2008

High Order Spectral Volume and Spectral Difference Methods on Unstructured Grids

Ravishekar Kannan
Iowa State University

Recommended Citation

Kannan, Ravishekar, "High Order Spectral Volume and Spectral Difference Methods on Unstructured Grids" (2008). *Graduate Theses and Dissertations*. 11264.
https://lib.dr.iastate.edu/etd/11264

High order spectral volume and spectral difference methods on unstructured grids

by

Ravishekar Kannan

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:
Zhi Jian Wang, Major Professor
Tom Shih
Paul Durbin
Richard Pletcher
Hailiang Liu

Iowa State University

Ames, Iowa

2008

DEDICATION

I would like to dedicate this thesis to my parents.

TABLE OF CONTENTS

LIST OF FIGURES......vi

LIST OF TABLES......xiii

LIST OF SYMBOLS......xiv

ACKNOWLEDGEMENTS......xix

ABSTRACT......xx

CHAPTER 1. INTRODUCTION......1
1.1 Motivation......2
1.2 Objectives of the present study......6
1.2.1 Primary objective......6
1.2.2 Secondary objective......7
1.3 Outline of the thesis......7

CHAPTER 2. INTRODUCTION TO THE SPECTRAL DIFFERENCE AND SPECTRAL VOLUME METHODS......9
2.1 Introduction......9
2.2 The SD formulation......11
2.2.1 Location of flux and solution points......15
2.3 The SV formulation......15
2.3.1 Basic formulation......15
2.3.2 Discretization of viscous fluxes......17
2.3.3 Curved boundary implementation......18
2.3.4 Simplified curved boundary formulation......21

CHAPTER 3. TIME INTEGRATION ALGORITHMS......23
3.1 Introduction......23
3.2 The explicit method 25
3.3 The implicit method......25

CHAPTER 4. THE P-MULTIGRID METHOD......29
4.1 Introduction......29
4.1.1 An example of geometric multigrid......30
4.2 The p-multigrid method......31
4.2.1 P-Multigrid for higher order methods......31
4.2.2 Implementation of the p-multigrid algorithm......34
4.2.3 V cycles and the full multigrid (FMG)......38

CHAPTER 5. VISCOUS FLUX FORMULATIONS FOR THE SV METHOD..........40
5.1 Introduction..........40
5.2 The three viscous flux formulations for the SV method..........41
5.2.1 The LDG formulation..........41
5.2.2 The penalty formulation..........43
5.2.3 The second approach of Bassi and Rebay(BR2)..........45
5.3 Analysis for uniform grids..........47
5.3.1 One dimensional analysis..........47
5.3.1.1 Spatial analysis..........49
5.3.1.2 Temporal analysis..........52
5.3.2 Two dimensional analysis..........53
5.3.2.1 Spatial analysis..........55
5.3.2.2 Temporal analysis..........57
5.4 Analysis for non uniform grids..........57
5.5 Discussion on stability and spurious modes..........58
5.6 Dissipative properties using multigrid techniques..........61

CHAPTER 6. RESULTS AND DISCUSSIONS 65
6.1 SD results..........65
6.1.1 Validation using 2D conservation laws..........65
6.1.1.1 Linear straight wave..........65
6.1.1.2 Rotational wave..........66
6.1.1.3 Nonlinear wave..........69
6.1.2 Results for the Euler equations..........72
6.1.2.1 2D supersonic vortex flow..........72
6.1.2.2 2D subsonic flow over a bump..........75
6.1.2.3 2D subsonic flow over an airfoil..........79
6.2 SV results..........83
6.2.1 2D diffusion equation..........83
6.2.1.1 Accuracy study..........83
6.2.1.2 Convergence study 1..........86
6.2.1.3 Convergence study 2..........88
6.2.2 2D Navier Stokes equations..........88
6.2.2.1 The Couette flow problem..........88
6.2.2.2 Flow over an airfoil..........92
6.2.2.3 Laminar flow over a flat plate..........97
6.2.2.4 Turbulent flow over a flat plate..........100

CHAPTER 7. CONCLUSIONS AND FUTURE WORK..104

BIBLIOGRAPHY...107

LIST OF FIGURES

Figure 1.1 2nd order finite volume mesh (a): (192x64x2) and 4th order SV mesh (b) :(48x16x2). Case (a) has 24576 DOF and Case (b) has 15360 DOF. Taken from Wang et al [76]. 3

Figure 1.2 Entropy errors obtained using the 2nd order finite volume and the 4th order SV methods on meshes from Figure 1.1(a) and 1.1(b) respectively. 4

Figure 1.3 Error levels attained by different grids for the Couette flow problem..............................4

Figure 2.1 Placement of unknown (●) and flux (■) points for a triangular element. Case (a): First order; Case (b): Second order; Case (c): Third order. 13

Figure 2.2: Flux computation for a corner (●) and an edge point (■) using one-dimensional Riemann solvers. 14

Figure 2.3 Partitions of a triangular SV . Linear, quadratic and cubic reconstructions are shown in a, b and c respectively. 16

Figure 2.4 Transformation of a general SV in the physical domain to the standard triangle in the computational. Taken from Wang et al [76] 19

Figure 2.5 The partition in the standard triangle is transformed to the physical domain using an inverse transformation. Taken from Wang et al [76] 20

Figure 2.6 Simplified quadratic and cubic SVs with only one curved face . Taken from Wang et al [76] 21

Figure 4.1 Error distribution on the fine and coarse grids for the 1D Laplace problem. 30

Figure 5.1 1D SV with 2 CVs……………………………………………42

Figure 5.2 Compact stencil in 1D……………………………………………44

Figure 5.3 Non Compact stencil in1D……………………………………………45

Figure 5.4 Two SVs having a common SV face f. ……………………………………………46

Figure 5.5 Linear spectral volume in 1D……………………………………………48

Figure 5.6 Fourier footprint for 2nd order scheme……………………………………………49

Figure 5.7 Error associated at low wavenumbers for the 2nd order scheme……………………………………………50

Figure 5.8 Fourier footprint for 3rd order scheme……………………………………………50

Figure 5.9 Error associated at low wavenumbers for the 3rd order scheme……………………………………………51

Figure 5.10 Fourier footprint for 4th order scheme……………………………………………51

Figure 5.11 Error associated at low wavenumbers for the 4th order scheme……………………………………………52

Figure 5.12 Basic Building block colored in red; the neighbors are used in analysis……………………………………………54

Figure 5.13 Fourier footprint for the 2D 2nd order scheme for θ = 0 radians……………………………………………55

Figure 5.14 Fourier footprint for the 2D 2nd order scheme for θ = π/4 radians……………………………………………55

Figure 5.15 Error associated at low wavenumbers for the 2D 2nd order scheme for θ = 0 radians……………………………………………56

Figure 5.16 Error associated at low wavenumbers for the 2D 2nd order scheme for θ = π/4 radians……………………………………………56

Figure 5.17 Basic Building block for non uniform grids in 1D: consists of 2 SVs……………………………………………58

Figure 5.18 Eigen values of a 2nd order LDG growth matrix for RK 3stage scheme. Case (a): non dimensional time step =0.157, case(b): non dimensional time step =0.138 59

Figure 5.19 Damping occurring in a 2nd order LDG simulation using two different time steps 60

Figure 5.20 Eigen values of a 2nd order LDG growth matrix for Crank-Nicolson scheme. Case (a): non dimensional time step=0.5, case(b): non dimensional time step =0.25 61

Figure 5.21 Figure 5.21 Damping factor for a 3-2 mg using the L_2 norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post sweeps 63

Figure 5.22 Figure 5.21 Damping factor for a 3-2 mg using the L_{inf} norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post sweeps 63

Figure 5.23 Figure 5.21 Damping factor for a 4-3 mg using the L_2 norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post sweeps 63

Figure 5.24 Figure 5.21 Damping factor for a 4-3 mg using the L_{inf} norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post sweeps 64

Figure 6.1: Linear wave equation case. Case (a): Grid 40x40x2; Case (b): Contour Plot........65

Figure 6.2: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the 2D Linear wave case 66

Figure 6.3: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the 2D circular convection case 68

Figure 6.4: Contour plot obtained for 2D scalar circular convection case. 68

Figure 6.5: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the non linear wave case. 70

Figure 6.6: Non Linear wave case using a LU-SGS implicit smoother. Case (a): Convergence history as a function of Multigrid cycles. Case (b): Convergence History as a function of time....70

Figure 6.7: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the non linear wave case....71

Figure 6.8: Supersonic vortex flow case. Case (a): Computational grid. Case (b): Computed pressure contours. 74

Figure 6.9: Subsonic flow over a bump confined in a channel. Case (a): Computational grid. Case (b): Computed pressure contours. 75

Figure 6.10: Comparison of convergence history with respect to time for the bump testing case using explicit R-K and implicit LU-SGS schemes. 77

Figure 6.11: 2D subsonic flow over a bump case using implicit LU-SGS smoother. Case (a): Convergence history as a function of Multigrid cycles. Case (b): Convergence History as a function of time. 78

Figure 6.12: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the bump testing case....79

Figure 6.13: Grid(72*24*2) used for the subsonic flow over the NACA 0012 airfoil....80

Figure 6.14: Pressure Contours obtained for the subsonic flow over the NACA 0012 airfoil....80

Figure 6.15: Comparison of convergence history with respect to time for the airfoil testing case using explicit R-K and implicit LU-SGS schemes........81

Figure 6.16: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the airfoil testing case........82

Figure 6.17: Flow over an airfoil case. Case (a): Convergence history as a function of Multigrid cycles for two and three levels of FMG. Case (b): Convergence History as a function of time using two and three levels of FMG. 82

Figure 6.18 Regular grid(10x10x2) used for validation study........86

Figure 6.19. 4th order solution contours of the equation 5.1 using a 40x40x2 grid........86

Figure 6.20 Convergence history using explicit R-K and implicit LU-SGS schemes for the 2D diffusion equation using the penalty scheme. Case (a): 2nd order; Case (b): 3rd order; Case (c): 4th order........86

Figure 6.21 Convergence history using single level and multilevel FMG LU-SGS schemes for the 2D diffusion equation using the penalty scheme. Case (a): 3rd order; Case (b): 4th order. 87

Figure 6.22 Convergence history using the three formulations for the 2D diffusion equation using a 2nd order scheme........88

Figure 6.23. Computational domain for the Couette Flow problem. 89

Figure 6.24. Convergence history as a function of time for the Couette flow problem. Case (a): 3rd order; Case (b): 4th order........ 91

Figure 6.25. Convergence history as a function of iterations for the Couette flow problem. Case (a): 3rd order; Case (b): 4th order.. 92

Figure 6.26. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over naca0012 airfoil using the penalty scheme. Case (a): 2ndorder; Case (b): 3rd order; Case (c): 4th order.. 94

Figure 6.27. Convergence history as a function of finest order iterations (LU-SGS) scheme using single level and FMG schemes for the flow over a naca0012 airfoil. Case (a): 3rd order; Case (b): 4th order... 95

Figure 6.28. Convergence history using single level and FMG schemes for the flow over naca0012 airfoil. Case (a): 3rd order; Case (b): 4th order.. 96

Figure 6.29. Mach contours around the NACA0012 airfoil at zero degree of attack, Re=5000, M=0.5. Case (a): 2ndorder; Case (b): 3rd order.. 96

Figure 6.30 Convergence history using explicit R-K and implicit LU-SGS schemes for the laminar flow over a flat plate using the penalty scheme. Case (a): 2ndorder; Case (b): 3rd order; Case (c): 4th order... 97

Figure 6.31 Convergence history using single level and FMG schemes for the laminar flow over a flat plate using the penalty scheme. Case (a): 3rd order; Case (b): 4th order.......................... 98

Figure 6.32 Convergence history for the 3rd order multigrid system FMG scheme for laminar flow over a flat plate. Case (a): Convergence as a function of time; Case (b): Convergence as a function of highest order iterations.. 99

Figure 6.33 Convergence history for the 4th order multigrid system FMG scheme for laminar flow over a flat plate. Case (a): Convergence as a function of time; Case (b): Convergence as a function of highest order iterations.. 100

Figure 6.34 Variation of yplus versus uplus for turbulent flow over a flat plate. The simulations are compared with the indirect turbulence model... 103

LIST OF TABLES

Table I. Maximum non-dimensional time step for obtaining stable solutions for the LDG, penalty and BR2 methods. 53

Table II. Maximum non-dimensional time step for obtaining stable solutions for the 2nd order, 2D LDG, penalty and BR2 methods 57

Table III. Maximum non-dimensional time step for obtaining stable solutions for the LDG, penalty and BR2 methods for different clustering factors 58

Table IV. Non dimensional time step criteria for obtaining stable and physically relevant solutions for the LDG, penalty and BR2 schemes. 60

Table V. L_1 and L_∞ errors and orders of accuracy of the scalar circular advection case using the third order SD method. 67

Table VI. L_1 and L_2 errors and orders of accuracy of the supersonic vortex flow case using the third order SD method. 74

Table VII. L_1 and L_2 errors and orders of accuracy of the supersonic vortex flow case using the second order SD method. 75

Table VIII. Accuracy of 2nd order 2D diffusion equation on regular grids 84

Table IX. Accuracy of 3rd order 2D diffusion equation on regular grids 84

Table X. Accuracy of 4th order 2D diffusion equation on regular grids 85

Table XI. Temperature error for the Couette flow problem (3rd order) 94

Table XII. Temperature error for the Couette flow problem (4th order) 94

Table XIII. Comparison of pressure and viscous drag coefficients and location of separation point between current simulations and other existing simulations in the literature........ 93

LIST OF SYMBOLS

A_r	=	r^{th} face of C_{ij}
C_{ij}	=	Control volume j of spectral volume i
$\Delta\tilde{Q}_c$	=	$\tilde{Q}_c^{n+1} - \tilde{Q}_c^n$
d_p	=	Defect at the finest level
d_{p-1}	=	Defect at the intermediate level
D	=	Domain in the standard coordinate space
E	=	Total energy
$\tilde{f}$	=	Vector of inviscid fluxes in the x direction
$\tilde{f}_v$	=	Vector of viscous fluxes in the x direction
F_n	=	Unique normal components of the two fluxes, i.e. F_{n1} and F_{n2}
F_{n1}	=	Normal component of the flux obtained from the left cell
F_{nR}	=	Normal component of the flux obtained from the right cell
$\tilde{g}$	=	Vector of inviscid fluxes in the y direction
$\tilde{g}_v$	=	Vector of viscous fluxes in the y direction
γ	=	Ratio of specific heats for air
G	=	Amplification matrix
H	=	Growth matrix
I_p^{p-1}	=	Restriction operator used to restrict from polynomial space p to polynomial space p-1

I_{p-1}^{p-2} = Restriction operator used to restrict from polynomial space p-1 to polynomial space p-2

I_{p-1}^{p} = Prolongation operator used to prolongate from polynomial space p-1 to polynomial space p

I_{p-2}^{p-1} = Prolongation operator used to prolongate from polynomial space p-2 to polynomial space p-1

$L_{j,i}(r)$ = Lagrange-type polynomial basis function constructed using the solution values at solution points (in spectral difference)

$-$ = Refers to the neighboring cell

μ = Positive diffusion coefficient

M_i = Mach number at the inner arc for the 2D supersonic vortex problem

$M_{k,i}(r)$ = The set of shape functions defined uniquely by the flux point locations (in spectral difference)

$\vec{n}$ = Outward unit normal vector of A_r

Nb = All the neighboring cells contributing to the residual of cell *c*

$\vec{n}^+$ = Outward normal w.r.t the V^+ SV for the face in consideration

$\vec{n}^-$ = Outward normal w.r.t the V^- SV for the face in consideration

$\tilde{\nu}$ = Principal unknown variable to be solved for in the SA model.

ν_t = Turbulent viscosity used in the SA model

N_p = Number of unknown points required to support a degree p polynomial construction in 2D

$+$ = Refers to the current cell

$r_{j,i}$ = Vector of solution points for cell i (in spectral difference)

$r_{k,i}$ = Vector of flux points for cell i (in spectral difference)

$R_c(\tilde{Q})$ = Vector of residuals

ρ = Density

$\tilde{Q}$ = Vector of conservative variables

u = Velocity component in the x direction

$\underline{u}$ = Numerical solution at the SV boundaries for the 2D diffusion equation

$\overline{u}_{ij}$ = CV averaged solution in C_{ij} for the 2D diffusion equation

u_L = Left state solution of the face in consideration for the 2D diffusion equation

u_R = Right state solution of the face in consideration for the 2D diffusion equation

v = Velocity component in the y direction

$\underline{\vec{q}}$ = Numerical gradient at the SV boundaries for the 2D diffusion equation

$\overline{q}_{ij}$ = CV averaged gradient in C_{ij} for the 2D diffusion equation

q_L = Left state solution of the face in consideration for the 2D diffusion equation

q_R = Right state solution of the face in consideration for the 2D diffusion equation

$\tilde{Q}$ = Vector of conservative variables

$\tilde{Q}^*$ = Most current value of $\tilde{Q}$

$\overline{Q}_{i,j}$ = Control volume averaged solution of C_{ij}

Q_{p-1}^0 = Restricted value of Q_p to the coarser level

Q_L	=	Solution on the left side of the face
Q_R	=	Solution on the right side of the face
$\vec{r}^{+}$	=	Lifting function obtained from the current cell
$\vec{r}^{-}$	=	Lifting function obtained from the neighboring cell
$\overline{\vec{r}^{+}}$	=	CV averaged lifting functions in the current SV
$\overline{\vec{r}^{-}}$	=	CV averaged lifting functions in the neighboring SV
r_i	=	Radius at the inner arc for the 2D supersonic vortex problem
S_i	=	Spectral volume i
τ_{BR2}	=	Maximum allowable (stable) time step for the BR2 scheme
τ_{LDG}	=	Maximum allowable (stable) time step for the LDG scheme
$\tau_{penalty}$	=	Maximum allowable (stable) time step for the Penalty scheme
V_{ij}	=	Volume of C_{ij}
V+	=	Volume of CVs in the current SV
V-	=	Volume of CVs in the neighboring SV
χ	=	Non-dimensional frequency (used in SV analysis)

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Z. J. Wang, for introducing me to the research arena and for his constant support and his relentless motivation during this endeavor. It was due to him that I could surmount the various obstacles in my path. Without his perpetual encouragement, this project would have lagged by eons.

This work was funded by the Air Force Office of Scientific Research (Grant Number FA9550-04-1-0053). I am grateful to the Technical Monitor Dr. Fariba Fahroo for her support during the last two years. I am also grateful to this country for providing me with a platform for education and scientific research.

I would like to thank my committee members Drs. Shih, Durbin, Pletcher and Liu for their valuable time and for their constructive comments and suggestions. I would also like to thank the scholars in the CFD lab at ISU and my close friends for their invaluable help during the course of my research.

This research could not have been successful without the basic knowledge of heat transfer, fluids and computing. I am thankful to my undergraduate faculty in Indian Institute of Technology Madras (IITM), India and the graduate faculty in MSU for strengthening my foundation. Last but definitely not the least, I thank my family for their help and continuous support throughout my career.

ABSTRACT

The spectral volume (SV) and the spectral difference (SD) methods were developed by Wang and Liu and their collaborators for conservation laws on unstructured grids. They were introduced to achieve high-order accuracy in an efficient manner. Recently, these methods were extended to three-dimensional systems and to the Navier Stokes equations. The simplicity and robustness of these methods have made them competitive against other higher order methods such as the discontinuous Galerkin and residual distribution methods.

Although explicit TVD Runge-Kutta schemes for the temporal advancement are easy to implement, they suffer from small time step limited by the Courant-Friedrichs-Lewy (CFL) condition. When the polynomial order is high or when the grid is stretched due to complex geometries or boundary layers, the convergence rate of explicit schemes slows down rapidly. Solution strategies to remedy this problem include implicit methods and multigrid methods. A novel implicit lower-upper symmetric Gauss-Seidel (LU-SGS) relaxation method is employed as an iterative smoother. It is compared to the explicit TVD Runge-Kutta smoothers. For some p-multigrid calculations, combining implicit and explicit smoothers for different p-levels is also studied. The multigrid method considered is nonlinear and uses Full Approximation Scheme (FAS). An overall speed-up factor of up to 150 is obtained using a three-level p-multigrid LU-SGS approach in comparison with the single level explicit method for the Euler equations for the 3rd order SD method.

A study of viscous flux formulations was carried out for the SV method. Three formulations were used to discretize the viscous fluxes: local discontinuous Galerkin (LDG), a penalty method and the 2nd method of Bassi and Rebay. Fourier analysis revealed some interesting advantages for the penalty method. These were implemented in the Navier Stokes solver. An implicit and p-multigrid method was also implemented for the above. An overall speed-up factor of up to 1500 is obtained using a three-level p-multigrid LU-SGS approach in comparison with the single level explicit method for the Navier-Stokes equations. The SV method was also extended to turbulent flows. The RANS based SA model was used to close the Reynolds stresses. The numerical results are very promising and indicate that the approaches have great potentials for 3D flow problems.

CHAPTER 1. INTRODUCTION

Computational fluid dynamics (CFD) has been used to simulate practical flow problems for three decades. The performance of numerical methods in CFD tools with respect to both reliability and accuracy has been the main area of concern. Extensive research efforts have led to the development of more accurate and reliable methods. Various research projects have been carried out in the last decade to develop and improve high order accurate algorithms for conservation laws. These endeavors have led to the development of many methods including the k-exact finite volume [7,23] methods, the essentially non-oscillatory (ENO) [29,1] methods, weighted ENO [33,85] methods, discontinuous Galerkin method [18-21], residual distribution method [25] and the spectral volume (SV) [77-81] and spectral difference (SD) [42,82,83] methods. In spite of the above developments, most of the current calculations in computational aerodynamics, aero-acoustics and CFD are performed using methods that are at most 2nd order accurate. Not only are high order methods more accurate, they also have the potential to reduce the CPU time required to obtain the solutions. The above serves as the most important motivational argument for conducting the present research. The goal of this thesis is to take part in the development of higher order methods for conservation laws on unstructured grids. In particular, the thesis deals with the SV and SD methods of discretization of the wave, the Euler and the Navier Stokes equations, development of an implicit and p-multigrid for them, obtaining stable, accurate and robust viscous flux formulations, a stability analysis for the above and finally extending it to turbulent flows.

1.1 Motivation

CFD has undergone an explosive growth in the last two to three decades. This growth was facilitated by increases in computational resources, progresses in geometric modeling and grid generation and adaptation, improvements in physical and turbulence models and better and faster numerical methods. Most of the current commercial CFD packages are now capable of simulating complex large scale problems. These packages use the standard finite volume approach for discretizing the Euler and the Navier-Stokes equations. Upwinding [71, 56,72,57,74] was integrated into these algorithms in the 1980's. Improved techniques for viscous flows gave the solvers enough robustness to handle complex ”real world” applications. Major advances in the applications of finite volume methods to RANS (Reynolds –Averaged Navier-Stokes) based simulations happened in the 1990's. The difficulty in generating smooth structured grids for complex geometries has prompted the research and development of unstructured grid algorithms for the last 2 decades [5,53,45,48]. In spite of all these advances, most of the algorithms used are at best second order accurate. In other words, the error decreases as O (h^2), with h being the mesh size

Even though second order methods have achieved successes, there exist several areas where their performance is unsatisfactory, e.g., for vortex-dominated flows including helicopter blade vortex interactions and flow over high lift configurations. Higher order methods are more successful in handling these flows. For example, high order compact methods have demonstrated much better results than their lower order counterparts [76].

The advantage of higher order methods is evident. By definition, the error of a numerical scheme is said to be of order k+1 if the solution error e decreases with the mesh size h according to the following law:

$$e = ch^{k+1} \tag{1.1}$$

where c is the constant of proportionality. Thus for a third order scheme, if the grid size halves, the error should be 1/8 of that on the coarse mesh. In general, a higher order method could reduce the degrees of freedom (DOFs) required to obtain a desired accuracy level. This corresponds to potentially obtaining the desired result in a shorter time span. Figure 1.1 shows the 2nd order finite volume and 4th order spectral volume meshes around an airfoil [76]. The 2nd order simulation has more DOFs than the 4th order solution. Figure 1.2 shows the entropy error associated with the above methods. The 4th order simulation yielded errors much lower than the 2nd order simulation.

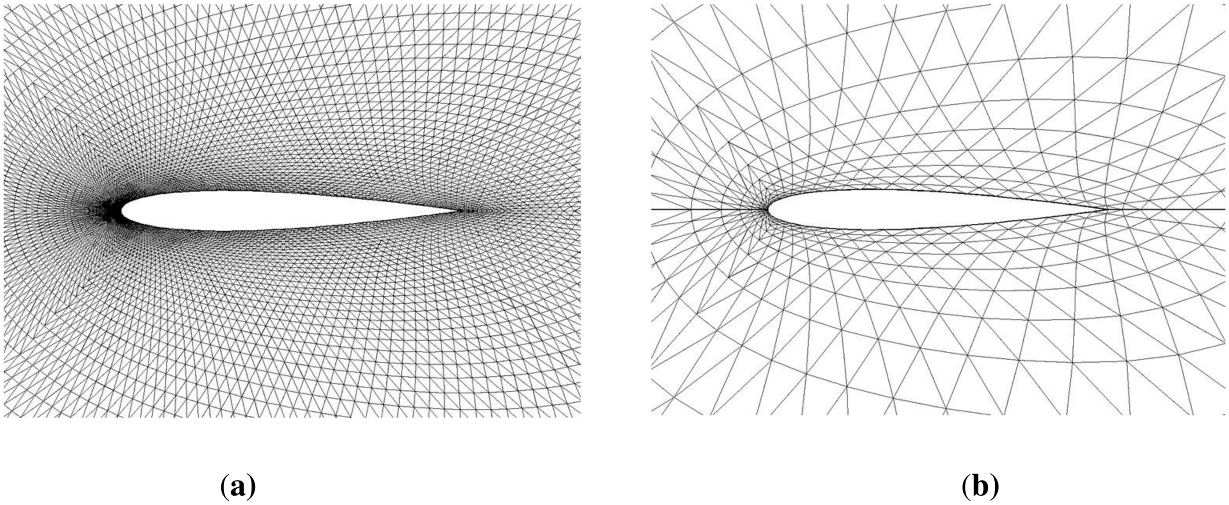


Figure 1.1. 2nd order finite volume mesh (a): (192x64x2) and 4th order SV mesh (b) :(48x16x2). Case (a) has 24576 DOF and Case (b) has 15360 DOF. Taken from Wang et al [76].

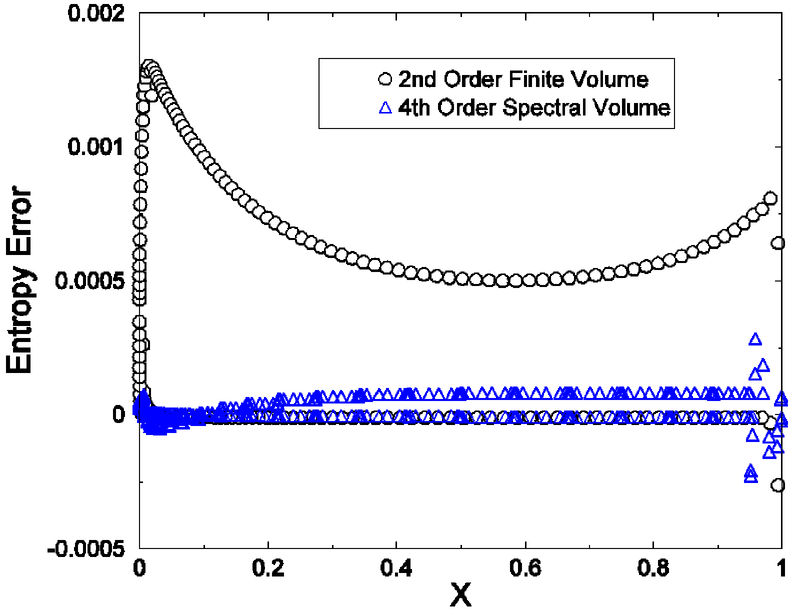


Figure 1.2. Entropy errors obtained using the 2^{nd} order finite volume and the 4^{th} order SV methods on meshes from Figure 1.1(a) and 1.1(b) respectively.

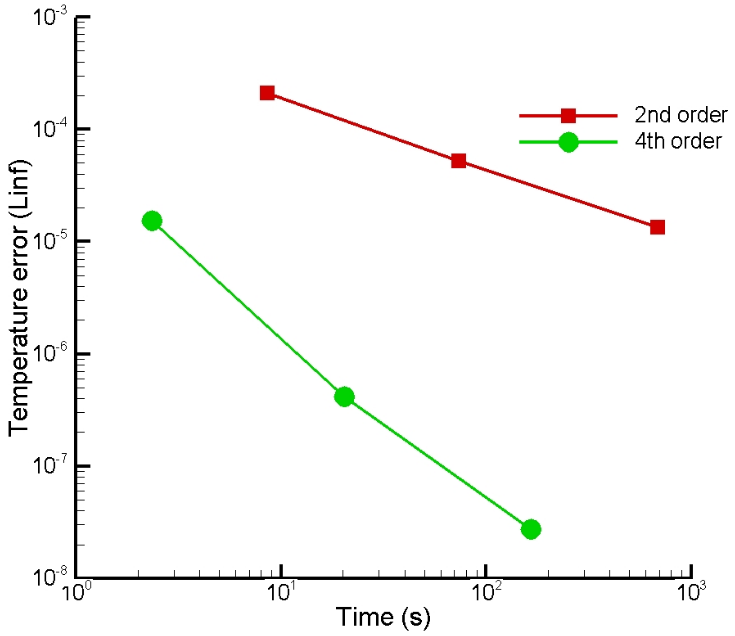


Figure 1.3. Error levels attained by different grids for the Couette flow problem

In many cases, higher order methods also consume lesser CPU time (for a given error) than their lower order counterparts. Figure 1.3 shows the error associated with successively refined grids and their corresponding time using the 2nd order FV and the 4th order SV method for the Couette flow problem. It can be seen that the 4th order SV method can attain a much lower error level than the 2nd order FV if both of them take nearly the same time to converge.

In theory, the traditional finite volume methods can be used to obtain high-order accuracy. This can be done by using an extended stencil. However it is difficult to obtain a stable scheme using the above approach. In addition, it requires enormous memory to store the reconstruction coefficients, especially in three dimensions. In a recent implementation of a third order finite volume method with a quadratic reconstruction by Delanaye and Liu [23], the average size of the reconstruction stencils was around 50-70. The average stencil size for a fourth order finite volume scheme is estimated to be around 120. Moreover it is tough to implement the above approach on unstructured grids. An alternative approach namely the discontinuous Galerkin (DG) method [18, 19 and 22] has been in use for the last decade and a half. It has had success in handling complex flows. Nevertheless, its implementation on unstructured grids is quite complex, involving both volume and surface integrals.

Wang and Liu came up with the spectral volume [77,78,79,80,81] and spectral difference [42,82,49] methods during the early 2000's. They share many similar properties with the DG method, such as discontinuous solution space and compactness. They mainly differ on how the DOFs are updated. The DG being a derivative of the finite element method has unknown the elemental nodal values as DOF. The spectral volume being a derivative of the finite volume has subcell averages as its DOF while the spectral difference has point wise values as DOF. In terms of complexity, DG requires both volume and surface integrations. In contrast, SV requires only surface integrations and the SD requires differentiations. They extended these methods to three –dimensional hyperbolic systems on unstructured tetrahedral meshes. Recently Sun et al. [68] implemented the method for the laminar Navier-Stokes equations. This thesis deals exclusively with these two methods.

