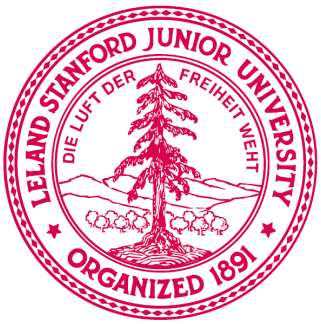
A p-Multigrid Spectral Difference method for viscous compressible flow using 2D quadrilateral meshes

Report ACL 2008-5

by

Sachin Premasuthan, Chunlei Liang & Antony Jameson



Aerospace Computing Laboratory

Aeronautics and Astronautics

Stanford University

May 2008

Abstract

The work focuses on the application of p-multigrid technique to accelerate convergence of viscous flow calulations using the Spectral Difference (SD) method. This paper extends the previous work by Liang et al (2007) on p-multigrid method for 2D inviscid compressible flow to viscous flows. The high-order spectral difference solver for unstructured quadrilateral meshes is based on the formulation of Sun et al Sun et al. (2007b) for unstructured hexahedral elements. The p-multigrid method operates on a sequence of solution approximations of different polynomial orders ranging from one upto four. The spectral difference method with p-multigrid will be applied to a variety of compressible flow problems to demonstrate the accuracy and speed-up using this technique.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Until recently, compressible flow computations on unstructured meshes have generally been dominated by schemes restricted to second order accuracy. However, the need for highly accurate methods in applications such as large eddy simulation, direct numerical simulation, computational aeroacoustics etc., has seen the development of higher order schemes for unstructured meshes such as the Discontinuous Galerkin (DG) Method (Bassi and Rebay (1997); Luo et al. (2006)), Spectral Volume (SV) method (Liu et al. (2006*b*); Wang and Liu (2006)) and Spectral Difference (SD) Method (Liu et al. (2006*a*); Wang et al. (2007)). The SD method is a newly developed efficient high-order approach based on differential form of the governing equation. It was originally proposed by Liu et al. (2006*a*) and developed for wave equations in their paper on triangular grids. Wang et al. (2007)(2007) extended it to 2D Euler equations on triangular grids and Sun et al. (2007*b*)(2007) further developed it for three-dimensinal Navier-Stokes equations on hexahedral unstructured meshes. The SD method combines elements from finite-volume and finite-difference techniques, and is particularly attractive because it is conservative, and has a simple formulation and implementation.

One of the difficulties encountered in large-scale simulations and high-order solutions is that the rate of convergence slows down dramatically as order is increased. In order to speed up convergence, an implicit temporal discretisation or a multigrid strategy is required. The issue with implicit solution methods is that they require a considerable amount of memory to store the Jacobian matrix, which becomes prohibitive for large scale 3D problems and high-order solutions. The p-multigrid method is an iterative scheme

which solves the discretized equations by recursively iterating on solution approximations of different polynomial orders. This method was initially proposed by Ronquist and Patera Ronquist and A.T.Patera (1987), and extended by Maday and Munoz Maday and Munoz (1988). The convergence acceleration by the p-multigrid algorithm on Euler equations was demonstrated by Liang et al. (2007) for unstructured triangular mesh. In this paper, we extend the previous work by Liang et al to viscous flow problems.

Chapter 2

Formulation of 2D Spectral Difference Scheme on quadrilateral meshes

The formulation of the equations for the 2D spectral difference scheme on quadrilateral meshes is similiar to the formulation of Sun et al Sun et al. (2007*b*) for unstructured hexahedral grids

Consider the unsteady compressible 2D Navier Stokes equations in conservative form

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{2.1}$$

where Q is the vector of conserved variables; F and G are the total fluxes including both inviscid and viscous flux vectors.

