-1} = \frac{1}{-1 + e^{-khi}} \begin{bmatrix} 1 & -1 \\ -1 & e^{-khi} \end{bmatrix}. \quad (4.30)$$

We can obtain explicitly the solution as

$$\begin{bmatrix} \hat{u}_{k,1}(t) \\ \hat{u}_{k,2}(t) \end{bmatrix} = e^{G(k,h)t} \begin{bmatrix} \hat{u}_{k,1}(0) \\ \hat{u}_{k,2}(0) \end{bmatrix}, \quad (4.31)$$

with

$$e^{G(k,h)t} = R \cdot \begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & 1 \end{bmatrix} \cdot R^{-1}.$$

The L_2 norm of the matrix $e^{G(k,h)t}$ can be computed explicitly, which reveals the stability property. The two eigenvalues of the symmetric matrix $(e^{G(k,h)t})^H (e^{G(k,h)t})$ (where $A^H = (\bar{A})^T$, A^T denotes the transpose of A , and $\bar{A}$ is the conjugate of A) are found to be

$$\Lambda_1 = \{(1-\mu)^2 + \mu\sigma - (1-\mu)\sqrt{(1-\mu)^2 + 2\mu\sigma}\} / \sigma,$$

$$\Lambda_2 = \{(1-\mu)^2 + \mu\sigma + (1-\mu)\sqrt{(1-\mu)^2 + 2\mu\sigma}\} / \sigma,$$

with $\mu = e^{\lambda_1 t}$, $\sigma = 1 - \cos(kh)$. Then the L_2 norm of $e^{G(k,h)t}$ is given by

$\|e^{G(k,h)t}\| = \sqrt{\max(|\Lambda_1|, |\Lambda_2|)}$. If $\|e^{G(k,h)t}\|$ is uniformly bounded with respect to k and

h , (4.32) is said to be stable. However, if we take $\sigma = h^2/t$, we can easily get

$$\Lambda_1 = \{1 - 2 \cdot e^{-4} + e^{-8} - (1 - e^{-4})\sqrt{1 + e^{-8} - 2 \cdot e^{-4} + 2 \cdot e^{-4} h^2/t + e^{-4} h^2/t}\} \cdot t/h^2,$$

$$\Lambda_2 = \{1 - 2 \cdot e^{-4} + e^{-8} + (1 - e^{-4})\sqrt{1 + e^{-8} - 2 \cdot e^{-4} + 2 \cdot e^{-4} h^2/t + e^{-4} h^2/t}\} \cdot t/h^2,$$

and obviously

$$\|e^{G(k,h)t}\| = O(1/h),$$

which is unbounded when $h \rightarrow 0$. Therefore system (4.17) for the naïve SV scheme is unstable.

4.2.2 Local SV Formulation

The local SV formulation can be written as

$$\begin{cases} \frac{d\bar{u}_{j,1}}{dt} = \frac{1}{h^2} \bar{u}_{j-1,1} + \frac{5}{h^2} \bar{u}_{j-1,2} - \frac{12}{h^2} \bar{u}_{j,1} + \frac{4}{h^2} \bar{u}_{j,2} + \frac{3}{h^2} \bar{u}_{j+1,1} - \frac{1}{h^2} \bar{u}_{j+1,2} \\ \frac{d\bar{u}_{j,2}}{dt} = \frac{2}{h^2} \bar{u}_{j,1} - \frac{6}{h^2} \bar{u}_{j,2} + \frac{6}{h^2} \bar{u}_{j+1,1} - \frac{2}{h^2} \bar{u}_{j+1,2} \end{cases} \quad (4.32)$$

The corresponding coefficient matrices A , B , and C are

$$A = \begin{bmatrix} \frac{1}{h^2} & \frac{5}{h^2} \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{12}{h^2} & \frac{4}{h^2} \\ \frac{2}{h^2} & -\frac{6}{h^2} \end{bmatrix}, \quad C = \begin{bmatrix} \frac{3}{h^2} & -\frac{1}{h^2} \\ \frac{6}{h^2} & -\frac{2}{h^2} \end{bmatrix}. \quad (4.33)$$

The amplification matrix, its eigenvalues and eigenvectors are

$$G(k, h) = \begin{bmatrix} \frac{1}{h^2}e^{-ikh} + \frac{3}{h^2}e^{ikh} - \frac{12}{h^2} & \frac{5}{h^2}e^{-ikh} - \frac{1}{h^2}e^{ikh} + \frac{4}{h^2} \\ \frac{6}{h^2}e^{ikh} + \frac{2}{h^2} & -\frac{2}{h^2}e^{ikh} - \frac{6}{h^2} \end{bmatrix} \quad (4.34)$$

$$\lambda_1 = -\frac{16}{h^2}, \quad \lambda_2 = -\frac{2}{h^2}(1 - \cos(kh)) \quad (4.35)$$

$$V_1 = \begin{bmatrix} \frac{-5 + e^{ikh}}{1 + 3e^{ikh}} \\ 1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} \frac{1}{2}(e^{-ikh} + 1) \\ 1 \end{bmatrix}. \quad (4.36)$$

Clearly both eigenvalues are real and negative. To study the accuracy and consistency, we again examine the lowest mode $k = 1$. By applying the initial condition Eq. (4.20), the coefficients α and β in Eq. (4.21) are found to be

$$\alpha = \frac{2e^{-ih/2}(1 + 3e^{ih})(-1 + e^{ih/2})(-2e^{ih} + e^{i3h/2} + e^{ih/2})}{(1 + 14e^{ih} + e^{2ih})ih} \quad (4.37)$$

$$\beta = \frac{4e^{ih/2}(-1 + e^{ih/2})(1 + 3e^{ih} - e^{i3h/2} + 5e^{ih/2})}{(1 + 14e^{ih} + e^{2ih})ih}. \quad (4.38)$$

We thus have the following explicit solution for scheme (4.32)

$$\bar{u}_{j,1}(t) = (\alpha \cdot e^{\lambda_1 t} \cdot \frac{-5 + e^{ih}}{1 + 3e^{ih}} + \beta \cdot e^{\lambda_2 t} \cdot \frac{1}{2}(e^{-ih} + 1)) \cdot e^{ix_{j,3/2}}. \quad (4.39)$$

Applying a Taylor expansion assuming small h , the imaginary part of $\bar{u}_{j,1}(t)$ is found to be

$$\text{Im}\{\bar{u}_{j,1}(t)\} = \sin(x_{j,1})e^{-t} + O(h^2).$$

Clearly, the numerical solution converges to the exact solution with second order accuracy.

The matrix composed of the eigenvectors of $G(k, h)$ is

$$R = \begin{bmatrix} \frac{-5 + e^{ikh}}{1 + 3e^{ikh}} & \frac{1}{2}(e^{-ikh} + 1) \\ 1 & 1 \end{bmatrix}, \quad (4.40)$$

with its inverse given by

$$R^{-1} = \frac{1}{7 + \cos(kh)} \begin{bmatrix} -1 - 3 \cdot e^{ikh} & (1 + e^{-ikh}) \cdot (1 + 3 \cdot e^{ikh}) / 2 \\ 1 + 3 \cdot e^{ikh} & 5 - e^{ikh} \end{bmatrix}. \quad (4.41)$$

In order to prove the stability of scheme (4.32), it is sufficient to show that the L_2 norms of R and R^{-1} are uniformly bounded with respect to k and h since both eigenvalues of $G(k, h)$ are negative. In fact, the eigenvalues of $R^H R$ are

$$\Lambda_1 = (51 + 10 \cdot \cos(kh) + 3 \cdot (\cos(kh))^2 - \sqrt{641 - 716 \cdot \cos(kh) + 30 \cdot (\cos(kh))^2 + 36 \cdot (\cos(kh))^3 + 9 \cdot (\cos(kh))^4}) / (2(10 + 6 \cos(kh)))$$

$$\Lambda_2 = (51 + 10 \cdot \cos(kh) + 3 \cdot (\cos(kh))^2 + \sqrt{641 - 716 \cdot \cos(kh) + 30 \cdot (\cos(kh))^2 + 36 \cdot (\cos(kh))^3 + 9 \cdot (\cos(kh))^4}) / (2(10 + 6 \cos(kh))),$$

and the eigenvalues of $(R^{-1})^H (R^{-1})$ are

$$\omega_1 = (1225 + 1330 \cdot \cos(kh) + 452 \cdot (\cos(kh))^2 + 62 \cdot (\cos(kh))^3 + 3 \cdot (\cos(kh))^4 - (7 + \cos(kh))^2 \cdot \sqrt{745 - 288 \cdot \cos(kh) - 306 \cdot (\cos(kh))^2 + 96 \cdot (\cos(kh))^3 + 9 \cdot (\cos(kh))^4}) / D$$

$$\omega_2 = (1225 + 1330 \cdot \cos(kh) + 452 \cdot (\cos(kh))^2 + 62 \cdot (\cos(kh))^3 + 3 \cdot (\cos(kh))^4 + (7 + \cos(kh))^2 \cdot \sqrt{745 - 288 \cdot \cos(kh) - 306 \cdot (\cos(kh))^2 + 96 \cdot (\cos(kh))^3 + 9 \cdot (\cos(kh))^4}) / D$$

where $D = 2 \cdot (2401 + 1372 \cdot \cos(kh) + 294 \cdot (\cos(kh))^2 + 28 \cdot (\cos(kh))^3 + (\cos(kh))^4)$.

Hence $\|R\| = \sqrt{\max(|\Lambda_1|, |\Lambda_2|)}$ and $\|R^{-1}\| = \sqrt{\max(|\omega_1|, |\omega_2|)}$. It is easy to see that both $\|R\|$

and $\|R^{-1}\|$ are uniformly bounded with respect to kh . Thus the stability of scheme (4.32)

is established.

4.2.3 Penalty SV Formulation

Finally we turn to the analysis of the third formulation, for which we obtain the following scheme based on the linear reconstruction

$$\begin{cases} \frac{d\bar{u}_{j,1}}{dt} = \frac{2}{h^2}(1-\varepsilon)\bar{u}_{j-1,1} - \frac{2}{h^2}(1-3\varepsilon)\bar{u}_{j-1,2} - \frac{2}{h^2}(1+3\varepsilon)\bar{u}_{j,1} + \frac{2}{h^2}(1+\varepsilon)\bar{u}_{j,2} \\ \frac{d\bar{u}_{j,2}}{dt} = \frac{2}{h^2}(1+\varepsilon)\bar{u}_{j,1} - \frac{2}{h^2}(1+3\varepsilon)\bar{u}_{j,2} - \frac{2}{h^2}(1-3\varepsilon)\bar{u}_{j+1,1} + \frac{2}{h^2}(1-\varepsilon)\bar{u}_{j+1,2} \end{cases} \quad (4.42)$$

Choosing $\varepsilon = 1$, the scheme (4.42) reduces to

$$\begin{cases} \frac{d\bar{u}_{j,1}}{dt} = \frac{4}{h^2}\bar{u}_{j-1,2} - \frac{8}{h^2}\bar{u}_{j,1} + \frac{4}{h^2}\bar{u}_{j,2} \\ \frac{d\bar{u}_{j,2}}{dt} = \frac{4}{h^2}\bar{u}_{j,1} - \frac{8}{h^2}\bar{u}_{j,2} + \frac{4}{h^2}\bar{u}_{j+1,1} \end{cases} . \quad (4.43)$$

The coefficient matrices are

$$A = \begin{bmatrix} 0 & \frac{4}{h^2} \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{8}{h^2} & \frac{4}{h^2} \\ \frac{4}{h^2} & -\frac{8}{h^2} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 \\ \frac{4}{h^2} & 0 \end{bmatrix}. \quad (4.44)$$

The amplification matrix $G(k, h)$ is

$$G(k, h) = \begin{bmatrix} -\frac{8}{h^2} & \frac{4}{h^2}e^{-ikh} + \frac{4}{h^2} \\ \frac{4}{h^2}e^{ikh} + \frac{4}{h^2} & -\frac{8}{h^2} \end{bmatrix}. \quad (4.45)$$

The two eigenvalues of $G(k, h)$ are

$$\lambda_1 = -\frac{8}{h^2}(1 + \cos(kh/2)), \quad \lambda_2 = -\frac{8}{h^2}(1 - \cos(kh/2)). \quad (4.46)$$

Clearly both eigenvalues are real and negative. The corresponding eigenvectors are

$$V_1 = \begin{bmatrix} -e^{-ikh/2} \\ 1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} e^{-ikh/2} \\ 1 \end{bmatrix}. \quad (4.47)$$

The coefficients α and β in Eq. (4.21) are

$$\alpha = 0, \quad \beta = \frac{2(e^{ih/2} - 1)}{ih}. \quad (4.48)$$

We thus have the explicit solutions of scheme (4.42). For example

$$\bar{u}_{j,1}(t) = (\alpha \cdot e^{\lambda_1 t} \cdot (-e^{-ih/2}) + \beta \cdot e^{\lambda_2 t} \cdot e^{-ih/2}) \cdot e^{ix_{j,3/2}}. \quad (4.49)$$

Using a Taylor expansion, we obtain the imaginary part of $\bar{u}_{j,1}(t)$ to be

$$\text{Im}\{\bar{u}_{j,1}(t)\} = \sin(x_{j,1})e^{-t} + O(h^2).$$

Clearly, the scheme is consistent and second order accurate. Note that we may take

$\varepsilon > O(h)$ for consistency, but we only have $\text{Im}\{\bar{u}_{j,1}(t)\} = \sin(x_{j,1})e^{-t} + O(h)$ if $\varepsilon \neq 1$,

which is why we suggest $\varepsilon = 1$.

The matrix composed of the eigenvectors (4.47) of $G(k, h)$ is

$$R = \begin{bmatrix} -e^{-ikh/2} & e^{-ikh/2} \\ 1 & 1 \end{bmatrix}, \quad (4.50)$$

with its inverse given by

$$R^{-1} = \frac{1}{2} \begin{bmatrix} -e^{ikh/2} & 1 \\ e^{ikh/2} & 1 \end{bmatrix}. \quad (4.51)$$

Finally the L_2 norms of R and R^{-1} can be computed. They take the following form

$$\|R\| = \sqrt{2} \quad \text{and} \quad \|R^{-1}\| = \sqrt{2}/2.$$

It is clear that both $\|R\|$ and $\|R^{-1}\|$ are uniformly bounded with respect to kh . Thus the stability of scheme (4.42) is established.

4.3 Conclusions

Three different formulations of the spectral volume method are presented for the diffusion equation. Numerical tests and analysis are performed for these formulations. We have found that both the local SV and penalty-like SV formulations are stable and consistent while the naïve SV formulation is neither consistent nor stable. Numerical results agree well with the analysis. It appears that the local SV formulation achieves $(m+1)^{th}$ order accuracy with a degree m polynomial reconstruction, while the penalty SV formulation achieves $(m+1)^{th}$ order accuracy if m is odd, but m^{th} order accuracy if m is even.

Since the local SV formulation achieves the highest accuracy for a given polynomial reconstruction, we will extend it to the 1D/2D convection-diffusion equations in Chapter 5, to the Navier-Stokes equations in Chapter 6.

CHAPTER 5

EXTENSION OF SV METHOD TO CONVECTION-DIFFUSION EQUATIONS

5.1 Formulation for 1D Convection-Diffusion Equation

Consider the following 1D convection-diffusion equation with fixed boundary conditions

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) - \frac{\partial}{\partial x} g\left(u, \frac{\partial u}{\partial x}\right) = 0 \quad \text{in } (a, b) \subset R^1 \quad (5.1a)$$

$$u(a, t) = u_a, \quad u(b, t) = u_b \quad (5.1b)$$

where $f(u)$ and $g(u, \frac{\partial u}{\partial x})$ might be linear or nonlinear continuous functions, u_a and u_b are constant.