1.2 Objectives of the present study

1.2.1 Primary objective

The recent efforts in the development of the spectral volume and difference methods are primarily focused on the spatial discretization. The temporal discretization methods have lagged behind. Usually, explicit temporal discretizations such as multi-stage SSP (Strong Stability Preserving) Runge–Kutta schemes are used to advance the solution in time. In general, explicit schemes and their boundary conditions are easy to implement, vectorize and parallelize, and require only limited memory storage. However, for large-scale simulations and especially for high-order solutions, the rate of convergence slows down dramatically, resulting in inefficient solution techniques to steady state solution. In order to speed up convergence and to improve the robustness, a multigrid strategy and/or an implicit temporal discretization is required.

The primary objective of this thesis is to develop a fast, low storage p-multigrid method for the spectral volume and spectral difference methods to solve the compressible Euler (spectral difference) and Navier Stokes (spectral volume) equations on unstructured grids. Explicit SSP multi-stage Runge–Kutta method and implicit lower–upper symmetric Gauss–Seidel (LU-SGS) method are implemented, used, and discussed as iterative smoothers. Unlike the traditional p-multigrid methods where the same time integration scheme is used on all approximation levels, we sometimes implement the multi-stage Runge–Kutta scheme as the iterative smoother on the higher level approximations, and the LU- SGS method as the iterative smoother on the lowest level approximation in an attempt to significantly reduce the storage requirements

1.2.2 Secondary objective

While the spectral volume method has been well studied for the discretization of first order hyperbolic problems, their extension to elliptic problems is less obvious. The implementation of the local discontinuous Galerkin (LDG) method by Sun et al [68] has been the only significant work in this area. More recent numerical tests indicated that the computational results using LDG are somewhat dependent on how the faces are oriented, especially for unstructured and non uniform grids.

The secondary objective of this thesis is to test another technique to discretize the viscous fluxes. The new procedure is a penalty-like method based on the algorithm developed by Bassi and Rebay. This procedure is more symmetrical than LDG and hence is better suited for unstructured and non uniform grids. Fourier analysis revealed better accuracy, higher time step limit (for explicit simulations) and faster convergence for the penalty approach when compared to LDG. These techniques are also compared with the 2nd approach of Bassy and Rebay [11].

1.3 Outline of the thesis

This thesis is organized as follows: Chapter 2 gives a basic introduction to the spectral difference and spectral volume methods. This includes some historical developments prior to the SD and SV methods, their formulation and presentation of the governing equations.

Temporal relaxation schemes are discussed in chapter 3. In chapter 4, we focus on the nonlinear FAS p-Multigrid method. The various viscous flux formulations are presented in chapter 5. This includes detailed linear analysis for the stability, accuracy, convergence and multigrid convergence properties. Chapter 6 has results for the SD and SV formulations. This includes validation test cases using 1D wave propagations and diffusion equations and solutions to Euler (SD) and Navier-Stokes (SV) equations. Chapter 7 summarizes the entire work done during the doctoral stage and suggests possible work for the future.

CHAPTER 2. INTRODUCTION TO THE SPECTRAL DIFFERENCE AND SPECTRAL VOLUME METHODS

2.1 Introduction

Wang and Liu et al [77-81,42,82,49] came up with the concept of spectral difference and spectral volume methods in the early 2000's. They were developed to solve conservation laws with the following properties:

A. Conservative

B. High-order accurate

C. Geometrically flexible

D. Computationally efficient

E. Simply formulated.

The earliest, easiest and most widely used numerical method for conservation laws is the finite-difference (FD) method employing a body-fitted curvilinear coordinate system. The spatial differences are essentially one dimensional and the spatial differencing is carried out along coordinate directions. It is imperative that the stencil be modified with uni-sided formulas near boundaries. The unknowns are the solution values at the grid nodes. The differencing can degrade the accuracy if the grid is not sufficiently smooth. In addition, the integral conservation laws are only satisfied to second order accuracy.

Finite volume methods were developed to satisfy the integral conservation laws. The unknowns (or DOFs) are the cell averaged quantities. The cell averages of the neighboring cells are used to compute the gradients. This process is called reconstruction. The flux integral over each control volume boundary is computed using Riemann solvers. The flux and the volume integrals are computed by a single point quadrature and hence are 2nd order accurate. In addition, the reconstructions done are along coordinate directions. In general, the finite volume and finite difference methods suffer loss of accuracy for curved and non smooth grids.

Unstructured grids comprising of triangles in 2D and tetrahedral in 3D are used to achieve geometric flexibility. Most commercial packages use the unstructured finite volume method for solving conservation laws. Theoretically, a polynomial reconstruction of any desired accuracy can be obtained using the neighboring cell averages. However, it is possible to obtain a singular stencil. In addition, the unstructured nature of the grid may warrant a least squares type approach. Since each cell employs a different stencil, storage becomes a costly issue, especially in three dimensions. As a result, most unstructured finite volume solvers are at most 2nd order accurate.

One could use the finite difference method on unstructured grids, with the conservative values at grid nodes acting as the unknowns. The reconstruction is local in terms of the neighboring values determined by the unknowns. However this method suffers from the disadvantages listed in the above paragraphs.

The spectral difference (SD) method combines the salient features of structured and unstructured grid methods to achieve high computational efficiency and geometric flexibility. It utilizes the concept of discontinuous and high-order local representations to achieve conservation and high accuracy. Universal reconstructions are obtained by distributing unknown and flux points in a geometrically similar manner for all unstructured cells. In this thesis, we consider first order (*p0*), second order (*p1*) and third order (*p2*) schemes. The unknowns are updated using the differential form of the conservation law equations by approximating the flux derivatives at these unknown points. In order to obtain the flux derivatives, we use a polynomial reconstruction of the fluxes from their values at available flux points to the unknown points. As a result, the method is defined as a difference method. The SD method is similar to the multi-domain spectral method developed by Kopriva [37, 38] and can be viewed as an extension of the multi-domain spectral method to a simplex unstructured grid. Details of the SD discretization procedure are given in the next section.

Spectral volume is a numerical method for conservation laws. It has all the desired properties of a higher order scheme. The spectral volume (SV) method can be interpreted as a derivative of the finite volume method where each cell (called spectral volume) is sub divided into a structured set of smaller subcells(called control volume). This scheme exhibits many of the desirable properties of the Discontinuous Galerkin (DG) method. It achieves high-order accuracy on unstructured grids and has a compact stencil (as each SV only communicates only with its nearest neighbors), thus making it suitable for parallel applications. As the DOFs are the control volume averages, the limiting phenomenon can be applied at a control volume level. Thus the SV scheme can attain a higher resolution for discontinuities than the DG method. In this thesis, we consider 2nd, 3rd and 4th order SV schemes. Details of the SV discretization procedure are given in the subsequent sections.

2.2 The SD formulation

We could write most linear or nonlinear equations in 2D conservative form

$$\frac{\partial \tilde{Q}}{\partial t}+\frac{\partial(\tilde{f}(\tilde{Q})-\tilde{f}_v(\tilde{Q}))}{\partial x}+\frac{\partial(\tilde{g}(\tilde{Q})-\tilde{g}_v(\tilde{Q}))}{\partial y}=0 \quad \text{and} \quad R_c(\tilde{Q})=-\frac{\partial(\tilde{f}-\tilde{f}_v)}{\partial x}-\frac{\partial(\tilde{g}-\tilde{g}_v)}{\partial y}. \tag{2.1}$$

If the unsteady 2D Euler equations are considered, $\tilde{Q}$ is a vector of conservative variables, the viscous fluxes are zero vectors and $\tilde{F}$ and $\tilde{G}$ are the inviscid fluxes and are given by

$$\tilde{Q}=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{Bmatrix}, \quad \tilde{f}=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{Bmatrix}, \quad \tilde{g}=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{Bmatrix}, \tag{2.2}$$

where 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 $E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2)$ with constant ratio of specific heats $\gamma=1.4$ for air.

The current cell residual term $R_c(\tilde{Q})$ can be evaluated once the neighboring three cells are known. We can denote the unknown and flux points for cell i as $r_{j,i}$ and $r_{k,i}$ respectively. The solutions of $\tilde{Q}$ at flux points can be conveniently constructed using a Lagrange-type polynomial basis function $L_{j,i}(r)$ as

$$\tilde{Q}(r_{k,i})=\sum_{j=1}^{N_p}L_{j,i}(r_{k,i})\tilde{Q}_{j,i}, \tag{2.3}$$

where $N_p=\frac{(p+1)(p+2)}{2}$ is the number of unknown points required to support a degree p polynomial construction as already illustrated in figure 2.1. As a result, $\tilde{Q}(r_k)$ is continuous inside individual cell element, while across the element interfaces, it is discontinuous and the inviscid fluxes $\tilde{f}(r_k)$ and $\tilde{g}(r_k)$ are not uniquely determined.

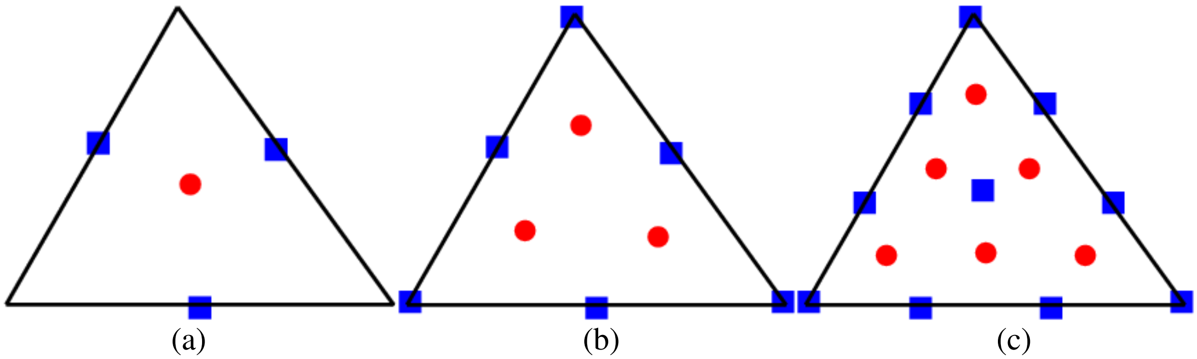


Figure 2.1: Placement of unknown (●) and flux (■) points for a triangular element. Case (a): First order; Case (b): Second order; Case (c): Third order.

We employ one-dimensional Riemann solvers, namely, the Rusanov [59] or Roe [56] flux to obtain a unique normal component of the flux $F(r_k)=\tilde{f}(r_k)+\tilde{g}(r_k)$ at the element boundary interface for an edge point. The local cell values are used for the tangential components as shown in figure 2.2, i.e. $(F(Q_L)\bullet l, F_n)$ and $(F(Q_R)\bullet l, F_n)$ for the left cell and right cell respectively. However, for the corner flux points, multiple values are allowed for different cells using the procedure as shown in [82]. In other words, two faces are associated with a particular corner point of a cell and we can use either the Rusanov or Roe flux to compute unique normal components of the two fluxes, i.e. F_{n1} and F_{n2}.

where F_{n1} and F_{nR} are the normal component of the flux obtained from the left and right cells respectively.

A flux vector $F(r_{k,i})$ at this corner point for this particular cell can be constructed using F_{n1} and F_{n2} i.e. two Riemann fluxes are computed (as there are two faces). The actual flux is obtained by the vectorial sum of the above.

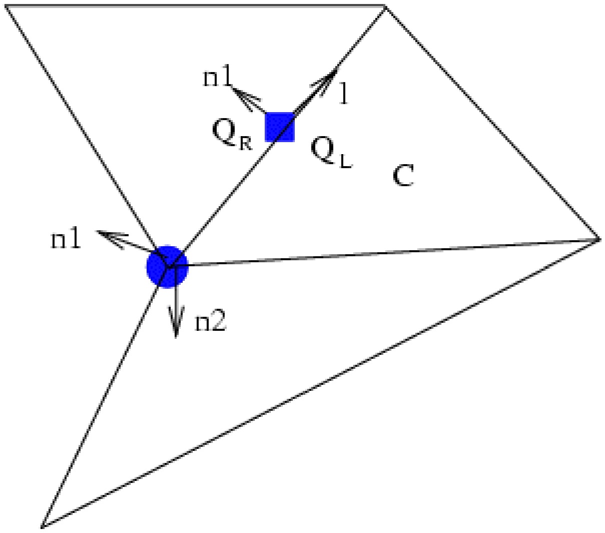


Figure 2.2: Flux computation for a corner (●) and an edge point (■) using one-dimensional Riemann solvers.

Once all the flux vectors are determined, they are used to form a degree *p+1* polynomial, i.e., one order higher than the polynomial used in (2.3). The flux at any location can be expressed as follows,

$$F_i(r) = \sum_{k=1}^{N_{p+1}} M_{k,i}(r) F_{k,i} \,, \tag{2.4}$$

where $M_{k,i}(r)$ are the set of shape functions defined uniquely by the flux point locations. We are now ready to compute the divergence of the flux at any locations inside the particular cell and for the unknown point locations; they can easily computed according to

$$\nabla \bullet F_i(r_{j,i}) = \sum_{k=1}^{N_{p+1}} \nabla M_{k,i}(r_{j,i}) \bullet F_{k,i} \,. \tag{2.5}$$

The residual term $R_c(\tilde{Q})$ used in (2.1) is simply the negative value of the divergence. Once the SD spatial discretization is completed, temporal relaxation schemes are used to update the solution vector.

2.2.1 Location of flux and solution points

It is the location of the solution and the flux points is an important aspect of SD. If these points are distributed in a geometrically similar fashion, the formulas for the flux derivatives become universal and can be obtained as a weighted combination of the geometric variables and the fluxes. Here initial concepts and conditions are explained and several hypotheses are mentioned in brief. The locations of these points are determined by symmetry restrictions, order of the method and by stability limitations. In addition, to reduce the number of computations, one can relocate the solution points on the boundary to coincide with the Flux points. Moreover, it would be wise to reduce the number of flux points located on the boundary (as these require more Riemann computations). These suggestions should be implemented only after making sure that the resulting reconstruction matrix is non singular, well conditioned and the entire system is stable.

2.3 The SV formulation

2.3.1 Basic formulation

Equation (2.1) is considered in domain Ω with the appropriate initial and boundary conditions. Domain Ω is discretized into I non-overlapping triangular (2D) cells. In the SV method, the simplex grid cells are called SVs, denoted S_i, which are further partitioned into

CVs, denoted C_{ij}, which depend on the degree of the polynomial reconstruction. Figure 2.3 shows linear, quadratic and cubic partitions in 2D.

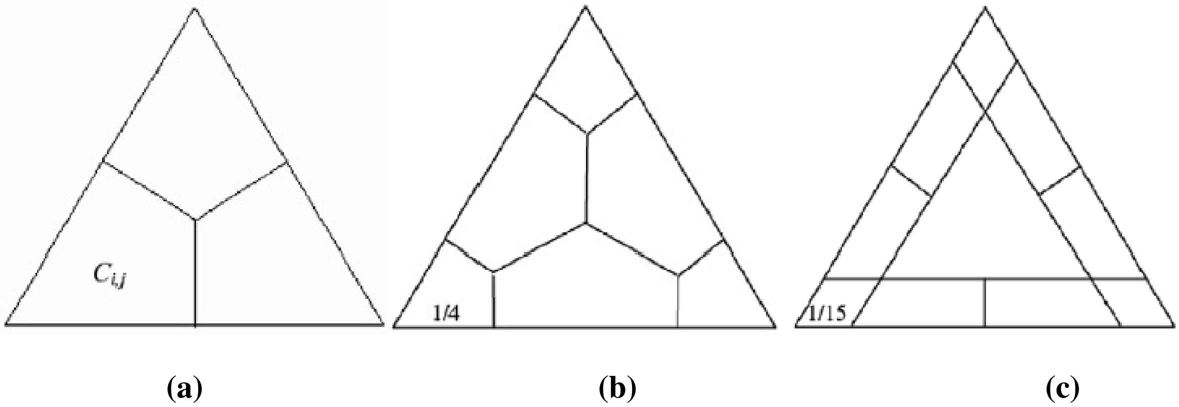


Figure 2.3 Partitions of a triangular SV . Linear, quadratic and cubic reconstructions are shown in Case (a), Case(b) and Case(c) respectively.

We need N unknown control volume solution averages (or DOFs). N_p is calculated using the below formula

$$N_p = \frac{(m+1)(m+2)}{2} \tag{2.6}$$

where m is the degrees of the polynomial, constructed using the CV solution averages. The CV averaged conserved variable for C_{ij} is defined as

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q dV \text{ , j=1…N}_p\text{, i=1…I,} \tag{2.7}$$

where $V_{i,j}$ is the volume of C_{ij}. Given the CV averaged conserved variables, a m^{th} degree polynomial can be constructed such that it is $(m+1)^{th}$ order approximation to $\tilde{Q}$. In other words, we can write the polynomial as

$$p_i(x, y) = \sum_{j=1}^{N_p} L_j(x, y)\overline{Q}_{i,j} \text{ ,} \tag{2.8}$$

where the shape functions $L_j(x,y)$ satisfy

$$\frac{1}{V_{i,j}}\int_{C_{i,j}} L_n(x,y)dV = \delta_{j,n}\,. \tag{2.9}$$

Equation 2.1 is integrated over the C_{ij}. This results in the below equation

$$\frac{\partial \overline{Q}}{\partial t} + \frac{1}{V_{i,j}}\sum_{r=1}^{K}\int_{Ar}(\vec{F}.\vec{n})dA = 0\,, \tag{2.10}$$

where $\vec{F} = (\tilde{f} - \tilde{f}_v, \tilde{g} - \tilde{g}_v)$ where A_r represents the r^{th} face of C_{ij}, $\vec{n}$ is the outward unit normal vector of A_r and K is the number of faces in C_{ij}. The surface integration on each face is done using a $(m+1)^{th}$ order accurate Gauss quadrature formula. The fluxes are discontinuous across the SV interfaces. The inviscid fluxes are handled using a numerical Riemann flux like the Rusanov flux [59], the Roe flux [56] or AUSM flux [41]. The handling of the viscous fluxes is discussed in the next section.

2.3.2 Discretization of viscous fluxes

The following 2D diffusion equation is considered first in domain Ω with the appropriate initial and boundary conditions

$$\frac{\partial u}{\partial t} - \nabla\cdot(\mu\nabla u) = 0\,, \tag{2.11}$$

where μ is a positive diffusion coefficient. We define an auxiliary variable

$$\vec{q} = \nabla u\,. \tag{2.12}$$

Equation 2.11 then becomes

$$\frac{\partial u}{\partial t} - \nabla\cdot(\mu\vec{q}) = 0\,. \tag{2.13}$$

Using the Gauss-divergence theorem, we obtain

$$\bar{\vec{q}}_{ij}V_{ij} = \sum_{r=1}^{K} \int_{Ar} u \cdot \vec{n} dA , \tag{2.14}$$

$$\frac{d\bar{u}_{ij}}{dt}V_{ij} - \sum_{r=1}^{K} \int_{A_r} \mu\vec{q} \cdot \vec{n} dA = 0 . \tag{2.15}$$

where $\bar{\vec{q}}_{ij}$ and $\bar{u}_{ij}$ are the CV averaged gradient and solution in C_{ij}, K is the number of faces in C_{ij}, A_r represents the r^{th} face of C_{ij} and $\vec{n}$ is the unit surface normal vector. As the solution u is cell-wise continuous, u and q at SV boundaries are replaced by numerical fluxes $\underline{\vec{q}}$ and $\underline{u}$. The above equations thus become

$$\bar{\vec{q}}_{ij}V_{ij} = \sum_{r=1}^{K} \int_{Ar} \underline{u} \cdot \vec{n} dA , \tag{2.16}$$

$$\frac{d\bar{u}_{ij}}{dt}V_{ij} - \sum_{r=1}^{K} \int \mu\underline{\vec{q}} \cdot \vec{n} dA = 0 . \tag{2.17}$$

Chapter 5 discusses the various viscous formulations available for $\underline{\vec{q}}$ and $\underline{u}$.

2.3.3 Curved boundary implementation

As mentioned in the earlier chapter, it is possible to achieve high-order with fewer DOF than with lower order methods. In all the traditional second order simulations, all the curved boundaries are represented with line-segments or planar facets (in 3D). This is compatible with the linear data interpolations used in second-order solvers. Thus in order to minimize the solution errors produced by this ‘‘crude’’ approximation of curved boundaries, many elements may be required to just preserve the geometry with a reasonable precision. If the above concept is used in high-order simulations, unnecessarily fine grids may be required near curved boundaries to represent the boundary with high fidelity. Obviously this practice can waste significant computational resources. In addition, the whole approach goes against the basic principle of high order methods i.e. to achieving high-order with fewer DOF A much more desirable approach is to represent curved boundaries with higher order polynomials compatible with the order of the data reconstruction.

We use quadratic or cubic polynomials to approximate boundaries in third and fourth-order SV schemes respectively, which employ quadratic and cubic polynomials (respectively) to represent the solution variables. Following the usual practice in high-order finite element method [78], isoparametric SVs can be used to map SVs with curved boundaries into the standard SV. The basic assumption is that a one-to-one transformation exists between a general SV in the physical space (x, y) and the standard triangle in the computational domain (ξ,η) as shown in Fig. 2.4 (taken from Wang et al [76]). In other words,

$$\xi=\xi(x,y), \tag{2.18}$$

$$\eta=\eta(x,y). \tag{2.19}$$

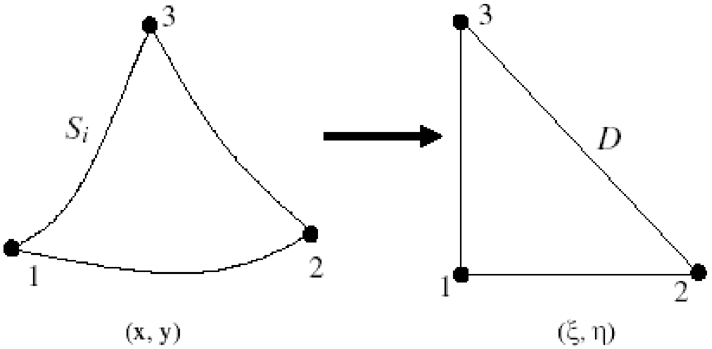


Figure 2.4 Transformation of a general SV in the physical domain to the standard triangle in the computational. Taken from Wang et al [76]

The partition is now performed in the standard triangle (D). An inverse transformation is used to obtain the corresponding partition in the physical domain as shown in Fig. 2.5 (taken from Wang et al [76]). In other words,

$$x=x(\xi,\eta), \tag{2.20}$$

$$y=y(\xi,\eta). \tag{2.21}$$

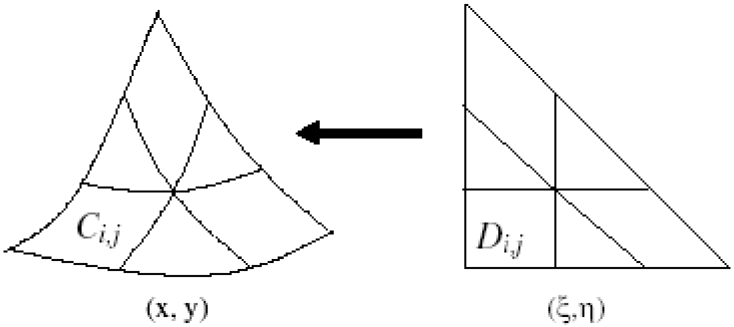


Figure 2.5 The partition in the standard triangle is transformed to the physical domain using an inverse transformation. Taken from Wang et al [76]

By definition, the DOFs are the CV averaged solution variables i.e.

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q(x,y)dV = \frac{1}{\int_{C_{i,j}} dxdy} \int_{C_{i,j}} Q(x,y)dxdy\,. \tag{2.22}$$

So if J is the Jacobian matrix of the transformation,

$$\overline{Q}_{i,j} = \frac{1}{\int_D |J|\, d\xi d\eta} \int_D Q(\xi,\eta)\,|J|\,d\xi d\eta\,. \tag{2.23}$$

The above can be rewritten as follows:

$$\overline{Q}_{i,j} \int_D |J|\,d\xi d\eta = \int_D Q(\xi,\eta)\,|J|\,d\xi d\eta\,. \tag{2.24}$$

Equation 2.24 is basically a reconstruction problem. It can be solved given the reconstruction coefficients. However, the reconstruction coefficients are strictly dependent on the geometry. Therefore it is necessary to physically store these coefficients for SVs with curved boundaries. The number of curved SVs is expected to be small compared with the total number of SVs in any simulation.

2.3.4 Simplified curved boundary formulation

The basics of the general curved boundary formulation were given in 2.3.3. The general formulation assumes all the faces (in the physical domain) to be curved faces. This scenario seldom occurs in practice. Most CFD simulations require only one boundary of the SV as a curved boundary as shown in Fig 2.6. As a result, simplified curved boundary formulations can be used.

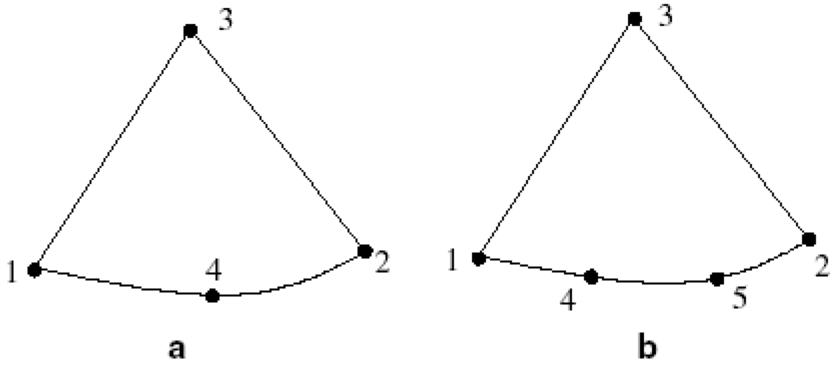


Figure 2.6 Simplified quadratic and cubic SVs with only one curved face. Taken from Wang et al [76]

For instance, the simplified quadratic SV has the following transformation

$$\vec{r} = \sum_{j=1}^{4} M_j \vec{r}_j \ , \tag{2.25}$$

where $\vec{r} = (\mathrm{x,y})$ and the above shape functions are

$$M_1(\xi,\eta) = 1 - 3\xi + 2\xi(\xi + \eta) - \eta \ ,$$

$$M_2(\xi,\eta) = -\xi + 2\xi(\xi + \eta) \ ,$$

$$M_3(\xi,\eta) = \eta \ ,$$

$$M_4(\xi,\eta) = 4\xi(1-\xi-\eta)\,. \tag{2.26}$$

CHAPTER 3. TIME INTEGRATION ALGORITHMS

3.1 Introduction

In general, the high-order spatial operators are much stiffer than lower-order ones. In addition, for time accurate problems, the maximum allowable CFL number decreases with increasing the order of accuracy for explicit schemes. This problem becomes even more severe for viscous flow problems with highly clustered meshes: explicit high-order methods are adversely limited by time step size. Hence the development of efficient time integration schemes for high-orders is essential. Some of the existing methods are reviewed in the next two paragraphs.

Explicit Runge Kutta schemes have been one of the most widely used methods of time integration. One particular class of the above is the strong-stability-preserving (SSP) Runge Kutta schemes. This class of time integration method was originally developed by Shu [64], and Shu and Osher [63] and named TVD Runge–Kutta schemes. It was also the topic of research for many other researchers e.g. [65,28]. This class of Runge Kutta schemes preserves the stability properties of forward Euler in any norm or semi-norm. They have been popular for high-order spatial operators because of its TVD or SSP property. The coefficients of these schemes are not unique. However, optimal versions with maximum CFL numbers have been derived in [64] for the second and third order schemes, and by Spiteri and Ruuth [65] for the fourth-order counterpart. More details can be found in a recent review article by Gottlieb [27].

The limitations of the explicit methods were mentioned in the first paragraph of this chapter. It is obvious that implicit algorithms are necessary to overcome the time step limit suffered by explicit algorithms especially for highly clustered viscous meshes. Many types of implicit algorithms have been successfully developed for unstructured grid-based solvers in the last two decades, e.g., the element Jacobi, Gauss–Seidel, precondition GMRES [8,60,75], matrix-free Krylov [55], lower–upper symmetry Gauss–Seidel [2,62], and line-implicit algorithms [45]. In addition, these implicit algorithms can serve as ‘‘smoothers’’ for geometric or p-multigrid approaches. Many of these algorithms have been successfully applied to high-order spatial discretizations (e.g. [43], [55] for DG). In almost all implicit approaches, the non-linear system of equations is linearized and then solved with an iterative solution algorithm. Even though these implicit methods offer extremely high speedup, they need huge memory to store the associated matrices. This is greatly felt for higher order methods. One intelligent way to mitigate the above problem is to use the traditional multi stage Runge-Kutta method as the higher level smoother and the implicit scheme for the lower levels [43].

The following unsteady equation is considered in the current thesis and the time integration schemes (either explicit or implicit schemes) can be used as a smoother.

$$\frac{\partial \tilde{Q}}{\partial t} = R_i^c(\tilde{Q}), \tag{3.1}$$

where $R_i^c(\tilde{Q})$ denotes the residual vector of the current cell and $\tilde{Q}$ is the solution vector.

The remainder of the chapter gives the formulations for the three stage SSP Runge kutta schemes and the implicit LU-SGS method.

3.2 The explicit method

We mainly concentrate on the three stage SSP SSP Runge kutta scheme. The three-stage explicit SSP Runge Kutta [64] can be written as follows:

$$\overline{u}_i^{(1)} = \overline{u}_i^{\,n} - \Delta t R_i(\overline{u}^{\,n});$$

$$\overline{u}_i^{(2)} = \frac{3}{4}\overline{u}_i^{\,n} + \frac{1}{4}[\overline{u}_i^{(1)} - \Delta t R_i(\overline{u}^{\,(1)})];$$

$$\overline{u}_i^{\,n+1} = \frac{1}{3}\overline{u}_i^{\,n} + \frac{2}{3}[\overline{u}_i^{(2)} - \Delta t R_i(\overline{u}^{\,(2)})] \tag{3.2}$$

The main advantage of the Runge Kutta scheme is that it requires little memory for storage. In addition, this method is inherently simple and so is easy to program. These are the main reasons for it being one of the most preferred methods of time integration.

The main bottleneck associated with the Runge Kutta scheme is the limit imposed on the time step. Euler (and Navier Stokes) equations for realistic geometries entail a rather strict limit on the time step. Even though the above can be circumvented by using a very higher order (several RK steps) scheme, it is seldom used as it required lots of storage and thus adversely affects its simplistic nature. This prompts us to switch over to implicit schemes.

3.3 The implicit method

It is well known that the explicit scheme is limited by the Courant-Friedrichs-Lewy (CFL) condition. When the polynomial order of the SD/SV method is high or the grid is stretched due to complex geometries, the convergence rate of the explicit scheme slows down rapidly.

The implicit methods are normally formulated by the linearization of a given set of equations.