To achieve an efficient implementation, all elements in the physical domain (x, y) are transformed into a standard square element. $0 < \xi < 1$, $0 < \eta < 1$. The transformation can be written as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi, \eta) \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{2.2}$$

where K is the number of points used to define the physical element, (x_i, y_i) are the cartesian coordinates at those points, and $M_i(\xi, \eta)$ are the shape functions. The metrics and the Jacobian of the transformation can be computed for the standard element. The governing equations in the physical domain are then transfered into the computational domain, and the transformed equations take the following form:

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0 \tag{2.3}$$

where $\bar{Q} = |J| \cdot Q$ and

$$\begin{pmatrix} \bar{F} \\ \bar{G} \end{pmatrix} = |J| \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix} \tag{2.4}$$

In the standard element, two sets of points are defined, namely the solution points and the flux points, illustrated in figure 2.1. In order to construct a degree $(N-1)$ polynomial in each coordinate direction, solution at N points are required. The solution points in 1D are chosen to be the Gauss points defined by:

$$X_s = \frac{1}{2}\left[1 - cos\left(\frac{2s-1}{2N} \cdot \pi\right)\right], s = 1, 2, \cdots, N. \tag{2.5}$$

The flux points are selected to be the Gauss-Lobatto points given by

$$X_{s+1/2} = \frac{1}{2}\left[1 - cos\left(\frac{s}{N} \cdot \pi\right)\right], s = 0, 1, \cdots, N. \tag{2.6}$$

Using the solutions at N solution points, a degree $(N-1)$ polynomial can be built using the following Lagrange basis defined as:

$$h_i(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s}\right) \tag{2.7}$$

Similiarly, using the fluxes at $(N+1)$ flux points, a degree N polynomial can be built for the flux using a similar Lagrange basis defined as:

$$l_{i+1/2}(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}}\right) \tag{2.8}$$

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

$$Q(\xi, \eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \frac{\tilde{Q}_{i,j}}{|J_{i,j}|} h_i(\xi) \cdot h_j(\eta) \tag{2.9}$$

Similiarly, the reconstructed flux polynomials take the following form:

$$\begin{aligned} \tilde{F}(\xi, \eta) &= \sum_{j=1}^{N}\sum_{i=0}^{N} \tilde{F}_{i+1/2,j} l_{i+1/2}(\xi) \cdot h_j(\eta), \\ \tilde{G}(\xi, \eta) &= \sum_{j=0}^{N}\sum_{i=1}^{N} \tilde{G}_{i,j+1/2} h_i(\xi) \cdot l_{j+1/2}(\eta) \end{aligned} \tag{2.10}$$

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, a Riemann solver is employed to compute a common

flux at interfaces to ensure conservation amd stability. In our case, we have used the Rusanov solver to compute the interface fluxes.

In summary, the algorithm to compute the inviscid flux derivatives consists of the following steps:

1. Given the conservative variables at the solution points, the conservative variables are computed at the flux points

2. The inviscid fluxes at the interior flux points are computed using the solutions computed at Step 1

3. The inviscid fluxes at the element interfaces are computed using the Rusanov solver.

4. The derivative of the fluxes are computed at the solution points according to (equation)

$$\left(\frac{\partial \tilde{F}}{\partial \xi}\right)_{i,j} = \sum_{r=0}^{N} \tilde{F}_{r+1/2,j} \cdot l'_{r+1/2}(\xi_i), \tag{2.11}$$

$$\left(\frac{\partial \tilde{G}}{\partial \eta}\right)_{i,j} = \sum_{r=0}^{N} \tilde{G}_{i,r+1/2} \cdot l'_{r+1/2}(\eta_j) \tag{2.12}$$

The viscous flux is a function of both the conserved variables and their gradients. Therefore, the solution gradients have to be calculated at the flux points. In our solver, the average approach described in reference Sun et al. (2007*b*) is used to compute the viscous fluxes.

A four-stage Runge Kutta scheme is used to advance the solution in time.

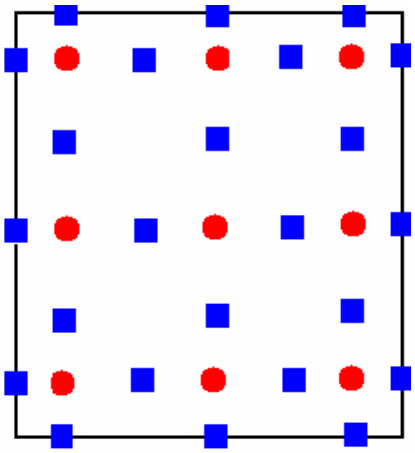


Figure 2.1: Position of solution (circles) and flux (squares) points on standard square element for 3rd order SD

Chapter 3

p-Multigrid method

The p-multigrid method is a natural extension of geometric multigrid methods to high-order accurate formulations. 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 at lower polynomial levels leads to the p-multigrid method.

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 Helenbrook and Atkins (2006); Luo et al. (2006). 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.