Introducing an auxiliary variable $v = \frac{\partial u}{\partial x}$, equation (5.1) becomes

$$v = \frac{\partial u}{\partial x} \quad \text{in } (a, b) \quad (5.2a)$$

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) - \frac{\partial}{\partial x} g(u, v) = 0 \quad \text{in } (a, b) \quad (5.2b)$$

$$u(a, t) = u_a, \quad u(b, t) = u_b. \quad (5.2c)$$

Integrating equations (5.2a-5.2b) on each control volume $C_{i,j}$, which is a sub-cell of a spectral volume $S_i = [x_{i-1/2}, x_{i+1/2}]$ depicted in Figure 5.1, and replacing the flux by the numerical flux, we obtain the following equations for $\bar{u}_{i,j}, \bar{v}_{i,j}$,

$$\left\{ \begin{array}{l} \bar{v}_{i,j} - \frac{1}{h_{i,j}} (\hat{u}|_{i,j+1/2} - \hat{u}|_{i,j-1/2}) = 0 \end{array} \right. \quad (5.3.1)$$

$$\left\{ \begin{array}{l} \frac{d\bar{u}_{i,j}}{dt} + \frac{1}{h_{i,j}} (\hat{f}|_{i,j+1/2} - \hat{f}|_{i,j-1/2}) - \frac{1}{h_{i,j}} (\hat{g}|_{i,j+1/2} - \hat{g}|_{i,j-1/2}) = 0 \end{array} \right. \quad (5.3.2)$$

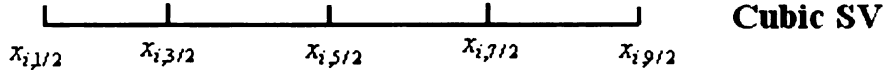
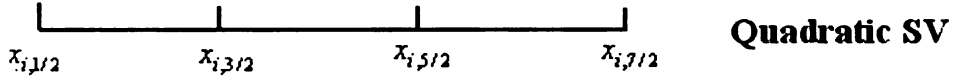
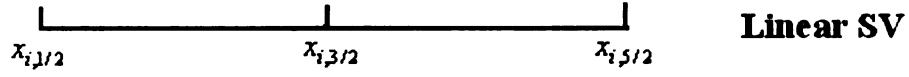


Figure 5.1 One-dimensional spectral volumes

The numerical fluxes are chosen as follows

At internal interfaces:

$$\hat{u}|_{i,j+1/2} = u|_{i,j+1/2}^+ \quad (\text{in (5.3.1)})$$

$$\hat{f}|_{i,j+1/2} = \begin{cases} f(u|_{i,j+1/2}^-) & \text{if } \partial f / \partial u > 0 \\ f(u|_{i,j+1/2}^+) & \text{if } \partial f / \partial u < 0 \end{cases} \quad \hat{g}|_{i,j+1/2} = g(\bar{u}|_{i,j+1/2}, v|_{i,j+1/2}^-) \quad (\text{in (5.3.2)})$$

or

$$\hat{u}|_{i,j+1/2} = u|_{i,j+1/2}^- \quad (\text{in (5.3.1)})$$

$$\hat{f}|_{i,j+1/2} = \begin{cases} f(u|_{i,j+1/2}^-) & \text{if } \partial f / \partial u > 0 \\ f(u|_{i,j+1/2}^+) & \text{if } \partial f / \partial u < 0 \end{cases} \quad \hat{g}|_{i,j+1/2} = g(\bar{u}|_{i,j+1/2}, v|_{i,j+1/2}^+) \quad (\text{in (5.3.2)})$$

where , $\bar{u}|_{i,j+1/2} = (u|_{i,j+1/2}^- + u|_{i,j+1/2}^+)/2$

At $x = a$:

$$\hat{u}|_a = u_a \quad (\text{in (5.3.1)})$$

$$\hat{f}|_a = f(u_a) \quad \hat{g}|_a = g(u_a, v|_a^+) \quad (\text{in (5.3.2)})$$

At $x = b$:

$$\hat{u}|_b = u_b \quad (\text{in the (5.3.1)})$$

$$\hat{f}|_b = f(u_b) \quad \hat{g}|_b = g(u_b, v|_b^-) \quad (\text{in (5.3.2)})$$

If the boundary is not fixed, for example, $\frac{\partial u}{\partial x}|_{x=a} = v_a$ is given instead of $u|_{x=a} = u_a$,

then the calculation of numerical flux at $x = a$ becomes

$$\hat{u}|_a = u|_a^+ \quad (\text{in (5.3.1)})$$

$$\hat{f}|_a = f(u|_a^+) \quad \hat{g}|_a = g(u|_a^+, v_a) \quad (\text{in (5.3.2)})$$

The flux calculation at internal interfaces is the same as before.

5.2 Formulation for 2D Linear Convection-Diffusion Equation

Consider a model second-order convection-diffusion equation with proper boundary conditions

$$u_t + \nabla \bullet (\beta \cdot u) - \nabla \bullet (A \nabla u) = 0 \quad \text{in } \Omega \subset R^2 \quad (5.4a)$$

$$u = f \quad \text{on } \Gamma_D \quad (5.4b)$$

$$(A \nabla u) \bullet n = g \quad \text{on } \Gamma_N \quad (5.4c)$$

where $\beta = (\beta_1, \beta_2)$ is a vector, $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is a matrix, and $n = (n_x, n_y)$ is the unit

outward normal of the boundary of the computational domain Ω .

By introducing auxiliary unknowns $v = \partial u / \partial x$, and $w = \partial u / \partial y$, Eq. (5.4) becomes

$$v = \partial u / \partial x \quad \text{in } \Omega \subset R^2 \quad (5.5a)$$

$$w = \partial u / \partial y \quad \text{in } \Omega \subset R^2 \quad (5.5b)$$

$$u_t + \nabla \bullet (\beta \cdot u) - \nabla \bullet (A(v, w)^T) = 0 \quad \text{in } \Omega \subset R^2 \quad (5.5c)$$

$$u = f \quad \text{on } \Gamma_D \quad (5.5d)$$

$$(A(v, w)^T) \bullet n = g \quad \text{on } \Gamma_N \quad (5.5e)$$

Integrating equations (5.5a-5.5c) on each control volume $C_{i,j}$, we obtain the following

integral form for the CV-averaged mean for $\bar{u}_{i,j}, \bar{v}_{i,j}, \bar{w}_{i,j}$,

$$\bar{v}_{i,j} - \frac{1}{V_{i,j}} \left(\sum_{r=1}^K \int_{A_r} u \cdot n_x dA \right) = 0 \quad (5.6a)$$

$$\bar{w}_{i,j} - \frac{1}{V_{i,j}} \left(\sum_{r=1}^K \int_{A_r} u \cdot n_y dA \right) = 0 \quad (5.6b)$$

$$\frac{d\bar{u}_{i,j}}{dt} + \frac{1}{V_{i,j}} \left\{ \sum_{r=1}^K \int_{A_r} [(\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y] dA - \sum_{r=1}^K \int_{A_r} (A(v, w)^T) \bullet n dA \right\} = 0 \quad (5.6c)$$

Note that we use the same reconstruction for the auxiliary unknowns v, w as that for the original unknown u . We use the following upwind numerical flux

$$u \cdot n_x \approx (u \cdot n_x)^R \quad (5.7)$$

$$u \cdot n_y \approx (u \cdot n_y)^R \quad (5.8)$$

$$(\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y \approx \begin{cases} ((\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y)^L, & \text{if } \beta \bullet n > 0 \\ ((\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y)^R, & \text{if } \beta \bullet n < 0 \end{cases} \quad (5.9)$$

$$(A(v, w)^T) \bullet n \approx ((A(v, w)^T) \bullet n)^L, \quad (5.10)$$

and for boundary faces

$$u \cdot n_x \approx \begin{cases} u \cdot n_x, & \text{on } \Gamma_D \\ (u \cdot n_x)^L, & \text{on } \Gamma_N \end{cases} \quad (5.11)$$

$$u \cdot n_y \approx \begin{cases} u \cdot n_y, & \text{on } \Gamma_D \\ (u \cdot n_y)^L, & \text{on } \Gamma_N \end{cases} \quad (5.12)$$

$$(\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y \approx \begin{cases} ((\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y)^L, & \text{if } \beta \bullet n > 0 \\ ((\beta_1 u)_x \cdot n_x + (\beta_2 u)_y \cdot n_y)^R, & \text{if } \beta \bullet n < 0 \end{cases} \quad (5.13)$$

$$(A(v, w)^T) \bullet n \approx \begin{cases} ((A(v, w)^T) \bullet n)^L, & \text{on } \Gamma_D \\ (A(v, w)^T) \bullet n = g, & \text{on } \Gamma_N \end{cases} \quad (5.14)$$

5.3 Accuracy study

Extensive accuracy studies were carried out for both 1D and 2D convection and diffusion equations. In 1D, both linear and non-linear equations are employed, and problems with exact solutions are designed to test the local spectral volume (LSV) approach. These accuracy studies are presented next.

5.3.1 1D linear Convection-Diffusion Equation

The following linear equation is solved with SV schemes of various orders

$$u_t + c \cdot u_x - \mu \cdot u_{xx} = 0$$

subject to the initial condition of $u(x,0) = \sin(x)$ and periodic boundary condition. The computational domain is $[-\pi, \pi]$. The exact solution is $u(x,t) = e^{-\pi^2 \mu t} \sin(\pi(x - ct))$.

The numerical simulation was carried out until $t = 1$. The L_1 and L_∞ errors are presented in Table 5.1 for $c = 1.0$, $\mu = 1.0$. Note that the LSV approach is capable of achieving the optimum orders of accuracy in all cases.

Table 5.1 L_1 and L_∞ errors and orders of accuracy on 1D linear convection-diffusion equation (at $t = 1.0$)

Order of accuracy	nTCell	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10	7.60e-03	--	9.97e-03	--
	20	2.07e-03	1.88	2.89e-03	1.79
	40	5.46e-04	1.92	7.86e-04	1.88
	80	1.40e-04	1.96	2.04e-04	1.95
3	10	3.83e-04	--	5.37e-04	--
	20	4.55e-05	3.07	6.64e-05	3.02
	40	5.57e-06	3.03	8.16e-06	3.02
	80	6.89e-07	3.02	1.01e-06	3.01
4	10	1.24e-05	--	1.85e-05	--
	20	7.92e-07	3.97	1.17e-06	3.98
	40	5.01e-08	3.98	7.43e-08	3.98
	80	3.15e-09	3.99	4.69e-09	3.99

5.3.2 1D Viscous Burger's Equation

Consider

$$u_t + u \cdot u_x - \mu u_{xx} = 0, \quad \mu = 0.1, \quad x \in (0,1)$$

with the following initial condition $u(x,0) = -\tanh(\frac{x}{2\mu})$ and boundary condition

$$u(0,t) = 0, \quad u(1,t) = -\tanh\left(\frac{1}{2\mu}\right).$$

The problem has the following exact solution

$$u(x,t) = -\tanh\left(\frac{x}{2\mu}\right).$$

The simulation is conducted until $t = 1.0$ with various SV schemes. The L_1 and L_∞ errors are presented in Table 5.2. Note that the LSV approach is again capable of achieving the optimum orders of accuracy in all cases.

5.3.3 1D Fully Nonlinear Case

Further, we consider the following fully nonlinear equation

$$u_t + u \cdot u_x - \frac{1}{2}(u \cdot u_x)_x = 0 \quad x \in (0,1)$$

with initial and boundary conditions as $u(x,0) = e^x$; $u(0,t) = 1$, $u(1,t) = e$. The problem has the following exact solution $u(x,t) = e^x$.

The simulation was conducted until $t = 1.0$ with various SV schemes. The L_1 and L_∞ errors are presented in Table 5.3. Note that the LSV approach is again capable of achieving the optimum order of accuracy in all cases.

Table 5.2 L_1 and L_∞ errors and orders of accuracy on 1D viscous Burger's equation
(at $t = 1.0$)

Order of accuracy	nTCell	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10	2.12e-03	--	1.22e-02	--
	20	6.16e-04	1.78	4.30e-03	1.50
	40	1.73e-04	1.83	1.33e-03	1.69
	80	4.76e-05	1.86	3.66e-04	1.86
3	10	4.84e-04	--	3.43e-03	--
	20	7.62e-05	2.67	5.25e-04	2.71
	40	1.04e-05	2.87	7.06e-05	2.89
	80	1.36e-06	2.93	9.12e-06	2.95
4	10	3.17e-05	--	2.16e-04	--
	20	1.49e-06	4.41	9.53e-06	4.50
	40	7.31e-08	4.35	8.62e-07	3.47
	80	4.22e-09	4.11	6.20e-08	3.80

Table 5.3 L_1 and L_∞ errors and orders of accuracy on the 1D fully nonlinear convection-diffusion equation ($t = 1.0$)

Order of accuracy	nTCell	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10	9.52e-04	--	2.1e-03	--
	20	2.53e-04	1.91	6.12e-04	1.78
	40	6.52e-05	1.96	1.65e-04	1.89
	80	1.65e-05	1.98	4.27e-05	1.95
3	10	9.32e-06	--	2.92e-05	--
	20	1.22e-06	2.93	3.99e-06	2.87
	40	1.56e-07	2.97	5.21e-07	2.94
	80	1.97e-08	2.99	6.65e-08	2.97
4	10	6.30e-08	--	2.37e-07	--
	20	4.07e-09	3.95	1.64e-08	3.85
	40	2.59e-10	3.97	1.07e-09	3.94
	80	1.63e-11	3.99	6.81e-11	3.97

5.3.4 2D Linear Convection-Diffusion Equation

We also tested the LSV method on a 2D linear equation written as

$$u_t + c(u_x + u_y) - \mu(u_{xx} + u_{yy}) = 0, \quad (x, y) \in (-1, 1) \times (-1, 1); \quad c = 1, \mu = 0.01$$

with the initial condition $u(x, y, 0) = \sin(\pi(x + y))$ and periodic boundary condition. The exact solution is $u(x, y, t) = e^{-2\pi^2 \mu t} \sin(\pi(x + y - 2ct))$. The recorded L_1 and L_∞ errors in Table 5.4 again show that the LSV approach is capable of achieving the optimum orders of accuracy in all cases.

Table 5.4 L_1 and L_∞ errors and orders of accuracy on the 2D linear convection-diffusion equation ($t = 1.0$)

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10x10x2	3.73e-02	--	5.35e-02	--
	20x20x2	8.60e-03	2.12	1.28e-02	2.06
	40x40x2	2.16e-03	1.99	3.25e-03	1.98
	80x80x2	5.36e-04	2.01	8.11e-04	2.00
3	10x10x2	2.79e-03	--	3.91e-03	--
	20x20x2	3.40e-04	3.04	5.10e-04	2.94
	40x40x2	4.16e-05	3.03	6.31e-05	3.01
	80x80x2	5.21e-06	3.00	7.97e-06	2.98
4	10x10x2	4.14e-05	--	5.72e-05	--
	20x20x2	2.47e-06	4.07	3.37e-06	4.09
	40x40x2	1.47e-07	4.07	1.95e-07	4.11
	80x80x2	8.45e-09	4.12	1.14e-08	4.10

5.4 Conclusions

We've extended the local spectral volume method to the 1D and 2D scalar convection-diffusion equations. Accuracy studies with 1D linear, viscous Burger's, fully nonlinear and 2D linear cases have been carried out, and the order of accuracy claim has been numerically verified, i.e. the local spectral volume method achieve 2nd, 3rd and 4th order of accuracy for linear, quadratic and cubic reconstructions.