At each current cell *c*, the backward Euler difference can be written as

$$\frac{\tilde{Q}_c^{n+1}-\tilde{Q}_c^{n}}{\Delta t}-\left[R_c(\tilde{Q}^{n+1})-R_c(\tilde{Q}^{n})\right]=R_c(\tilde{Q}^{n}), \tag{3.3}$$

where n refers to the current time level and n+1 the next time level

Let $\Delta\tilde{Q}_c=\tilde{Q}_c^{n+1}-\tilde{Q}_c^{n}$ and linearizing the residual, we obtain

$$R_c(\tilde{Q}^{n+1})-R_c(\tilde{Q}^{n})\approx\frac{\partial R_c}{\partial\tilde{Q}_c}\Delta\tilde{Q}_c+\sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb}, \tag{3.4}$$

where *nb* indicates all the neighboring cells contributing to the residual of cell *c*. Therefore, the fully linearized equations for (3.3) can be written as

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\tilde{Q}_c}\right)\Delta\tilde{Q}_c-\sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb}=R_c(\tilde{Q}^{n}). \tag{3.5}$$

However, it costs too much memory to store the entire LHS implicit Jacobian matrices. For instance, a 3rd order Euler system of equations in 2D would result in a [96x96] matrix per cell. In addition, the cost of performing a Gaussian Elimination of that matrix is O(96^3) i.e. O(900000) units. Therefore, we employ a LU-SGS scheme to solve (3.5), i.e., we use the most recent solution for the neighboring cells,

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\tilde{Q}_c}\right)\Delta\tilde{Q}_c^{(k+1)}=R_c(\tilde{Q}^{n})+\sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb}^{*}. \tag{3.6}$$

The matrix

$$D=\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\tilde{Q}_c}\right), \tag{3.7}$$

is the element (or cell) matrix, which is not quite small for 2nd to 3rd order SD schemes. For instance, D is [24x24] for the 3rd order SD Euler system of equations. The cost of performing a Gaussian Elimination for a matrix of this size is O(24^3) i.e. O(14000) units. These matrices also consume much lesser storage space.

Since we do not want to store the matrices $\frac{\partial R_c}{\partial Q_{nb}}$, (3.6) is further manipulated as follows

$$\begin{aligned} R_c(\tilde{Q}^n) + \sum_{nb\neq c} \frac{\partial R_c}{\partial \tilde{Q}_{nb}} \Delta \tilde{Q}^*_{nb} &= R_c(\tilde{Q}^n_c, \{\tilde{Q}^n_{nb}\}) + \sum_{nb\neq c} \frac{\partial R_c}{\partial \tilde{Q}_{nb}} \Delta \tilde{Q}^*_{nb} \\ \approx R_c(\tilde{Q}^n_c, \{\tilde{Q}^*_{nb}\}) &\approx R_c(\tilde{Q}^*_c, \{\tilde{Q}^*_{nb}\}) - \frac{\partial R_c}{\partial \tilde{Q}_c} \Delta Q^*_c = R_c(\tilde{Q}^*) - \frac{\partial R_c}{\partial \tilde{Q}_c} \Delta Q^*_c. \end{aligned} \tag{3.8}$$

Let $\Delta^2 Q_c^{(k+1)} = \Delta\, Q_c^{(k+1)} - \Delta\, Q_c^*$. We can combine (3.6) and (3.8) together to obtain

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c} \right) \Delta^2 \tilde{Q}_c^{(k+1)} = R_c(\tilde{Q}^*) - \frac{\Delta Q_c^*}{\Delta t}. \tag{3.9}$$

Equation (3.9) is then solved with a direct LU decomposition solver and sweeping symmetrically forward and backward through the computational grid. Note that once (3.9) is solved to machine zero, the unsteady residual is zero at each time step. For steady flow problem, we found that the term $\frac{\Delta Q_c^*}{\Delta t}$ in the right-hand-side of (3.9) can be neglected and leading to quicker convergence. Note that ΔQ_c^* is the difference between the current solution Q_c^* and the solution at the previous time level Q_c^n. In reality, the entire system is swept several times in order to proceed to the next time level. As a result, ΔQ_c^* is influenced by the solution occurred several sweeps ago. This introduces an under-relaxation effect. Hence neglecting the $\frac{\Delta Q_c^*}{\Delta t}$ term accelerates the convergence. We define the solver obtained using (3.9) as implicit normal approach. If $\frac{\Delta Q_c^*}{\Delta t}$ term is dropped, the iterative solver is defined as implicit simplified approach.

CHAPTER 4. THE P-MULTIGRID METHOD

4.1 Introduction

Multigrid methods were originally developed to efficiently solve linear systems that arise from the diffusion operators. These methods rely on the fact that there exist iterative methods, called smoothers, which rapidly damp out highly oscillatory components of the solution error.

The commonly used multigrid is the geometric multigrid (also referred to as the h-multigrid or sometimes as multigrid). The standard multigrid algorithm (geometric multigrid, or h-multigrid) has been used very effectively in CFD to accelerate the rate of convergence to steady state. Jameson popularized multigrid in CFD with his original combination of Runge Kutta time stepping and multigrid approaches [34]. This triggered numerous in the field, e.g., [43,45,35,3,4,32,46,50]. The standard multigrid techniques consist of projecting the solution error into a coarser grid and solve the coarser grid version of the original problem to obtain an error correction term. This constitutes a 2-grid method. By applying the 2-grid method recursively on a sequence of grids, one obtains a general multigrid method. In general, a coarser mesh is obtained by doubling the mesh size of the finer mesh in all directions. The effectiveness of a multigrid approach is determined by several factors viz

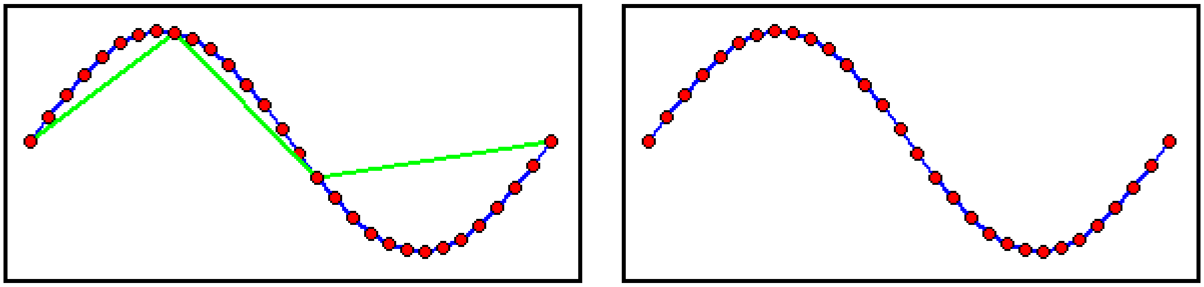
1. All smoothers are more efficient in damping high frequency errors than low frequency ones. However the errors on a fine mesh are represented on a coarser mesh at higher frequency, which can be damped more effectively.

2. Larger time steps can be used on the coarser mesh (in the case of explicit time stepping schemes) and errors can be driven out of the computational domain at a faster rate.

Novel work was also done to reduce the stiffness associated with the grid anisotropy and due to the low flow speed: by line implicit solvers [45] and semi-coarsening [50] for grid coarsening and including low speed preconditioning [69,17,73] for low speed flows. It is worth to mention the results of Jameson and Caughley [35] who demonstrated convergence to truncation error level using just a few multigrid cycles with a preconditioned non-linear LU-SGS smoother.

4.1.1 An example of geometric multigrid

Consider a Laplace equation in 1D. The solution in the interior is driven to steady state solely due to the values imposed on the boundaries (Dirichlet boundary conditions). Let us consider the case wherein the values at the two boundaries are set to zero. Let the initial guess be in the form of a sinusoidal wave which satisfies the boundary conditions. Fine and coarse grids are considered as shown in figure 4.1.



Low frequency errors (fine grid)

High frequency errors (coarse grid)

Figure 4.1: Error distribution on the fine and coarse grids for the 1D Laplace problem.

The imposed errors behave as low frequency errors in the fine grid and as high frequency errors in the coarse grid. As expected the correction equation in the coarse grid can be solved rapidly as the errors "appear to be of high frequency" in that grid. A speedup of 3 was attained for this simulation.

4.2 The p-multigrid method

Geometric multigrid is simple, easy to implement and can coded in a recursive fashion. It has been implemented in many 2nd order commercial packages. Nevertheless, there exist some practical issues with h-multigrid. These include

A. Geometric multigrids in an unstructured setting is a nightmare. Grid coarsening and grid agglomerations can be a nightmare due to the bookkeeping involved with the various data structures.

B. Attaining multiple levels of coarsening is even tougher for unstructured grids.

C. Implementing multigrid for viscous flows is always problematic. Preferential coarsening is needed near boundary layers. This becomes more problematic if the grid is unstructured.

These issues can be overcome by shifting to the p-multigrid method.

4.2.1 P-Multigrid for higher order methods

The p-multigrid method is a smoothing operator which hastens the convergence by using the coarse levels constructed by reducing the level of the interpolation polynomial p. This method was initially proposed by Ronquist and Patera [58] and extended by Maday and Munoz [44]. The acceleration of the convergence by the p-multigrid algorithm on Euler equations was demonstrated by Bassi and Rebay [12], Nastase and Mavriplis [51] and Hong et al. [43]. Helenbrook [31] examined the performance of the Laplace equation and the convection equations in two dimensions. Fidkowski et al. [26] developed a p-multigrid approach using an effective line-implicit smoother for their high-order DG Navier–Stokes solver. Luo et al [43] demonstrated a low storage p-multigrid approach for a 3D DG Euler solver, in which an explicit Runge–Kutta scheme is used for the highest order discretization, while implicit SGS smoothers are employed for the lower-order operators. Recently, Nastase and Mavriplis [51] developed an hp-multigrid approach for their high-order inviscid DG solver, and demonstrated h-independent and p-independent convergence rates.

The Gauss-Seidel or Jacobi iterations produce smooth errors when applied on the above mentioned nonlinear equations. The error vector has its high frequencies nearly removed in a few iterations using a higher order polynomial; but low frequencies are removed very slowly. The key idea of the p-Multigrid method is to solve the nonlinear equations using a lower order polynomial such that “smooth becomes rough” and low frequencies act like high frequencies. Such a p-Multigrid method has been used for high-order discontinuous Galerkin method; see [51, 31, 26]. The p-Multigrid method operates on a sequence of solution approximations of different polynomial orders. Therefore it offers the flexibility of switching between higher and lower polynomial levels without changing the actual geometrical nodal grid points.

Furthermore, p-multigrid is a natural fit to the high-order SV/SD unstructured grid framework. Unlike h-multigrid, spatially coarser meshes are not required. Thus, no element agglomeration or re-meshing procedures are necessary. Only prolongation and restriction between orders are required. These are calculated during the pre-processor stage using mathematical kits like Matlab/Mathematica or Maple and are hardcoded into the system.

To accomplish the communication between different levels, restriction (I_p^{p-1}, I_{p-1}^{p-2}) and prolongation ($I_{p-1}^{p}, I_{p-2}^{p-1}$) operators are required in addition to the aforementioned relaxation scheme as a smoother. Restriction consists of moving solutions and their residuals at the unknown points from a space of higher polynomial order to a lower polynomial order. Prolongation refers to a reverse procedure in which lower order polynomial solution correction is redistributed as corrections to the solutions of the unknown points at a higher polynomial order.

I_p^{p-1} is an operator which restrict from a degree p polynomial space to a degree p-1 polynomial space. This operator is essentially a matrix. For instance, I_2^1 is a [3x6] matrix and I_1^0 is a [1x3] matrix in 2D. Thus if [u_p] is a vector in degree p polynomial space, its restricted component in the p-1 degree polynomial space is given by

$$[u_{p-1}] = I_p^{p-1} \,.\, [u_p]. \tag{4.1}$$

Conversely, I_{p-1}^{p} is an operator which is used to prolongate from a degree p-1 polynomial space to a degree p polynomial space. As expected, I_1^2 is a [6x3] matrix and I_0^1 is a [3x1] matrix in 2D. Thus if [u_{p-1}] is a vector in degree p-1 polynomial space, its prolongated component in the p degree polynomial space is given by

$$[u_p] = I_{p-1}^{p} \,.\, [u_{p-1}]. \tag{4.2}$$

For instance, I_2^1 for a standard SV partition is given by:

2.75e-01	-1.18e-01	-1.18e-01	4.71e-01	1.92e-02	4.71e-01
-1.18e-01	2.75e-01	-1.18e-01	4.71e-01	4.71e-01	1.92e-02
-1.18e-01	-1.18e-01	2.75e-01	1.92e-02	4.71e-01	4.71e-01

and I_1^2 for a standard SV partition is given by:

1.46e+00	-2.33e-01	-2.33e-01
-2.33e-01	1.46e+00	-2.33e-01
1.46e+00	-2.33e-01	-2.33e-01
5.77e-01	5.77e-01	-1.55e-01
5.77e-01	5.77e-01	-1.55e-01
5.77e-01	-1.55e-01	5.77e-01

4.2.2 Implementation of the p-multigrid algorithm

Classical multigrid method begins with a two-level process. First, iterative relaxation is applied using the higher order polynomial, thus basically reducing high-frequency errors. Then, a “coarse-grid” correction is applied, in which the smooth error is determined at the lower polynomial level. This error is interpolated to the higher polynomial level and used to correct the existing higher order solutions. Applying this method recursively to solve the lower polynomial level problems leads to multigrid.

The p-Multigrid procedure is summarized below:

Defining three polynomial levels from the highest to the lowest as *p*, *p-1* and *p-2*, we want to solve:

$$R_p(Q_p) = r_p \,. \tag{4.3}$$

The rhs r_p is zero for the highest level polynomial.

$$R_{p-1}(Q_{p-1}) = r_{p-1} \,, \tag{4.4}$$

$$R_{p-2}(Q_{p-2}) = r_{p-2} \,. \tag{4.5}$$

We employ the implicit LU-SGS schemes as the smoothers for all three levels. The following steps are used to update the solutions on the highest p level in one big V cycle.

- Improve Q_p by application of a few times the smoother similar as equation (3.9)

$$\left[\frac{I}{\Delta t} - \frac{\partial R_c^p}{\partial Q_c^p}\right]^n \Delta^2 Q_c^p = R_p(Q_p^*) - r_p \,. \tag{4.6}$$

- Restrict the latest solution Q_p to the coarser level for an approximate solution Q_{p-1}

$$Q_{p-1}^0 = I_p^{p-1}(Q_p). \tag{4.7}$$

- Compute the defect on the finest level

$$d_p = r_p - R_p(Q_p) = -R_p(Q_p). \tag{4.8}$$

- Compute the right hand side of equation (4.4) as

$$r_{p-1} = R_{p-1}(Q_{p-1}^0) + I_p^{p-1} d_p \,. \tag{4.9}$$

- Improve Q_{p-1} by application of a few times the smoother

$$\left[\frac{I}{\Delta t}-\frac{\partial R_c^{p-2}}{\partial Q_c^{p-2}}\right]^n \Delta^2 Q_c^{p-2} = R_{p-2}(Q_{p-2}^*)-r_{p-2}. \tag{4.10}$$

- Restrict the latest solution Q_{p-1} obtained by $Q_{p-1}^{k+1}=Q_{p-1}^{k}+\Delta^2 Q_c^{p-1}$ to the coarser level for an approximate solution Q_{p-2}^s

$$Q_{p-2}^s = I_{p-1}^{p-2}\left(Q_{p-1}\right). \tag{4.11}$$

- Define the defect on the intermediate level as

$$d_{p-1}=r_{p-1}-R_{p-1}\left(Q_{p-1}\right). \tag{4.12}$$

- Compute the right hand side of equation (4.3) as

$$r_{p-2}=R_{p-2}\left(Q_{p-2}^s\right)+I_{p-1}^{p-2}d_{p-1}. \tag{4.13}$$

- Improve Q_{p-2} by application of a few times the smoother

$$\left[\frac{I}{\Delta t}-\frac{\partial R_c^{p-2}}{\partial Q_c^{p-2}}\right]^n \Delta^2 Q_c^{p-2} = R_{p-2}(Q_{p-2}^*)-r_{p-2}. \tag{4.14}$$

- Correct the current solution on the intermediate level by

$$\tilde{Q}_{p-1}=Q_{p-1}+I_{p-2}^{p-1}(\tilde{Q}_{p-2}-Q_{p-2}^s). \tag{4.15}$$

- Improve $\tilde{Q}_{p-1}$ by application of a few times the smoother (the same as equation 4.10) to get $\overline{Q}_{p-1}$

- Correct the current solution on the finest level by

$$\tilde{Q}_p = Q_p + I_{p-1}^{p}(\overline{Q}_{p-1}-Q_{p-1}^0) \tag{4.16}$$

- Improve $\tilde{Q}_p$ by application of a few times the iterative smoother (the same as equation 4.6) to get $\overline{Q}_p$

Note that only the implicit LU-SGS smoother is described in the above procedure for simplicity. In practice, we can replace any of the implicit smoothers (4.6, 4.10 or 4.14) with a three-stage or five-stage TVD Runge-Kutta scheme. If extension to 3D solver is considered, we intend to use the p2 level with an explicit scheme since its storage is small and implicit LU-SGS for the p1 and p0 levels. We define this as a mixed-smoother 3-level scheme. In this paper, we also consider using explicit schemes as smoothers for all three levels. It is defined as an R-K smoother 3-level scheme.

If we iterate the steps marked with hollow circle bullets (○) for one more time, a big W cycle is formed. If these steps are removed, it will be a classical two-level scheme as aforementioned.

Note that to numerically compute the D matrix at different p levels, we adopted a rediscretization approach based on the distribution of unknown and flux points at different levels as shown in figure 2.1. The procedure produces very stable results. However, Helenbrook and Atkins [30] used an algebraic approach to get a subset matrix from a higher order polynomial level, i.e., $A_{p-1} = I_{p-1}^{p} A_p I_p^{p-1}$ for the DG method. They found that it works better than the rediscretization approach for the Poisson equation. In the SD method, we need to take into account the effect of neighboring cells (at local polynomial level) during the calculations of fluxes. The rediscretization approach utilizes the most recently reconstructed solutions at the flux points. In contrast, the algebraic approach is very local and only takes information from the current cell. This phenomenon also occurs in SV. Therefore, the more accurate rediscretization approach is used for all the calculations in this thesis unless otherwise stated.

4.2.3 V cycles and the full multigrid(FMG)

To make multigrid more practical, the basic two level correction scheme is extended to a Vcycle and also to full multigrid (FMG). In a V-cycle, a sequence of one or more coarse levels is used to correct the solution on the fine level. Proceeding from the finest level to the coarsest, a certain number of pre-smoothing steps is performed on each level before the problem is restricted to the next coarser level. The problem is solved using the conventional methods in the coarsest level. Now while proceeding the other way around (coarsest to finest), post smoothing steps are performed on each level after prolongation. Each such V-cycle constitutes a multigrid iteration.

We need to start at the finest level to start using the V-cycle idea. The finest level contains the largest number of DOF. So smoothing operations in this level are most expensive. In addition, the system is also at a very highly ill conditioned state as the initial guess is way different from the actual solution. For instance, the free stream conditions serve as the initial guess for the solution in Euler/ Navier Stokes simulations. Clearly, this results in a severely ill conditioned system for the cells near the wall. The situation is even worse at high orders: resulting in a severe limitation for the time steps/CFL number.

An alternative is to first obtain an approximation to the solution using the coarser levels before smoothing on the finest level. This is the premise behind FMG in which V-cycles on successively finer levels are used to approximate the solution on the finest level. By the time the solution is prolongated to the finest level, it is usually a close approximation to the final solution with the exception of certain high frequency errors that can be smoothed efficiently on that level. This way, the system becomes relatively well conditioned and the user can start to use relatively high time steps. In practice, FMG should require only a few V-cycles on each level before prolongating to the next finer level.

CHAPTER 5 VISCOUS FLUX FORMULATIONS FOR THE SV METHOD

5.1 Introduction

The ultimate goal of this research study is to extend the SV method to the Navier-Stokes equations to perform DNS and large eddy simulations of turbulence flow for problems with complex geometries. Finding a way to properly discretize the second-order viscous terms can be regarded as the first step toward achieving the above goal. In the classical second-order finite volume method, the solution gradients at an interface are computed by averaging the gradients of the neighboring cells sharing the face in consideration. However, for higher order elements, special care has to be taken in computing the solution gradients

In the late 1970s and early 1980s, Arnold [6] and Wheeler [84] introduced the discontinuous finite element methods known as penalty methods for elliptic operators. More recently, many researchers [13,11,9,20,21,15] have applied DG methods to diffusive operators. One procedure was the famous local discontinuous Galerkin method, developed by Cockburn and Shu [20,21]. This method dealt with rewriting a second-order equation as a first-order system and then discretize the first-order system using the DG formulation. Their simplicity and effectiveness have made them the main choice for discretizing the viscous fluxes. Baumann and Oden [13] came up with different DG methods for the second order viscous terms. Recently, Zhang and Shu [86] conducted some Fourier analysis for the different formulations.

In this thesis, we extend three of the above formulations to the SV method setting. The first is the LDG method, the second is the penalty method and the third is the 2nd method of Bassi and Rebay(also referred to as BR2). Fourier analysis were performed on the above methods and yielded some interesting new results.

5.2 The three viscous flux formulations for the SV method

Equations (2.17) and (2.18) can be solved if the numerical fluxes $\underline{q}$ and $\underline{u}$ (discussed in section 2.3.2) can be determined. These three formulations are some of the ways to obtain the above mentioned numerical fluxes.

5.2.1 The LDG formulation

This is the easiest formulation. In this approach, the numerical fluxes are defined by alternating the direction in the following manner [68]

$$\underline{u} = u_L, \tag{5.1}$$

$$\underline{\vec{q}} = \vec{q}_R. \tag{5.2}$$

where u_L and u_R are the left and right state solutions of the CV face in consideration and $\vec{q}_L$ and $\vec{q}_R$ are the left and right state solutions gradients of the face (of the CV) in consideration. Thus if the CV face lies on the SV boundary, $u_L \neq u_R$ and $\vec{q}_L \neq \vec{q}_R$.

It must be noted that the above style of discretization is valid only if the face in consideration is a non-boundary face. Using the above style of discretization for a boundary face can result in an inconsistent system of equations. This can be explained using a numerical example. Consider a 2nd order simulation in 1D. The entire computational domain consists of only one SV i.e. 2 CVs. A schematic of this can be seen in figure 5.1.

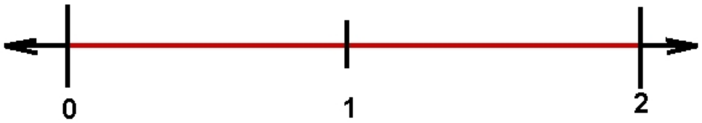


Figure 5.1 1D SV with 2 CVs

The CVs in the SV have a length (i.e. volume) of unity. The normals on the domain boundaries point outward (as shown in the figure). Now consider an analytical function,

$$u=Ax+B. \tag{5.3}$$

The above function is used to set values on the boundaries i.e.

$$u_{R\_leftboundary} = A\,, \tag{5.4}$$

$$u_{R\_rightboundary} = 2A+B\,. \tag{5.5}$$

Let $\overline{u}_1$ and $\overline{u}_2$ be the CV averages of the solution for the 1st and 2nd CVs respectively. After some algebra, we arrive at the following time evolution differential equations for $\overline{u}_1$ and $\overline{u}_2$:

$$\frac{d\overline{u}_1}{dt} = 2A+2B-\overline{u}_1-\overline{u}_2 \tag{5.6}$$

$$\frac{d\overline{u}_2}{dt} = 2A+2B-\overline{u}_1-\overline{u}_2 \tag{5.7}$$

Equations (5.6) and (5.7) have a non-unique set of solutions. This means that the 2nd order SV is incapable of capturing a linear function with the LDG style of discretization. So some special treatment is needed at the boundaries to circumvent this. One way is to use a different formulation like the penalty formulation for the boundary faces.

5.2.2 The penalty approach

It can be seen (from equations 5.1 and 5.2) that LDG is inherently unsymmetric. A symmetric approach was given by Bassi and Rebay [9], in which the numerical fluxes are defined by

$$\underline{u} = 0.5 * (u_R + u_L), \tag{5.8}$$

$$\underline{\vec{q}} = 0.5 * (\vec{q}_R + \vec{q}_L). \tag{5.9}$$

Analysis by Brezzi et al [15] showed that the approach might be unstable in some situations. In this paper, we suggest the following the penalty approach to obtain the numerical fluxes:

$$\underline{u} = 0.5 * (u_R + u_L), \tag{5.10}$$

$$\underline{\vec{q}} = 0.5 * (\vec{q}_R + \vec{q}_L) + (u_R - u_L)\vec{n}\frac{A_r}{V_{ij}}, \tag{5.11}$$

where $\vec{q}_L$ and $\vec{q}_R$ are the left and right state solutions gradients of the face (of the CV) in consideration, A_r is the area of the face (of the CV) in consideration, V_{ij} is the CV volume and $\vec{n}$ is the unit normal of the face.

One can see a similarity between equation 5.11 and an approximate Riemann (like Roe, Rusanov or AUSM) flux. The approximate Riemann flux is obtained by averaging the left and right state fluxes and then adding a dissipation term. This dissipation term is

1. Proportional to the jump in the solution between the right and left states.

2. Proportional to the Jacobian term/matrix or its eigen values (The Jacobian term in 1D is $\frac{\partial \tilde{f}}{\partial \tilde{Q}}$). For instance, in Rusanov flux, it is the maximum eigen value of the Jacobian matrix.

Equation 5.11 is obtained by averaging the left and right states and then penalizing it with the penalty term. This is similar to the structure of the approximate Riemann flux. The Jacobian term in this case has a dimension of 1/length. So we picked $\frac{A_r}{V_{ij}}$ as an approximation to the eigen value. The penalty term has a sign, which is opposite to the dissipation term. This is because the dissipation terms come on the RHS.

It can be seen that the current formulation is still non compact. By definition, a non compact system is one wherein the residual of a cell is dependent on the solution of cells which are neither the current cell nor its adjacent neighbors. Compactness is generally a desired attribute as it has serious desirable properties:

1. Very desirable when the cells are clustered. This is because the reconstructions are done from information obtained from a restricted region and hence is more accurate.
2. Efficient for parallel applications: lesser information is needed w.r.t storage and message transfers

Figures 5.2 and 5.3 show compact and non compact stencils in 1D. The red and blue colors indicate the domain of dependence.

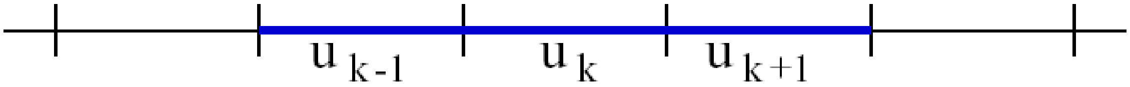


Figure 5.2 Compact stencil in 1D

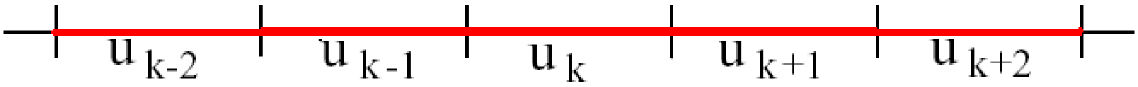


Figure 5.3 Non Compact stencil in 1D

5.2.3 The second approach of Bassi and Rebay (BR2)

The penalty approach is definitely more symmetrical than LDG. However, the implementation results in a non-compact stencil. We now give the formulation of BR2 [10], which is compact. The BR2 formulation is slightly different from the above two methods. Two different gradients are evaluated for residual computation:

I. The first gradient is referred to as the self gradient. It is computed using the solution from the cell in consideration. In other words, there is no contribution from the neighboring cells. This gradient is utilized for computing the viscous fluxes through the CV faces, which lie on the SV boundary.

II. The second gradient is referred to as the averaged gradient. It is computed by using the averaged solution obtained from the left and the right sides of the CV face. This gradient is used for computing the viscous fluxes through the CV faces, which do not lie on the SV boundary.

BR2 uses the concept of lifting functions. These lifting functions, $\vec{r}^{+}$ and $\vec{r}^{-}$ (which are actually polynomials) can be viewed as corrections to the self gradient. Every SV face has the above lifting functions, $\vec{r}^{+}$ and $\vec{r}^{-}$ associated with it. CV averaged lifting functions are first computed for the SV face in consideration. The actual procedure can be explained using an example. Consider the two 2nd order SVs, shown in figure 5.4. The SV boundary face in consideration is f and is using in red. This comprises of two CV faces, f1 and f2. The CVs of the SV in consideration are also marked in the figure. The CV averaged lifting functions $\overline{\vec{r}_1^{\,+}}$ and $\overline{\vec{r}_1^{\,-}}$ for CV1 can be obtained using equations 5.12 and 5.13.

$$V^+\overline{\vec{r}_1^{\,+}} = \int_{f1} \frac{(u^- - u^+)}{2}\vec{n}^+ dA, \tag{5.12}$$

$$V^-\overline{\vec{r}_1^{\,-}} = \int_{f1} \frac{(u^+ - u^-)}{2}\vec{n}^- dA. \tag{5.13}$$

where + and – refer to the current CV(in this case CV1) and the neighboring CV respectively, V^+ and $V^-$are the volumes of the current and neighboring CVs respectively, $\vec{n}^+$ and $\vec{n}^-$are the outward normal w.r.t SVs (V^+ and V^-respectively) for the face in consideration. Note: $\vec{n}^+ = -\vec{n}^-$.

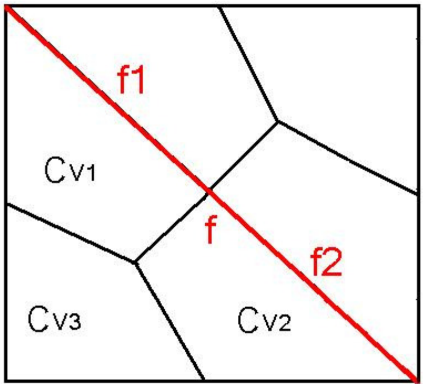


Figure 5.4 Two SVs having a common SV face f.

Similarly, one could obtain the CV averaged lifting functions $\overline{\vec{r}_2^{\,+}}$ and $\overline{\vec{r}_2^{\,-}}$ for CV2. The CV averaged lifting functions for the CV3 is zero. Given the CV averaged lifting functions, $\overline{\vec{r}_1^{\,+}}$, $\overline{\vec{r}_2^{\,+}}$ and $\overline{\vec{r}_3^{\,+}}$ (which is zero!), a mth degree polynomials can be constructed such that they are (m+1)th order approximation to the above described CV averaged lifting functions. This is the actual lifting functions $\vec{r}^{\,+}$ corresponding to the faces f1 and f2.