In this study a multi-level V-cycle has been used to drive the iterations. To accomplish the communication between different levels, we use restriction and prolongation operators derived from cardinal basis functions of the solution points for the starting p-level. Currently a four-stage Runge-Kutta explicit scheme is employed as a smoother at all levels

The following steps summarize the standard two-level scheme with a V-cycle at p and $p-1$ levels:

Firstly, we are trying to solve $R_p(Q_p) = r_p$, where $R_p(Q_p)$ corresponds to the residual

of current solution Q_p, r_p is the rhs and equals 0 for the higher polynomial level p.

1. Apply smoothing iterations at level p to improve the solution Q_p.

2. Compute the defect at level p.

$$d_p = r_p - R_p\left(Q_p\right) = -R_p\left(Q_p\right) \tag{3.1}$$

3. Restrict the latest solution and the defect to the lower approximation level $p-1$

$$Q_{p-1}^0 = I_p^{p-1}\left(Q_p\right) \tag{3.2}$$

4. Compute the rhs of equation at level $p-1$.

$$r_{p-1} = R_{p-1}\left(Q_{p-1}^0\right) + I_p^{p-1} d_p \tag{3.3}$$

5. Apply smoothing iterations at level $p-1$ to improve Q_{p-1}

6. Compute the correction at level $p-1$.

$$C_{p-1} = \bar{Q}_{p-1} - Q_{p-1}^0 \tag{3.4}$$

7. Extrapolate the correction from level $p-1$ to update the solution at level p.

$$\tilde{Q}_p = Q_p + I_{p-1}^p C_{p-1} \tag{3.5}$$

8. Improve Q_p by application of iterative smoother

The p-multigrid has been used to accelerate convergence for DG method Luo et al. (2006). For SD schemes, the p-multigrid has been used for both inviscid and viscous flows on triangular grids. This paper intends to study the performance and effectiveness of p-multigrid with quadrilateral grids using upto four-level multigrid cycles. Earlier studies for Euler flow calculations have indicated that the use of explicit smoothing iterations at all levels adversely affects the performance of the p-multigrid. This has been attributed to the severely anisotropic (hyperbolic) nature of the Euler equations and the isotropic (elliptic) nature of p-multigrid iterations Luo et al. (2006). Efforts will be made to address this issue with regards to the viscous flow equations. The possible use of different time-integration schemes at different levels: explicit RK smoothers at the higher levels, and implicit schemes at the lower levels, will be examined.

Chapter 4

Results

In the following, the results obtained from the testing and validation of the inviscid flow solver are presented. The supersonic vortex test case and the inviscid flow past circle case are discussed in this context. In the next section, the testing and validation of the viscous flow solver is discussed. The test cases demonstrated include the planar Couette flow test case and the viscous flow past circle case. We then discuss (in section *IV.C*), the application of p-multigrid method to the inviscid flow computations, using the inviscid flow past bump test case. Finally, the application of p-multigrid to viscous flow computation is demonstrated, with the planar Couette flow test case.

4.1 Results for inviscid flow

4.1.1 Validation using supersonic vortex flow

This test case is used to verify the implementation of the developed computer code and assess the order of accuracy of the spectral difference method used. The supersonic vortex flow problem is one of the few non-trivial problems of the steady compressible 2D Euler equations for which a smooth analytical solution is known. The inviscid, isentropic, supersonic flow of a compressible fluid between concentric circular arcs presents a flow where the velocity varies inversely with radius. The expression for density as a function of radius r is given by:

$$\rho\left(r\right) = \rho_i \left\{ 1 + \frac{\gamma - 1}{2} M_i^2 \left[1 - \left(\frac{r_i}{r} \right)^2 \right] \right\}^{\frac{1}{\gamma - 1}} \tag{4.1}$$

where M_i and r_i are the Mach number and the radius at the inner arc. In the present calculation, the Mach number, density and pressure at the inner radius r_i are specified to be 2.25, 1 and $1/\gamma$ respectively. The inner and outer radii are 1 and 1.384. The outer arc and bottom boundaries are fixed with analytical solutions. The zero-gradient extrapolation boundary is employed for the exit. In the following, the numerical solution to this problem are computed for 2nd, 3rd and 4th order SD method on succesively refined grids. All the computations are initialized using constant density and pressure. The L^2-error of the density is evaluated.