CHAPTER 6

EXTENSION OF SV METHOD TO THE NAVIER-STOKES EQUATIONS

6.1 Formulation for Navier-Stokes Equations

We consider the two-dimensional Navier-Stokes equations written in conservation form

$$\frac{\partial Q}{\partial t} + \nabla \cdot \vec{F}_e(Q) - \nabla \cdot \vec{F}_v(Q, \nabla Q) = 0 \quad (6.1a)$$

in the computational domain Ω subject to suitable initial and boundary conditions. The conservative variables Q and the Cartesian components $f_e(Q)$ and $g_e(Q)$ of the inviscid flux vector $\vec{F}_e(Q)$ are given by

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix}, \quad f_e(Q) = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{Bmatrix}, \quad g_e(Q) = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{Bmatrix}. \quad (6.1b)$$

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

$$E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho (u^2 + v^2) \quad (6.1c)$$

with ratio of specific heats γ , which is taken to be 1.4 in all the simulations in this research. The Cartesian components $f_v(Q, \nabla Q)$ and $g_v(Q, \nabla Q)$ of the viscous flux vector $\vec{F}_v(Q, \nabla Q)$ are given by

$$f_v(Q, \nabla Q) = \mu \cdot \left\{ \begin{array}{c} 0 \\ 2u_x + \lambda(u_x + v_y) \\ v_x + u_y \\ u[2u_x + \lambda(u_x + v_y)] + v(v_x + u_y) + \frac{C_p}{P_r} T_x \end{array} \right\} \quad (6.1d)$$

$$g_v(Q, \nabla Q) = \mu \cdot \left\{ \begin{array}{c} 0 \\ v_x + u_y \\ 2v_y + \lambda(u_x + v_y) \\ u(v_x + u_y) + v[2v_y + \lambda(u_x + v_y)] + \frac{C_p}{P_r} T_y \end{array} \right\} \quad (6.1e)$$

where μ is the dynamic viscosity, C_p is the specific heat at constant pressure, P_r is the Prandtl number, T is the temperature and using the Stokes hypothesis, $\lambda = -2/3$.

Integrating Eq. (6.1a) in $C_{i,j}$, we obtain the following integral equation for the CV-averaged mean

$$\frac{d\bar{Q}_{i,j}}{dt} + \frac{1}{V_{i,j}} \sum_{r=1}^K \int_{A_r} \bar{F}(Q, \nabla Q) \cdot \mathbf{n} dA = 0 \quad (6.2)$$

where $\bar{Q}_{i,j}$ is the vector of the CV-averaged conservative variables in $C_{i,j}$, K is the number of faces in $C_{i,j}$, and A_r represents the r -th face of $C_{i,j}$,

$\bar{F}(Q, \nabla Q) = \bar{F}_e(Q) - \bar{F}_v(Q, \nabla Q)$. We treat the gradient of the conservative variables $\nabla Q = G(Q)$ as auxiliary unknowns of the Navier-Stokes equations following [7,10,19], which are therefore reformulated as the following coupled system for the unknowns G and Q ,

$$G - \nabla Q = 0 \quad (6.3a)$$

$$\frac{\partial Q}{\partial t} + \nabla \bullet \vec{F}_e(Q) - \nabla \bullet \vec{F}_v(Q, G) = 0 \quad (6.3b)$$

Integrating system (6.3) over each control volume $C_{i,j}$, we obtain the following integral form for the CV-averaged mean $\bar{G}_{i,j}$ and $\bar{Q}_{i,j}$,

$$\bar{G}_{i,j} - \frac{1}{V_{i,j}} \left(\sum_{r=1}^K \int_{A_r} Q \bullet \mathbf{n} dA \right) = 0 \quad (6.4a)$$

$$\frac{d\bar{Q}_{i,j}}{dt} + \frac{1}{V_{i,j}} \left(\sum_{r=1}^K \int_{A_r} \vec{F}_e(Q) \bullet \mathbf{n} dA - \sum_{r=1}^K \int_{A_r} \vec{F}_v(Q, G) \bullet \mathbf{n} dA \right) = 0. \quad (6.4b)$$

Note that at the SV boundaries, reconstructed solutions Q and G are not continuous. It is necessary to substitute those fluxes with interface numerical fluxes. In this research, we use the following numerical fluxes for internal interfaces:

$$Q \bullet \mathbf{n} \approx Q^L \bullet \mathbf{n} \quad (6.5a)$$

$$\vec{F}_e(Q) \bullet \mathbf{n} \approx \text{Roe flux splitting} \quad (6.5b)$$

$$\vec{F}_v(Q, G) \bullet \mathbf{n} \approx \vec{F}_v(\bar{Q}, G^R) \bullet \mathbf{n} \quad (6.5c)$$

where $\bar{Q} = (Q^L + Q^R)/2$. Combining the Gauss quadrature formulae and numerical fluxes, all the integrals appearing in equations (6.4a-6.4b) can be evaluated. An important issue is that the reconstructions for the auxiliary variable G have the same structure as the one employed for the original conservative variable Q . For boundary faces, we borrow the idea of the reference [7], but the form is different.

For inviscid flux:

$$\vec{F}_e(Q) \bullet \mathbf{n} = \begin{bmatrix} \rho \vec{V} \bullet \mathbf{n} \\ \rho u \vec{V} \bullet \mathbf{n} + p n_x \\ \rho v \vec{V} \bullet \mathbf{n} + p n_y \\ \rho (E + p) \vec{V} \bullet \mathbf{n} \end{bmatrix} \quad (6.6)$$

On both inviscid and no-slip surfaces, the normal velocity $\vec{V} \bullet \mathbf{n}$ vanishes. The flux is equal to the pressure contribution of the inviscid flux function in the normal direction to the surface, with the pressure p being taken from the internal boundary state.

At inflow (outflow), $\vec{F}_e(Q) \bullet \mathbf{n} \approx \vec{F}_e(Q_b) \bullet \mathbf{n}$, Q_b is computed by imposing the available data and the Riemann invariant associated to outgoing characteristics.

For the flux in the auxiliary equation:

$$Q \bullet \mathbf{n} \approx Q_b \bullet \mathbf{n}$$

where Q_b has the same value as that for the inviscid flux.

For the viscous boundary flux:

On the no-slip surface, $\vec{F}_v(Q, G) \bullet \mathbf{n} \approx \vec{F}_v(Q_b, G_b) \bullet \mathbf{n}$, where Q_b is that for the inviscid flux; $G_b = G^L$ if there are no boundary conditions on $\nabla Q \bullet \mathbf{n}$. When such conditions are instead prescribed, the value of G_b is modified accordingly.

At inflow (outflow), $\vec{F}_v(Q, G) \bullet \mathbf{n} \approx \vec{F}_v(Q^L, G^L) \bullet \mathbf{n}$

6.2 Numerical Experiments

6.2.1 The Couette Flow

To verify the formal accuracy of the local SV method on Navier-Stokes equations, we consider the Couette flow between two parallel walls. The lower is stationary with temperature T_0 and the upper is moving at speed of U with temperature T_1 . The distance between the two walls is H .

The steady analytic solution is

$$u = \frac{U}{H} y, \quad v = 0$$

$$T = T_0 + \frac{y}{H} \cdot (T_1 - T_0) + \frac{\mu \cdot U^2}{2k} \cdot \frac{y}{H} \cdot \left(1 - \frac{y}{H}\right)$$

$$p = \text{const } t, \quad \rho = \frac{p}{R \cdot T}$$

The parameters were chosen as $U = 1.0$, $H = 2.0$, $T_0 = 0.8$, $T_1 = 0.85$, $\mu = 0.01$. The formulation described above was tested with this problem on the two dimensional domain $[0,2] \times [0,2]$. The computational grid is displayed as Figure 6.1.

Convergence test: When the simulation was started with the following initial condition:

$$u = 0, v = 0, p = 1, T = 1.$$

Figure 6.2 shows that the numerical solution converges to the steady analytical solution.

Error estimation 1: The simulation was started with the steady solution. We record the L_1 and L_∞ errors of the numerical solution away from the steady solution until the

residuals approach to the *machine zero*. Tables 6.1 and 6.2 demonstrate the L_1 and L_∞ errors for various reconstructions with mesh refinements in density and temperature in this sense.

Error estimation 2: The simulation was also started with the steady solution. We record the L_1 and L_∞ errors of the numerical solution away from the steady solution until the physical time $t=1.0$ for all reconstructions and grids. The numerical results are displayed in Tables 6.3 and 6.4.

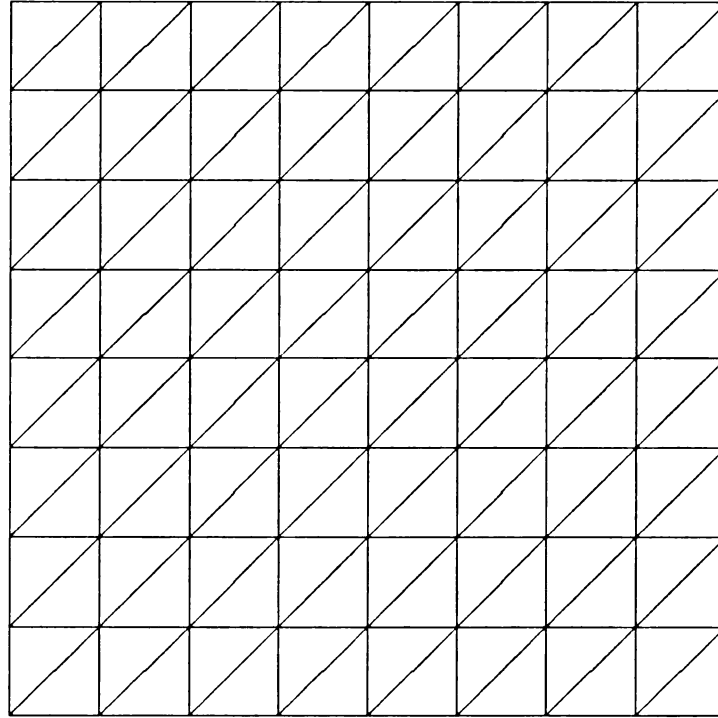


Figure 6.1 Computational grid for Couette flow case

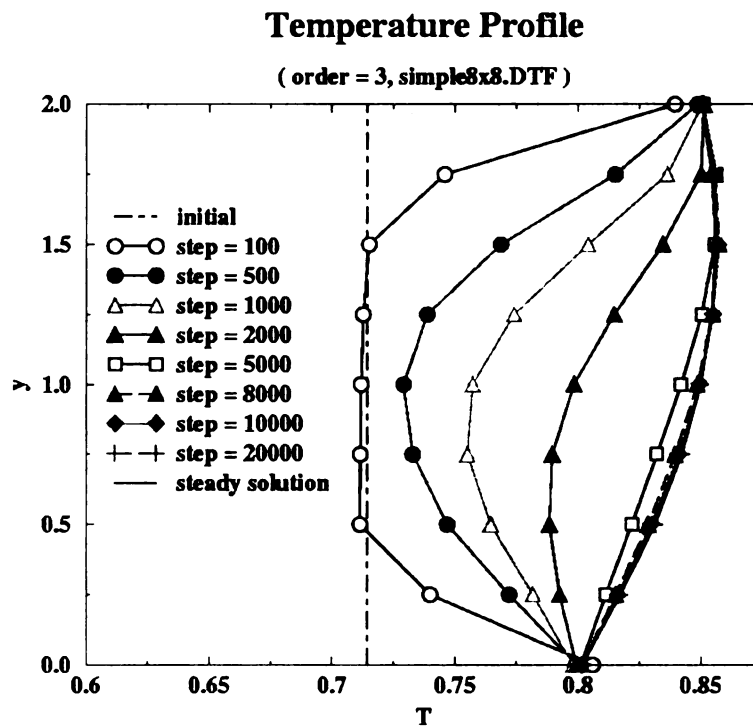
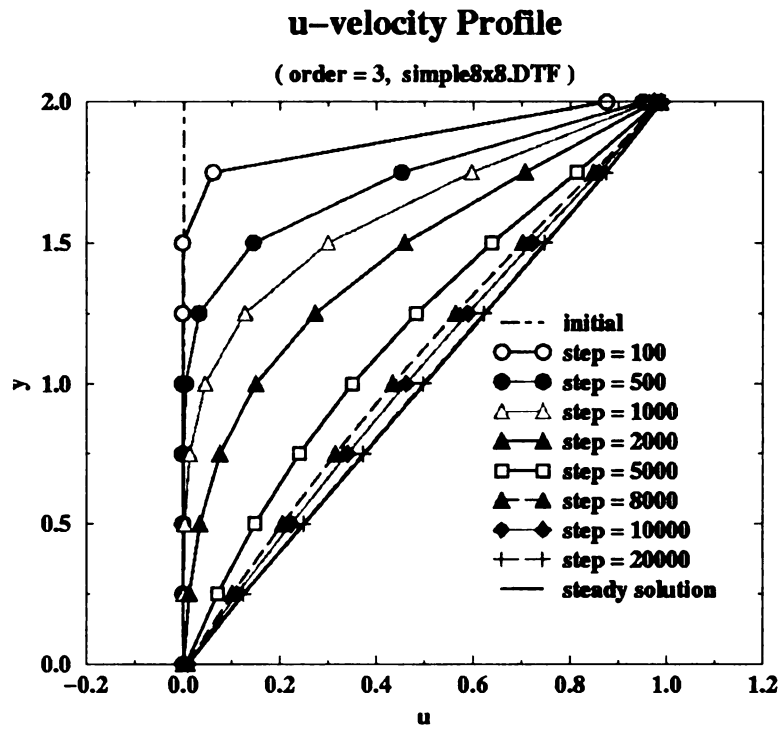


Figure 6.2 Convergence trend of numerical solution to the steady exact solution in Couette flow test

Table 6.1 L_1 and L_∞ errors and orders of accuracy on Couette flow
(Density in Estimation 1)

Accuracy of order	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	2x2x2	1.5279e-02	---	2.4263e-02	---
	4x4x2	4.0045e-03	1.93	8.6840e-03	1.48
	8x8x2	1.0767e-03	1.90	3.2561e-03	1.42
	16x16x2	3.0054e-04	1.84	1.3237e-03	1.30
3	2x2x2	1.5493e-04	---	6.3243e-04	---
	4x4x2	3.8310e-05	2.02	1.2701e-04	2.32
	8x8x2	7.5935e-06	2.33	2.1812e-05	2.54
	16x16x2	1.3064e-06	2.54	3.4697e-06	2.65
4	2x2x2	4.9086e-05	--	9.4826e-05	---
	4x4x2	4.0150e-06	3.61	8.1247e-06	3.54
	8x8x2	3.2535e-07	3.63	7.6764e-07	3.40
	16x16x2	2.6444e-08	3.62	1.0771e-07	2.83

Table 6.2 L_1 and L_∞ errors and orders of accuracy on Couette flow
(Temperature in Estimation 1)

Accuracy of order	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	2x2x2	3.2154e-03	---	9.1184e-03	---
	4x4x2	8.6193e-04	1.90	2.2878e-03	1.99
	8x8x2	2.1349e-04	2.01	8.9539e-04	1.35
	16x16x2	5.1577e-05	2.05	4.6119e-04	0.96
3	2x2x2	4.0687e-05	---	1.7787e-04	---
	4x4x2	5.4795e-06	2.89	1.9263e-05	3.21
	8x8x2	8.2204e-07	2.74	2.5200e-06	2.93
	16x16x2	1.1357e-07	2.86	4.2090e-07	2.58
4	2x2x2	5.5108e-06	---	1.9615e-05	---
	4x4x2	3.9574e-07	3.80	1.4941e-06	3.71
	8x8x2	2.6944e-08	3.88	1.3252e-07	3.49
	16x16x2	2.1046e-09	3.68	1.1550e-07	0.2

Table 6.3 L_1 and L_∞ errors and orders of accuracy on Couette flow
(Density in Estimation 2)

Accuracy of order	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	2x2x2	5.6506e-03	---	1.2276e-02	---
	4x4x2	1.7948e-03	1.65	4.3519e-03	1.50
	8x8x2	4.7120e-04	1.93	1.4161e-03	1.62
	16x16x2	1.1585e-04	2.02	5.4656e-04	1.37
3	2x2x2	9.1819e-05	---	4.3684e-04	---
	4x4x2	1.7355e-05	2.40	9.2720e-05	2.24
	8x8x2	2.8273e-06	2.62	1.7454e-05	2.41
	16x16x2	4.0887e-07	2.79	2.8176e-06	2.63
4	2x2x2	2.0969e-05	---	5.4657e-05	---
	4x4x2	1.5542e-06	3.75	5.1630e-06	3.40
	8x8x2	1.0483e-07	3.89	4.5225e-07	3.51
	16x16x2	6.7874e-09	3.95	4.4748e-08	3.34

Table 6.4 L_1 and L_∞ errors and orders of accuracy on Couette flow
(Temperature in Estimation 2)

Accuracy of order	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	2x2x2	1.7303e-03	---	5.0913e-03	---
	4x4x2	5.2609e-04	1.72	1.6907e-03	1.59
	8x8x2	1.4576e-04	1.85	4.2863e-04	1.98
	16x16x2	3.7475e-05	1.96	1.5568e-04	1.46
3	2x2x2	2.6304e-05	---	7.9946e-05	---
	4x4x2	3.5649e-06	2.88	1.3643e-05	2.55
	8x8x2	5.2462e-07	2.76	1.8414e-06	2.89
	16x16x2	7.5620e-08	2.79	2.6495e-07	2.80
4	2x2x2	4.7828e-06	---	1.5984e-05	---
	4x4x2	3.3793e-07	3.82	1.2703e-06	3.65
	8x8x2	2.3936e-08	3.82	9.6706e-08	3.72
	16x16x2	1.5990e-09	3.90	6.6968e-09	3.85

6.2.2 Laminar Flow along a Flat Plate

We consider the laminar flow on an adiabatic flat plate characterized by a free stream Mach number $Ma = 0.3$ and by a Reynolds number based on the free stream condition and on the plate length $Re = 10000$. The length of the plate is set to $L = 1.0$ as shown in Figure 6.3. At $x = 1.0$, based on the Blasius solution, the thickness of the boundary layer δ is

$$\delta|_{x=1.0} = 5 \cdot \sqrt{\frac{\mu \cdot x}{\rho_{\infty} \cdot u_{\infty}}} \Big|_{x=1.0} = 5 \cdot \frac{x}{\sqrt{Re}|_{x=1.0}} = 0.05$$

Therefore, the size of computational domain in y -direction is chosen to be 20 times of the boundary layer thickness at $x = 1.0$ such that the flow at the top boundary is nearly inviscid. The range of computational domain in x -direction is $[-1, 1]$. Figure 6.3 shows the computational domain. The red point $(0, 0)$ indicates the leading edge of the flat plate. Adiabatic wall boundary is used on the plate surface. From $(-1, 0)$ to $(0, 0)$, a symmetry boundary condition is applied. At the inlet, the free stream condition is imposed. At the top and exit boundaries, the static pressure is fixed.