The above procedure is repeated for the other SV faces so as to obtain the lifting functions corresponding the CV faces located in the other two SV faces. The other lifting function $\vec{r}^{\,-}$ can be obtained in a similar way. The gradients used for the CV faces located on the SV boundary are given by

$$\underline{\vec{q}} = \frac{1}{2}(\vec{q}_R + \vec{r}^{\,+} + \vec{q}_L + \vec{r}^{\,-})\,. \tag{5.14}$$

It must be noted that BR2 requires nearly twice the number of CV averaged gradient computations and their reconstructions. In addition, the computations involved in evaluating the lifting function have more Floating Point Operations (FLOPs) than the penalty or the LDG scheme. As expected, there exist additional storage requirements for the BR2 scheme (additional set of gradients and lifting functions)

5.3 Analysis for uniform grids

5.3.1 One dimensional analysis

In this analysis, we follow a technique described by Zhang and Shu [86] and focus on linear, quadratic and cubic reconstructions. The SV is partitioned into two equal CVs for the second order, three CVs for the third order and four CVs for the fourth order. For the sake of simplicity, let us first consider a linear partition (shown in figure 5.5). In this case, the formulations can be cast in the following form:

$$\frac{d}{dt}\begin{bmatrix}\bar{u}_{j,1}\\ \bar{u}_{j,2}\end{bmatrix} = \mathrm{A}\begin{bmatrix}\bar{u}_{j-2,1}\\ \bar{u}_{j-2,2}\end{bmatrix} + \mathrm{B}\begin{bmatrix}\bar{u}_{j-1,1}\\ \bar{u}_{j-1,2}\end{bmatrix} + \mathrm{C}\begin{bmatrix}\bar{u}_{j,1}\\ \bar{u}_{j,2}\end{bmatrix} + \mathrm{D}\begin{bmatrix}\bar{u}_{j+1,1}\\ \bar{u}_{j+1,2}\end{bmatrix} + \mathrm{E}\begin{bmatrix}\bar{u}_{j+2,1}\\ \bar{u}_{j+2,2}\end{bmatrix}, \tag{5.15}$$

where A,B,C,D and E are constant matrices. We now seek general solution of the following form

$$u(x,t) = \hat{u}_k(t)e^{ikx}, \tag{5.16}$$

where k is the index of modes (k=1,2..) representing the wave number. Obviously, the analytical solution for equation 2.12 is $u(x,t) = e^{ikx-k^2t}$. The solution we are looking for can be expressed as

$$\begin{bmatrix} \bar{u}_{j,1} \\ \bar{u}_{j,2} \end{bmatrix} = \begin{bmatrix} \hat{u}_{k,1} \\ \hat{u}_{k,2} \end{bmatrix} e^{ikx_{j,3/2}}. \tag{5.17}$$

Substituting 5.17 into 5.15, we obtain the advancement equation:

$$\begin{bmatrix} \hat{u}'_{k,1} \\ \hat{u}'_{k,2} \end{bmatrix} = \mathrm{G(k,h)} \begin{bmatrix} \hat{u}_{k,1} \\ \hat{u}_{k,2} \end{bmatrix}, \tag{5.18}$$

where the amplification matrix is given by

$$\mathrm{G} = e^{-2ikh}\mathrm{A} + e^{-ikh}\mathrm{B} + \mathrm{C} + e^{ikh}\mathrm{D} + e^{2ikh}\mathrm{E}. \tag{5.19}$$

The above method can be easily extended to 3rd and 4th orders. In general, all but one of the eigen values of G is made up of spurious modes and is damped rapidly. The error associated with the scheme, convergence properties can be determined by analyzing the non spurious mode

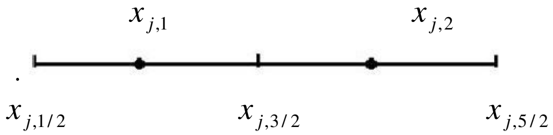


Figure 5.5 Linear spectral volume in 1D

5.3.1.1 Spatial analysis

Figure 5.6 shows the variation of the principal eigen value with respect to the non dimensional frequency χ (=kh) for the 2nd order case. This is also referred to as the Fourier footprint. Figure 5.7 shows the error at lower wave numbers. It is clear that the errors associated with the penalty formulation are lower than that of LDG. Moreover, the error associated with the penalty scheme is lower than the LDG error. This means that the system is damped faster in the penalty case and hence converges faster. The situation is similar for the 3rd order as is seen from figures 5.8 and 5.9. Thus in general, we expect the penalty scheme to converge faster than the LDG for the 2nd and 3rd orders.

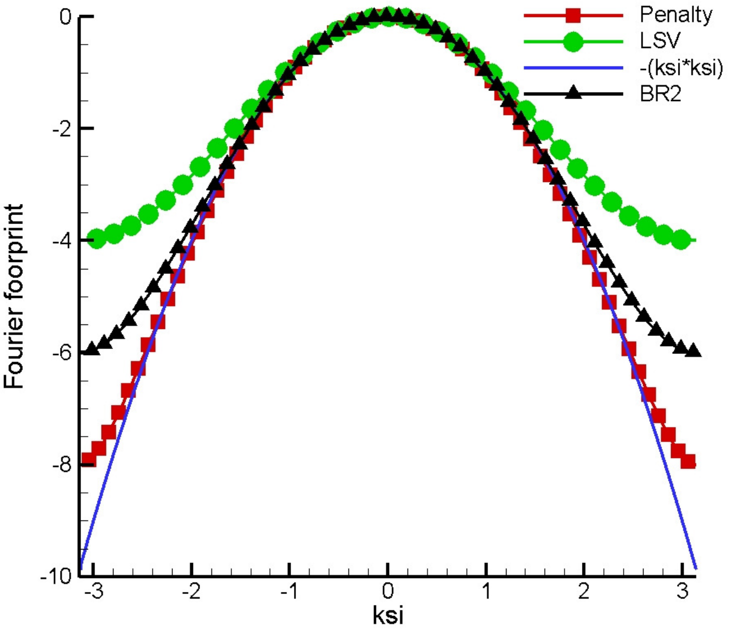


Figure 5.6 Fourier footprint for 2nd order scheme

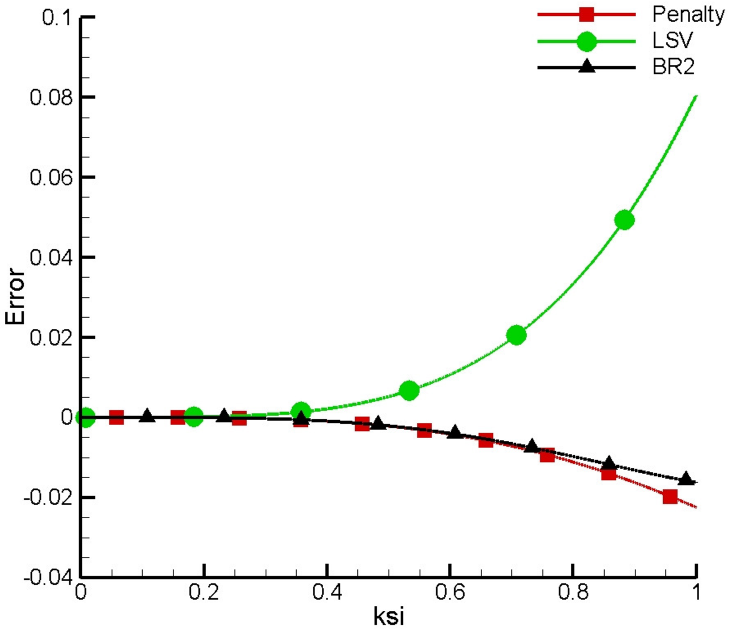


Figure 5.7 Error associated at low wavenumbers for the 2nd order scheme

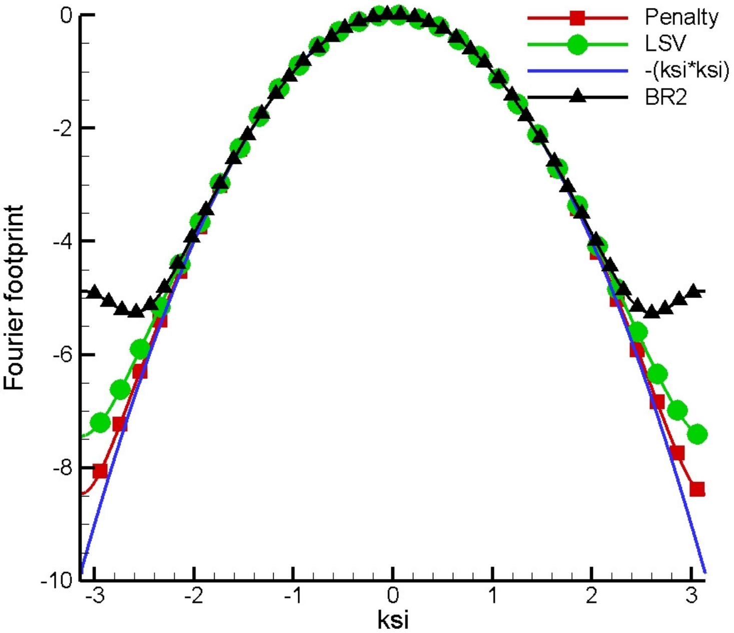


Figure 5.8 Fourier footprint for 3rd order scheme

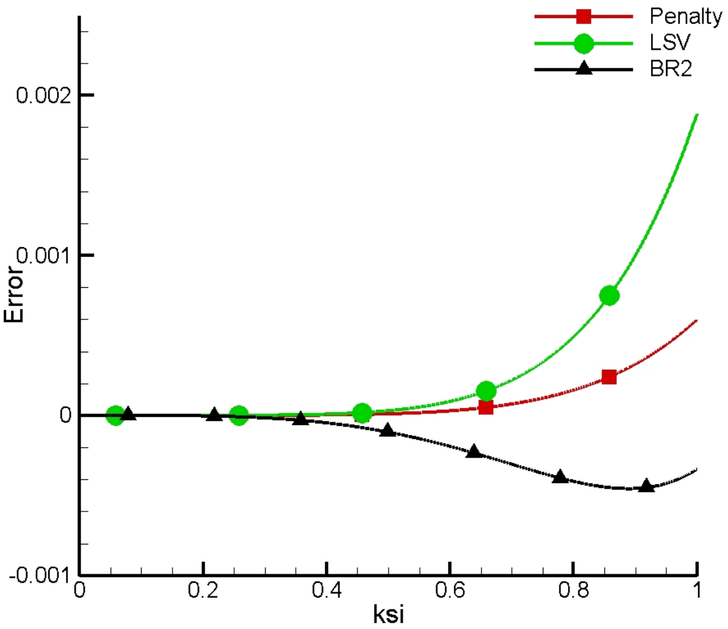


Figure 5.9 Error associated at low wavenumbers for the 3rd order scheme

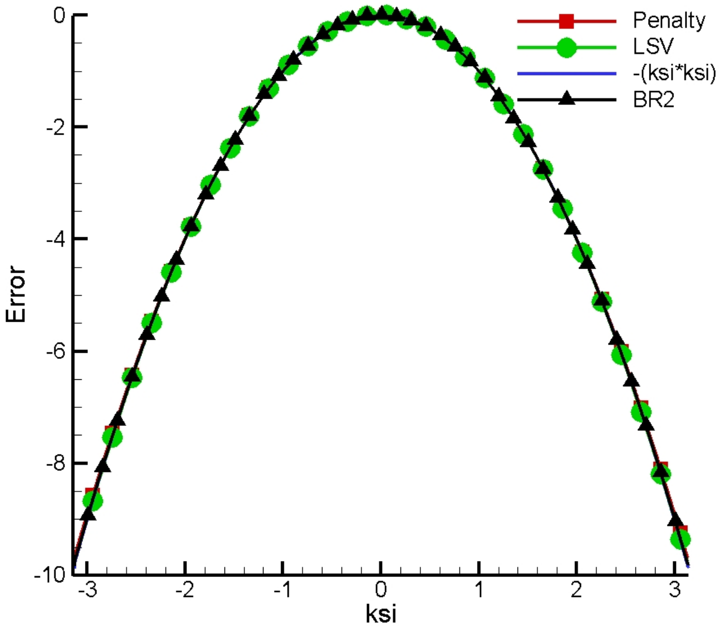


Figure 5.10 Fourier footprint for 4th order scheme

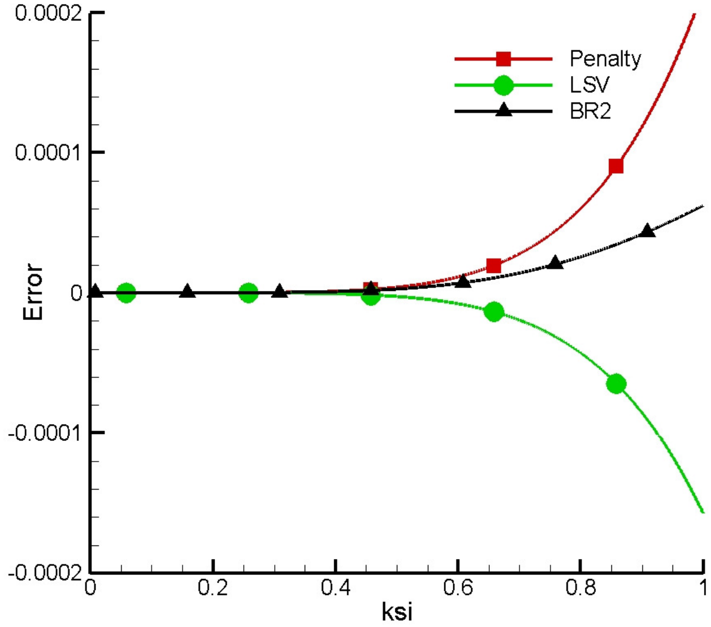


Figure 5.11 Error associated at low wavenumbers for the 4th order scheme

The principal eigen value lies on top of the y=--x^2 parabola for the fourth order (figure 5.10). The fourth order errors are too negligible (figure 5.11) to comment on convergence phenomena. It must be noted that the matrices A and E are zero matrices for the LDG and BR2 formulations.

5.3.1.2 Temporal analysis

In this section, we compute the time step requirements of the 3-stage Runge Kutta scheme. The 3-stage Runge Kutta can be simplified and can be written (for a 2nd order SV) as follows:

$$\begin{bmatrix} \hat{u}'_{k,1} \\ \hat{u}'_{k,2} \end{bmatrix} = \begin{bmatrix} \hat{u}_{k,1} \\ \hat{u}_{k,2} \end{bmatrix} . (\mathrm{I} + \mathrm{G}\,dt \ + \mathrm{G}^2 \frac{dt^2}{2} \ + \mathrm{G}^3 \frac{dt^3}{6} \ ). \qquad (5.20)$$

The above method can be easily extended to 3rd and 4th orders.

Table I lists the maximum non-dimensional time step ($\tau = \mu \frac{dt}{dx^2}$) required for obtaining a stable solution. It can be seen that the penalty and BR2 methods permit a higher time-step limit than the LDG method. In fact, the time step permitted by the BR2 method is more than double of that of the penalty method. Numerical experiments verified the above conclusion.

Table I. Maximum non-dimensional time step for obtaining stable solutions for the LDG, penalty and BR2 methods.

Case	2nd order (stability)	3rd order (stability)	4th order (stability)
LDG	0.157	0.0266	0.0109
Penalty	0.182	0.0322	0.0133
BR2	0.314	0.0784	0.0321

5.3.2 Two dimensional analysis

In this analysis, we follow a technique described by Abeele et al [70] and focus on the linear reconstruction. The SV is partitioned into 3 CVs. We had to use a basic unit, for imposing periodicity. In a one-dimensional case, the basic unit was the spectral volume itself. It comprises of 2 SVs in 2D. The basic unit (only SVs are shown) is shown in figure 5.12. This solution process would result in solving a [6*6] set of equations. The rate of change of the solution (in the basic unit) can be written as a linear combination of solutions from the left (L), right(R), top (T), bottom (B) and central(C) units. The formulations can be cast in the following form:

$$\frac{d}{dt}\left[\bar{u}_{i,j}\right] = \mathrm{L}\left[\bar{u}_{i-1,j}\right] + \mathrm{R}\left[\bar{u}_{i+1,j}\right] + \mathrm{C}\left[\bar{u}_{i,j}\right] + \mathrm{B}\left[\bar{u}_{i,j-1}\right] + \mathrm{T}\left[\bar{u}_{i,j+1}\right], \tag{5.21}$$

where L,R,C,B and T are constant matrices and $\left[\overline{u}_{i,\,j}\right]$ is the solution vector of the basic building unit.

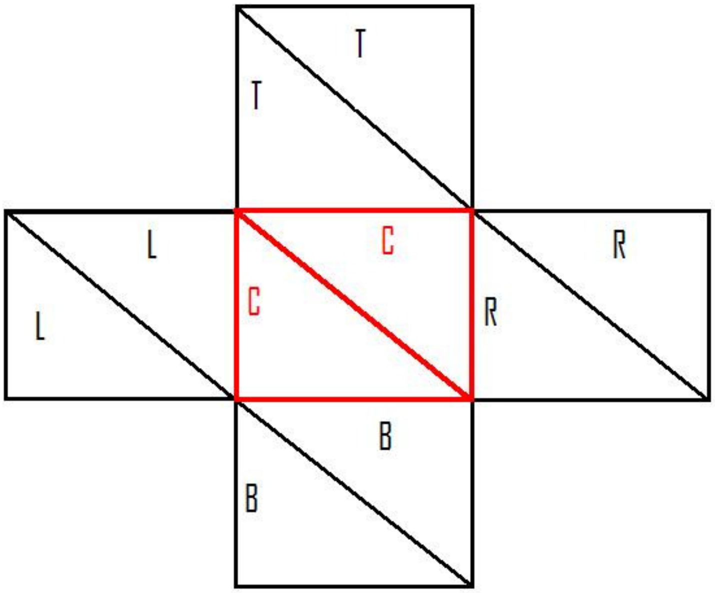


Figure 5.12 Basic Building block colored in red; the neighbors are used in analysis

We now seek general solution of the following form

$$u(x,t) = \hat{u}_k(t)e^{ik(x\cos(\theta)+y\sin(\theta))}, \tag{5.22}$$

where k is the index of modes (k=1,2..) representing the wave number and θ is the direction of wave propagation. Obviously, the analytical solution for equation 2.12 is $u(x,t) = e^{ikx-k^2t}$. We follow the method described in the 1D analysis section and obtain the following result:

$$\left[\hat{u}_k'\right]=\mathrm{G(k,h)}\ \left[\hat{u}_k\right], \tag{5.23}$$

where the amplification matrix is given by

$$\mathrm{G} = e^{-ikh\cos(\theta)}\,\mathrm{L} + e^{ikh\cos(\theta)}\,\mathrm{R} + \mathrm{C} + e^{-ikh\sin(\theta)}\,\mathrm{B} + e^{ikh\sin(\theta)}\,\mathrm{T} \tag{5.24}$$

5.3.2.1 Spatial analysis

There is an extra degree of freedom, θ. Due to symmetry of the system and of the building unit, we need to focus on θ varying from 0 to π/4 radians. In this thesis, we focus on these extreme cases.

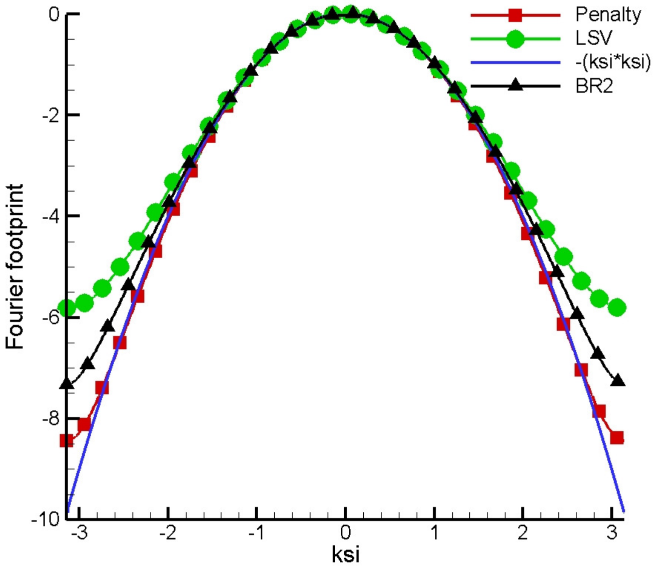


Figure 5.13 Fourier footprint for the 2D 2^{nd} order scheme for θ = 0 radians

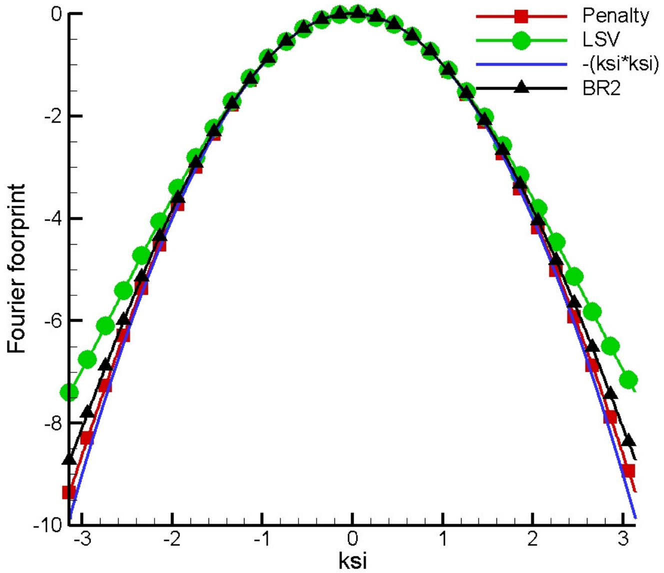


Figure 5.14 Fourier footprint for the 2D 2^{nd} order scheme for θ = π/4 radians

Figures 5.13 and 5.14 show the variation of the principal eigen value with respect to the non dimensional frequency χ (=kh) for θ = 0 and π/4 radians respectively. Figures 5.15 and 5.16 show the errors at low frequencies.

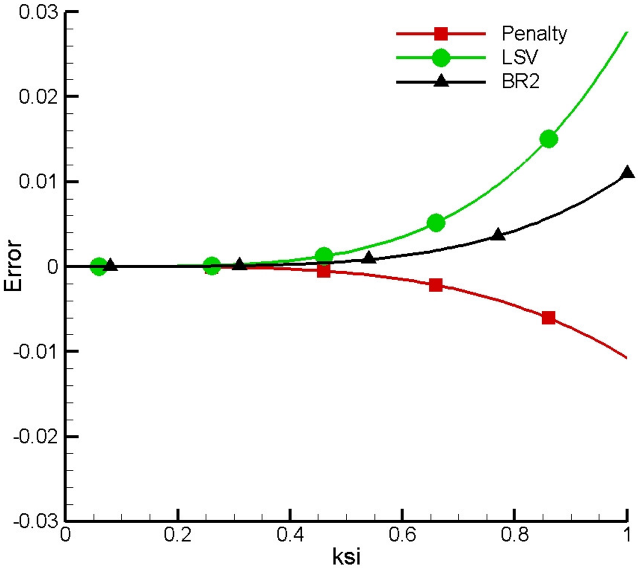


Figure 5.15 Error associated at low wavenumbers for the 2D 2^{nd} order scheme for θ = 0 radians

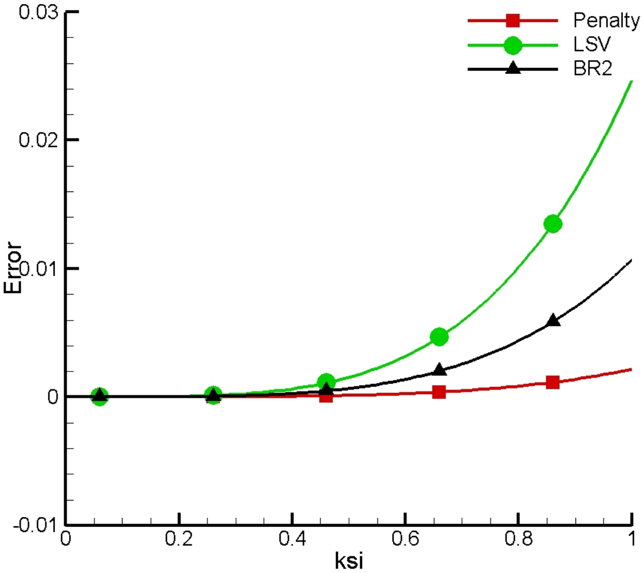


Figure 5.16 Error associated at low wavenumbers for the 2D 2^{nd} order scheme for θ = π/4 radians

In general, the trends are qualitatively similar to those obtained using 1D analysis. In addition, it can be seen that the errors are much lower when θ = π/4 radians. This is probably due to the way our SVs are defined (right triangles).

5.3.2.2 Temporal analysis

As in the 1D analysis, we stick to the 3-stage SSP Runge Kutta scheme. Table II lists the maximum non dimensional time step required for obtaining a stable solution. Once again, the penalty and BR2 methods permit a higher time-step limit than the LDG method. In addition, the time step permitted by the BR2 method is more than that of the penalty method.

Table II. Maximum non-dimensional time step for obtaining stable solutions for the 2^{nd} order, 2D LDG, penalty and BR2 methods

Case	θ = 0	θ = π/4
LDG	0.0267	0.0267
Penalty	0.0465	0.0465
BR2	0.0635	0.0635

5.4 Analysis for non uniform grids

Analysis was also done for non uniform grids in 1D. We used the building block concept to do the analysis. Each building block consists of 2 SVs. The ratio of the lengths of the 2 SVs is the clustering factor. This can be seen from figure 5.17. The first SV has a length of one unit and the length of the second SV is varied. Table III shows the maximum non dimensional time step which can be used for the different formulations and for different clustering factors. The trend is similar to the uniform grids scenario. The main conclusions are

1. Stable even for very high clustering ratios
2. Maximum possible non dimensional time step increases with increasing clustering
3. The maximum allowable time step follows a pattern similar to the structured grids i.e.

$\tau_{LDG} < \tau_{penalty} < \tau_{BR2}$

Figure 5.17 Basic Building block for non uniform grids in 1D: consists of 2 SVs

Table III. Maximum non-dimensional time step for obtaining stable solutions for the LDG, penalty and BR2 methods for different clustering factors

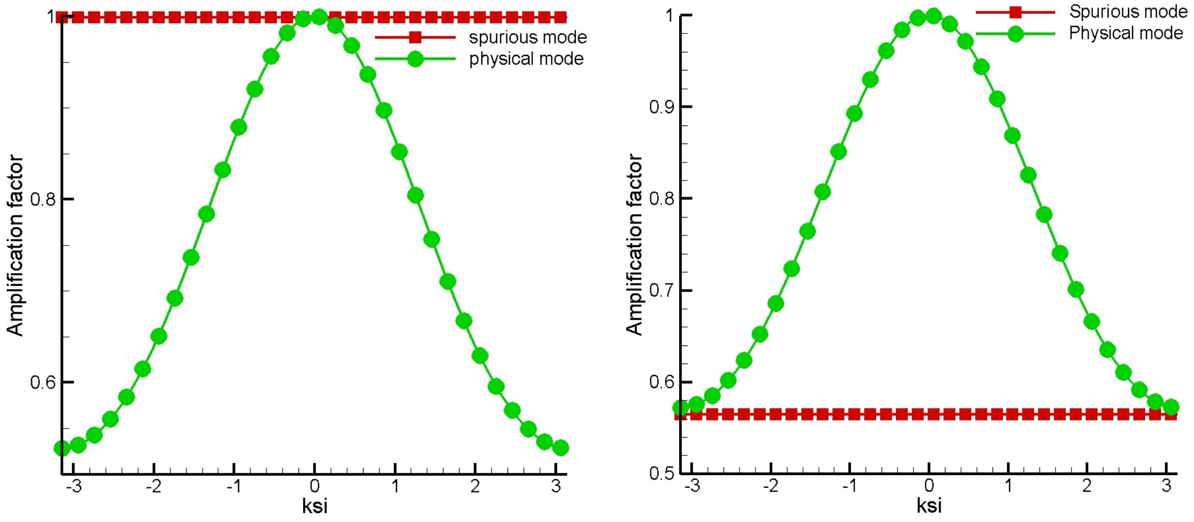
Order	Clustering factor	LDG	Penalty	BR2
2	0.5	0.246	0.332	0.508
2	0.1	0.352	0.545	0.752
3	0.5	0.0464	0.034	0.093
3	0.1	0.082	0.11	0.17
4	0.5	0.018	0.027	0.040
4	0.1	0.031	0.049	0.069

5.5 Discussion on stability and spurious modes

In the previous sections, we analyzed the stability requirements of the 3 stage SSP Runge Kutta smoother. It has been a common practice in CFD to use the maximum possible time step to drive a system to convergence. This time step is limited by the stability requirements (as determined in the earlier sections). A different trend was observed during the course of this study. Consider the 2nd order LDG case: the amplification matrix G has the eigen values -16/h^2 (spurious mode) and -2(1-cos (χ))/h^2 (physical mode). Thus if

$$\begin{bmatrix} \bar{u}^{n+1}_{j,1} \\ \bar{u}^{n+1}_{j,2} \end{bmatrix} = \mathrm{H}\,(\chi,\mathrm{dt}) \begin{bmatrix} \bar{u}^{n}_{j,1} \\ \bar{u}^{n}_{j,2} \end{bmatrix}. \tag{5.25}$$

Obviously for stability, the eigen values of the growth matrix H(χ,dt) need to lie between -1 and 1. Figure 5.18(a) shows the eigen values for this system for a non dimensional time step of 0.157. The system satisfies the requirement for stability. However, the spurious mode was never damped. The numerical solution overshoots the analytical solution and one can expect some oscillations. Therefore steady-state problems may never converge. The time step needs to be lowered to reduce the magnitude of the eigen value of the spurious mode. Figure 5.18(b) shows the eigen values for this system for a non dimensional time step of 0.138. In other words, there exists a range, wherein the system converges to a wrong solution. In general, this phenomenon can be seen in higher orders for LDG, Penalty and BR2 schemes.



(a) (b)

Figure 5.18 Eigen values of a 2nd order LDG growth matrix for RK 3stage scheme. Case (a): non dimensional time step=0.157; Case (b): non dimensional time step =0.138

The above conclusion was tested using a numerical simulation. Figure 5.19 shows the variation of magnitude (L1 norm) of the solution as a function of time for a 2nd order LDG. The simulation with CFL=0.157 is stable but has no damping. This is the case wherein the spurious mode dominates over the physical mode. The simulation with CFL=0.138 is both stable and consistent.

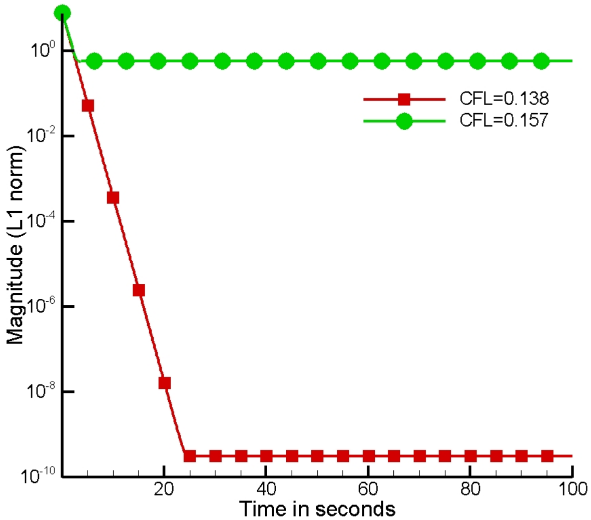


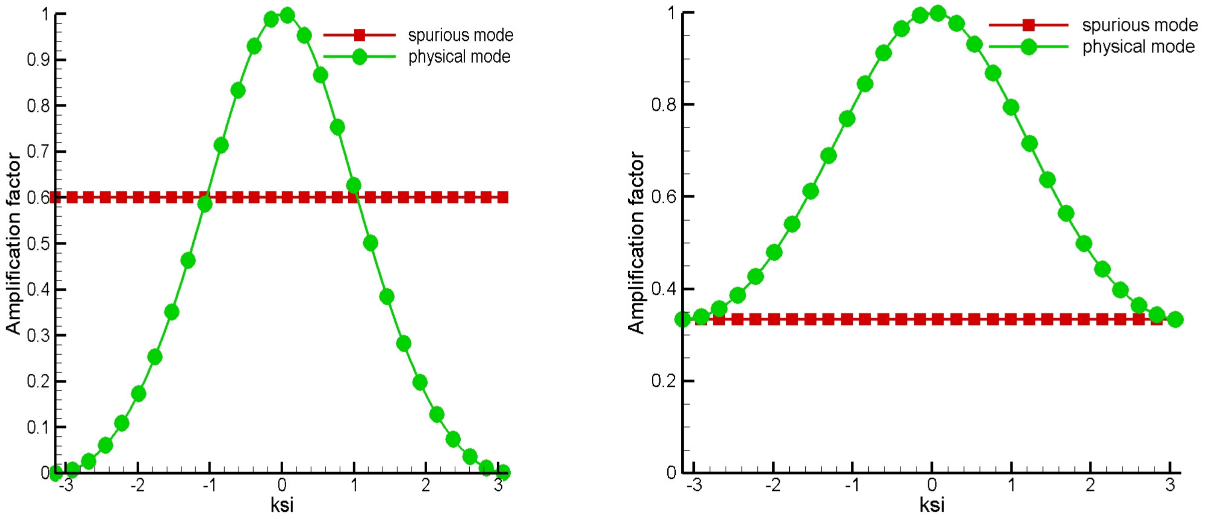
Figure 5.19 Damping occurring in a 2nd order LDG simulation using two different time steps

Table IV lists the non dimensional time steps required for obtaining a stable solution as well as a physically relevant solution. Thus in an explicit simulation, it is generally necessary to run the case with a time step lower than the cut-off stability limit.