The four meshes used in the computation were of sizes 10×4, 15×6, 30×12, and 60×24. A sample 15×6 mesh is shown in figure 4.1 (a). Figure 4.1 (b) shows the density contours in the flow field obtained by 3rd order SD method. The details of the order calculation and verification are shown in Table 1. The table clearly indicates that the SD method applied to the steady compressible Euler equations exhibits a full order of convergence on smooth solutions. Figure 4.2 (a) provides the details of the spatial accuracy of the SD method for different orders for this numerical experiment. Figure 4.2 (b) shows the L^2-error of the SD method at different order SD method plotted against the number of degrees of freedom. One can clearly see that a higher order SD method requires a lesser number of degrees of freedom than a lower order SD method to achieve the same accuracy.

4.1.2 Subsonic flow past a circle

In this section, the results obtained for 2D steady, subsonic flow around a circle at Mach number M_∞=0.2, are presented. Computations were made for second, third and fourth order approximation of unknowns, and linear, quadratic and cubic geometric mapping for the curved boundary elements. The computations have been performed on a 32×32 grid . This test case demonstrates the significance of higher order representation of boundary elements to obtain accurate solutions.

Figures 4.3 (a) and (b) show the Mach number contours using 3rd and 4th order SD with linear representation of boundary. These solutions are very inaccurate, as put in evidence by the non-physical “boundary layer" which develops along the solid wall, and by the associated “wake" in the downstream region of the circle. Moreover the compu-

tations do not converge to a steady solution due to the unsteadiness of the unphysical wake. Figure 4.3 (c) shows the Mach number contours using 3rd order SD with quadratic boundary, and figure 4.3 (d) corresponds to using 4th order SD with cubic boundary. It clearly indicates that the use of higher order representation for the boundary, results in a smoother and more accurate solution.

4.2 Results for viscous flow

4.2.1 Validation using planar Couette flow

The planar Couette flow test case was used to test and validate the viscous solver, and to confirm the order of accuracy of viscous computations. This problem models the viscous flow between two parallel plates located at $y = 0$ and $y = H$. The plate at $y = 0$ is stationary and kept at a temperature T_0, and the plate at $y = H$ is moving at speed U in the x-direction and is at fixed temperature T_1. Under the assumption of constant viscosity coefficient, this problem has an exact solution, which can be expressed as

$$u(y) = \frac{U}{H}y, v = 0, w = 0, \tag{4.2}$$

$$T(y) = T_0 + \frac{y}{H}(T_1 - T_0) + \frac{\mu U^2}{2k} \cdot \frac{y}{H}\left(1 - \frac{y}{H}\right), \tag{4.3}$$

$$p = const, \rho = \frac{p}{R \cdot T} \tag{4.4}$$

The numerical solution to this problem was computed for 2nd, 3rd and 4th order SD method on succesively refined grids. All the computations are initialized using constant density and pressure. Isothermal wall BC was used for top and bottom walls, and periodic BC was used for left and right boundary. The L^2-error of the density is evaluated. The four meshes used in the computation were of sizes 2×1, 4×2, 8×4, and 16×8. Figure 4.4 shows the density contours in the flow field obtained by 3rd order SD method using the 4×2 grid. The details of the order calculation and verification are shown in Table 2. The table clearly indicates that the SD method applied to the steady compressible Navier Stokes equations exhibits a full order of convergence on smooth solutions.

4.2.2 Viscous flow past a circle

The viscous solver was also tested for flow past a circle. The results obtained for viscous flow past a circle are discussed below. The grid used for viscous computation had 800 elements and is shown in figure 4.5 (a). The freestream Mach number is 0.2 and the Reynolds number is 160. Computations were performed using both 3rd and 4th order schemes with corresponding quadratic and cubic boundary representations. The computed Mach number contours for 4th order unsteady calculation are shown in figure 4.5 (b). The unsteady von Karman vortex street was predicted by both the 3rd and 4th order schemes. It was observed that the flow reached a periodic state in both the cases. For the fourth order solution, the frequency of the periodic variation of lift coefficient C_L gave a Strouhal number of 0.170.

Further efforts will be made to test and validate the viscous solver for a variety of flow problems.