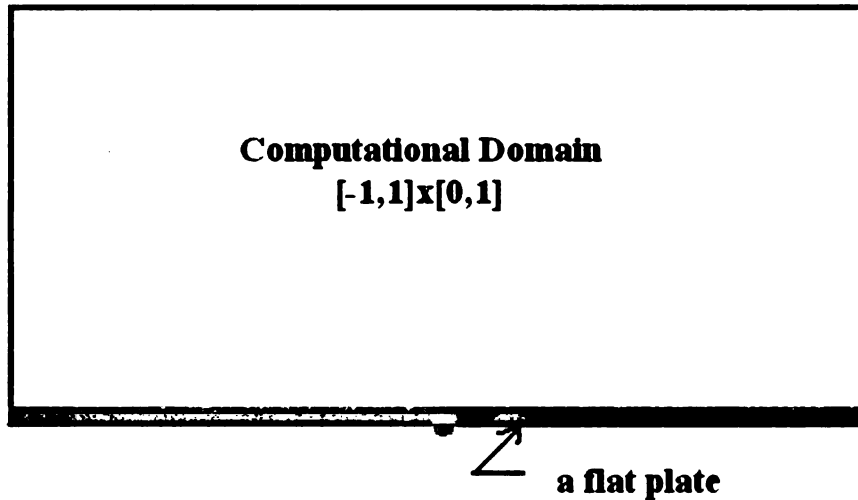
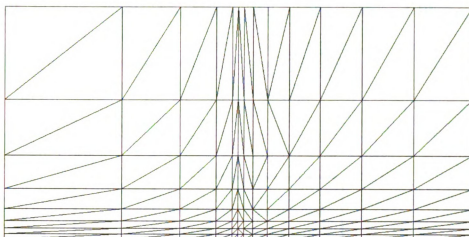


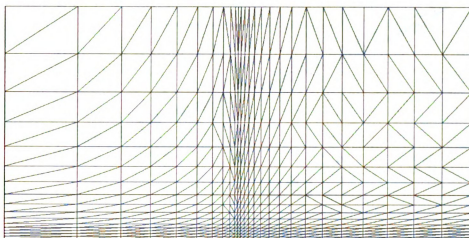
Figure 6.3 Computational domains for the flat plate case
(this figure presented by color)

Since the singularity around the leading edge of the flat plate, we employ cluster meshes both in x -direction and y -direction. The computations have been performed on three triangular meshes, coarse mesh (208 cells) with 8 cells along the plate, medium mesh (832 cells) with 16 cells along the plate, and fine mesh (3328 cells) with 32 cells along the plate.

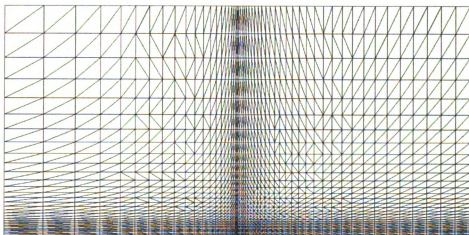
In y -direction of the coarse mesh, we set two cells within the boundary layer, and do a proper transmission to the rest. In x -direction we set the proper clustering factor such that the grids almost have the same length in x and y direction around the leading edge of the plate. Figure 6.4 shows three different meshes used in our numerical simulations.



(a) Coarse mesh



(b) Medium mesh



(c) Fine mesh

Figure 6.4 Meshes for the flat plate case

Figure 6.5 shows the profiles based on the numerical solution and Blasius solution.

Where u is the velocity in x -direction, and $y^* = y \cdot \sqrt{\frac{\rho_\infty \cdot U_\infty}{\mu \cdot x}}$.

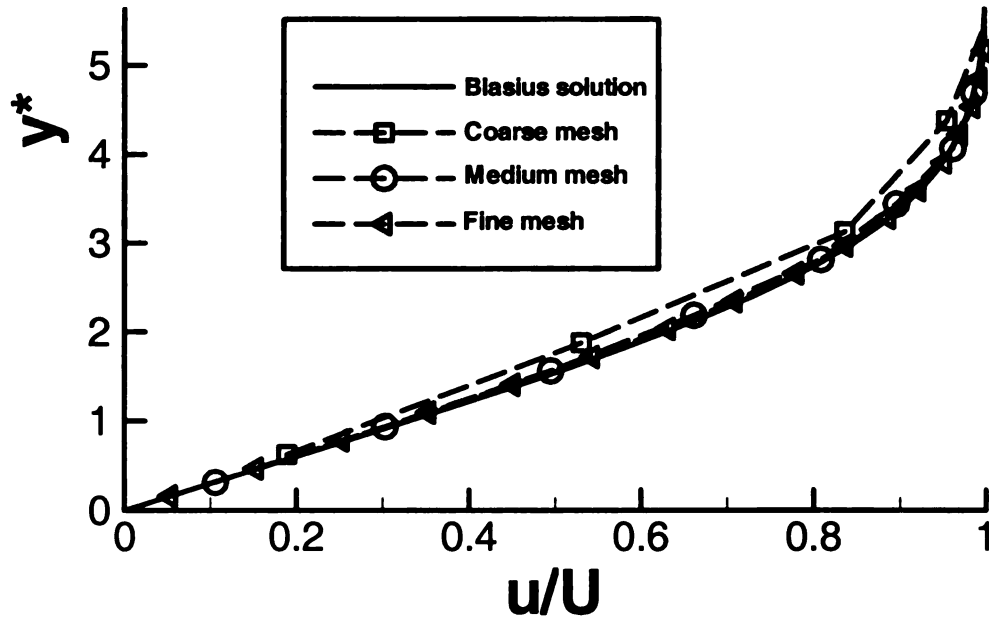


Figure 6.5a u -velocity profiles for the flat plate case with linear SV

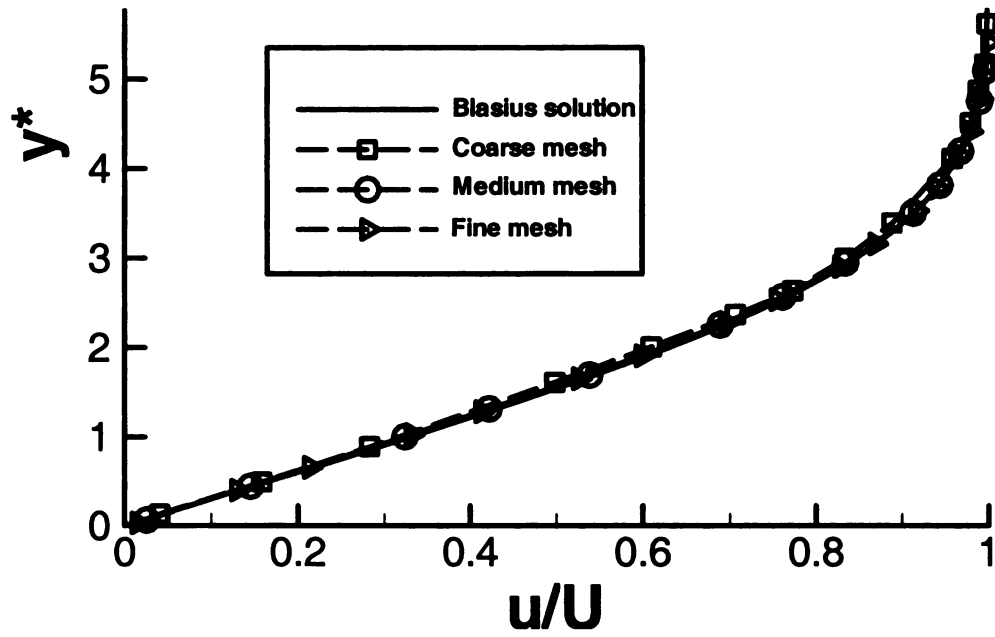


Figure 6.5b u -velocity profiles for the flat plate case with quadratic SV

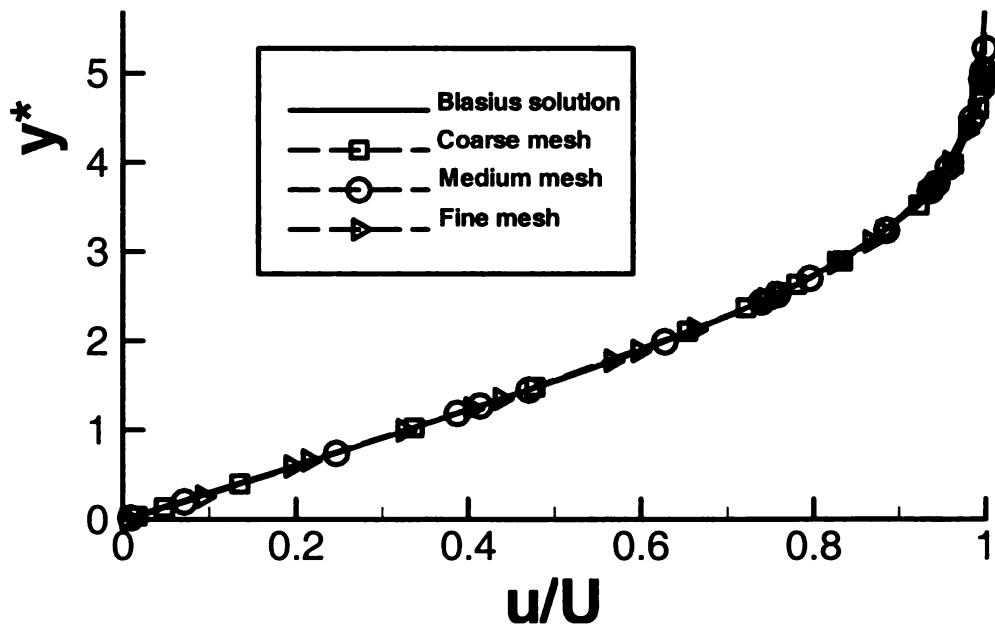


Figure 6.5c u -velocity profiles for the flat plate case with cubic SV

Furthermore, we compute skin friction coefficient C_f by both numerical solution and the well-known Blasius formula for the C_f distribution along a flat plate in the case of incompressible flow in Figure 6.6. The computed results show a good agreement with the Blasius solution. Also, the residual history plot in Figure 6.7 shows that we achieve the steady state solution numerically.

Numerical solution:

$$C_f = \frac{\tau_w}{\frac{1}{2} \rho_\infty \cdot U_\infty^2}$$

Blasius solution:

$$C_f = \frac{0.664}{\sqrt{\text{Re}_x}}$$

where, shear stress $\tau_w = \mu \cdot \left. \frac{\partial u}{\partial y} \right|_{y=0}$, Reynolds number $\text{Re}_x = \frac{\rho_\infty \cdot U_\infty \cdot x}{\mu}$, U_∞ is the

speed of the free stream, and x is the distance away from the leading edge of the plate.