Table IV. Non dimensional time step criteria for obtaining stable and physically relevant solutions for the LDG, penalty and BR2 schemes.

Case	2nd order (Stability)	2nd order (Relevant)	3rd order (Stability)	3rd order (Relevant)	4th order (Stability)	4th order (Relevant)
LDG	0.157	0.138	0.0266	0.0266	0.0109	0.0106
Penalty	0.182	0.175	0.0322	0.0322	0.0133	0.0129
BR2	0.314	0.224	0.0784	0.0784	0.0321	0.0300

The above pattern was also observed while using the Crank-Nicolson scheme. There were no stability issues as it is an implicit scheme. However, the spurious modes start to dominate after a critical value. Figures 5.20(a) and 5.20(b) show the eigen values of the H matrix using a non dimensional time step of 0.5 and 0.25 respectively. It is clear that the former has inadequate damping in most parts of the spectrum, while the latter is sufficiently damped.



(a) (b)

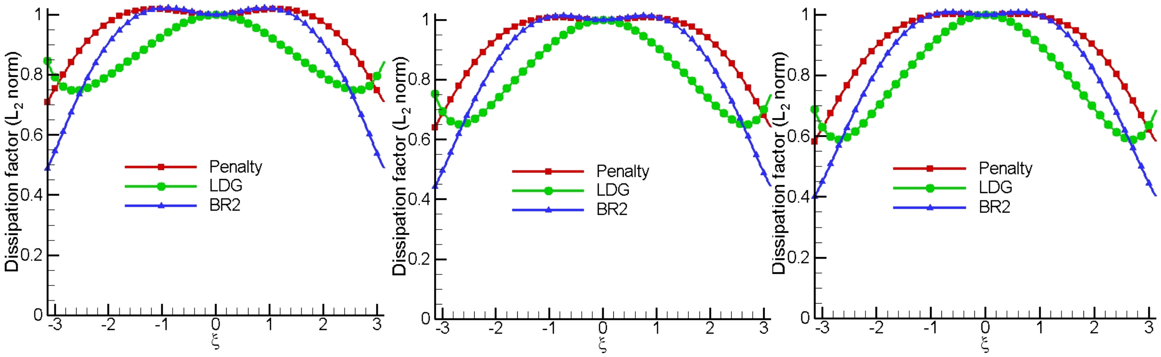
Figure 5.20 Eigen values of a 2nd order LDG growth matrix for Crank-Nicolson scheme. Case (a): non dimensional time step=0.5, case(b): non dimensional time step =0.25

5.6 Dissipative properties using multigrid techniques

The effect of dissipative properties of the three methods is compared using multigrid method. We consider the 3-2 and 4-3 order two level schemes in 1D. We use ideas similar to the ones listed in section 5.3.1.1. We use the multigrid idea to solve equation 2.17, 2.18. We analyze the effect of a single V cycle. This V cycle consists of 0 pre-sweeps in the higher polynomial level. By definition, the pre-sweeps are the sweeps performed before restricting the solution to the lower polynomial level. The solution and the defect are restricted to the lower polynomial level and the correction equation is constructed. We make use of the linear property of the system and solve the correction equation exactly. This is akin to solving the correction equation to steady state (using lots of iterations). The correction to the solution is prolongated to the higher polynomial level and is added to the existing value of the solution. A finite number of post-sweeps (0-3) are performed. By definition, the post-sweeps are the sweeps performed after correcting the solution using the lower polynomial level corrections.

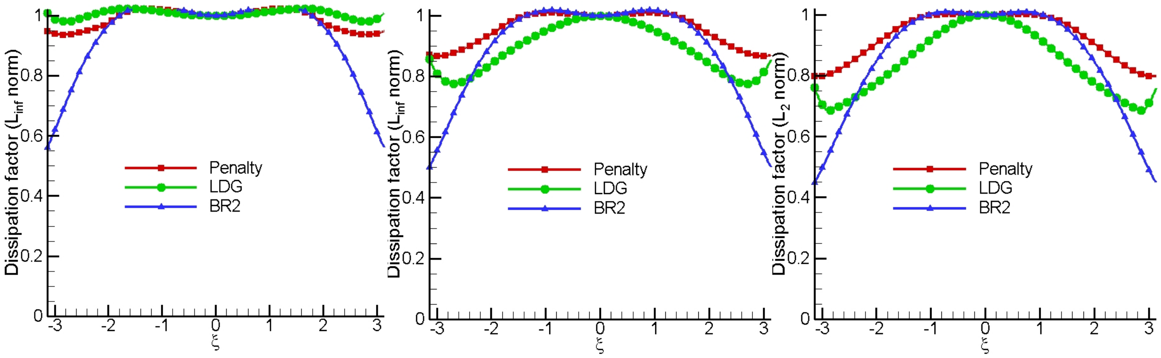
Figures 5.21 and 5.22 show the L_2 and L_{inf} damping factors for the 3-2 system. It can be seen that there exists some wave numbers where the solution experiences growth (rather than damping). A (small) non dimensional time step of 0.01 was used for the current simulations. In addition, the solution at high wave numbers is damped faster in the BR2 and the penalty simulations. Hence, we expect the LDG to have the most inefficient damping property for a 3-2 system.

Figures 5.23 and 5.24 show the L_2 and L_{inf} damping factors for the 4-3 system. It can be seen that the 4-3 system has instabilities at a much larger range of wave numbers, Numerical tests were performed and their results were in accord with the above. A non dimensional time step of 0.005 was used for the current simulations. The figures indicate that the LDG has the least amount of instability and might be the best choice for a 4-3 system! However, LDG has a very strict stability constraint and this is strong enough to offset the above advantage. BR2 has better dissipation at higher wave numbers than the penalty scheme and is expected to converge faster.



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

Figure 5.21 Damping factor for a 3-2 mg using the L$_2$ norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post-sweeps



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

Figure 5.22 Damping factor for a 3-2 mg using the L$_{inf}$ norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post-sweeps

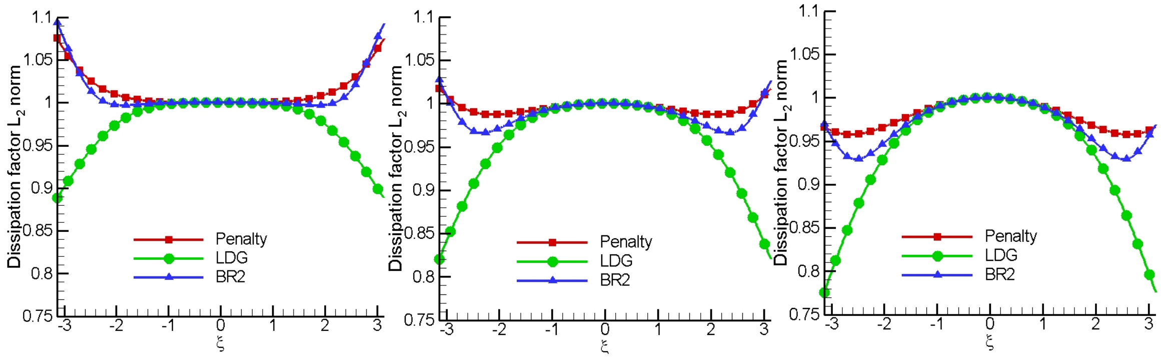


Figure 5.23 Damping factor for a 4-3 mg using the L$_2$ norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post-sweeps

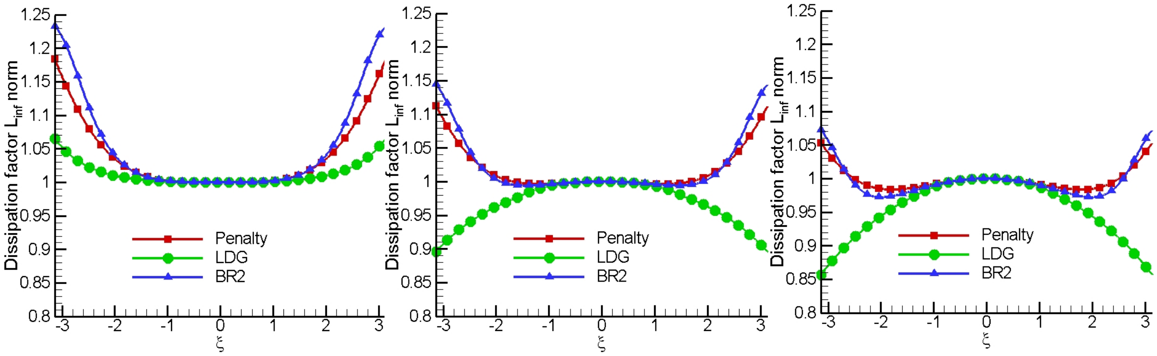


Figure 5.24 Damping factor for a 4-3 mg using the L$_{inf}$ norm. Case (a) Zero post-sweeps; Case (b): One post-sweep; Case (c): Two post-sweeps

CHAPTER 6 RESULTS AND DISCUSSIONS

6.1 SD results

6.1.1 Validation using 2D scalar conservation laws

6.1.1.1 Linear straight wave

We consider the linear straight wave over a square domain. The equation can be described as

$$\frac{\partial q}{\partial t}+\frac{\partial q}{\partial x}+\frac{\partial q}{\partial y}=0\,,\ (x,y)\in[0,1]^2 \text{ and } t>0\ . \tag{6.1}$$

The computational unstructured grid with 40x40x2 elements is used as shown in figure 6.1(a). Figure 6.1(b) shows the converged solution obtained using the normal implicit scheme on the same grid

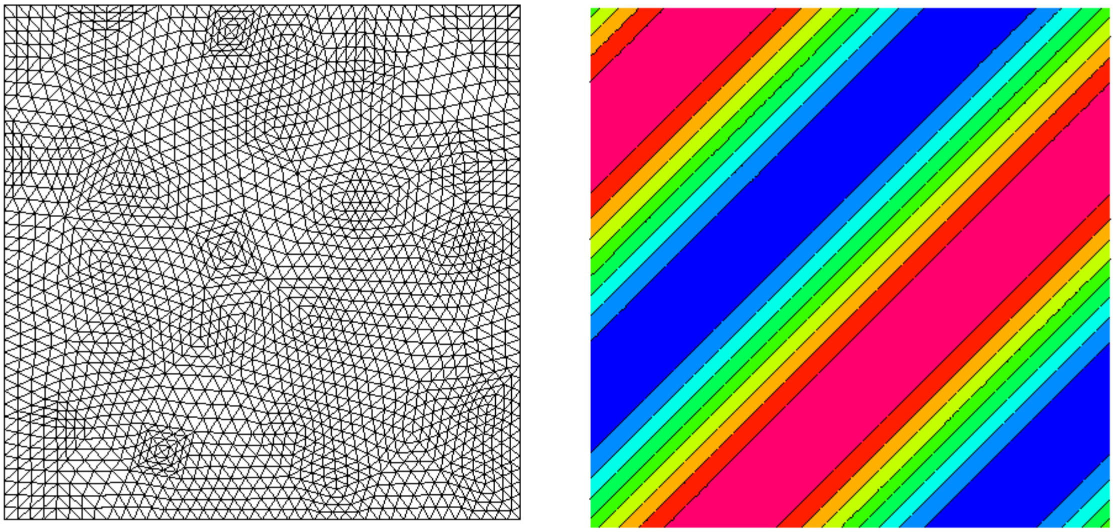

(a) (b)

Figure 6.1: Linear wave equation case. Case (a): Grid 40x40x2; Case (b): Contour Plot

Subsequently, we compared the convergence histories obtained by the explicit scheme and implicit schemes. Note that the computational grid is relatively coarse and the equation is very simple. Figure 6.2 demonstrates that the simplified implicit scheme without the $\frac{\Delta Q_c^*}{\Delta t}$ term in equation (3.9) converging speed is about 7 times as fast as the one of the explicit scheme, while the normal implicit scheme converging speed is only twice as fast as the explicit scheme.

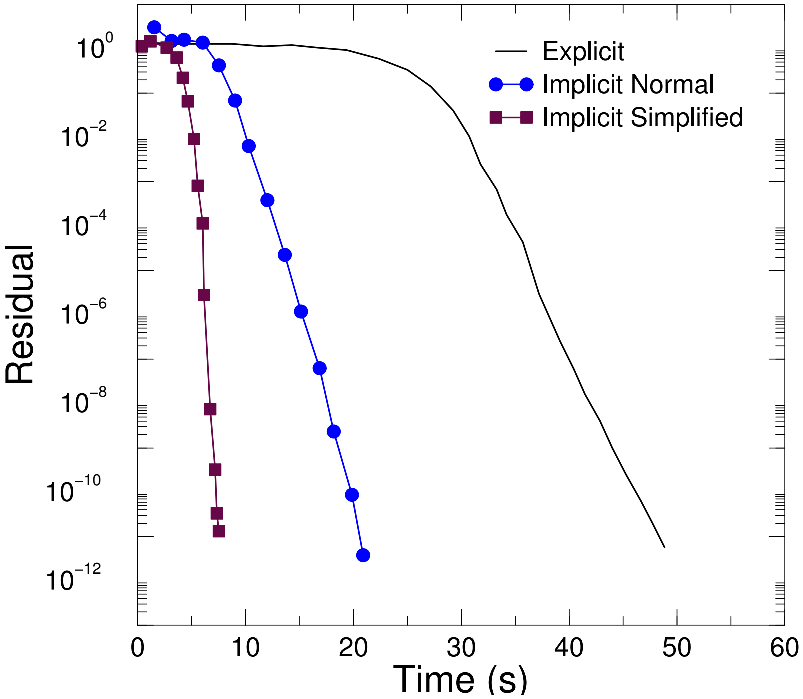


Figure 6.2: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the 2D linear wave case

6.1.1.2 Rotational wave

The second case we worked on is the linear scalar circular advection of a Gaussian profile over a square domain. This initial boundary value problem is conveniently expressed as

$$\frac{\partial q}{\partial t}+y\frac{\partial q}{\partial x}+(1-x)\frac{\partial q}{\partial y}=0, \quad (x,y)\in[0,1]^2 \text{ and } t>0. \tag{6.2}$$

$$q(x,0,t)=e^{-50(x-0.5)^2},\ x\in[0,1] \text{ and } t\geq 0.$$

$$q(x,1,t)=0,\ x\in[0,1] \text{ and } t\geq 0.$$

$$q(0,y,t)=0,\ y\in[0,1] \text{ and } t\geq 0.$$

The above has exact steady solution $q_{exact}(x,y)=e^{[-50(0.5-(1-\sqrt{(1-x)^2+y^2}))^2]}$, $(x,y)\in[0,1]^2$. It facilitates the accuracy study as shown in Table V. The third order accuracy of the SD method with implicit LU-SGS relaxation scheme for a steady flow problem is clearly demonstrated.

Table V. L_1 and L_∞ errors and orders of accuracy of the scalar circular advection case using the third order SD method.

Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
10x10	0.00166705	--	0.0245699	--
20x20	0.000272822	2.61	0.00357253	2.78
40x40	3.97545e-05	2.78	0.000516237	2.79
80x80	5.29239e-06	2.91	6.43088e-05	3.00

This example can conveniently assess the speed-up of the implicit LU-SGS simplified relaxation scheme on the same unstructured grid shown in figure 6.1(a). The simplified implicit scheme without $\frac{\Delta Q_c^*}{\Delta t}$ in equation (3.9) is about 1.5 times as fast as the normal implicit scheme (3.9) from figure 6.3. Figure 6.4 shows the converged smooth solution obtained using the simplified implicit scheme. Although the left-hand-side Jacobian matrix is pretty small, for the third order scheme it is only 6x6, the advantage of the simplified implicit over the three-stage Runge-Kutta explicit method is clear, and for this particular case, the speedup is around 10. The p-Multigrid method is also applied to the linear wave case. The acceleration by the p-Multigrid method is small, possibly due to the simple form of the governing equation and the nonlinear nature of the FAS.

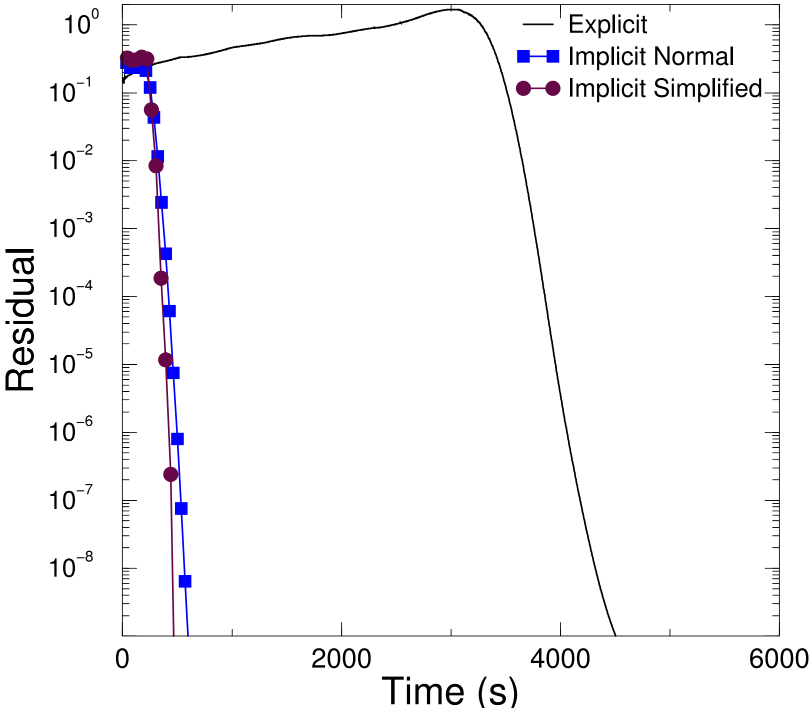


Figure 6.3: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the 2D circular convection case

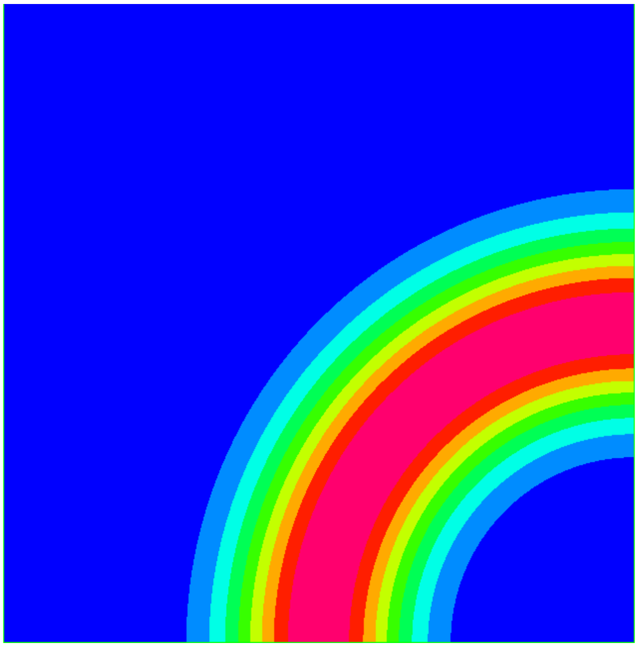


Figure 6.4: Contour plot obtained for 2D scalar circular convection case

6.1.1.3 Nonlinear wave

Application of the p-Multigrid method to nonlinear wave equations can improve the above situation. This case was formulated with the sole purpose of testing our implicit full Multigrid ideology. Unlike the linear wave test cases, the initial time step is limited even for the implicit simulations. The full Multigrid approach starts by solving the problem using lower order polynomials, and using these solutions as initial approximations for the next higher level. This process continues until it reaches the finest level where the solution of the problem is required. It results in a smooth, good initial guess. Hence a much higher time step can be used for the initial V or W cycles.

We consider a nonlinear 2D scalar initial boundary value problem

$$\frac{\partial \tilde{Q}}{\partial t}+\frac{\partial \tilde{f}}{\partial x}+\frac{\partial \tilde{g}}{\partial y}=0 \text{ and } \tilde{f}=\tilde{g}=\frac{1}{3}\tilde{Q}^3. \tag{6.3}$$

Equation (6.3) has an analytical solution $\tilde{Q}(x,y)=\sin(\pi(x-y))$ and $(x,y)\in[-1,1]^2$. As shown in figure 6.5, the speedup factor obtained by the three-level full p-Multigrid scheme is about 5 when compared to the single level implicit scheme. As expected, the computational CPU time of the single level explicit scheme is more than one and a half orders magnitude higher than the three-level implicit p-Multigrid.

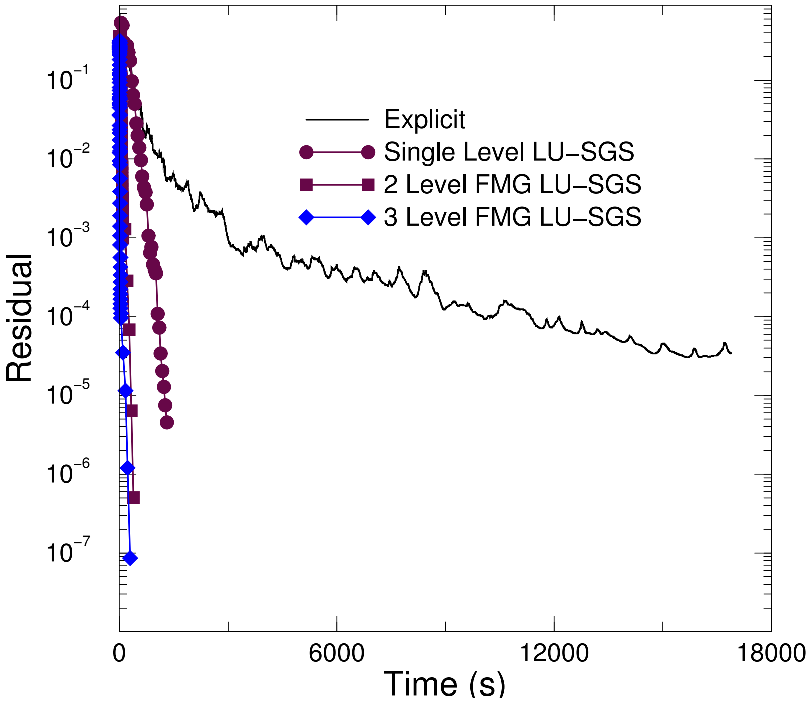
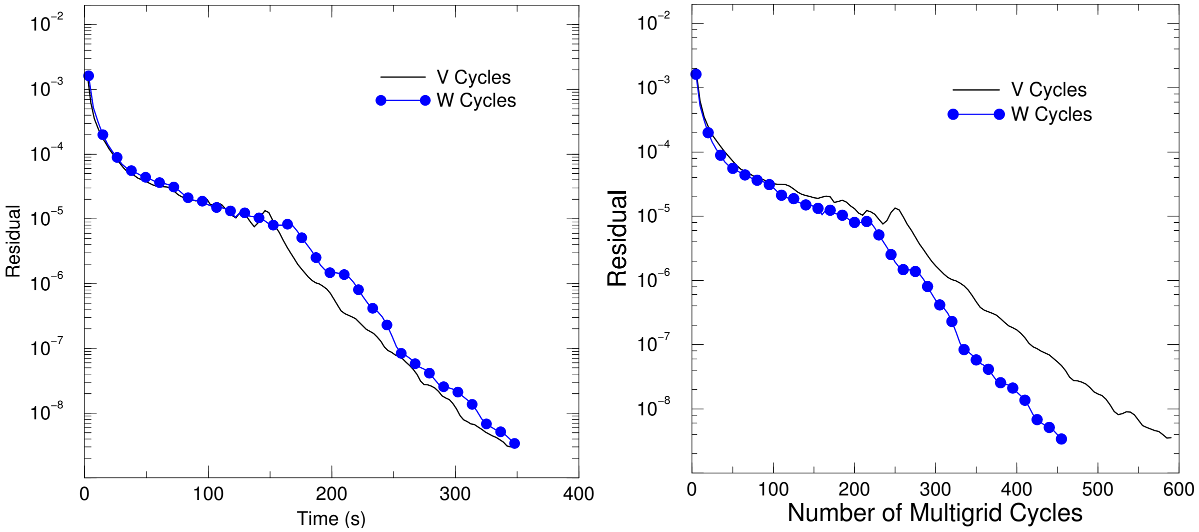


Figure 6.5: Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the non linear wave case.

It is easy to extend the above p-Multigrid method with V cycles to include W cycles if p0 and p1 levels are used once again before proceeding to the p2 level. The problem requires lesser W cycles than V cycles to attain convergence. This is shown is figure 6.6(a). However W cycles consumed slightly more time than the V cycles. This is shown is figure 6.6(b).



(a) (b)

Figure 6.6: Non Linear wave case using a LU-SGS implicit smoother. Case (a): Convergence history as a function of Multigrid cycles. Case (b): Convergence History as a function of time.

Both three-stage and five-stage TVD Runge-Kutta schemes are tested for this problem. Five-stage one is proven more effective when it is applied to all three p-levels. Therefore, in the subsequent part of the paper, we only discuss the results obtained using the five-stage Runge-Kutta scheme for explicit smoothers.

The problem was also tackled by combining the explicit and implicit smoothers. Since the implicit smoother is more effective than the explicit smoother, we use 20-30 iterations of explicit smoother at the p2 level to attenuate the prolongation errors generated by 6 LU-SGS iterations at p0 level and a single LU-SGS iteration at p1 level. The speedup factor is still more than 20 compared to a single level explicit scheme for the normalized residual reducing to 10^{-5} (fig 6.7).

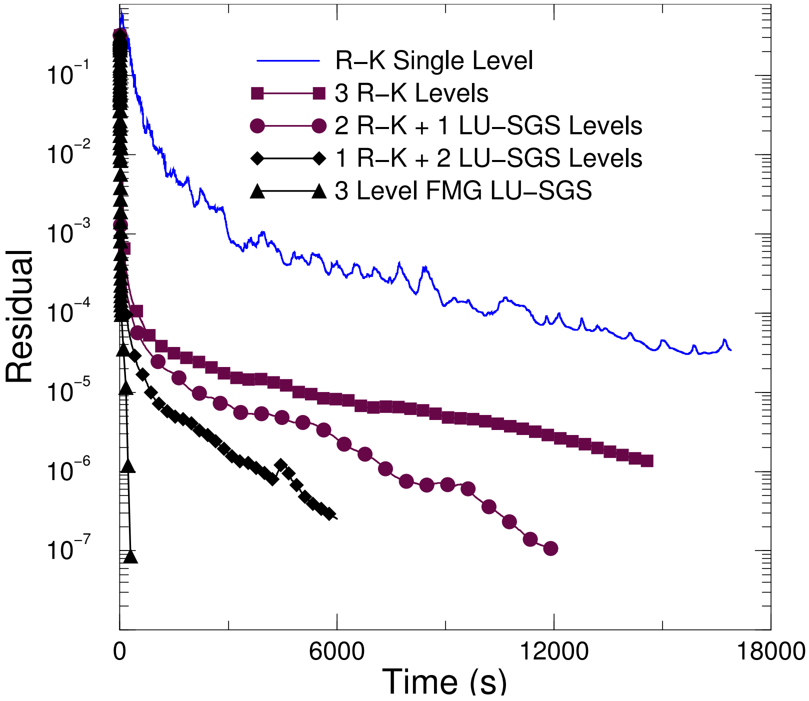


Figure 6.7: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the non linear wave case

The second approach is to use explicit smoothers for all the three levels. The speed is about 4 times that of a single level to reach residual level of10^{-5}. The last approach is maintaining explicit smoothers for p2 and p1 levels, but using implicit smoother for the p0 level. As expected, the speed of residual convergence to 10^{-5} lies in between the above two approaches and is about 8 times as fast as that of single level. Note that the convergence history curve using implicit LU-SGS 3 level full multigrid is also plotted. This has a convergence rate, which is faster than any of the above schemes

6.1.2 Results for the Euler equations

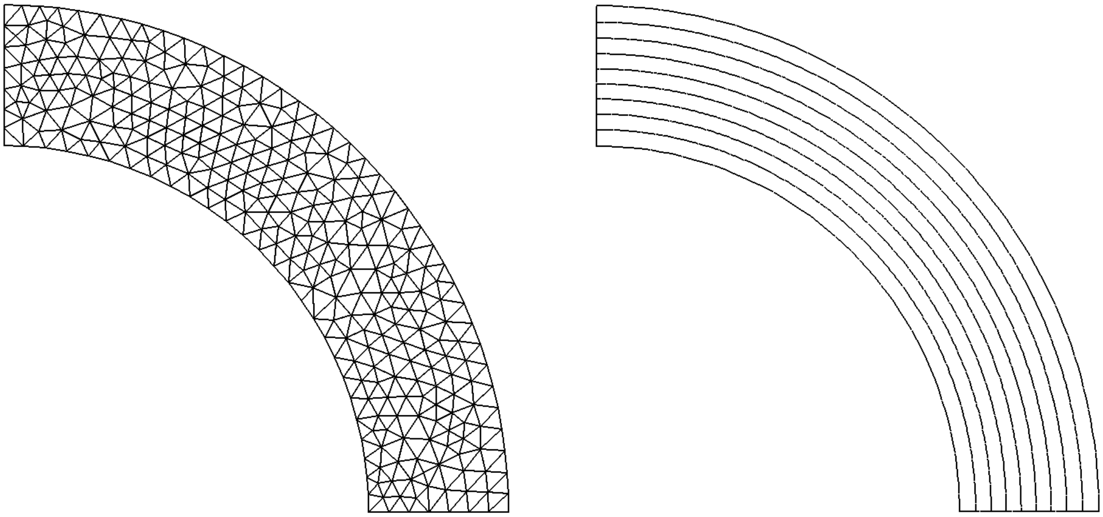
The Jacobian D matrix for the Euler equations is bigger than the scalar equations, since we have four conservative variables at each unknown point. For example, D is a 24x24 matrix for a third order SD formulation of the 2D Euler equation. The D matrix is frozen every 4 Multigrid cycles. The D matrix is frozen for around 20 Multigrid cycles when the steady residual drops to below5×10^{-4}. The Rusanov flux is used for the flux vector computation at the element boundary interfaces. A quadratic curved boundary condition is adopted [78] for non-straight wall boundary surfaces. The above was originally published in [39,40].

6.1.2.1 2D supersonic vortex flow

This test case is used to assess the order of accuracy of the p-Multigrid method for the SD scheme. An inviscid, isentropic, supersonic flow between concentric circular arcs presents a situation where the velocity and Mach number vary inversely with radius and the entropy is constant. This is a shock-free compressible flow and hence the measured order of accuracy is not contaminated by limiter action near shocks. The behavior will correspond to what one may expect of smooth regions of inviscid subsonic compressible flow. This particular type of flow was used in [43,2] to verify the order of accuracy of their unstructured grid methods. The expression for density ρ as a function of radius r is given by

$$\rho(r)=\rho_i\left\{1+\frac{\gamma-1}{2}M_i^2\left[1-\left(\frac{r_i}{r}\right)^2\right]\right\}^{\frac{1}{\gamma-1}},$$

where M_i and r_i are the Mach number and radius at the inner arc. In the present calculations, the inner and outer radii are 1 and 1.384 respectively. A typical computational grid is shown in figure 6.8(a). The Mach number, density and pressure are specified as constants 2.25, 1, and $1/\gamma$ respectively for the inner arc boundary. The outer arc and bottom inlet boundaries are fixed with analytical solutions. The zero-gradient extrapolation boundary is employed for the exit. Figure 6.8(b) shows the solution of pressure obtained using the single level implicit simplified method.