4.3 Application of p-multigrid to subsonic flow past bump

This test case consists of subsonic flow in a channel with a 10% thick circular bump on the bottom. The length of the channel is 3 units and its height 1 unit. The inlet Mach number is 0.5. This test case has been used by p-multigrid for DG formulation Luo et al. (2006) for Euler equation as well as for SD formulations Liang et al. (2007). The mesh which contains 3201 gridpoints, 3072 elements in total and 32 nodes to resolve the bump, is depicted in figure 4.6. It must be mentioned that for this grid, a 3rd order computation would involve $(3072 \times 9 =)$ 27648 computational nodes. The circular surface of the bump is represented as a quadratic boundary. The pressure contours obtained using 4th order SD scheme with 4-level multigrid is shown in figure 4.7, and is almost identical to those obtained in references Luo et al. (2006); Liang et al. (2007). All multigrid calculations for the bump case are done using 4th order RK scheme for time marching at all levels.

Figure 4.8 shows the comparison for convergence history of 2nd order and 1st order SD schemes. It is observed that the convergence using explicit RK4 scheme deteriorates

drastically when the SD order is increased to 2. For 3rd and 4th order SD, the RK scheme failed to converge. These convergence trends for higher order SD using RK time-stepping for the subsonic bump case have been documented earlier Luo et al. (2006); Sun et al. (2007*a*). However the application of the *p*-multigrid method results in superior convergence characteristics for second, third and fourth order SD schemes.

Figure 4.8 shows the convergence acceleration in terms of number of iterations as well as CPU time for the second order SD scheme. A 2-level V-cycle *p*-multigrid method has been used with single smoothing iteration at the higher level (p1) and 2 smoothing iterations at the lower level (p0). (It should be noted that p0 corresponds to 1st order SD, p1 to 2nd order and so on.) It is evident that speed-up is obtained using the 2-level multigrid cycle. Application of similiar multi-level *p*-multigrid method to 3rd and 4th order results in convergence acceleration as seen in figures 4.9 (a) and (b). In the absence of p-multigrid both 3rd and 4th order fail to converge. However, the use of p-multigrid results in favourable convergence characteristics. For the 3rd order SD, a 3 level V-cycle is used with 1-1-2-1-0 smoothing iterations at p2-p1-p0-p1-p2 levels. For the 4th order SD case, a 4-level V-cycle with 1-1-1-2-1-1-0 smoothing iterations at p3-p2-p1-p0-p1-p2-p3 levels is used.

The effect of changing the V-cycle on the convergence of 3rd order SD was studied. As expected the p2-p1-p0 3-level multigrid was found to give best results, as seen in figure 4.10. However, the p2-p0 2-level (3,1 (a)) V-cycle produces comparable convergence acceleration. This can be attributed to the fact that for a single level, p0 has the fastest convergence. The p2-p1 2-level cycle has poor convergence characteristics due to the slow convergence at p1 level. When a 2-level V-cycle at is used with 1-6-0 smoothing iterations at p2-p0-p2 levels (3,1 (b)), there is a drastic drop in residual initially, but then the convergence deteriorates.

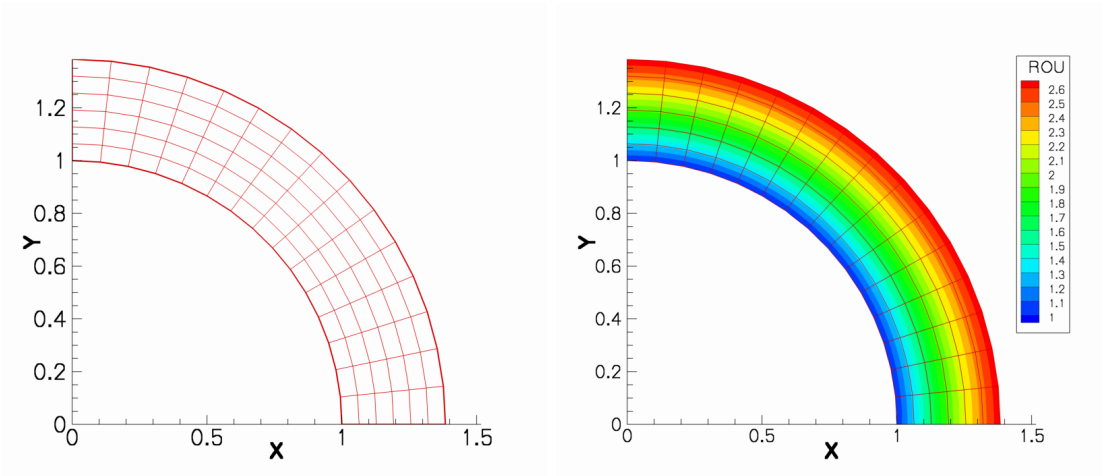
In the case of the 4th order SD method, the 4-level V-cycle (4,3,2,1 p-MG) with 1-1-1-2-1-1-0 iterations at p3-p2-p1-p0-p1-p2-p3 is found to be most effective for convergence acceleration, as seen in figures 4.11 (a) and (b). The 3-level V-cycle with 1-1-2-1-0 iterations at p3-p1-p0-p1-p3 (4,2,1 p-MG) gives the next best convergence. The use of p3-p2-p0-p2-p3 (4,3,1 p-MG) and the p3-p2-p1-p2-p3 (not shown in figure) 3-level cycles does not show good convergence characteristics. This could be attributed to the fact that