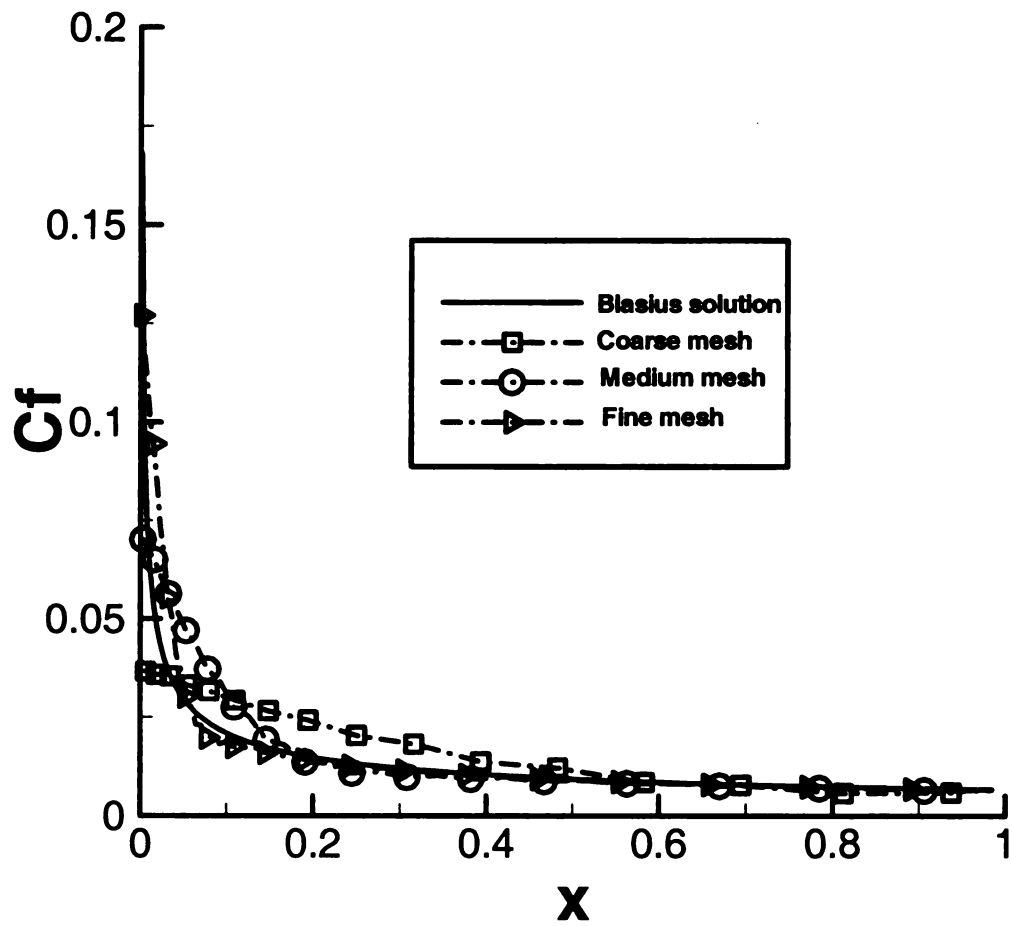


Figure 6.6a Skin friction coefficient along a flat plate with linear SV

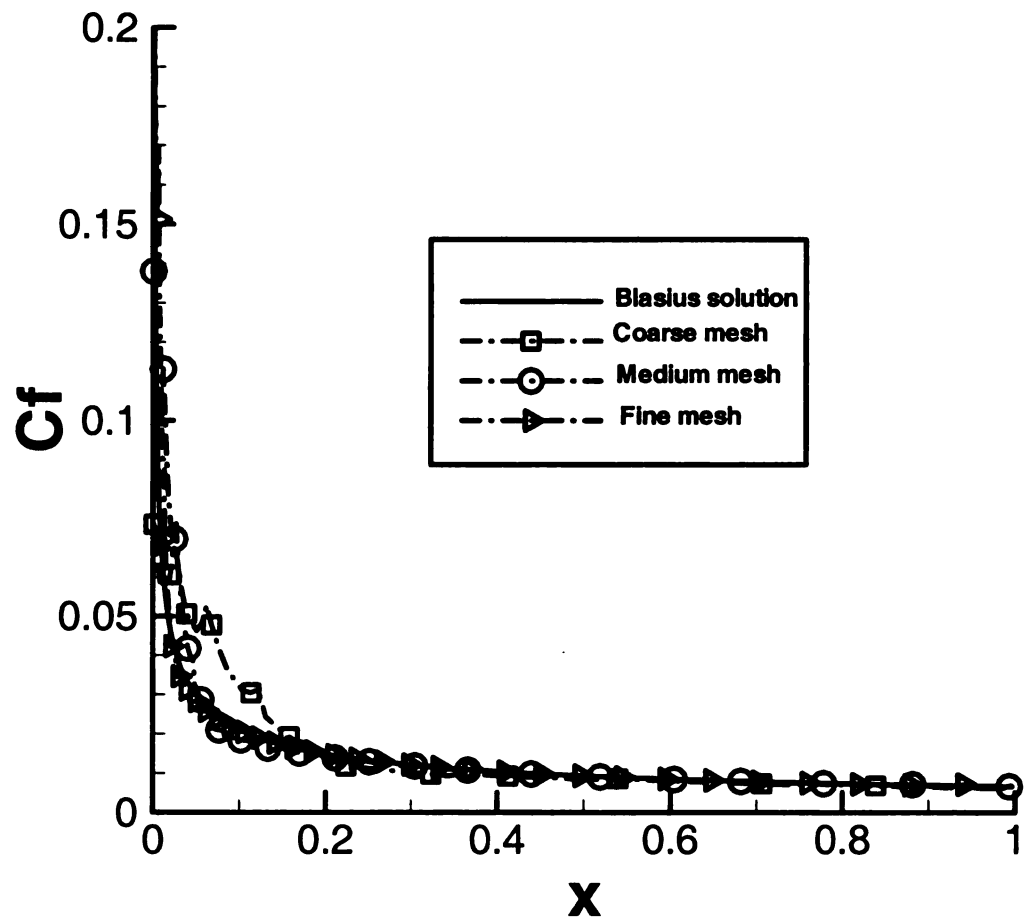


Figure 6.6b Skin friction coefficient along a flat plate with quadratic SV

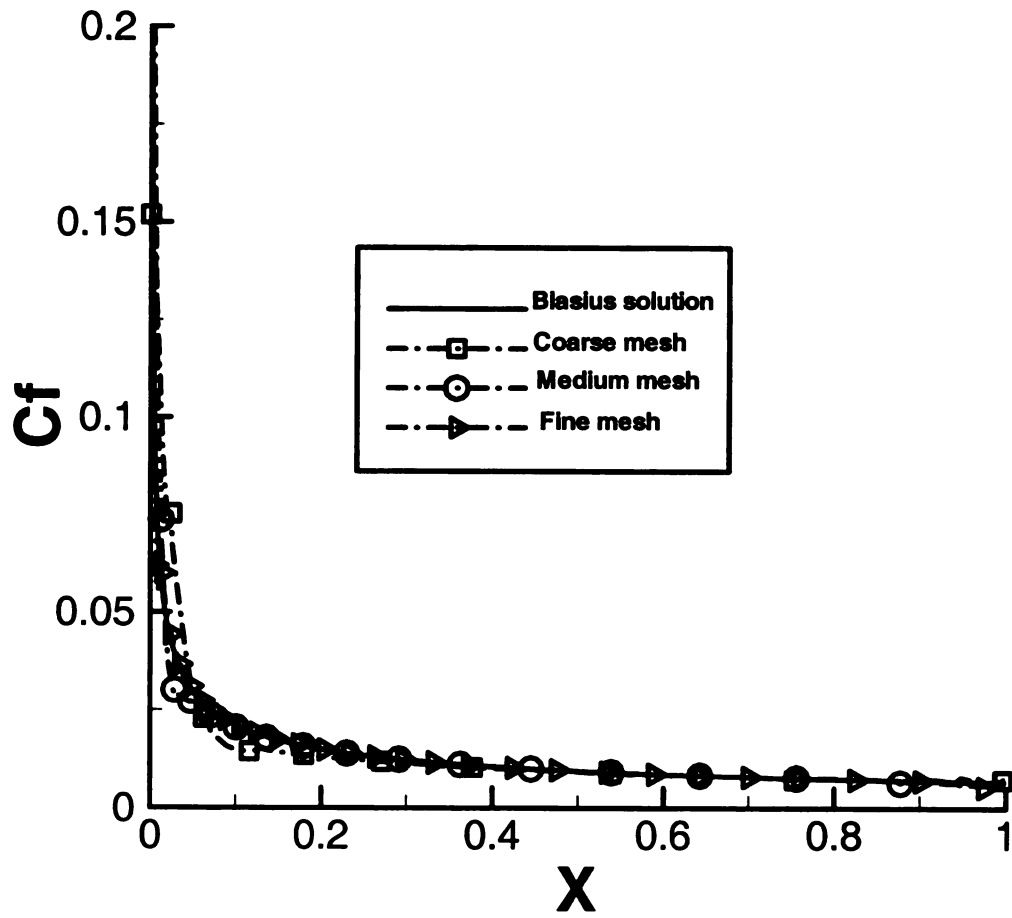


Figure 6.6c Skin friction coefficient along a flat plate with cubic SV

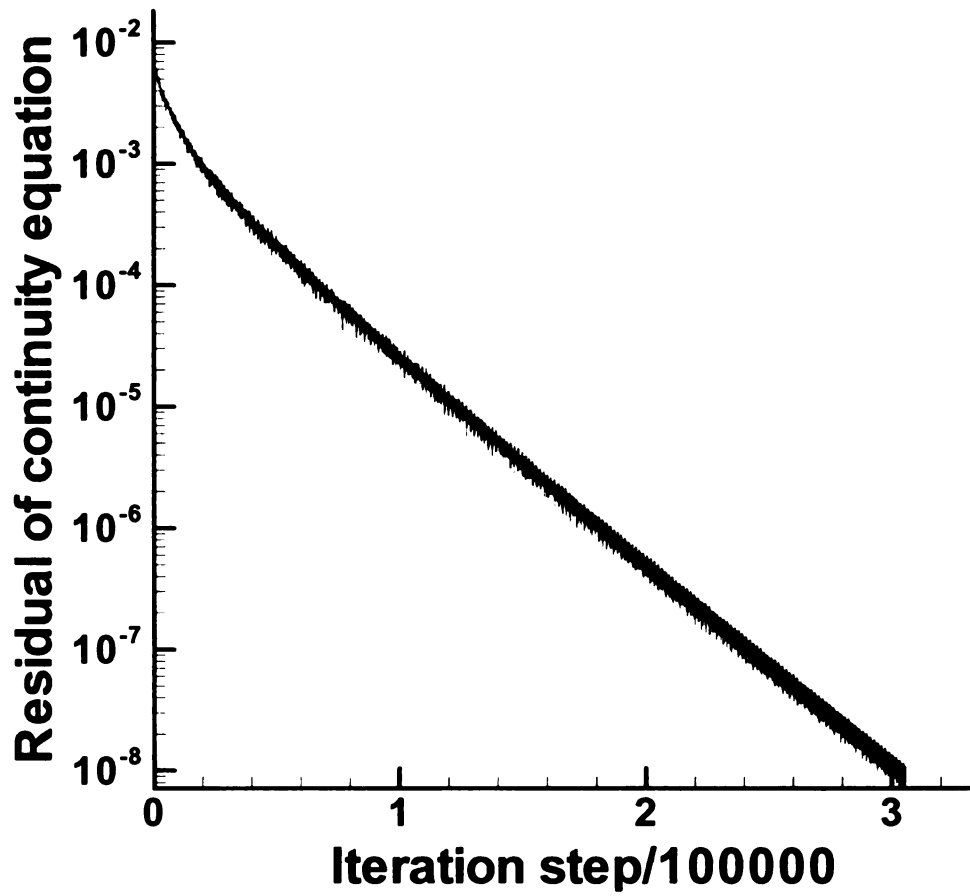


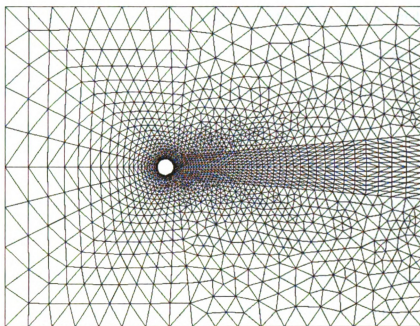
Figure 6.7 Residual history of continuity equation with quadratic SV

6.2.3 Subsonic Flow over a Circular Cylinder

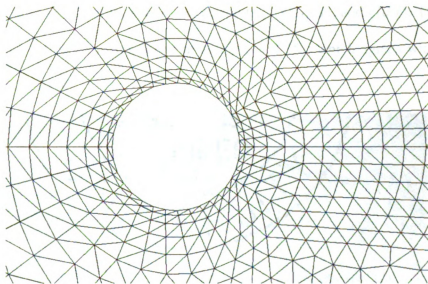
When a fluid flows over an isolated cylindrical solid barrier and the Reynolds number is great than about 50, vortices are shed on the down streamside. The vortices trail behind the cylinder in two rolls, alternatively from the top or the bottom of the cylinder. This vortex trail is called the Von Karman vortex street or Karmn street after Von Karman's 1912 mathematical description of the phenomenon. Since then, many numerical and experimental studies have focused on the dynamics of vortex street formation in the near wake. Measured value of the Strouhal number of the vortex shedding frequency can be found in Reference [70]. Spectral solutions of cylinder wake flows include those of [21,28,38]. A recent detailed review of the problem can be found in Reference [9].

Where, the Strouhal number $S = \frac{f_v \cdot D_c}{V}$, D_c is the diameter of the cylinder, f_v is the frequency of vortex shedding, and V is the flow velocity. To demonstrate the capability of LSV method in dealing with complex geometry (with curve wall effect), and complex flow properties, we choose this as one of our test cases.

The computations have been performed on the following unstructured grid shown in Figure 6.8. The center of cylinder is located at the origin and its diameter is equal to 1. The computational domain are $(-10, 16) \times (-10, 10)$.



(a)



(b)

Figure 6.8 Grid for the case of subsonic flow over a circular cylinder
(a) global grid, (b) grid near the cylinder

We consider a subsonic flow over an adiabatic circular cylinder at a small angle, free stream Mach number $Ma = 0.2$, Reynolds number based on the free stream condition and on the cylinder diameter $Re = 75$. We fix pressure P on the right boundary of the rectangular and fix every thing on the rest boundary of the rectangular.

After a sufficiently long time, the effects of the initial condition propagate out of the computational domain, and the periodic shedding of vortices is observed. Instantaneous contours of the Mach number, entropy, and vorticity showing the Von Karman vortex street generated by the cylinder are presented in Figures 6.9, 6.10 and 6.11. The periodic nature of the flow is show in Figure 6.12, which plot as a function of time the pressure calculated at three different locations, i.e. at points $(1,1)$, $(5,1)$, and $(10,1)$. The period of the oscillations corresponds to a Strouhal number of 0.151 , which agrees with Reference [63, 65, 66] very well.

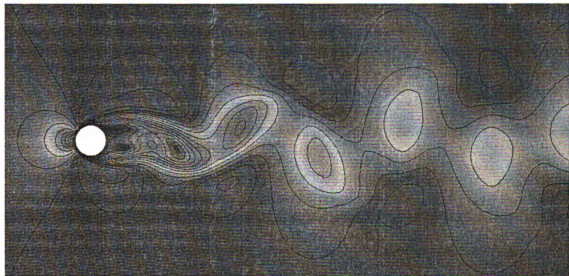


Figure 6.9 Instantaneous Mach contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

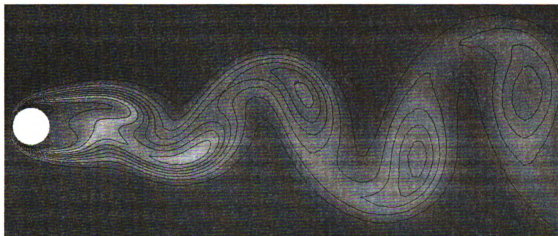


Figure 6.10 Instantaneous entropy contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

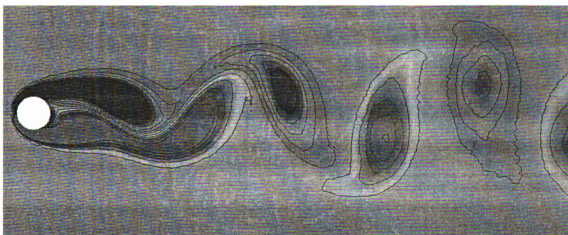


Figure 6.11 Instantaneous vorticity contours for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

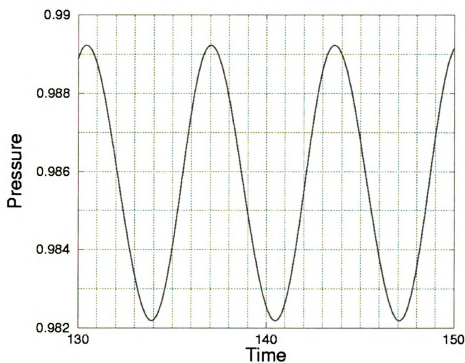
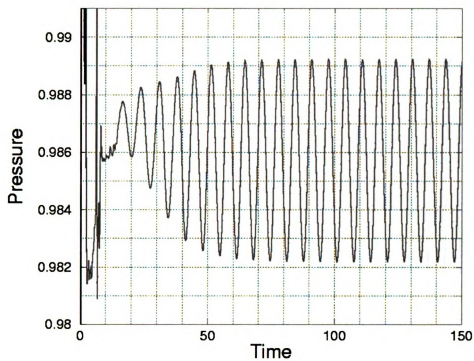


Figure 6.12a Pressure history at $(x, y) = (1, 1)$ for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