(a) Grid with 528 elements (b) pressure contours

Figure 6.8: Supersonic vortex flow case. Case (a): Computational grid. Case (b): Computed pressure contours.

Four different grids are used for the calculations. There is no difference in terms of L_1 and L_2 errors for the implicit LU-SGS and the explicit R-K methods. The results are summarized in Tables VI and VII. Nearly third order is demonstrated using p2 polynomials of unknown points for the SD method and second order is achieved using p1 polynomials. Most importantly, implicit scheme and p-Multigrid method are not found to corrupt the measured order of accuracy.

Table VI. L_1 and L_2 errors and orders of accuracy of the supersonic vortex flow case using the third order SD method.

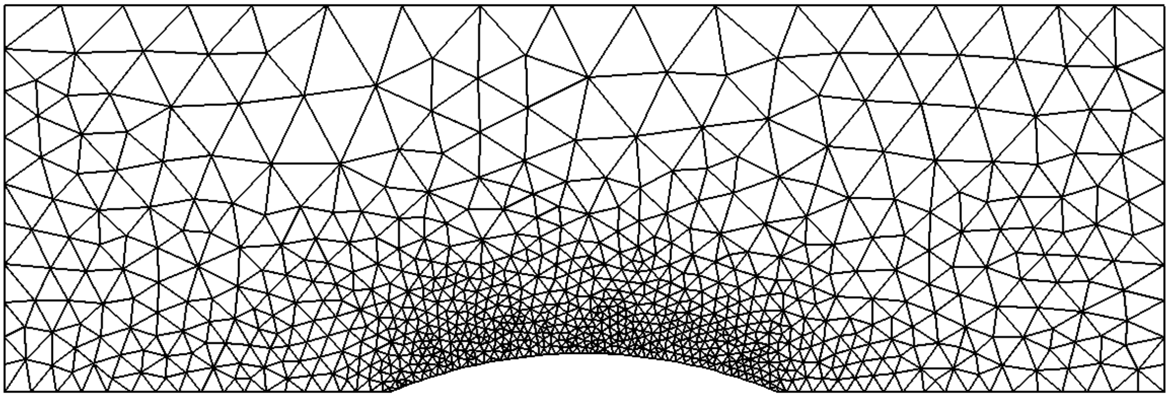
Cell no.	L_1 error	L_1 order	L_2 error	L_2 order
131	3.42e-4	--	5.09e-4	--
528	4.97e-5	2.77	7.83e-5	2.68
1892	8.93e-6	2.79	1.67e-5	2.52
7590	1.32e-6	2.77	2.74e-6	2.61

Table VII. L_1 and L_2 errors and orders of accuracy of the supersonic vortex flow case using the second order SD method.

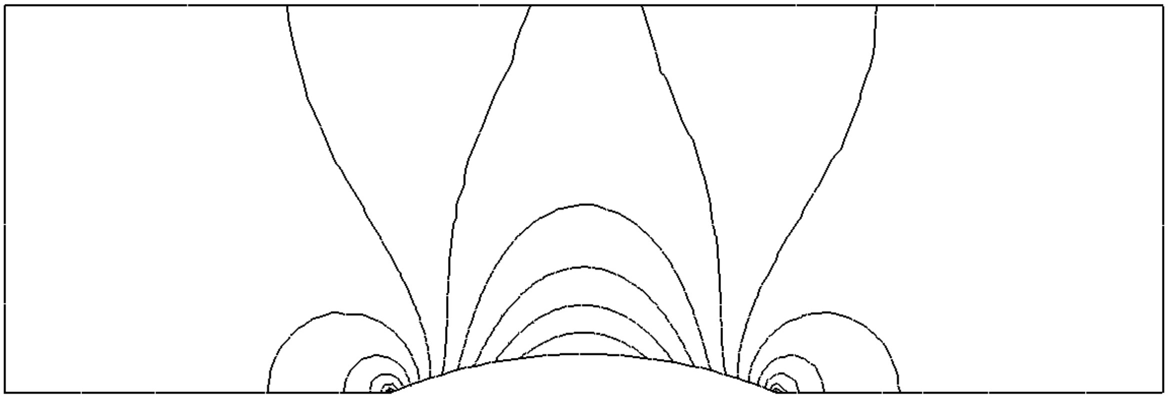
Cell no.	L_1 error	L_1 order	L_2 error	L_2 order
131	3.90e-3	--	4.98e-3	--
528	1.00e-3	1.95	1.34e-3	1.88
1892	3.25e-4	1.83	4.52e-4	1.76
7590	7.68e-5	2.08	1.03e-4	2.13

6.1.2.2 2D subsonic flow over a bump

We chose a testing case of the subsonic flow over a bump at Mach=0.5. This case has been used by p-Multigrid method for DG formulations of Euler equations in [43,51]. A 10% thick circular bump is mounted on the center of the channel bottom. The length of the channel is 3, its height 1, and its width 0.5. The computational grid with 3140 elements is shown in figure 6.9(a). The circular surface of the bump needs a higher-order boundary treatment and a quadratic boundary as described in [78] is adopted. Figure 6.9(b) shows the pressure contour obtained by the three-level p-Multigrid method. It is approximately identical to the pressure contour shown in [51].



(a)



(b)

Figure 6.9: Subsonic flow over a bump confined in a channel. Case (a): Computational grid. Case (b): Computed pressure contours.

The maximum CFL number used for all the implicit computations is around 8. Figure 6.10 shows the residual convergence history of the implicit schemes. The speedup obtained by the two-level p-Multigrid method is around 1.6 compared to the single level implicit scheme. The three-level p-Multigrid method accelerates the convergence further and the speed is 3.5 times as fast as the single level implicit method. The overall speedup i.e. the speedup attained by the three-level p-Multigrid is about 25 compared to the single level explicit scheme. Note that all the Multigrid methods mentioned so far is based V cycles for this particular case.

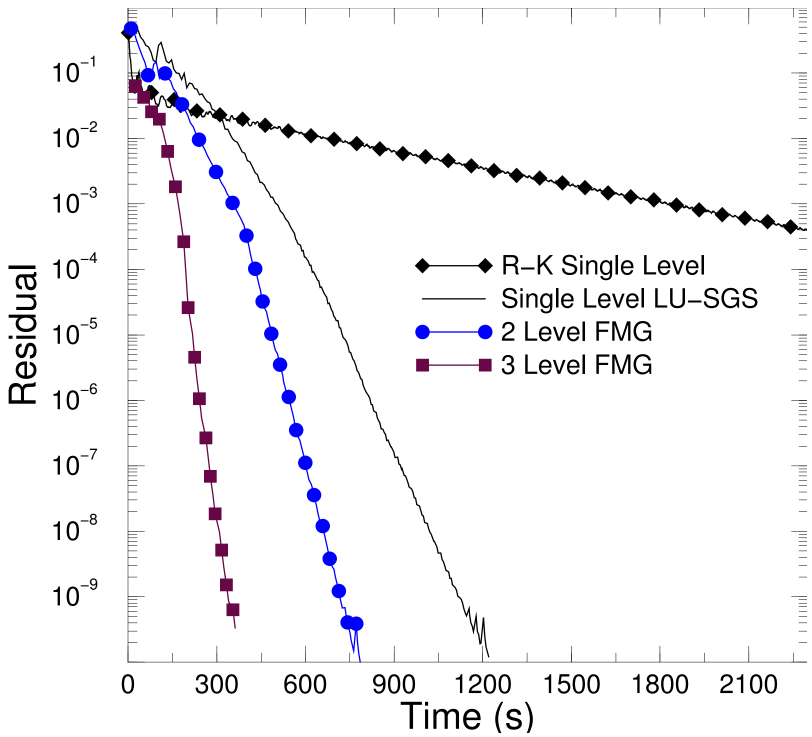
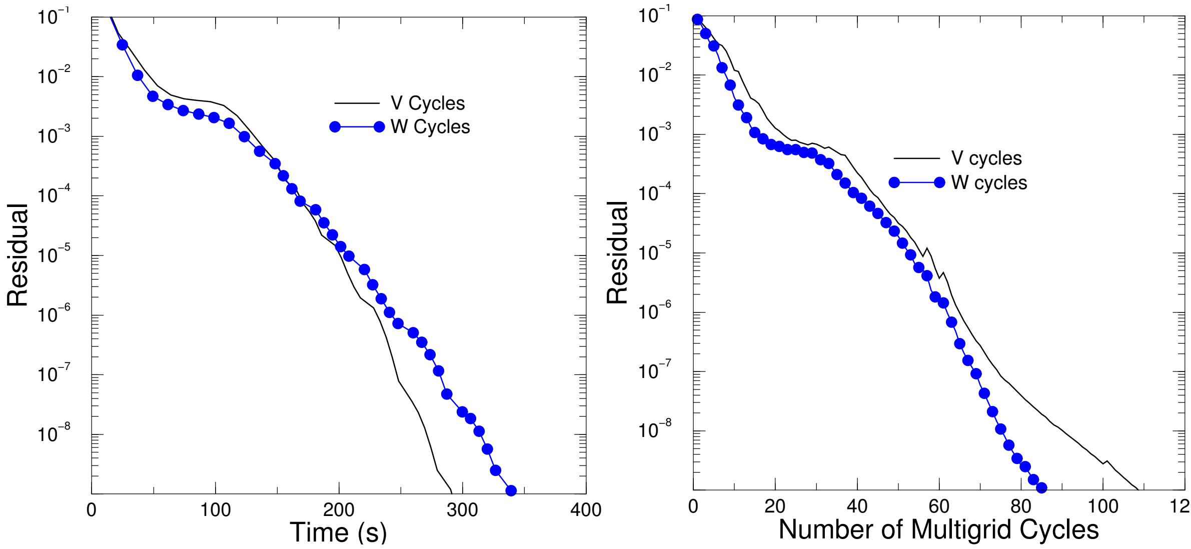


Figure 6.10: Comparison of convergence history with respect to time for the bump testing case using explicit R-K and implicit LU-SGS schemes.

As far as the implicit smoothers are considered for all three p-levels, we also examined the difference between V cycles and W cycles for the three-level p-Multigrid method. As expected, the current case requires lesser W cycles than V cycles to converge to machine zero as shown in figure 6.11(a). However the total time consumed is lesser when the V cycles were employed. This is shown in fig 5.11(b). Note that, the number of iterative smoothers performed for V cycles is 1-1-20-1-1 for third, second, first, second and third order levels respectively. More coarse level iterative smoothers are used in the W cycles, i.e., one, one and 16 iterations for p2, p1 and p0 levels respectively.



(a) **(b)**

Figure 6.11: 2D subsonic flow over a bump case using implicit LU-SGS smoother. Case (a): Convergence history as a function of Multigrid cycles. Case (b): Convergence History as a function of time.

All the above Multigrid calculations for the bump case are using implicit smoothers. If the explicit smoothers are used for all three p-levels (fig 5.12), a speedup of 8 was attained over the explicit single level case. As expected, the 3 level explicit scheme is slower than the mixed scheme (1 R-K + 2 LU-SGS). Note that, we use one-one-six-one-one-iteration explicit smoothers at the p2-p1-p0-p1-p2 levels to form a standard V cycle. The curve with the second fastest convergence rate in figure 6.12 is obtained using a calculation employing explicit smoothers only for the p2 levels and implicit smoothers for both p1 and p0 levels. The 3 level FMG LU-SGS is also plotted and is much faster than any of the above discussed methods.

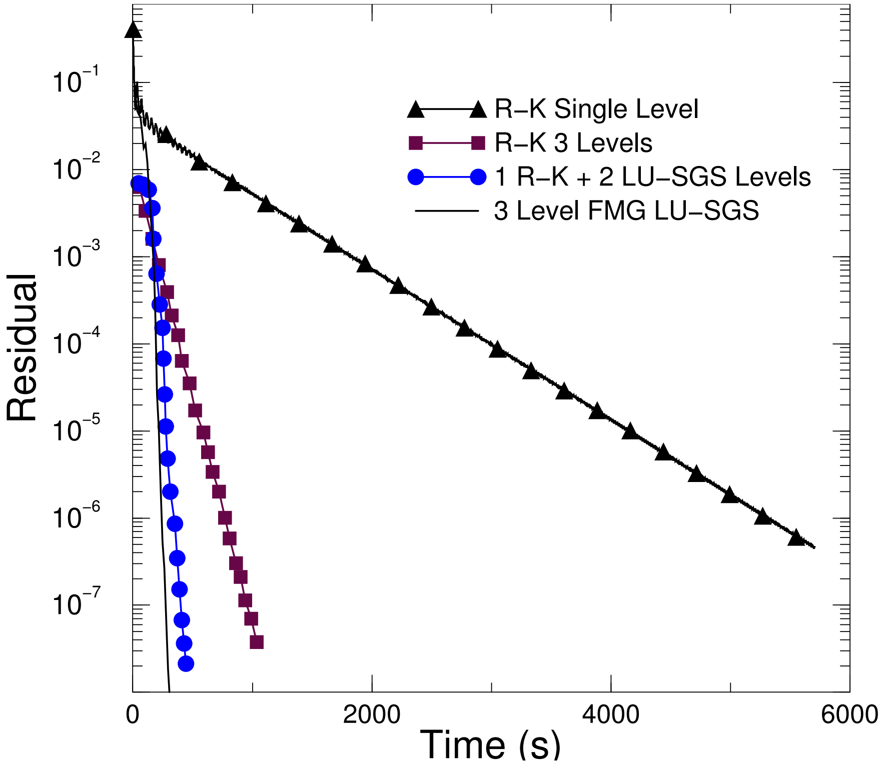


Figure 6.12: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the bump testing case

6.1.2.3 2D subsonic flow over an airfoil

The final testing case of the Euler equations is the subsonic flow over a NACA0012 airfoil at Mach=0.4 and angle of attack of zero degree. The computational grid is shown in figure 6.13. The outer boundary is 20 chords away from the airfoil centre. Figure 6.14 shows pressure contours obtained using two-level p-Multigrid scheme, i.e. using the p2 and p1 polynomials. The maximum CFL number used for the implicit computations is around 6.5. The explicit scheme limits the maximum CFL number to 0.06.

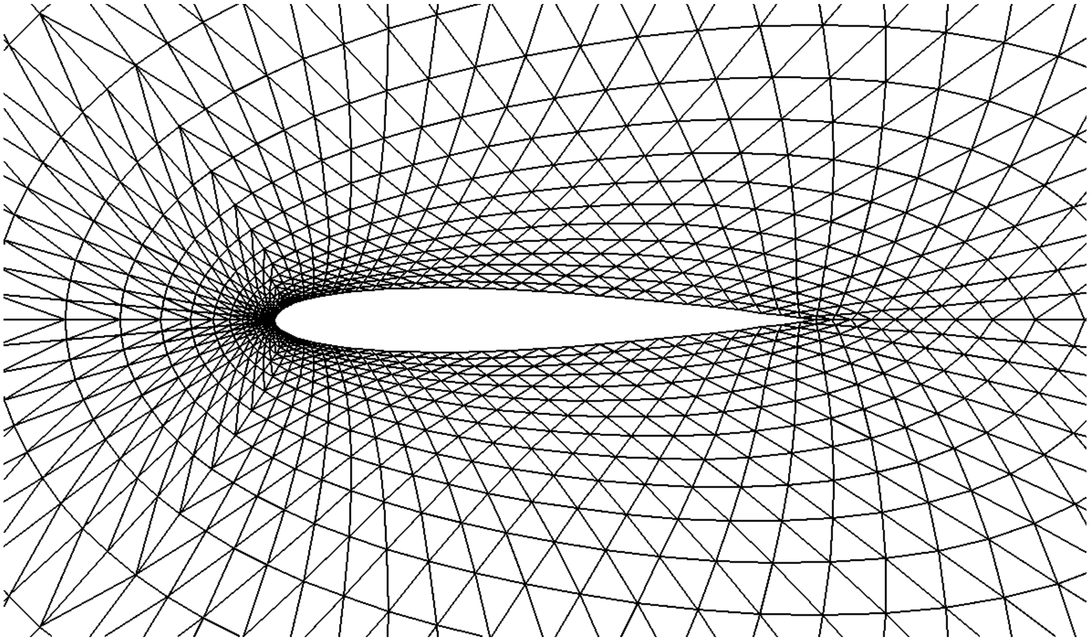


Figure 6.13: Grid(72*24*2) used for the subsonic flow over the NACA 0012

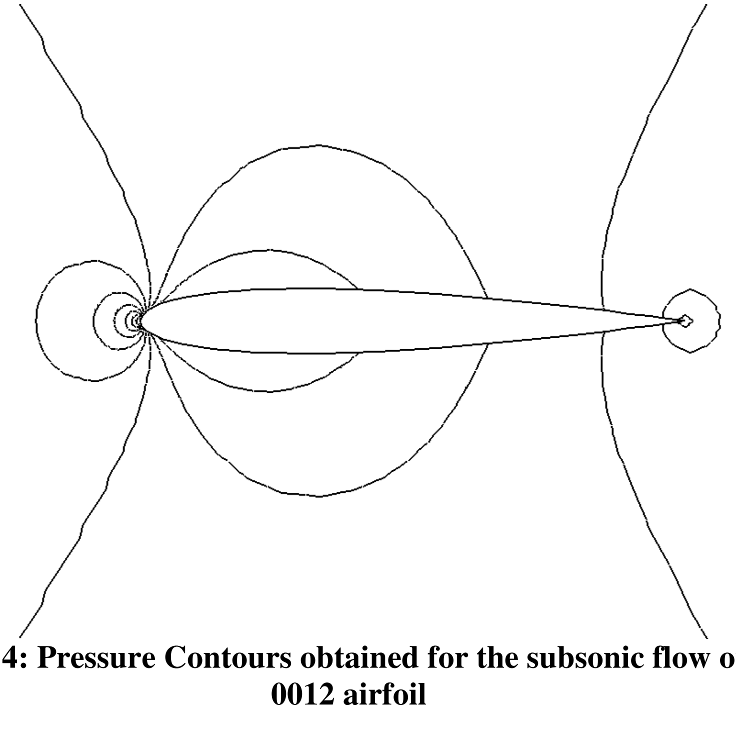


Figure 6.14: Pressure Contours obtained for the subsonic flow over the NACA 0012 airfoil

From figure 6.15, we can see that the highest speedup factor obtained using the three-level p-Multigrid is around 100 compared with the single level explicit scheme. The three-level p-Multigrid method is about 4 times as fast as the single level implicit method. The two-level scheme using the implicit smoothers for p2 and p1 levels shows a fast convergence too. The speedup factor of the two-level scheme is around 70 compared to the single level explicit scheme.

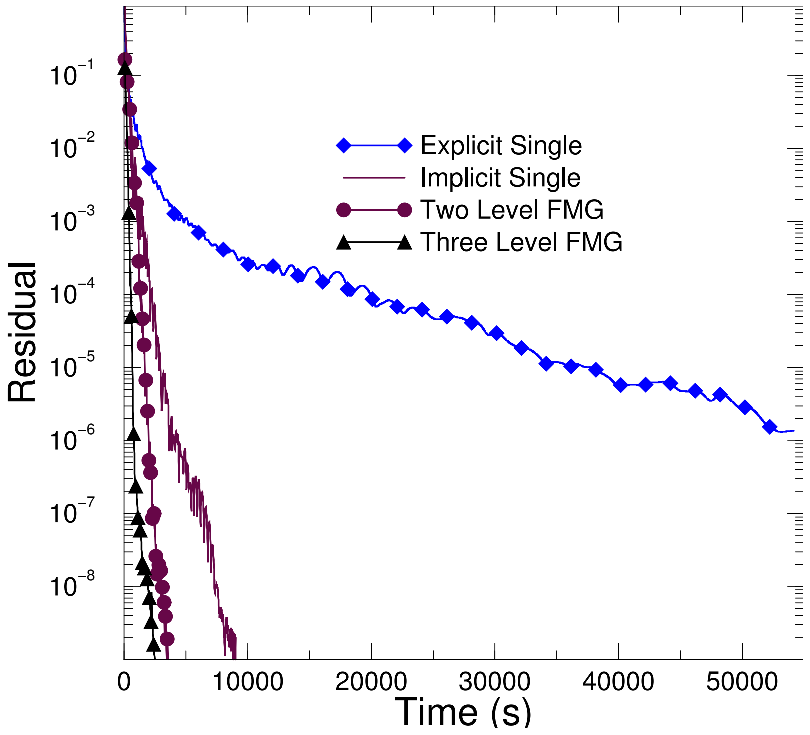


Figure 6.15: Comparison of convergence history with respect to time for the airfoil testing case using explicit R-K and implicit LU-SGS schemes

The effect of the explicit smoothers is also studied and is shown in figure 6.16. Full Multigrid approach is used for the three level calculations. The first calculation (3 level R-K) uses explicit smoothers for all p-levels. One-one-six-one-one-iteration explicit smoothers are employed for p2-p1-p0-p1-p2 levels to form a three-level V cycle. The second calculation is defined as mixed three-level approach. It employs explicit smoothers only for the p2 level. For stability reason, we use around 30 iterations of explicit smoother at the p2 level to smooth out the prolongation noises generated by the implicit smoothers of 6 iterations on the p0 level and a single iteration on the p1 level. The mixed approach is about 1.4 times as fast as the approach of R-K on 3 levels. The R-K 3-level approach is around 8 times as fast as the single level explicit method to reach residual level of10^{-6}. As expected, the 3 level implicit smoother is much faster than any of the above discussed schemes.

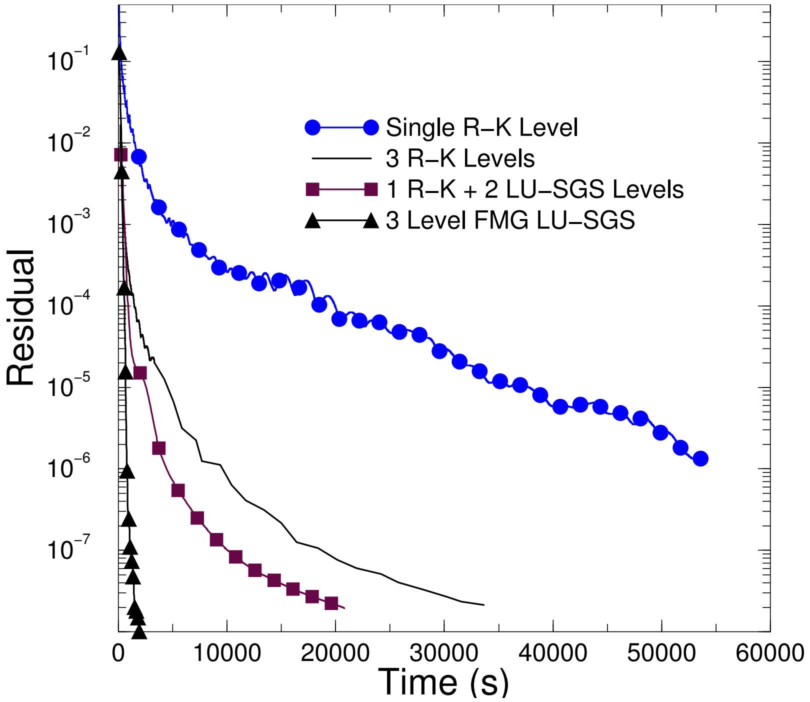
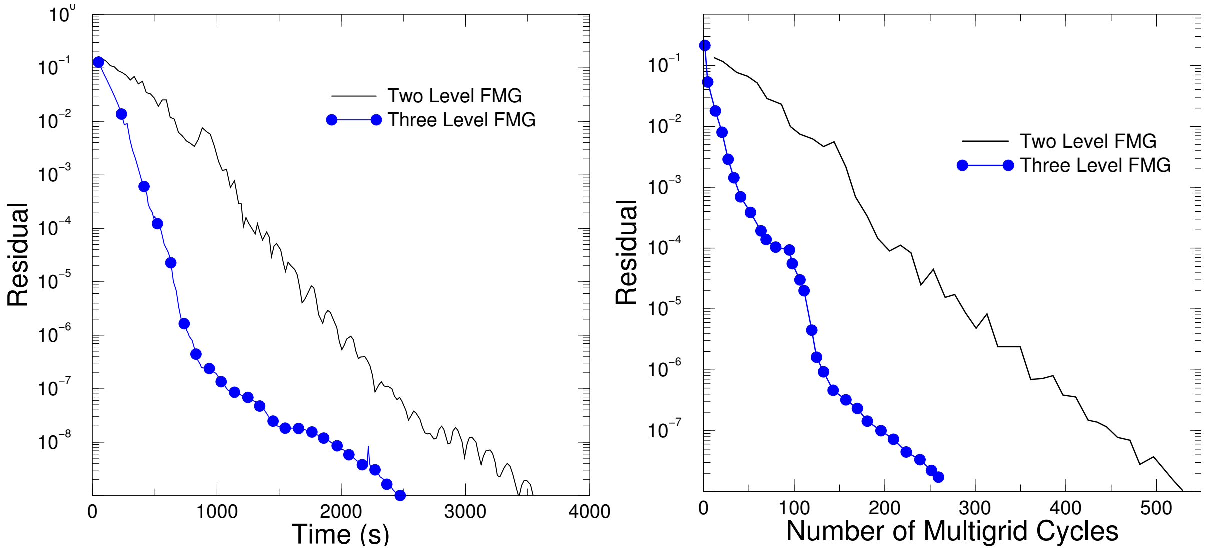


Figure 6.16: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the airfoil testing case



(a) **(b)**

Figure 6.17: Flow over an airfoil case. Case (a): Convergence history as a function of Multigrid cycles for two and three levels of FMG; Case (b): Convergence History as a function of time using two and three levels of FMG.

Note that for three-level p-Multigrid method, two iterations are sufficient for the p2 and p1 levels and 20 iterations are performed on the p0 level during every V cycle. For the two-level p-Multigrid method, a smaller V-cycle is constructed using one relaxation iteration at the p2 level, 2 relaxation iterations at the p1 level and two more smoothing iterations at the p2 level to remove the prolongation noise. The three level scheme requires lesser cycles than the two level scheme. In addition, it also converges faster than the two level scheme. These are evident from figures 6.17(a) and 6.17(b) respectively.

6.2 SV results

6.2.1 2D diffusion equation

6.2.1.1 Accuracy study

We test the accuracy of the different schemes using the 2D diffusion equation. The equation can be described as

$$\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}=0 \ , \qquad (x,y)\in[0,1]^2 . \tag{6.4}$$

$$u_{analytical}=\sin(x)e^{y} . \tag{6.5}$$

The steady state analytical solutions are specified at all the boundaries. In tables VIII, IX and X, we present the SV averaged L1 and Linf errors for the 2nd, 3rd and 4th orders respectively using the LDG, penalty and BR2 formulations. It can be seen that the 2nd and the

4th order schemes reach their respective orders asymptotically for all the schemes. We do experience a drop in order for the 3rd order (odd ordered) scheme. This is under investigation. It can be observed that the LDG method produces the maximum errors in general. Regular grids were used for the study (figure 6.18). The solution obtained using a 4th order stencil on a 40x40x2 grid is shown in figure 6.19.