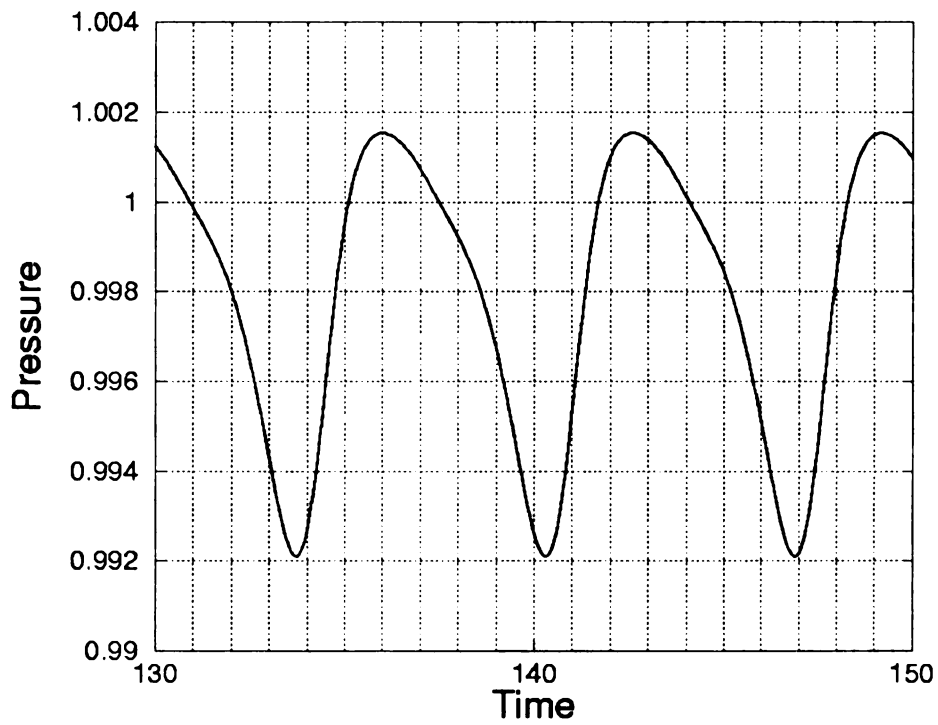
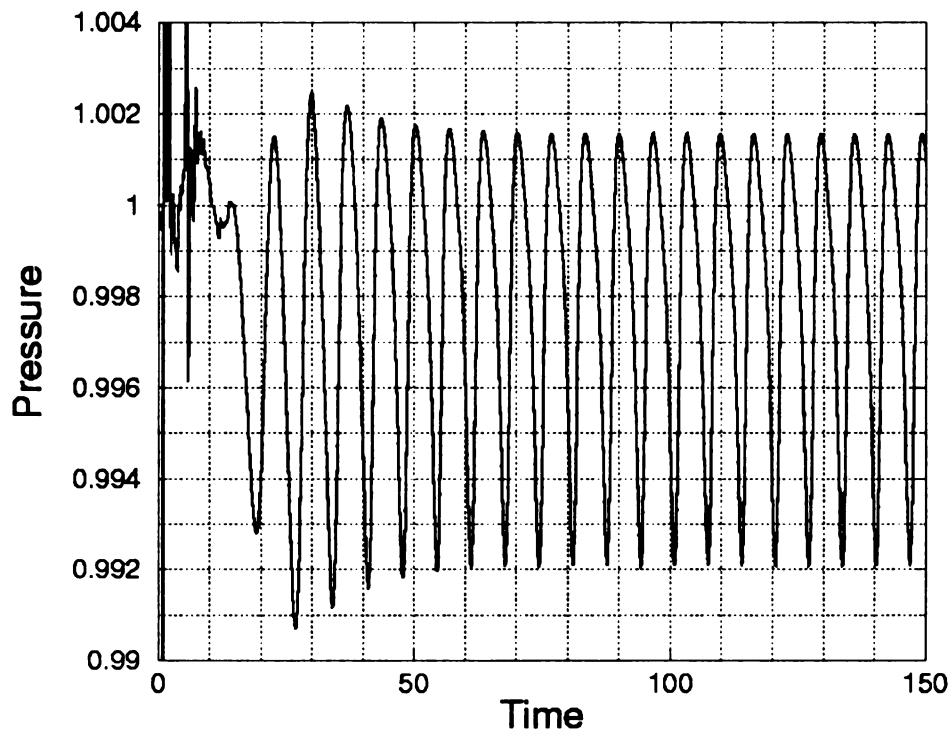


Figure 6.12b. Pressure history at $(x, y) = (5, 1)$ for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

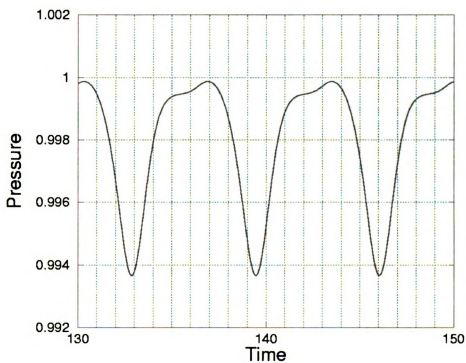
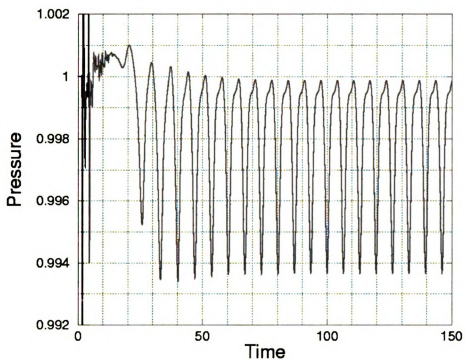


Figure 6.12c Pressure history at $(x, y) = (10, 1)$ for $Ma = 0.2$ flow over a circular cylinder at $Re = 75$

6.2.4 Laminar Subsonic Flow around the NACA0012 Airfoil

This test has been considered as validation test cases for shock capturing Navier-Stokes codes in a GAMM work shop, and is very well documented in the literature [7, 41, 43, 47, and 73].

The computations have been performed on the relative coarse grid shown in Figure 6.13, which is an unstructured triangulation of a 64x16 O-grid. The larger number refers to the number of cells distributed along the airfoil surface and smaller one to the number of cells in the radial direction. The grid extends about 20 chords away from the airfoil. The computations have been performed using linear, quadratic and cubic SVs

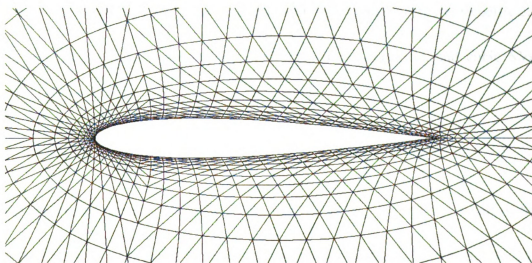


Figure 6.13 Computational grid for the NACA0012 test case

We run a laminar subsonic flow at an angle of attack $\alpha = 0^\circ$, free stream Mach number $Ma = 0.5$, and Reynolds number $Re = 5000$. In the test, the wall is adiabatic. The Reynolds number is near the upper limit for steady laminar flow. A distinguishing feature of this test case is the separation region of the flow occurring near the trailing edge, which causes the formation of a small recirculation bubble that extend in the near-wake region of the airfoil.

Figure 6-14 shows the Mach isolines computed with linear, quadratic and cubic SVs. It is obvious that the solution is getting smoother and smoother with the increasing of the order of polynomial reconstruction.

Figures 6.15 and 6.16 show the pressure coefficient (C_p) and skin friction coefficient (C_f) distributions along the airfoil computed with linear, quadratic, and cubic SVs, where

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty \cdot U_\infty^2} \quad C_f = \frac{\tau_w}{\frac{1}{2} \rho_\infty \cdot U_\infty^2}$$

Table 6.5 reports the pressure part of the drag coefficient ($C_{d,p}$) and stress part of the drag coefficient ($C_{d,v}$) computed with linear, quadratic and cubic SVs, respectively. The comparison of the values of the drag coefficients computed with cubic SVs and those obtained by other authors with both structured and unstructured solvers [7, 41, 43, 47] is given in Table 6.6, where

$$C_{d,p} = \frac{\oint p \, dA}{\frac{1}{2} \rho_{\infty} \cdot U_{\infty}^2 \cdot A} \quad C_{d,v} = \frac{\oint \tau_w \, dA}{\frac{1}{2} \rho_{\infty} \cdot U_{\infty}^2 \cdot A}$$

A is a characteristic area of the object. In addition the residual converge history is displayed in Figure 6.17, which demonstrates that we have reached the steady solution.

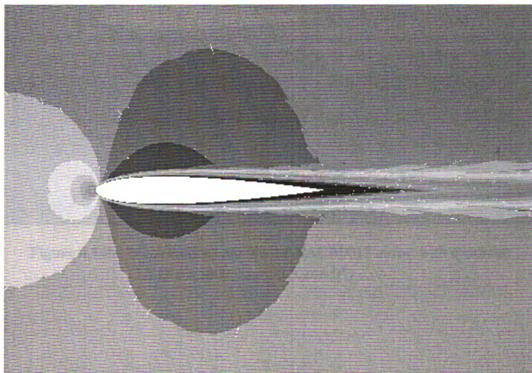


Figure 6.14a Mach isolines around the NACA0012 airfoil with linear SV ($Re = 5000$, $Ma = 0.5$, $\alpha = 0^\circ$)

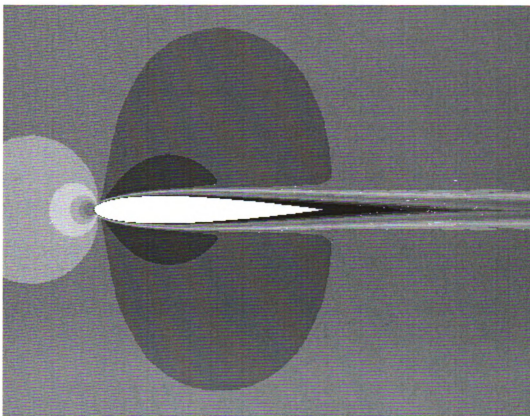


Figure 6.14b Mach isolines around the NACA0012 airfoil with quadratic SV ($Re = 5000$, $Ma = 0.5$, $\alpha = 0^\circ$)

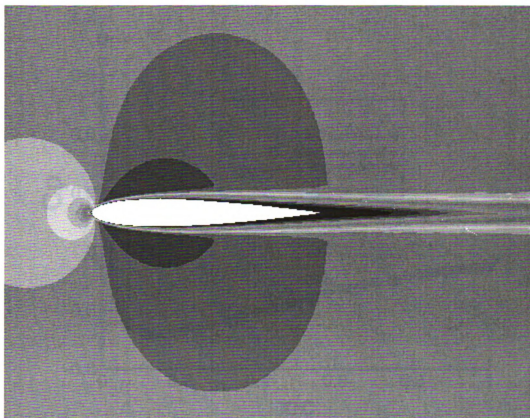


Figure 6.14c Mach isolines around the NACA0012 airfoil with cubic SV ($Re = 5000$, $Ma = 0.5$, $\alpha = 0^\circ$)

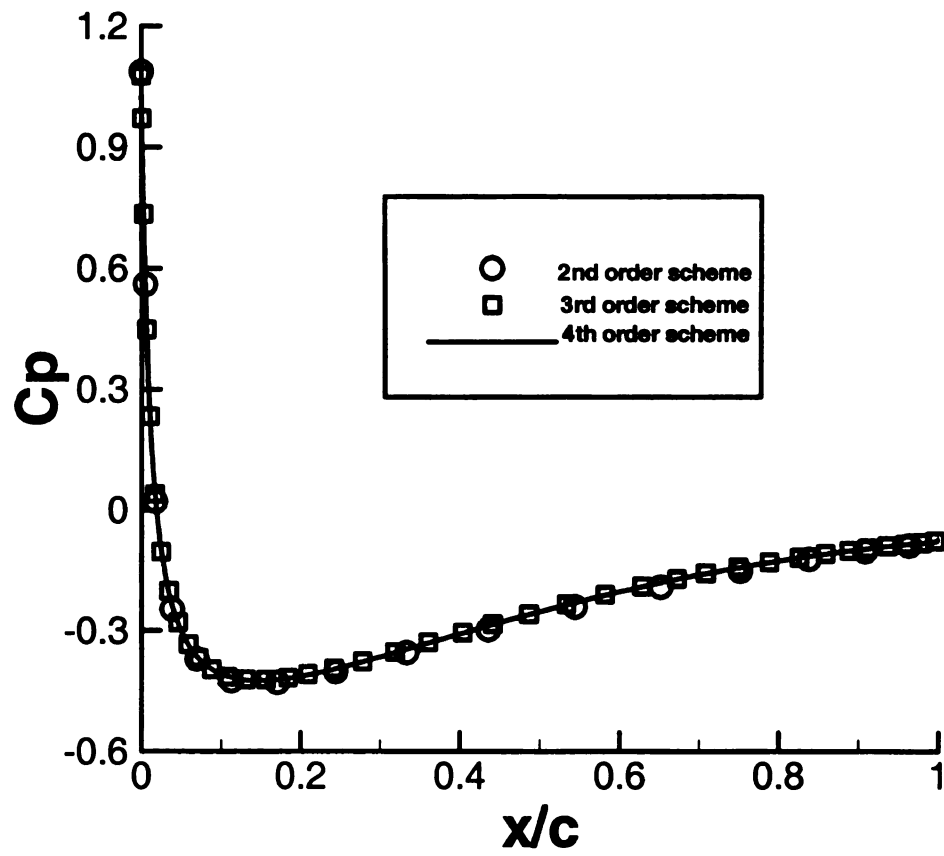


Figure 6.15 Pressure coefficient distribution along the NACA0012 airfoil ($Re = 5000$, $Ma = 0.5$, $\alpha = 0^\circ$)

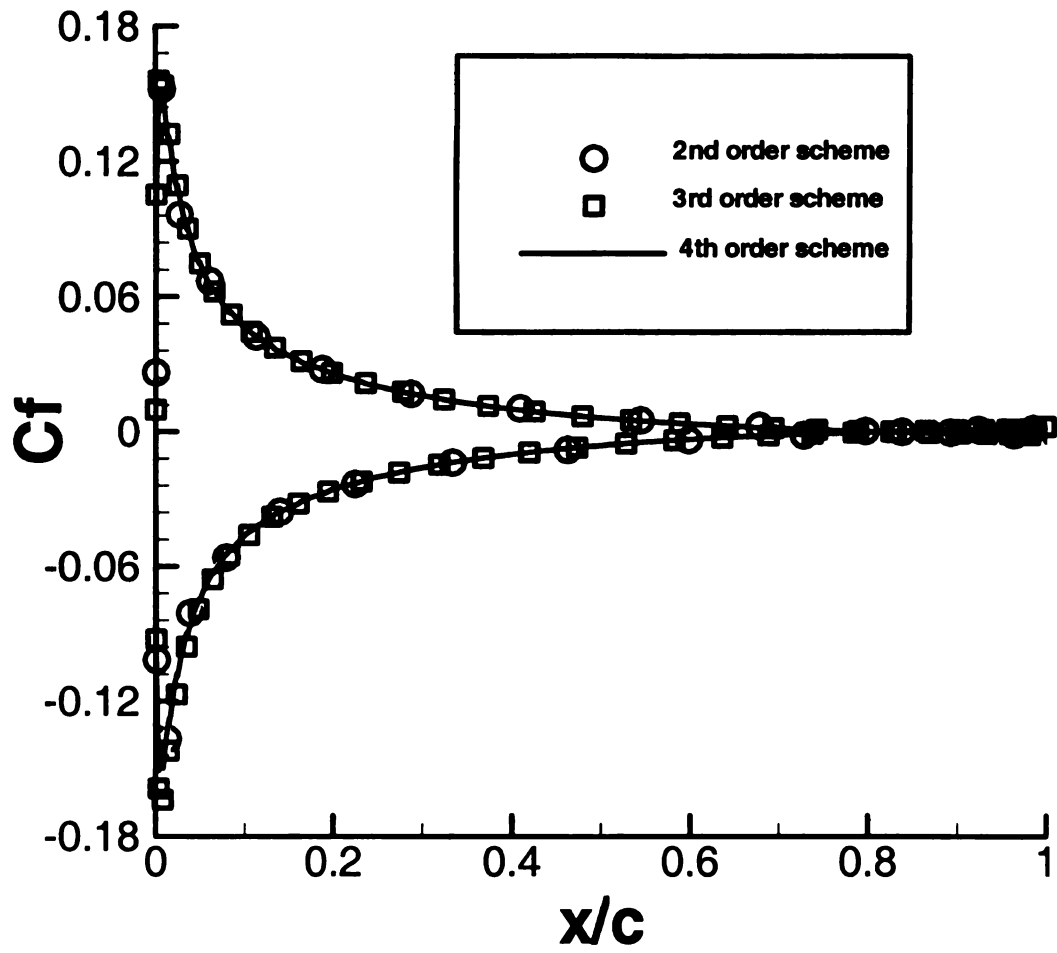


Figure 6.16 Skin friction coefficient distribution along the NACA0012 airfoil ($Re = 5000$, $Ma = 0.5$, $\alpha = 0^\circ$)

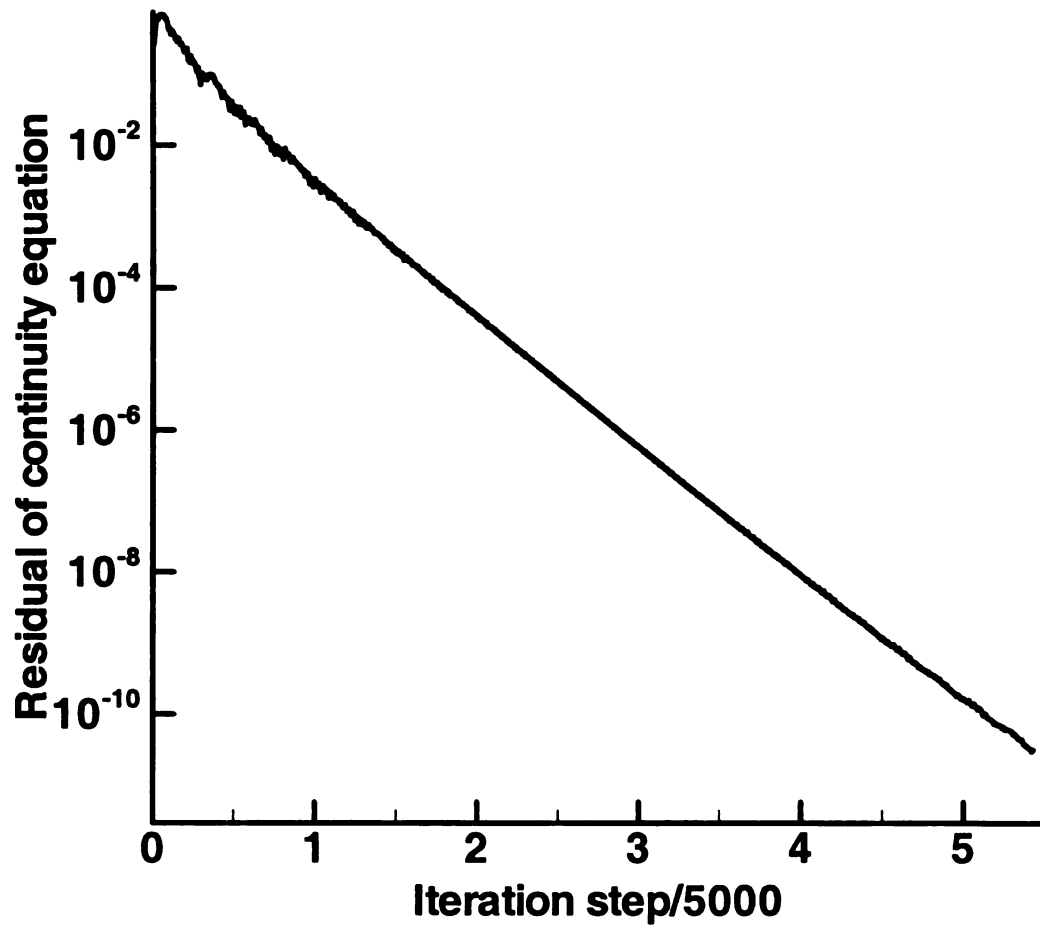


Figure 6.17 Residual history of continuity equation with linear reconstruction for NACA0012 test case

Table 6.5 Pressure Part ($C_{d,p}$) and Viscous Part ($C_{d,v}$) of the Drag Coefficient for the NACA0012 Airfoil

SV-type	Accuracy	$C_{d,p}$	$C_{d,v}$
Linear	2 nd	2.325e-02	3.311e-02
Quadratic	3 rd	2.246e-02	3.275e-02
Cubic	4 th	2.231e-02	3.302e-02

Table 6.6 Comparisons of Pressure Part ($C_{d,p}$) and Viscous Part ($C_{d,v}$) of the Drag Coefficient for the NACA0012 Airfoil Computed by the Present Method with Various Structured and Unstructured Solvers

Method	Grid	Degree of freedom	$C_{d,p}$	$C_{d,v}$
Cubic SV	64 x 16	10240	0.02231	0.03302
Cubic DG [36]	64 x 16	10240	0.02208	0.03303
Triangle scheme from Ref. [43]	320x64	20480	0.0229	0.0332
Cell-center scheme form Ref. [41]	320x64	20480	0.0219	0.0337
Cell-vertex scheme form Ref. [47]	256x64	16384	0.0227	0.0327
Cell-center scheme form Ref. [47]	256x64	16384	0.02256	0.03301
Cell-center scheme form Ref. [47]	512x128	65536	0.02235	0.03299

6.3 Conclusions

The SV method is successfully extended to the Navier-Stokes equations by following a mixed formulation named the local discontinuous Galerkin approach originally developed for the DG method. The approach, which is named the local SV (LSV) approach, has been tested extensively for 1D and 2D convection-diffusion equations using a series of accuracy studies. All tests indicated that the formulation is capable of achieving the formal optimum order of accuracy in both L_1 and L_∞ norms. The LSV approach is then implemented and tested for the Navier-Stokes equations, and was able to achieve the formal order of accuracy for the compressible Couette flow problem. Also, we have test on more complex problems such as laminar viscous flow along a flat plate, subsonic flow over a circular cylinder, and laminar viscous flow around NACA0012 airfoil. The case of laminar flow over a flat plate was simulated successfully with good agreement with the Blasius solution. The numerical results based on LSV approach, for the case of subsonic flow over a circular cylinder, match with those in experiments and

some other method. In the test case of NACA0012 airfoil, the physical solutions have been achieved, and the numerical solution is getting smoother and smoother with the increasing of degree of reconstruction polynomials.

CHAPTER 7

SUMMARY AND FUTURE WORK

7.1 Summary

Two major research areas have been undertaken in this study: the evaluation of the DG and SV methods, and the extension of the SV method to the Navier-Stokes equations. In the first research area, the DG and SV methods have been evaluated for the scalar conservation laws and the Euler equations in both 1D and 2D. The overall performance of both methods in terms of the efficiency, accuracy and memory requirement has been evaluated. In the second research area, several different algorithms have been suggested and tested for second-order derivatives. Fourier analysis was employed to analyze the accuracy, consistency and stability of the proposed algorithms for the 1D heat equation. The analysis has been numerically verified. Then the best performing algorithm, the local SV approach, has been extended to multi-dimensional scalar convection and diffusion equation and to the Navier-Stokes equations. Extensive numerical tests have been carried out to test the overall performance.

Generally speaking, both the DG and SV methods are capable of achieving the formal order of accuracy, i.e. they both achieve 2nd, 3rd, and 4th order accuracy for the corresponding linear, quadratic and cubic reconstructions respectively, while the DG method usually has a lower error magnitude and takes more memory. In the scalar case, the SV schemes are consistently faster than the DG schemes of the same order of accuracy for each residual evaluation. For the Euler equations, the 2nd-order SV scheme is faster than the 2nd-order DG scheme. However, 3rd and 4th order SV schemes are quite

similar to the corresponding DG schemes in terms of efficiency ($<12\%$ in difference). It is also clear that the SV method has a higher resolution for discontinuities than the DG method because of the sub-cell average based data limiting. We also confirm that the SV method takes less memory and allows larger time steps than the DG method for both the 2D scalar conservation laws and the Euler equations.

In order to test the SV method for second-order derivative terms, the 1D heat equation was first employed to evaluate the SV method in Chapter 4, in which three formulations are presented, i.e. Naïve SV formulation, Local SV formulation and Penalty SV formulation. A Fourier type analysis has been performed and it has been proved that the local SV method and penalty SV method can achieve the formal order of accuracy for linear reconstructions on pure diffusion equation. Second, the local SV method has been applied to scalar convection-diffusion equations in Chapter 5 based on the optimal accuracy it has achieved for the 1D heat equation. Extensive accuracy studies were carried out for both 1D and 2D convection-diffusion equations. The numerical results have demonstrated that the local SV method can achieve the optimal order of accuracy for convection-diffusion equation. Finally, the local SV method has been implemented for Navier-Stokes equations in Chapter 6. We have successfully solved several viscous flow problems such as the Couette flow, laminar flow along a flat plate, unsteady subsonic flow over a circular cylinder and laminar subsonic flow around the NACA0012 airfoil, by using local SV method. For the Couette flow problem, the numerical solution converged to the steady analytical solution and achieved the formal order of accuracy. For the case of laminar flow along a flat plate, it was demonstrate that the numerical solution agrees well with the Blasius solution. For the case of subsonic flow over a

circular cylinder, the numerical experiment has shown that an unsteady, stable vortex street formed behind the cylinder at a Reynolds number of $Re = 75$. The Strouhal number of the vortex shedding frequency based on the local SV method matches those based experimental measurement and a staggered-grid multidomain spectral method. For the NACA0012 airfoil case, the numerical solution was getting smoother and smoother with increasing order of reconstructions. In addition, the drag based on the local SV method, due to pressure and shear stress, was quite similar to that computed with several 2nd order finite volume methods. However, the total number of degrees of freedom is much less than those used in the 2nd order finite volume methods.

In summary, the SV method has some unique properties comparing with other high-order methods such as the DG method, and is capable of achieving high-order accuracy for the Navier-Stokes equations. Accurate numerical results can be computed with much coarser meshes than those used in second-order finite volume methods. For steady state computations, however, the explicit high-order SV schemes are still not competitive to implicit second-order finite volume method because of the time step limit and the slow convergence of the method. Much more efficient time integration algorithms must be developed for these high-order methods to be used routinely in engineering design.

7.2 Future Work

Although the feasibility of the SV method for the Navier-Stokes equations has been successfully demonstrated in this study, major obstacles still remain to apply the method

for “real world” engineering flow problems. In order to achieve that goal, the following activities are planned:

- Test the SV method for more complex viscous flow problems involving more complex geometries, higher Reynolds numbers;
- Develop an efficient time marching algorithm for the SV method. Two approaches are possible: implicit solution approach and multi-grid approach;
- Parallelize the SV method on distributed memory machines using domain decomposition and message passing;
- Finally the method must be extended to three-dimensions since all real flow problems are 3D.

APPENDIX A

GAUSS QUADRATURE FORMULAS

We consider integrals and quadrature rules of the form

$$I = \int_A f(p) dA \approx \sum_{i=1}^n W_i f(P_i) \quad (\text{A.1})$$

where, A is integral domain, W_i are the quadrature rule's *weights* and P_i are the *evaluation points*, $i = 1, 2, \dots, n$. The integration rule (A.1) is called *exact to order q* if it is exact when f is any polynomial of degree q or less. In References [1,22], we can see the detail.

A-1. 1D Quadrature Rule

Considering the one-dimensional quadrature rules on the canonical $[-1,1]$ element

$$I = \int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^n W_i f(\xi_i) \quad (\text{A.2})$$

Gaussian quadrature is preferred for numerical integration because they have fewer evaluation points for a given order. Actually, if we choose n evaluation points, then we can get an integration rule be exact to order $2n-1$. The flowing Table A.1 gives the evaluation points ξ_i , $i = 1, 2, \dots, n$, which are the roots of the Legendre polynomial of degree n . The weights W_i , $i = 1, 2, \dots, n$, called Christoffel weights are also shown for n ranging from 1 to 6.

Table A.1 Christoffel weights W_i and roots $\xi_i, i = 1, 2, \dots, n$, for Legendre polynomials of degree 1 to 6

N	$\pm \xi_i$	W_i
1	0.00000 00000 00000	2.00000 00000 00000
2	0.57735 02691 89626	1.00000 00000 00000
3	0.00000 00000 00000	0.88888 88888 88889
	0.77459 66692 41483	0.55555 55555 55556
4	0.33998 10435 84856	0.65214 51548 62546
	0.86113 63115 94053	0.34785 48451 37454
5	0.00000 00000 00000	0.56888 88888 88889
	0.53846 93101 05683	0.47862 86704 99366
	0.90617 98459 38664	0.23692 68850 56189
6	0.23861 91860 83197	0.46791 39345 72691
	0.66120 93864 66265	0.36076 15730 48139
	0.93246 95142 03152	0.17132 44923 79170

Example. Consider evaluating the integral

$$I = \int_a^b g(x) dx$$

by Gauss quadrature. Let us transform the integral to $[-1,1]$ using the mapping

$$\xi = \frac{2}{b-a} \left(x - \frac{a+b}{2} \right) \quad (\text{or } x = \frac{b-a}{2} \xi + \frac{a+b}{2}) \quad (\text{A.3})$$

to get

$$I = \frac{b-a}{2} \int_{-1}^1 f(\xi) d\xi \approx \frac{b-a}{2} \sum_{i=1}^n W_i f(\xi_i)$$

where, we could find W_i and ξ_i in Table A.1 for a given n ranging from 1 to 6, and

$f(\xi_i) = g(x_i), i = 1, 2, \dots, n$. ξ_i and x_i are related each other by relation (A.3). So we get

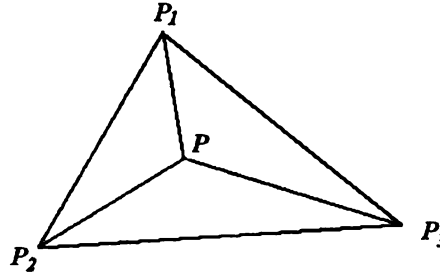
$$I \approx \frac{b-a}{2} \sum_{i=1}^n W_i g(x_i)$$

A-2. 2D Quadrature Rule

Two-dimensional integrals on triangles are conveniently expressed in terms of triangular coordinates as

$$\iint_{\Omega_e} f(x, y) dx dy \approx A_e \sum_{i=1}^n W_i f(\zeta_1^i, \zeta_2^i, \zeta_3^i)$$

where $(\zeta_1^i, \zeta_2^i, \zeta_3^i)$ are the triangular coordinates of evaluation point i and A_e is the area of triangle e . The relation between the coordinate of (x, y) and the triangular coordinate is described as follows:



Let $(x, y), (x_i, y_i)$ denote the triangular coordinate of points $P, P_i, i = 1, 2, 3$, then the triangular coordinates $(\zeta_1, \zeta_2, \zeta_3)$ of point P can be described as

$$\zeta_1 = \frac{\text{area of } \Delta PP_2P_3}{\text{area of } \Delta P_1P_2P_3} = \frac{1}{2A_e} \cdot \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}, \quad (\text{A.4.1})$$

$$\zeta_2 = \frac{\text{area of } \Delta PP_3P_1}{\text{area of } \Delta P_1P_2P_3} = \frac{1}{2A_e} \cdot \begin{vmatrix} 1 & x & y \\ 1 & x_3 & y_3 \\ 1 & x_1 & y_1 \end{vmatrix}, \quad (\text{A.4.2})$$

$$\zeta_3 = \frac{\text{area of } \Delta PP_1P_2}{\text{area of } \Delta P_1P_2P_3} = \frac{1}{2A_e} \cdot \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix}. \quad (\text{A.4.3})$$

Note that there is an identity $\zeta_1 + \zeta_2 + \zeta_3 \equiv 1$. Where A_e denotes the area of $\Delta P_1 P_2 P_3$, $| * |$ denotes a determinant. Therefore (A.4.1-A.4.3) give a relation between the coordinate of (x, y) and the coordinate of $(\zeta_1, \zeta_2, \zeta_3)$. We also can express (x, y) by using $(\zeta_1, \zeta_2, \zeta_3)$ as

$$x = x_3 + (x_1 - x_3) \cdot \zeta_1 + (x_2 - x_3) \cdot \zeta_2 \quad (\text{A.5.1})$$

$$y = y_3 + (y_1 - y_3) \cdot \zeta_1 + (y_2 - y_3) \cdot \zeta_2 \quad (\text{A.5.2})$$

We list some quadrature rules in Table A.2. A multiplication factor M indicates the number of permutations associated with an evaluation point having a weight W_i . The factor p indicates the order of the quadrature rule, i.e. the quadrature formula is exact integration for any polynomial of degree p or less. The error between the exact integration and quadrature formula is $O(h^{p+1})$ where h is the maximum edge of the triangle.