Table VIII. Accuracy of 2nd order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	1.72e-03	-	5.99e-03	-
20x20x2	4.40e-04	1.97	1.52e-03	1.98
40x40x2	1.12e-04	1.97	3.85e-04	1.98
80x80x2	2.83e-05	1.99	9.71e-05	1.99
Grid	L1 error (LDG)	L1 order (LDG)	Linf error (LDG)	Linf order (LDG)
10x10x2	4.95e-03	-	1.41e-02	-
20x20x2	1.26e-03	1.98	3.72e-03	1.95
40x40x2	3.48e-04	1.90	9.78e-04	1.95
80x80x2	1.28e-04	1.65	2.63e-04	1.93
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	2.28e-03		1.20e-02	
20x20x2	4.90e-04	2.22	3.19e-03	1.91
40x40x2	1.12e-04	2.13	8.19e-04	1.96
80x80x2	2.65e-05	2.07	2.07e-04	1.98

Table IX. Accuracy of 3rd order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	8.67e-05	-	3.19e-04	-
20x20x2	1.43e-05	2.6	4.66e-05	2.78

40x40x2	2.66e-06	2.43	6.39e-06	2.86
80x80x2	5.53e-07	2.27	1.23e-06	2.38
Grid	L1 error (LDG)	L1 order (LDG)	Linf error (LDG)	Linf order (LDG)
10x10x2	1.27e-04	-	2.81e-04	-
20x20x2	2.43e-05	2.38	5.41e-05	2.37
40x40x2	5.44e-06	2.16	1.27e-05	2.09
80x80x2	1.23e-06	2.10	2.96e-06	2.07
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	9.49e-05	-	2.68e-04	-
20x20x2	1.31e-05	2.85	3.80e-05	2.82
40x40x2	2.14e-06	2.61	5.21e-06	2.88
80x80x2	3.90e-07	2.46	9.71e-07	2.42

Table X. Accuracy of 4th order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	3.69e-06	-	9.56e-06	-
20x20x2	2.35e-07	3.97	6.01e-07	3.99
40x40x2	1.49e-08	3.98	3.74e-08	4.00
Grid	L1 error (LDG)	L1 order (LDG)	Linf error (LDG)	Linf order (LDG)
10x10x2	4.30e-06	-	9.74e-06	-
20x20x2	2.97e-07	3.8	6.74e-07	3.8
40x40x2	2.06e-08	3.9	4.43e-08	3.9
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	4.17e-06	-	9.99e-06	-
20x20x2	2.76e-07	3.91	6.68e-07	3.90
40x40x2	1.78e-08	3.95	4.28e-08	3.95

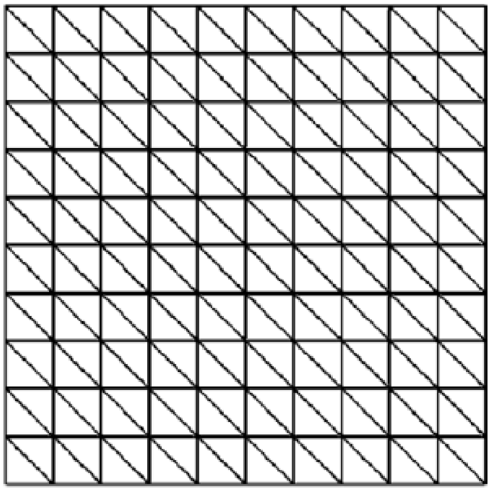


Figure 6.18. regular grid(10x10x2) used for validation study

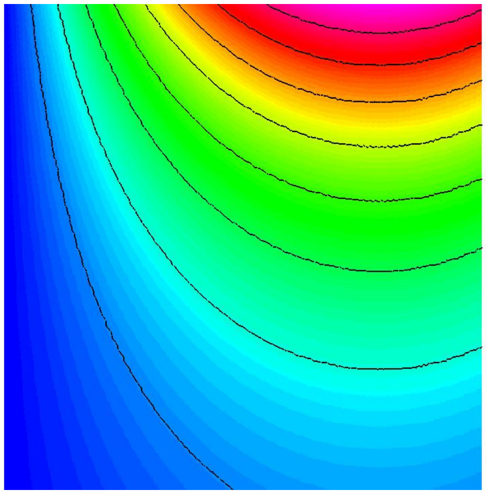
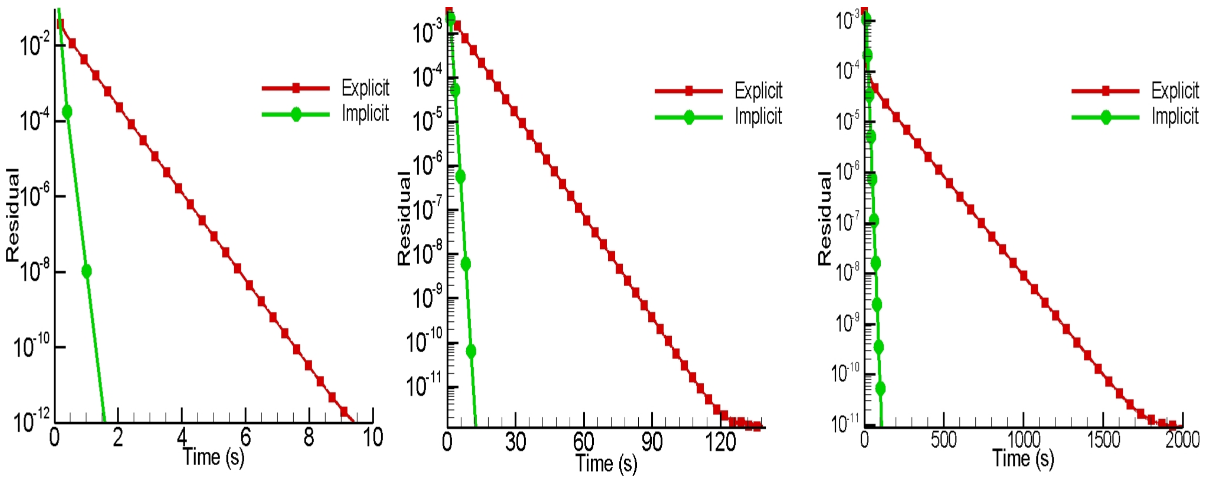


Figure 6.19. 4th order solution contours of the equation 5.1 using a 40x40x2 grid

6.2.1.2 Convergence study 1.

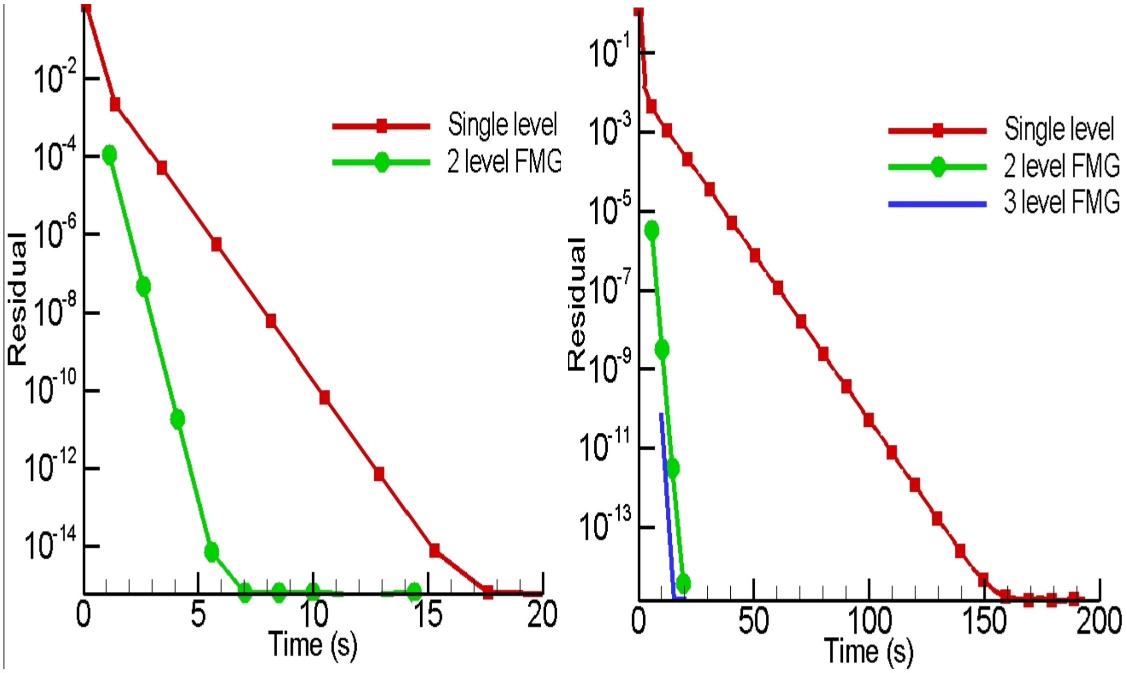
In this section, we perform numerical experiments to study the effect of the implicit LU-SGS smoother and the p-multigrid algorithm. The comparison of the convergence speeds between the implicit and the explicit Runge Kutta schemes using the penalty method is shown in Figure 6.20.



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

Figure 6.20. Convergence history using explicit R-K and implicit LU-SGS schemes for the 2D diffusion equation using the penalty scheme. Case a: 2nd order; Case b: 3rd order; Case c: 4th order

It can be seen that the speedup increases with increasing the order of the interpolation polynomial. We attain a speedup of nearly 20 for 4th order simulations. Thus it is more practical to use implicit algorithms for higher order methods. Figures 6.21(A) and 6.21(B) show the effect of the p-multigrid on the 3rd and 4th orders respectively. It can be seen that the multigrid results in a speed-up of 3 for the 3rd order and nearly 7 for the 4th order. Thus a combination of implicit LU-SGS and p-multigrid results in a speedup, which is more than 2 orders in magnitude!



(a) **(b)**

Figure 6.21. Convergence history using single level and multilevel FMG LU-SGS schemes for the 2D diffusion equation using the penalty scheme. Case a: 3rd order; Case b: 4th order

6.1.2.3 Convergence study 2.

In this section, we perform numerical experiments to study the convergence properties of the different flux formulations. Figure 6.22 shows the convergence history obtained using the three formulations for a 2nd order scheme. It can be seen that the LDG converges slower than the other 2 formulations. This phenomenon can be explained using the information provided by figure 5.3. It can be seen that LDG has a positive and higher error than the other schemes. This means that the numerical eigen value (physical mode) is higher than the other schemes. So it takes more time to decay and attain steady state and hence the slow convergence.

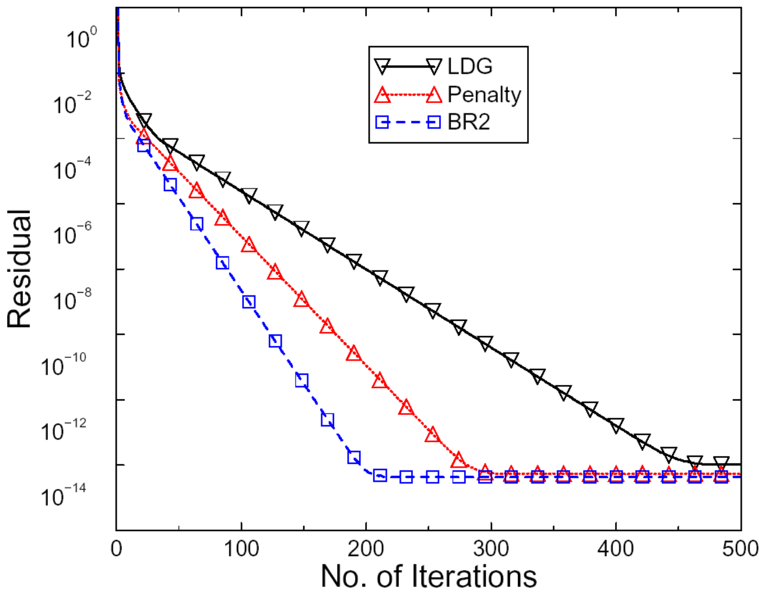


Figure 6.22 Convergence history using the three formulations for the 2D diffusion equation using a 2nd order scheme

6.2.2 2D Navier Stokes equations

6.2.2.1 The Couette flow problem

The Couette flow is an analytical solution of the Navier-Stokes equations, and was selected to study the accuracy for the 2D Navier-Stokes solver. This problem models the viscous flow between a stationary, fixed temperature, (at T_0) bottom plate, and a moving, fixed temperature (at T_1) top plate at speed of *U*. The distance between the two plates is *H*. A schematic of the above can be seen in figure 6.23.

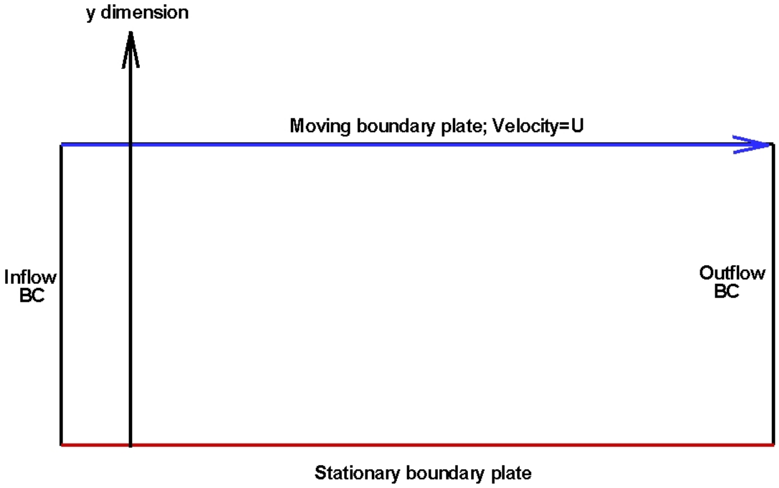


Figure 6.23. Computaional domain for the Couette Flow problem.

It has an exact solution under the simplification that the viscosity coefficient μ is a constant and the speed is low enough to ensure nearly incompressible effects. The steady analytic solution is

$$u = \frac{U}{H} y \text{, v=0,} \tag{6.6}$$

$$T = T_0 + \frac{y}{H}(T_1 - T_0) + \frac{\mu U^2}{2k}\frac{y}{H}(1 - \frac{y}{H}), \tag{6.7}$$

$$\text{p=constant,} \qquad \rho = \frac{p}{RT}, \tag{6.8}$$

where k is the thermal conductivity and R is the gas constant.

The accuracy of the three viscous flux formulations was tested. The L1 and Linf temperature errors are presented in tables XI and XII for 3rd and 4th orders respectively. It can be seen that the obtained orders are very close to the ones expected.

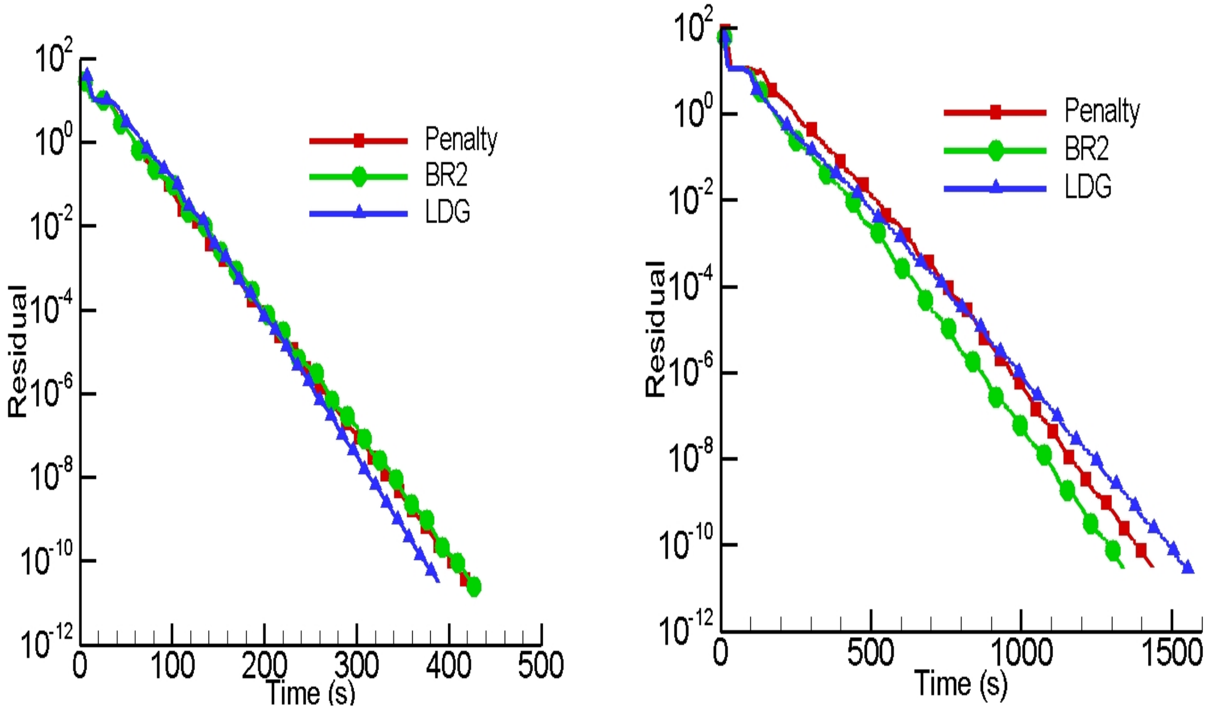
Table XI. Temperature error for the Couette flow problem (3rd order)

Grid	Linf error (Penalty)	Linf order (Penalty)	L1 error (Penalty)	L1 order (Penalty)
10x10x2	3.5465e-07	-	1.0437e-07	-
20x20x2	3.7694e-08	3.23	1.2785e-08	3.03
40x40x2	4.8249e-09	2.97	1.5994e-09	3.00
Grid	Linf error (LDG)	Linf order (LDG)	L1 error (LDG)	L1 order (LDG)
10x10x2	6.3494e-07	-	1.3324e-07	-
20x20x2	4.5975e-08	3.79	1.4709e-08	3.18
40x40x2	6.7767e-09	2.76	1.9260e-09	2.93
Grid	Linf error (BR2)	Linf order (BR2)	L1 error (BR2)	L1 order (BR2)
10x10x2	5.3522e-07	-	1.2798e-07	-
20x20x2	6.3011e-08	3.08	1.1615e-08	3.46
40x40x2	6.7305e-09	3.22	1.2038e-09	3.27

Table XII. Temperature error for the Couette flow problem (4th order)

Grid	Linf error (Penalty)	Linf order (Penalty)	L1 error (Penalty)	L1 order (Penalty)
10x10x2	2.7280e-08	-	1.2473e-08	-
20x20x2	1.7749e-09	3.94	7.6483e-10	4.03
Grid	Linf error (LDG)	Linf order (LDG)	L1 error (LDG)	L1 order (LDG)
10x10x2	2.6458e-08	-	7.7845e-09	-
20x20x2	1.8884e-09	3.81	4.4427e-10	4.13
Grid	Linf error (BR2)	Linf order (BR2)	L1 error (BR2)	L1 order (BR2)
10x10x2	2.2279e-08	-	1.2047e-08	-
20x20x2	1.5136e-09	3.88	7.9952e-10	3.92

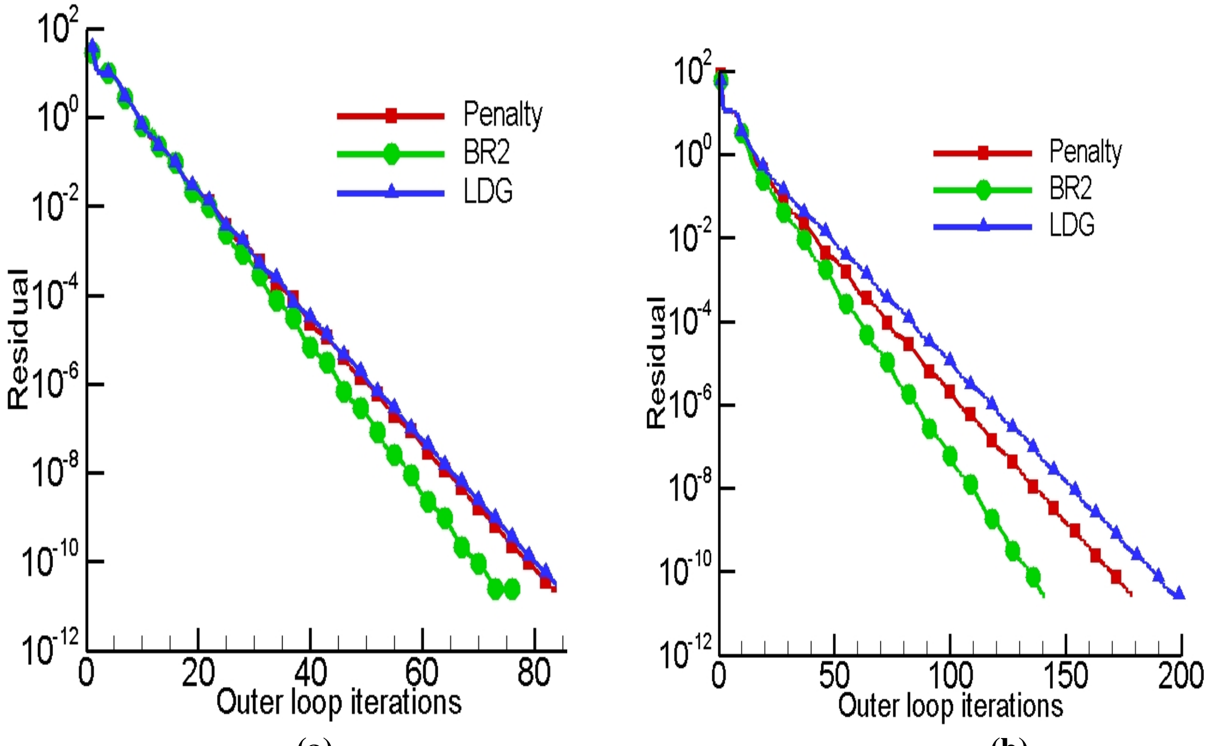
We now compare the convergence (using implicit LU-SGS method) properties using the three viscous flux formulations for 3rd and 4th order simulations. The residual history as a function of time is shown in figure 6.24 for the 3rd and 4th order simulations. The 20x20x2 grid was used for this purpose.



(a) **(b)**

Figure 6.24 Convergence history as a function of time for the Couette flow problem. Case (a): 3rd order; Case (b): 4th order

The residual history as a function of outer iterations is shown in figure 6.25. It can be seen that the BR2 scheme converges in lesser number of iterations. However, BR2 involves more computations per iteration (explained in section 5.2.3). Hence, the total time consumed by BR2 is nearly the same as the other two schemes.



(a) **(b)**

Figure 6.25 Convergence history as a function of iterations for the Couette flow problem. A: 3rd order; B: 4th order

6.2.2.2 Flow over an airfoil

In this section, we simulate flow over a naca0012 airfoil. The flow was subsonic at Mach 0.5, at Reynolds number of 5000 and had zero angle of attack. This has been a widely used validation case for viscous flow solvers and was used in [47,54,67]. The computational grid for this case is the one used for the SD simulation (figure 6.13). An important trait of this test case is the formation of a small recirculation bubble that extends in the near wake region of the airfoil. This is caused due to the separation of the flow near the trailing edge.

The current simulations were performed using the three viscous formulations for 2nd, 3rd and 4th orders. These are compared with the ones performed in [54 and 67]. These comparisons are summarized in table XIII. It is reasonable to assume that the 5th order SD results of Sun et al [67] are the most accurate of all the ones listed in table XIII. This table compares the separation point, drag coefficient due to pressure (CD_p) and the drag coefficient due to viscous stresses (CD_f). It can be seen that the 2nd order second order simulations over predict the separation point. The 3rd and 4th order results are very close the 5th order results of Sun et al [67].

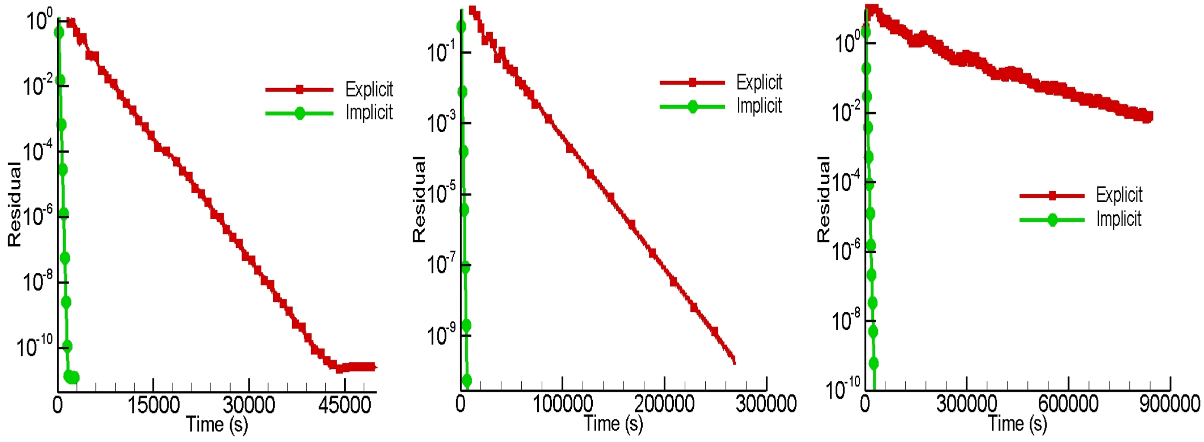
It is also clear that the penalty and the BR2 schemes display higher fidelity than the LDG scheme: for instance, it can be seen that penalty and the BR2 schemes yields CD_p readings which are closer to the actual value (than the under predicting LDG readings). The under-prediction for LDG's CD_p is observed for 2nd, 3rd and 4th orders.

Table XIII. Comparison of pressure and viscous drag coefficients and location of separation point between current simulations and other existing simulations in the literature

Method	NDOFs	Separation Point	CD_p	CD_f
2nd order SV(Pen)	10368	94.3%	2.0498e-2	3.5309e-2
2nd order SV(BR2)	10368	94.5%	2.0578e-2	3.4994e-2
2nd order SV(LDG)	10368	94.2%	2.0192e-2	3.5664e-02
3rd order SV(Pen)	20736	81.6%	2.2081e-2	3.2117e-2
3rd order SV(BR2)	20736	81.4%	2.2043e-2	3.2035e-2
3rd order SV(LDG)	20736	81.7%	2.2016e-2	3.2193e-02
4th order SV(Pen)	34560	81.3%	2.2270e-2	3.2345e-2
4th order SV(BR2)	34560	81.3%	2.2234e-2	3.2325e-2
4th order SV(LDG)	34560	81.2%	2.2219e-2	3.2354e-2
Cell Centered FV[54]	16384	80.9%	2.2560e-2	3.3010e-2
Cell Centered FV [54]	65536	81.4%	2.2350e-2	3.2990e-2

5th order SD [67]	43200	81.4%	2.2250e-2	3.2510e-2

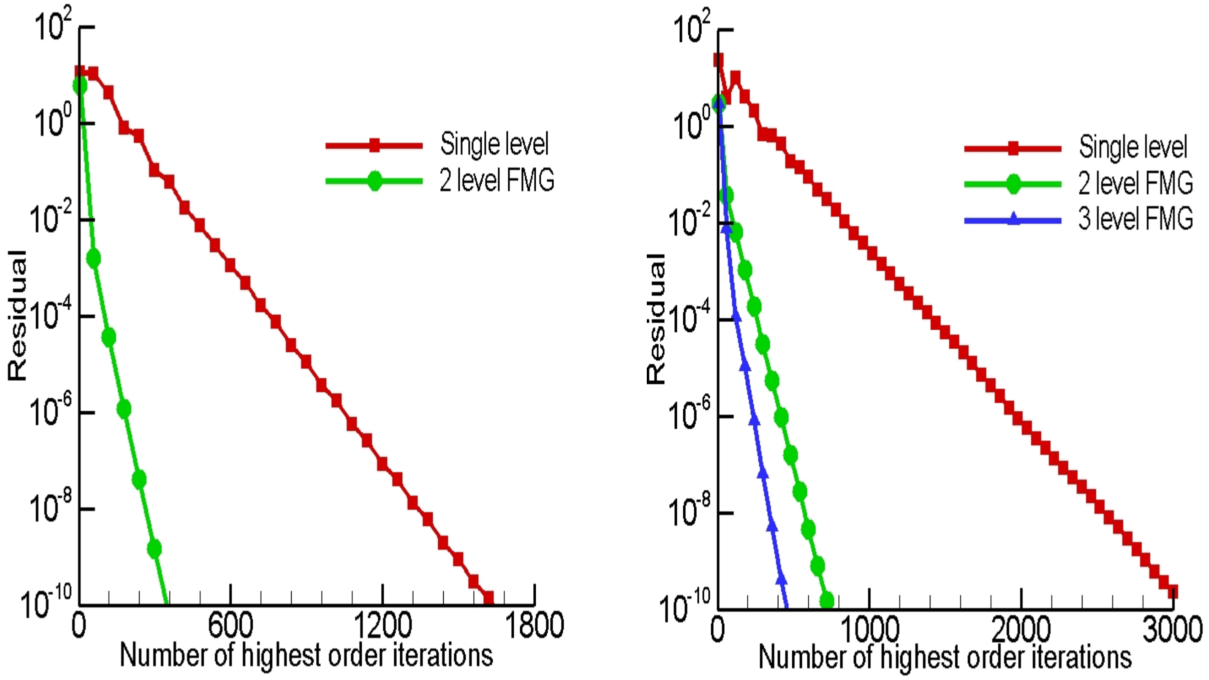
Figure 6.26 shows the convergence histories for 2nd, 3rd and 4th orders using the penalty approach. It can be seen that the speedup at higher orders is massive (more than 2 orders), especially for 4th order. The 3rd and 4th order explicit schemes limit the CFL to a few hundredths. In contrast, the implicit schemes have a CFL limit of more than a hundred. This is the main reason for obtaining the above-mentioned speedups.



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

Figure 6.26 Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over naca0012 airfoil using the penalty scheme. Case A: 2ndorder; Case B: 3rd order; Case C: 4th order

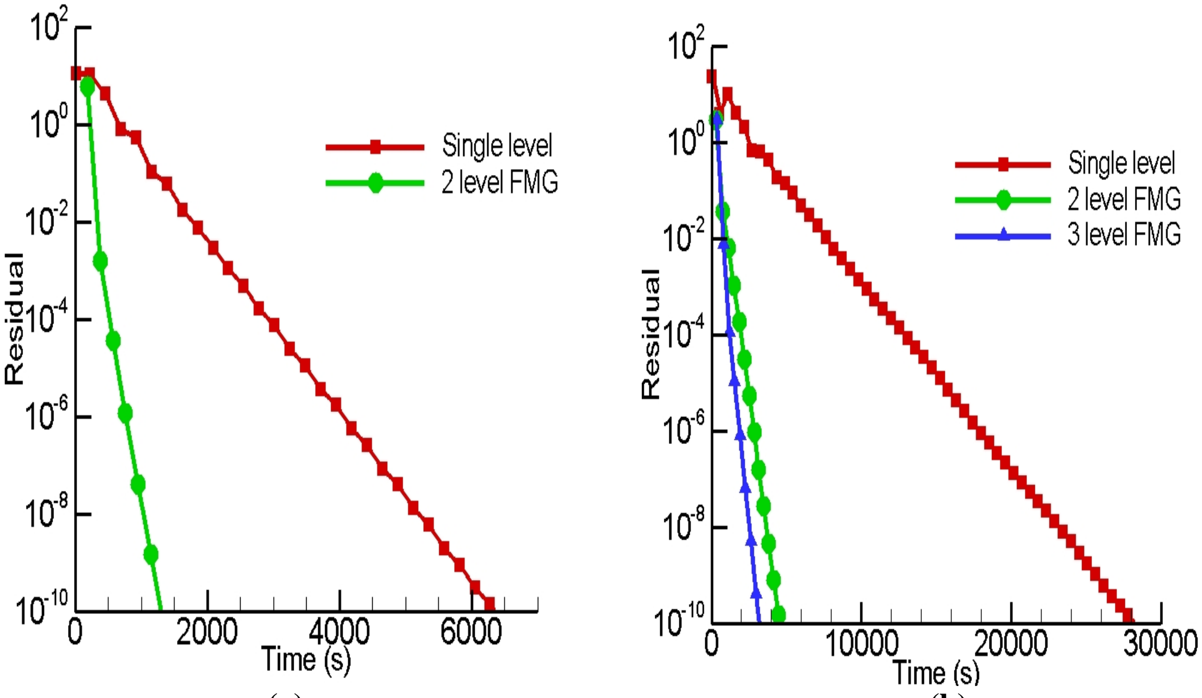
Figure 6.27 shows the convergence history as a function of highest order iterations for the single level and the multigrid simulations. It can be seen that the multigrid reduces the total number of required sweeps.



(a) **(b)**

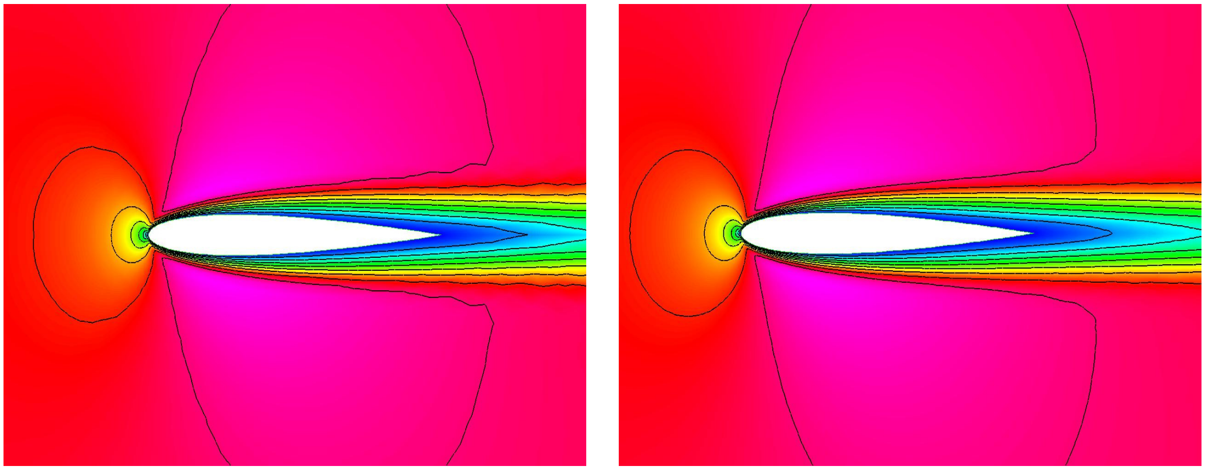
Figure 6.27 Convergence history as a function of finest order iterations (LU-SGS scheme) using single level and FMG schemes for the flow over naca0012 airfoil (penalty method). Case (a): 3rd order; Case (b): 4th order

Figure 6.28 shows the effect of p-multigrid for the 3rd and 4th order simulations. The lower level operations were carried out using a larger time step. This was done to eliminate the low frequency errors at a faster rate. Fig. 6.29 shows the Mach contours computed with linear and quadratic SVs. The solution gets smoother and smoother with the increasing of the order of the polynomial reconstruction. The wake region looks more refined and continuous for the 3rd order case. The difference between the 3rd and the 4th order plots were negligible.



(a) **(b)**

Figure 6.28. Convergence history using single level and FMG schemes for the flow over naca0012 airfoil. Case (a): 3rd order; Case (b): 4th order

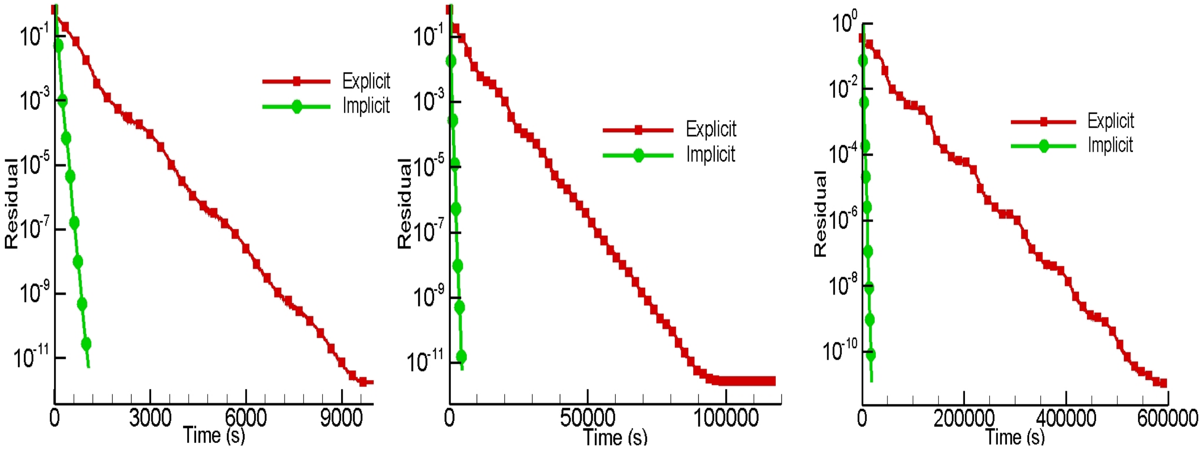


(a) **(b)**

Figure 6.29. Mach contours around the NACA0012 airfoil at zero degree of attack, Re=5000, M=0.5. Case (a): 2ndorder; Case (b): 3rd order

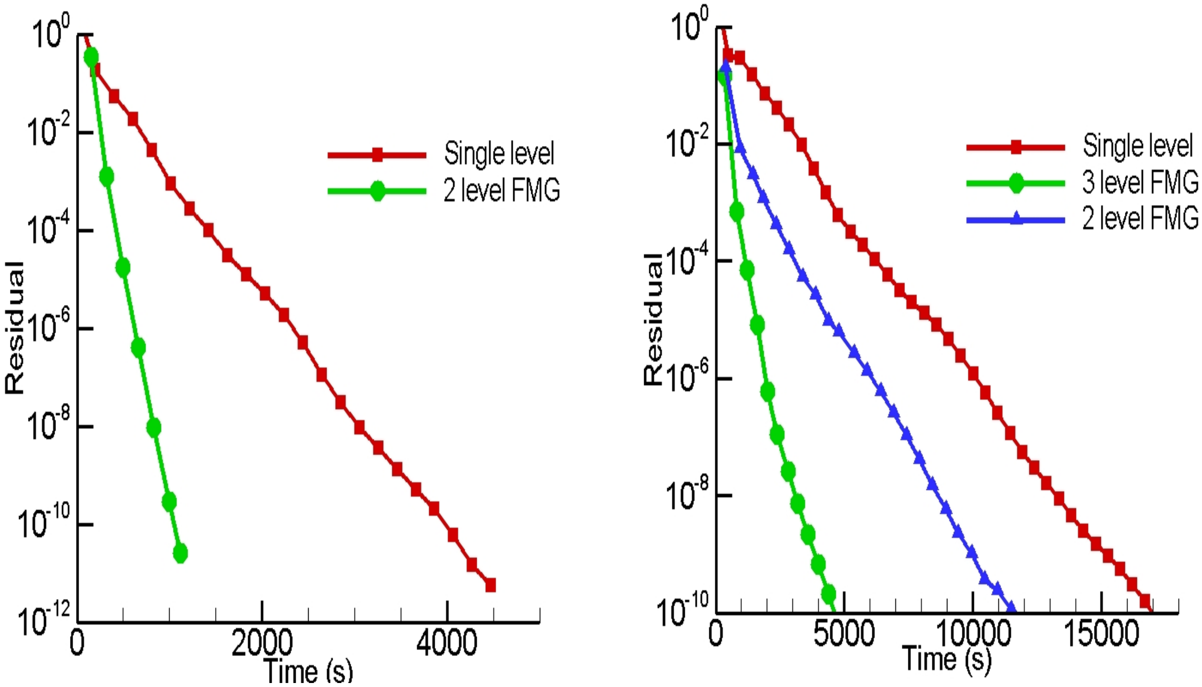
6.2.2.3 Laminar flow over a flatplate

In this section, we simulate laminar flow over a flat plate. The flow was transonic at Mach 0.5. Figure 6.30 shows the convergence histories for 2nd, 3rd and 4th orders using the penalty scheme.



(a) (b) (c)

Figure 6.30 Convergence history using explicit R-K and implicit LU-SGS schemes for the laminar flow over a flat plate using the penalty scheme. Case (a): 2ndorder; Case (b): 3rd order; Case (c): 4th order



(a) (b)

Figure 6.31. Convergence history using single level and FMG schemes for the laminar flow over a flat plate using the penalty scheme. Case (a): 3rd order; Case (b): 4th order

Figure 6.31 shows the effect of p-multigrid for the 3rd and 4th order simulations. These convergence plots look qualitatively similar to those obtained for the flow over naca0012 airfoil. It must be noted that the speedups obtained for this simulation are slightly lower than those obtained for the airfoil case. This is because of the surface grid: airfoil utilizes a curved boundary for the 3rd and 4th orders. This makes the system stiffer and the effect of implicit/multigrid schemes result in enormous speedups. All of the above was published in [36].

We also compared the multigrid performance for the three viscous discretizations. Figure 6.32(A) shows the convergence histories as a function of time for a 3rd order multigrid system. As expected, the LDG method performs badly. Due to the extra computations (i.e. computing the lifting functions), BR2 is slightly slower than the penalty scheme. Figure 6.32(B) shows the convergence history as a function of highest order iterations. As expected, the BR2 scheme converges in the least number of iterations.

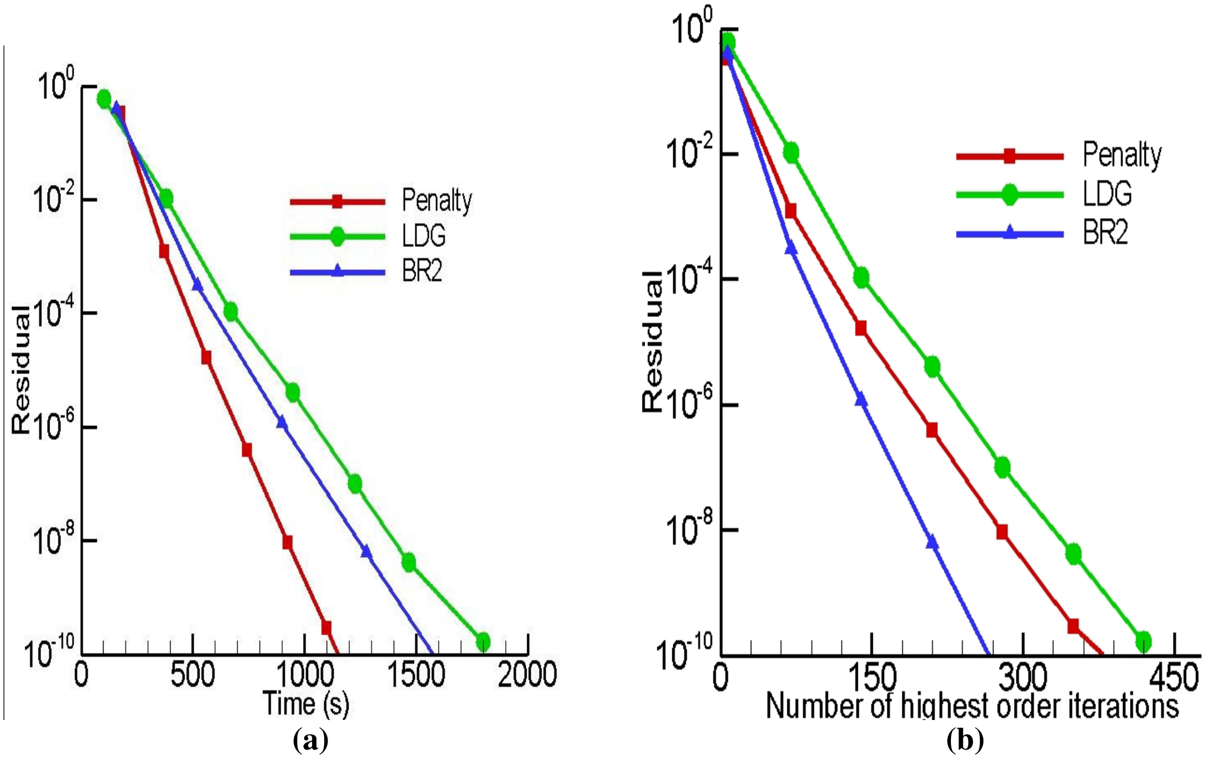
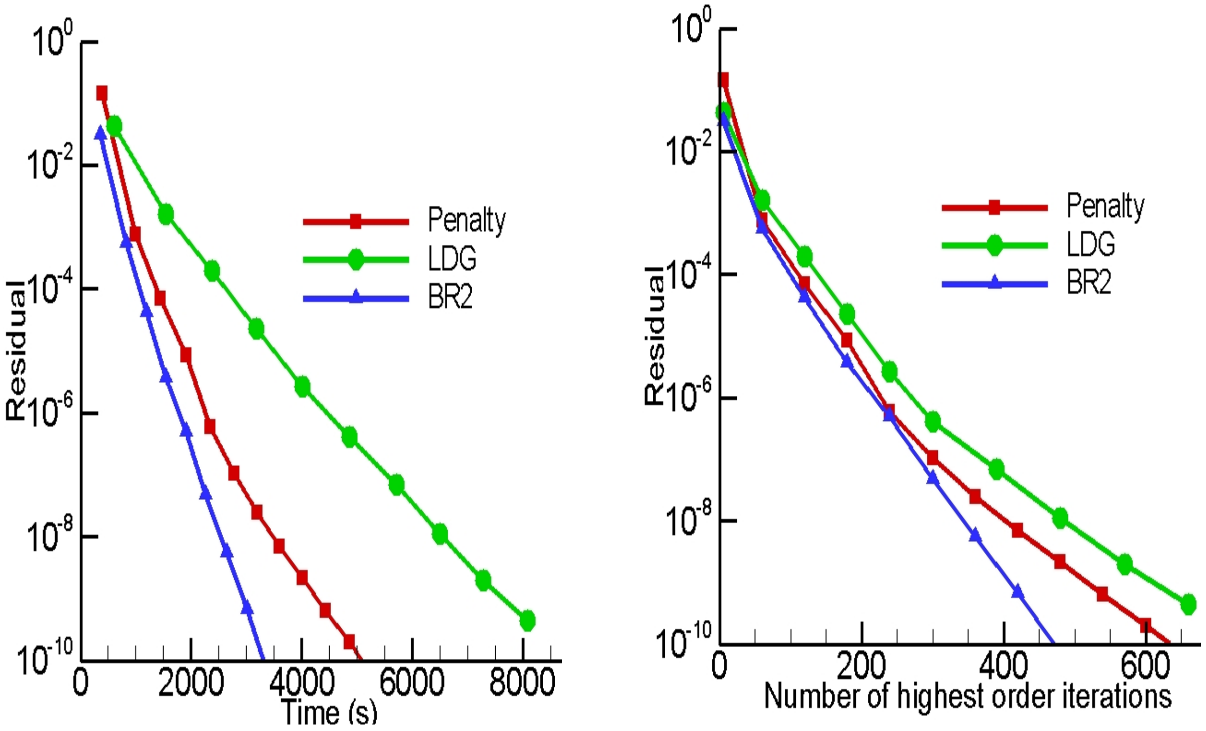


Figure 6.32. Convergence history for the 3rd order multigrid system FMG schemes for the laminar flow over a flat plate. Case (a): Convergence as a function of time; Case (b): Convergence as a function of highest order iterations

Figure 6.33(A) shows the convergence history as a function of time for the 2 level 4th order multigrid system. The LDG method was found to be extremely unstable and we had to use a much smaller time step to ensure stability. In addition, we also had to update the values of the gradients on the faces (CV faces in the SV boundary) while obtaining the residuals for every SV. In other words, the face gradients coming from the neighboring SVs have to be updated continuously. This makes the system more implicit and hence more stable. The BR2 method takes the least time to attain convergence. Figure 6.33(B) shows the convergence history as a function of highest order iterations. As expected, once again BR2 takes the least amount of iterations and LDG the most. Thus in spite of the additional computing cost (lifting function computation), BR2 outperforms the penalty and the LDG scheme.



(a) **(b)**

Figure 6.33. Convergence history for the 4th order multigrid system FMG schemes for the laminar flow over a flat plate. Case (a): Convergence as a function of time; Case (b): Convergence as a function of highest order iterations

6.2.2.4 Turbulent flow over a flatplate

To simulate flow turbulence, a RANS Spalart-Allmaras (S-A) model approach was employed. The S-A one-equation model [66] solves a single partial differential equation for a variable $\tilde{\nu}$ which is related to the turbulent viscosity. The differential equation is derived by using empiricism and arguments of dimensional analysis, Galilean invariance and selected dependence on the molecular viscosity. The model includes a wall destruction term that reduces the turbulent viscosity in the log layer and laminar sublayer. The equation can be written in the following form

$$\frac{D\tilde{\nu}}{Dt} = c_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w \left[\frac{\tilde{\nu}}{d}\right]^2 + \frac{1}{\sigma}\left[\nabla \bullet ((\nu+\tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2\right]. \qquad (6.9)$$

The turbulent viscosity is determined via,

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi \equiv \frac{\tilde{\nu}}{\nu}, \qquad (6.10)$$

where ν is the molecular viscosity. Using S to denote the magnitude of the vorticity, the modified vorticity is defined as

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \qquad (6.11)$$

where *d* is the distance to the closest wall. The wall destruction function is defined as

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right]^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}. \qquad (6.12)$$

The closure coefficients are given by:

$c_{b1} = 0.1355$

$\sigma = 2/3$

$c_{b2} = 0.622$

$k = 0.41$

$c_{w1} = \dfrac{c_{b1}}{k * k} + \dfrac{(1+c_{b2})}{\sigma}$

$c_{w2} = 0.3$

$c_{w3} = 2$

$c_{v1} = 7.1$

The following flux formulations were used

1. The inviscid fluxes were handled using upwinding. This term is generally not stiff and one could also use an approximate Riemann flux.
2. One of the most important terms in equation 5.26 is the diffusion term $\frac{1}{\sigma}\left[\nabla \bullet ((\nu + \tilde{\nu})\nabla \tilde{\nu})\right]$. A penalty formulation was used to obtain the above at the SV boundaries

The equation for $\tilde{\nu}$ also involves the computation of a source term. The production at the cell center of each control volume was obtained. The actual production was obtained by reconstructing the cell center valued production terms obtained from the CVs in each SV.

In this section, we simulate turbulent flow over a flat plate. The flow was transonic at Mach 0.5. The Reynolds number was 5 million. The Spalart Allmaras (SA) model was used to close the Reynolds stresses.

Simulating the turbulent flow over a flat plate can be regarded as the first step toward simulating turbulent flows. In other words, it is a starting location for computing complicated flows like rotor craft flows, aeroacoustics signal propagations etc. Figure 6.34 shows the variation of the non dimensional distance to the wall versus the non dimensional velocity uplus. This has been a classical problem and the well known log-law behavior is well documented [61]. The indirect turbulence model assumes an infinite Reynolds number and hence the log-law behavior continues till infinity. However the case in consideration has a finite Reynolds number and hence flattens out after some distance. It can be seen that the 3rd and the 4th order simulations display much higher fidelity than the 2nd order simulations. This may indicate that higher order methods might be the best alternative to experiments (w.r.t obtaining very accurate results).

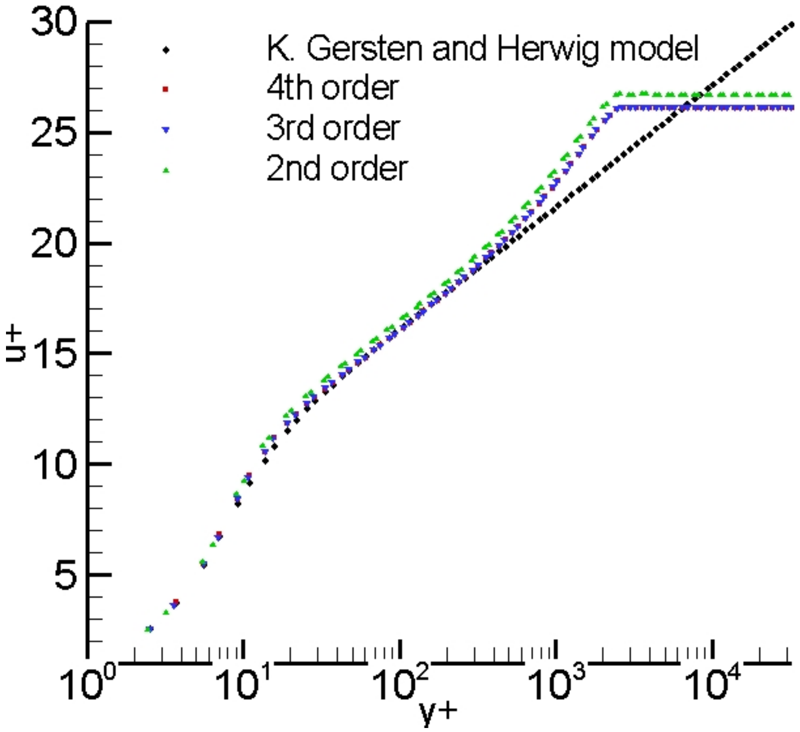


Figure 6.34. Variation of yplus versus uplus for turbulent flow over a flat plate. The simulations are compared with the indirect turbulence model.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

We have developed a p-Multigrid spectral difference solver for 2D Euler equations. The SD method is relaxed with an implicit lower-upper symmetric Gauss-Seidel approach. It was found that the simplified implicit scheme is very stable and produces a speedup of one or two orders magnitude compared to its explicit counterparts for the scalar wave equations as well as the Euler equations. The computational speed is further accelerated by a nonlinear p-multigrid approach in the context of Full Approximation Scheme. The combination of the implicit LU-SGS relaxation scheme with three-level p-Multigrid method achieved very good stability and speedup for both 2D wave cases and 2D Euler equation cases. In addition, the p-Multigrid scheme preserved the expected third order accuracy for the supersonic vortex flow. The p-multigrid method with implicit smoothers on all three p-levels achieves a speedup of around 4 over the single level implicit scheme. Explicit Runge-Kutta smoothers are also studied for the p-Multigrid method. They are not as effective as the implicit LU-SGS method. However, they can be used at the highest p-level for 3D problems to circumvent the problems associated with memory. Our calculations in two dimensions show that the approaches using explicit smoother at the high p-levels and implicit smoother at the low p-levels are stable and achieve considerable speedup. A speedup factor around 8 can be achieved using explicit smoothers for all three p-levels. In addition, the simulations using the V cycles were always faster than the simulations using the W cycles. Extending the SD p-Multigrid method with implicit LU-SGS smoothers to 2D compressible Navier-Stokes equations is one of the areas for future research.

In the present study, we suggested a new penalty approach and implemented the BR2 method for computing viscous fluxes for the SV method. Fourier analysis was done for both the new formulations and the previously used LDG scheme. Based on the Fourier analysis, the following conclusions were obtained

1. LDG has a severe time step limitation (based on stability). BR2 can handle a very high time step and the penalty scheme is in between the above two methods.
2. BR2 can converge in less iteration (compared to penalty and LDG). However, the time consumed by BR2 scheme is comparable to the penalty scheme as additional computations are required.
3. There exists some range of wave numbers where the multigrid is unstable. This range is very large for the 4th order multigrid system.
4. The BR2 and the penalty scheme are best suited for a 3rd order multigrid. Even though the LDG has the least instability for a 4th order multigrid system, it requires a very small time step and is very inefficient.

The accuracy of the three methods was tested by performing an accuracy study using the 2D Laplace equation. We also incorporated the implicit LU-SGS scheme into the SV method. This resulted in extraordinary speedups. A p-multigrid was also implemented for the diffusion and the NS equations. The implicit solver and the p-multigrid were successfully coupled in order to enable efficient simulation. A combination of the above gave speedups of nearly 3 orders of magnitude. Future research includes one or more of the following areas

1. A complete study of stability analysis for the SD method is needed. Currently SD has a mild instability for hyperbolic conservation laws. No analysis has been done for the diffusion problem for the SD method. It would be interesting to check

whether the three viscous flux formulations show the traits they displayed for the SV scheme.

2. Investigate the possibility of “All curved faces” approach i.e. all the faces in the computational domain are curved faces. This might be the only way to tackle the extremely skewed triangles in simulations, which have non-straight walls (which exist in most real world applications).
3. Implement h-p multigrid.

BIBLIOGRAPHY

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.

[2] M. Aftosmis, D. Gaitonde and T. S. Tavares, Behavior of linear reconstruction techniques on unstructured meshes, AIAA Journal, 33 (1995), pp 2038-2049.

[3] Allmaras S. Analysis of semi-implicit preconditioners for multigrid solution of the 2-D Navier Stokes equations. AIAA Paper 95–1651, AIAA 12th computational fluid dynamics conference, San Diego, CA, June 1995.

[4] Alonso JJ, Martinelli L, Jameson A. Multigrid unsteady Navier Stokes calculations with aeroelastic applications. AIAA Paper 95–0048, AIAA 33rd aerospace sciences meeting, Reno, NV, January 1995.

[5] W. K. Anderson, R. D. Rausch, and D. L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. J. Comput. Phys., 128:391–408, 1996.

[6] D. N. Arnold. An interior penalty finite element method with discontinuous elements. SIAM J. Numer. Anal., 19:742–760, 1982.

[7] Barth TJ, Frederickson PO. , High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.

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

[9] F. Bassi and S. Rebay. 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.

[10] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, Discontinuous Galerkin Methods: Theory, Computation and Applications, pages 197–208. Springer, Berlin, 2000.

[11] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d Euler equations. J. Comput. Phys., 138:251–285, 1997.

[12] F. Bassi, S. Rebay, Numerical solution of the Euler equations with a multiorder discontinuous finite element method, in: Proceedings of the Second International Conference on Computational Fluid Dynamics, Sydney, Australia, 15–19 July 2002.

[13] C. E. Baumann. An hp-Adaptive Discontinuous Finite Element Method for Computational Fluid Dynamics. PhD dissertation, University of Texas at Austin, December 1997.

[14] Braaten ME, Connell SD. Three dimensional unstructured adaptive multigrid scheme for the Navier–Stokes equations. AIAA J 1996;34(2):281–90.

[15] F. Brezzi; G. Manzini; D. Marini; P. Pietra; and A. Russo. Discontinuous Galerkin approximations for elliptic problems. Numer. Methods for Partial Differential Eqns., 16:365–378, 2000.

[16] Chen RF, Wang ZJ. Fast block lower-upper symmetric Gauss Seidel scheme for arbitrary grids. AIAA J 2000; 38(12): 2238–45.

[17] Choi YH, Merkle CL. The application of preconditioning in viscous flows. J Comput Phys 1993;105(2):207–23.

[18] B. Cockburn, S. Hou, and C.-W. Shu,TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, *Math. Comput.* **54**, 545 (1990).

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

[20] B. Cockburn and C. W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. of Sci. Comput., 16(3):173–261, September 2001.

[21] B. Cockburn and C. W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal., 35(6):2440–2463, December 1998.

[22] B. Cockburn and C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comput.* **52**, 411 (1989).

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

[24] V. Dolejsi, M. Feistauer, A semi-implicit discontinuous Galerkin Finite element method for the numerical solution of inviscid compressible flow, J. Comput. Phys. 198 (2004) 727–746.

[25] Nishikawa H. Higher order discretization of diffusion terms in residual distribution methods. In: Deconinck H,Ricchiuto M, editors. CFD—higher order discretization methods. von Karman Institute for Fluid Dynamics; 2006.

[26] K. J. Fidkowski, T. A. Oliver, J. Lu and D. L. Darmofal, p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, Journal of Computational Physics, 207 (2005) pp. 92-113.

[27] Gottlieb S. On high-order strong stability preserving Runge–Kutta and multi step time discretizations. J Sci Comput 2005;25(1/2):105–28.

[28] Gottlieb S, Shu CW. Total variation diminishing Runge–Kutta schemes. Math Comput 1998;67:73–85.

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

[30] B. T. Helenbrook and H. L. Atkins, Application of p-Multigrid to Discontinuous Galerkin formulations of the Poisson equation, AIAA Journal, 44 (2006) pp. 566-575.

[31] B. T. Helenbrook, D. J. Mavriplis and H. A. Atkins, Analysis of p-multigrid for continuous and discontinuous finite element discretizations. AIAA Paper 2003-3989, 2003.

[32] Hemker PW, Koren B. Defect correction and non-linear multi-grid for the steady Euler equations. Lecture Notes, 1988-05 von Karman Institute for Fluid Dynamics, 1988.

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

[34] Jameson A. Solution of the Euler equations by a multigrid method. Appl Math Comput 1983;13:327–56.

[35] Jameson A, Caughey DA. How many steps are required to solve the Euler equations of steady compressible flow: insearch of a fast solution algorithm. In: 15th AIAA computational fluid dynamics conference, Anaheim, CA, June 2001.

[36] Kannan, R., Sun. Y. and Wang, Z.J.," An Implicit Spectral Volume Method for the Navier Stokes Equations," AIAA Paper 2008-0783

[37] D. A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II semi-structured method, Journal of computational physics, 128 (1996), pp. 475-488.

[38] D. A. Kopriva, A staggered-grid multi-domain spectral method for the Euler and Navier-Stokes equations on unstructured grids, Journal of computational physics, 143 (1998), pp. 125-158.

[39] Liang, C., Kannan, R. and Wang, Z.J.," A p-MultiGrid Spectral Difference method for the Euler equations with implicit LU Symmetric Gauss Seidal relaxation scheme on unstructured grids," AIAA Paper 2007-4326.

[40] Liang, C., Kannan, R. and Wang, Z.J.," A p-multigrid Spectral Difference method with explicit and implicit smoothers on unstructured grids Computers and Fluids," Computers and Fluids (Accepted and in publication stage)

[41] M.-S.Liou, and C. Steffen, A New Flux Splitting Scheme, J. Comput. Phys., Vol. 107, 23-39, 1993.

[42] Y. Liu, M. Vinokur, and Z. J. Wang, Spectral difference method for unstructured grids I: Basic formulation, J. of Comput. Phys. 216 (2006), pp. 780-801.

[43] H. Luo, J. D. Baum and R. Löhner, A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, Journal of Computational Physics, 211 (2006), pp 767-783.

[44] Y. Maday, R. Munoz, Spectral element multigrid, Part 2: Theoretical justification, Tech. Rep. 88-73, ICASE, 1988.

[45] D. J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. Journal of Computational Physics, 145:141–165, 1998.

[46] Mavriplis DJ, Jameson A. Multigrid solution of the Navier–Stokes equations on triangular meshes. AIAA J 1990;28:1415–25.

[47] D. J. Mavriplis, A. Jameson, and L. Martinelli, "Multigrid solution of the Navier-Stokes equations on triangular meshes", AIAA paper 89-0120, 1989.

[48] D. J. Mavriplis and S. Pirzadeh. Large-scale parallel unstructured mesh computations for 3-D high-lift analysis. AIAA Journal of Aircraft, 36:987–998, 1999.

[49] G. May and A. Jameson, "A spectral difference method for the Euler and Navier-Stokes equations", AIAA paper No. 2006-304, 2006.

[50]Mulder WA. A high-resolution Euler solver based on multigrid, semi-coarsening, and defect correction. J Comput Phys 1992;100:91–104.

[51] C. R. Nastase and D. J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, Journal of Computational Physics, 213 (2006) pp. 330-357.

[52] R. H. Ni, A multiple grid scheme for solving the Euler equations, AIAA Journal, 20 (1982), pp.1565-1571.

[53] Niles A. Pierce and Michael B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. Journal of Computational Physics, 136:425–445, 1997.

[54] R. Radespiel and R.C. Swanson, "An investigation of cell-centered and cell vertex multigrid schemes for Navier-Stokes equations", AIAA Paper No. 89-0543, 1989.

[55] P. Rasetarinera, M.Y. Hussaini, An efficient implicit discontinuous spectral Galerkin method, Journal of Computational Physics 172 (2001) 718–738.

[56] P. L. Roe. Approximate Riemann solvers, parametric vectors, and difference schemes. Journal of Computational Physics, 43:357–372, 1981.

[57] P. L. Roe. Characteristic-based schemes for the Euler equations. Ann. Rev. Fluid Mech., 18:337–65, 1986.

[58] E.M. Ronquist, A.T. Patera, Spectral element multigrid, I. Formulation and numerical results, Journal of Scientific Computing 2 (4) (1987) 389–406.

[59] V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1, (1961), pp. 261-279.

[60] Saad Y, Schultz MH. GMRES: A genearlized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986;7:865.

[61] Schlichting, H., 1979, *Boundary Layer Theory*, seventh edition, McGraw-Hill, New York.

[62] Sharov D, Nakahashi K. Low speed preconditioning and LUSGS scheme for 3D viscous .low computations on unstructured grids. AIAA Paper No. 98–0614, January 1998.

[63] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77:439–71.

[64] Shu C-W. Total-variation-diminishing time discretizations. SIAM J Sci Stat Comput 1988;9:1073–84. Navier–Stokes equations on triangular meshes. AIAA J

[65] Spiteri RJ, Ruuth SJ. A new class of optimal high-order strong-stability preserving time discretization methods.

[66] Spalart, P.R., Allmaras, S.R., 1994. “A one-equation turbulence model for aerodynamic flows,” La Recherche Aerospatiale 1, 5–21.

[67] Y. Sun and Z.J. Wang,”Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method”, Communications in Comput. Phys (submitted)

[68] Y. Sun, Z.J. Wang and Y. Liu, “Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow”, J. Comput. Phys.2006; 215:41-58.

[69] E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations, Journal of Computational Physics, 72 (1987), pp. 277-298.

[70] Van Den Abeele, K., Broeckhoven, T., and Lacor, C., “Dispersion and Dissipation Properties of the 1d Spectral Volume Method and Application to a P-Multigrid Algorithm,” J. Comp. Phys., Vol. in press., 2007.

[71] B. Van Leer. Flux-vector splitting for the Euler equations. Technical Report 81-11, ICASE, 1981.

[72] B. Van Leer. Upwind-difference methods for aerodynamic problems governed by the Euler equations. Lectures in Applied Mathematics, 22, 1985.

[73] Van Leer B, Lee WT, Roe PL. Characteristic time-stepping or local preconditioning of the Euler equations. AIAA Paper No. 91–1552; 1991.

[74] B. Van Leer, J. Thomas, P. Roe, and R. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations. AIAA Paper 87-1104, 1987.

[75] Vassberg JC. A fast, implicit unstructured-mesh Euler method. AIAA Paper No. 92–2693, 1992.

[76] Z. J. Wang. Spectral volume and difference methods for unstructured grids. VKI lecture series, 2005

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

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

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

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

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

[82] Z. J. Wang and Y. Liu, The Spectral Difference Method for the 2D Euler Equations on unstructured grids, AIAA paper No. 2005-5112, 2005.

[83] Z. J. Wang, Y. Sun, C. Liang, and Y. Liu, “Extension of the spectral difference method to viscous flow on unstructured grids”, Proceedings of the 4th International conference on Computational Fluid Dynamics, Ghent, Belgium, July 2006.

[84] M.Wheeler. An elliptic collocation-finite element method with interior penalties. SIAM J. Numer. Anal., 15:152–161, 1978.

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

[86] Zhang M, Shu C-W, An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations, Mathematical Models and Methods in Applied Sciences 2003; 13:395-413