Table A.2 Weights and evaluation points for integration on triangles

n	W_i	ζ_1	ζ_2	M	p
1	1.0000000000000000	0.3333333333333333	0.3333333333333333	1	1
3	0.3333333333333333	0.6666666666666667	0.1666666666666667	3	2
4	-0.5625000000000000	0.3333333333333333	0.3333333333333333	1	3
	0.5208333333333333	0.6000000000000000	0.2000000000000000	3	
6	0.109951743655322	0.816847572980459	0.091576213509771	3	4
	0.223381589678011	0.108103018168070	0.445948490915965	3	
7	0.2250000000000000	0.3333333333333333	0.3333333333333333	1	5
	0.125939180544827	0.797426985353087	0.101286507323456	3	
	0.132394152788506	0.059715871789770	0.470142064105115	3	
12	0.050844906370207	0.873821971016996	0.063089014491502	3	6
	0.116786275726379	0.501426509658179	0.249286745170910	3	
	0.082851075618374	0.636502499121399	0.310352451033785	6	
13	-0.149570044467670	0.3333333333333333	0.3333333333333333	1	7
	0.175615257433204	0.479308067841923	0.260345966079038	3	
	0.053347235608839	0.869739794195568	0.065130102902216	3	
	0.077113760890257	0.638444188569809	0.312865496004875	6	

APPENDIX B

PUBLICATIONS ASSOCIATED WITH THIS WORK

Sun, Yuzhi and Wang, Z.J., "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Scalar and System Conservation Laws on Unstructured Grids," *International Journal for Numerical Methods in Fluids*, Vol. 45, pp819-838, 2004.

Sun, Yuzhi and Wang, Z.J., "Formulations and Analysis of the Spectral Volume Method for the Diffusion Equation," *Communications in Numerical Methods in Engineering*, Vol. 20, pp927-937, 2004.

Sun, Yuzhi and Wang, Z.J., "Extension of Spectral Volume Method to the Navier-Stokes Equations on Unstructured Grids," to be submitted.

Sun, Yuzhi and Wang, Z.J., "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Conservation Laws on Unstructured Grids," AIAA-2003-0253.

Sun, Yuzhi and Wang, Z.J., "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for 2D Euler Equations on Unstructured Grids," AIAA-2003-3680.

Sun, Yuzhi and Wang, Z.J., "High-Order Spectral Volume Method for the Navier-Stokes Equations on Unstructured Grids," AIAA-2004-2133.

BIBLIOGRAPHY

1. Abramowitz, M. and Stegun I. A., *Handbook of Mathematical Functions*, volume 55 of Applied Mathematics Series. National Bureau of Standards, Gathersburg, 1964.
2. Arnold, D.N., An interior penalty finite element method with discontinuous elements, *SIAM J. Numer. Anal.* , **19**, 742-760 (1982).
3. Barth, T.J., Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes, AIAA 29th Aerospace sciences meeting, AIAA 91-0721, January 1991.
4. Barth, T.J. and Frederickson, P.O., High order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA 90-0013, January 1990.
5. Barth, T.J. and Jespersen, D.C., The design and application of upwind schemes on unstructured meshes, AIAA 27th Aerospace sciences meeting, AIAA 89-0366, January 1989.
6. Bassi, F. and Rebay, S., High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys.*, **138**, 251(1997).
7. Bassi F. and Rebay, S., A High-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *J. Comput. Phys.* **131**, 267-279 (1997).
8. Baumann, C.E. and Oden, J.T., A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.* **175**, 311-341 (1999).
9. Beaudan, P. and Moin, P., Numerical experiments on the flow past a circular cylinder at sub-critical Reynolds Number, *Technical Report, TF-62*, Department of Mech. E., Stanford University, 1994.
10. Castilo, Paul, An optimal estimate for the local discontinuous Galerkin method, in Discontinuous Galerkin Methods, B. Cockburn, G.E. Karniadakis, and C. -W. Shu, *Lecture Notes in Computational Science and Engineering*, **11**, Springer, 285-290 (2000).
11. Cockburn, B., Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws, *J. Comput. Appl. Math.*, **128**, 187-204 (2001). *Numerical analysis 2000*, Vol. VII, Partial differential equations.
12. Cockburn, B., Gremaud, P. -A., and Yang, J.X., A priori error estimates for nonlinear scalar conservation laws, in hyperbolic problems: *theory numerics*,

applications, Vol. I (Zürich, 1998), vol. 129 of *Internat. Ser. Numer. Math.*, Birkhäuser, Basel, 1999, pp.167-176.

13. Cockburn, B., Hou, S. and Shu, C.-W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, *Mathematics of Computation* **54**, 545-581(1990).
14. Cockburn, B., Karniadakis, G. E., and C. -W. Shu, The development of discontinuous Galerkin methods, in Discontinuous Galerkin methods (Newport, RI, 1999), vol. 11 of *Lec. Notes Comput. Sci Eng.*, Springer, Berlin, 2000, pp.3-50.
15. Cockburn, B., Liu, S. -Y. and Shu, C. -W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, *J. Comput. Phys.* **84**,90-113 (1989).
16. Cockburn, B. and Shu, C. -W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Mathematics of Computation* **52**, 411-435 (1989).
17. Cockburn, B. and Shu, C.-W., The Runge-Kutta local projection P^1 -discontinuous-Galerkin finite element method for scalar conservation laws, *RAIRO Modél. Math. Anal. Numér.* , **25**, 337-361 (1991).
18. Cockburn, B. and Shu, C. -W., The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems, *J. Comput. Phys.*, **141**, 199-224 (1998)
19. Cockburn, B. and Shu, C. -W., The local discontinuous Galerkin method for time-dependent convection diffusion system. *SIAM J. Numer. Anal.* , **35**, 2440-2463(1998).
20. Colella, P. and Woodward, P., The piecewise parabolic method for gas-dynamical simulations, *J. Comput. Phys.* **54**, (1984).
21. Don, W. -S. and Gottlieb, D., Spectral simulation of unsteady compressible flow past a circular cylinder, *Comput. Methods Appl. Mech. Eng.* **80**, 39 (1990).
22. Dunavant, D.A., High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International Journal of Numerical Methods in Engineering*, **21**, 1129-1148(1985).
23. Falkkk, R. S. and Richter, G.R., Local error estimates for a finite element method for hyperbolic and convection-diffusion equations, *SIAM J. Numer. Anal.*, **29**,730-754. (1992).

24. Godunov, S.K., A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.* **47**,271(1959).
25. Harten, A., High resolution schemes for hyperbolic conservation laws, *J. of Comput. Phys.* **49**, 357-393(1983).
26. Haren, A., Engquist, B, Osher, S., and Chakravarthy, S., Uniformly high order essentially non-oscillatory schemes III, *J. Comput. Phys.* **71**, 231 (1987).
27. Hartmann, R. and Houston, P., Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws, *SIAM J. Sci. Comput.*, **24**, 979-1004 (2002).
28. Hesthaven, J. S., A stable penalty method for the compressible Navier-Stokes equations. ii. One dimensional domain decomposition schemes, *SIAM J. Sci. Comput.* **18**,658 (1997)
29. Houston, P., Schwab, C., and Süli, E., Discontinuous hp-finite element methods for convection-diffusion-reaction problems, *SIAM J. Numer. Anal.*, **39**, 2133-2163. (2002)
30. Hu, C. and Shu, C.-W., A discontinuous Galerkin finite element method for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.*, **21**, 666-690 (1999).
31. Hu, C. and Shu, C.-W., Weighted essentially non-oscillatory schemes on triangular meshes, *J. Comput. Phys.*, **150**,97 (1999).
32. Jameson, A. and Baker, T., Improvements to the aircraft Euler method, AIAA 25th Aerospace sciences meeting, AIAA 87-0452, January 1987.
33. Jameson, A. and Mavriplis, D., Finite volume solution in the two-dimensional Euler equations on a regular triangular mesh, AIAA 23rd Aerospace sciences meeting, AIAA 85-0435, January 1985.
34. Jameson, A., Baker, T., and Weatherill, N., Calculation of inviscid transonic flow over a complete aircraft, AIAA 24th Aerospace sciences meeting, AIAA 86-0103, January 1986.
35. Johnson, C., Nävert, U. and PTTKÄRANTA, J., Finite element methods for linear hyperbolic problems, *Comput. Methods Appl. Mech. Engrg.* , **45**, 285-312 (1984).
36. Johnson, C., Nävert, U. and PTTKÄRANTA, J., An analysis of the discontinuous Galerkin method for scalar hyperbolic equation, *Math. Comp.* , **46**,1-26 (1986).

37. Karakashian, O. and Makridakis, C., A space-time finite element method for the nonlinear Schrödinger equation: the discontinuous Galerkin method, *Math. Comp.*, **67**, 479-499 (1998).
38. Kopriva, D. A., A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, **143**, 125-158 (1998).
39. Lesaint, P. and Raviart, P. A., On a finite element method to solve the neutron transport equation, in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. de Boor (Academic Press, New York, 1974), P.89.
40. Liu, M. -S., Mass flux schemes and connection to shock instability, *J Comput. Phys.* **160**, 623-648 (2000).
41. Martinelli, L., Calculations of viscous flows with a multigrid method, *Ph.D thesis*, Dept. of Mechanical and Aerospace Engineering, Princeton University, 1987.
42. Mavriplis, D. and Jameson, A., Multigrid solution of the Navier-Stokes equations on triangular meshes, *AIAA Journal*, vol. 28, pp. 1415-1425, January 1986.
43. Mavriplis, D. J. and Jameson, A., Multigrid solution of the Navier-Stokes equations on triangular meshes, *AIAA J.* **28**, No. 8, 1415 (1990).
44. Oden, J.T., Babuska, Ivo and Baumann, C.E., A discontinuous hp finite element method for diffusion problems, *J. Comput. Phys.*, **146**, 491-519 (1998).
45. Osher, S., Riemann solvers, the entropy condition, and difference approximations, *SIAM J. on Numerical Analysis* **21**, 217-235 (1984).
46. Osher, S. and Chakravarthy, S., Upwind schemes and boundary conditions with applications to Euler equation in general geometries, *J. Comput. Phys.* **50** 447-481 (1983).
47. Radespiel, R. and Swanson, R. C., An investigation of cell centered and cell vertex multigrid schemes for the Navier-Stokes equations, *AIAA paper* No. 89-0453, 1989.
48. Reed, W. H. and Hill, T. R., Triangular mesh methods for the neutron transport equation, *Los Alamos Scientific Laboratory Report* LA-UR-73-479, 1973 (unpublished).
49. Richter, G. R., An optimal-order error estimate for the discontinuous Galerkin method, *Math. Comp.*, **50**, 75-88 (1988).
50. Richter, G. R., The discontinuous Galerkin method with diffusion, *Math. Comp.*, **58**, 631-643 (1992).

51. Riviere, B., Wheeler, M. and Girault, V., Apriori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems, *SIAM J. Numer. Anal.*, , **39**, 902-931 (2001).
52. Roe, P.L., Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43** 357-372 (1981).
53. Rusanov, VV, Calculation of interaction of non-steady shock waves with obstacles, *J. Comput. Math. Phys. USSR* **1**, 267-279 (1961).
54. Shu, C. -W., TVB uniformly high-order schemes for conservation laws. *Mathematics of computation*, **49**,105-121(1987).
55. Shu, C. -W., Total-variation-diminishing time discretization. *SIAM Journal on Scientific and Statistical Computing*, **9**, 1073-1084 (1988).
56. Shu, C. -W, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, B. Cockburn, C. Johnson, C.-W. Shu and E.Tadmor, *Lecture Notes in Mathematics* , volume 1697, Springer, 325-432 (1998).
57. Shu, C. -W., Different formulations of the discontinuous Galerkin method for the viscous terms. In *Advances in Scientific Computing*, Z. -C. Shi, M. Mu, W. Xue and J. Zou, editors, Science Press, 2001; pp. 144-145.
58. Shu, C. -W. and Osher, S., Efficient implementation of essentially non-oscillatory shock capturing schemes. *Journal of Computation Physics*, **77**, 439-471(1988).
59. Steger, J.L. and Warming, R.F., Flux vector splitting of the inviscid gasdynamics equations with application to finite difference methods, *J. Comput. Phys.* **40** 263(1981).
60. van Leer, B., Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme, *J. Comput. Phys.* **14**, 361(1974).
61. van Leer, B., Towards the ultimate conservative difference scheme V. a second order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101-136(1979).
62. van Leer, B., Flux-vector splitting for the Euler equations, *Lecture Notes in Physics*, **170**, 507 (1982).
63. Wang, Z.J., A fast flux-splitting for all speed flow, in *Proceedings of Fifteen International Conference on Numerical Methods in Fluid Dynamics*, pp141-146, 1997.

64. Wang, Z.J., Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation. *J. Comput. Phys.* **178**, 210 (2002).
65. Wang, Z.J. and Liu, Yen, Spectral (finite) volume method for conservation laws on unstructured grids II. Extension to two-dimensional scalar equation. *J. Comput. Phys.* **179**, 665-697 (2002).
66. Wang, Z.J. and Liu, Yen, Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization. *J. Scientific Computing*, **20**, 137-157 (2004).
67. Wang, Z.J., Zhang, L. and Liu, Y., Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids IV: Extension to Two-Dimensional Euler Equations, *J. of Comput. Physics*, **194**, 716-741 (2004).
68. Wheeler, M. F., An elliptic collocation-finite element method with interior penalties, *SIAM J. Numer. Anal.*, **15**, 152-161 (1978).
69. Whitaker, D.L., Slack, D.C., and Walters, R.W., Solution algorithms for the two-dimensional Euler Equations on Unstructured meshes, *AIAA 90-0697*, 1990.
70. Williamson, C. H. K., Oblique and parallel modes of vortex shedding in the wake of a cylinder at low Reynolds number, *J. Fluid Mech.* **206**, 579 (1989).
71. Woodward, P. and Colella, P., The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.*, **54**, 115-173 (1984).
72. Zhang, M. and Shu, C. -W., An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences*, **13**, 395-413 (2003).
73. "Numerical simulation of compressible Navier-Stokes equations-external 2D flows around a NACA0012 airfoil," in GAMM Workshop, December 4-6, 1985, Nice, France (Edt. INRIA, Centre de Rocqufort, de Rennes et de Sophia-Antipolis, 1986).

VITA

Yuzhi Sun, she received a M. Sc. in Mathematics in June of 1989, and worked as an assistant professor from 1989 to 1997 in the Mathematics Department at Hebei University in China. She began working with her major professor Z.J. Wang at CFD lab as a research assistant in Department of Mechanical Engineering at Michigan State University from the fall, 2001. She obtained a M.Sc. in Mechanical Engineering at Michigan State University in December of 2002. The following is her publications in this period. Yuzhi will be a post-doctoral research associate with Professor Z.J. Wang in the Department of Aerospace Engineering at Iowa State University.

Journal Papers:

- Wang, Z.J. and Sun, Yuzhi, "Curvature-Based Wall Boundary Condition for the Euler Equations on Unstructured Grids," *AIAA Journal*, Vol. 41, pp27-33, 2003.
- Sun, Yuzhi and Wang, Z.J., "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Scalar and System Conservation Laws on Unstructured Grids," *International Journal for Numerical Methods in Fluids*, Vol. 45, pp819-838, 2004.
- Sun, Yuzhi and Wang, Z.J., "Formulations and Analysis of the Spectral Volume Method for the Diffusion Equation," *Communications in Numerical Methods in Engineering*, Vol. 20, pp927-937, 2004.
- Sun, Yuzhi and Wichman, Indrek S. "On Transient Heat Conduction in One-Dimensional Composite Slab," *International Journal of Heat and Mass Transfer*, Vol. 47, pp1555-1559, 2004.

Conference Papers:

- Sun, Yuzhi and Wang, Z.J. "High-Order Spectral Volume Method for the Navier-Stokes Equations on Unstructured Grids," *AIAA-2004-2133*.
- Sun, Yuzhi and Wang, Z.J. "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for 2D Euler Equations on Unstructured Grids," *AIAA-2003-3680*.
- Sun, Yuzhi and Wang, Z.J., "Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Conservation Laws on Unstructured Grids," *AIAA-2003-0253*.
- Wang, Z.J. and Sun, Yuzhi, "A Curvature-Based Wall Boundary Condition for the Euler Equations on Unstructured Grids," *AIAA-2002-0966*.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02504 0043