two of the levels p3 and p2 both have very poor convergence.

4.4 Application of p-multigrid to planar Couette flow

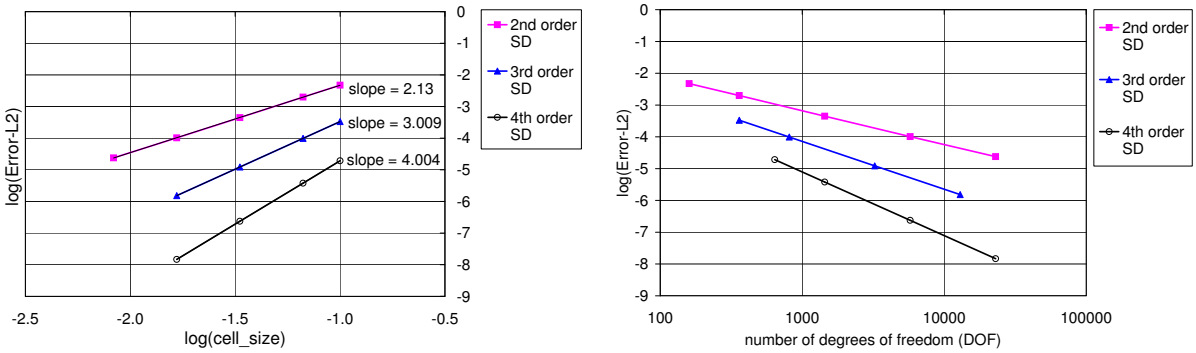
The p-multigrid method was used for convergence acceleration for the planar Couette flow test case. All multigrid calculations were done using 4th order RK scheme for time marching at all levels. It must be mentioned that for these computations, the multilevel p-multigrid cycle used has a single smoothing iteration at the higher levels and 2 smoothing iterations at the lowest level (p0). Also, the 2nd and 3rd order computations were done on the 8×4 grid (corresponding to 128 and 288 computational nodes respectively), while the 4th order computation was done on the 4×2 grid (128 computational nodes).

In figure 4.12, we see that for a second order computation, the application of p-multigrid accelerates convergence by about 3 times. For the third order computation, the 2-level V-cycle (3,1 p-MG) is found to be more effective for convergence acceleration than the 3-level V-cycle (3,2,1 p-MG) (see figure 4.13). The application of p-multigrid is able to reduce the computation time by a factor of about 5 in this case. In the case of the fourth order computation, the 3-level V-cycle (4,2,1 p-MG) gives the best convergence characteristics (see figure 4.14). A speed-up of about 8 is obtained for 4th order SD.



(a) (b)

Figure 4.1: (a) 15×6 mesh for supersonic vortex flow test case; (b) Density contours using 3rd order SD.



(a) (b)

Figure 4.2: Accuracy summary for the supersonic vortex flow case for SD method using 2nd, 3rd and 4th order solution approximations (a) L^2 error vs cell-size; (b) L^2 error vs degrees of freedom.

Table 4.1: L^2 errors and orders of accuracy for spersonic vortex case

No. of elements	No. of DOFs	L2-error	Order
2nd order SD			
40	160	4.7249E-03	-
90	360	1.9881E-03	2.135
360	1440	4.4721E-04	2.152
1440	5760	1.0196E-04	2.133
3rd order SD			
40	360	3.3393E-04	-
90	810	9.8833E-05	3.003
360	3240	1.2242E-05	3.013
1440	12960	1.5230E-06	3.007
4th order SD			
40	640	1.9238E-05	-
90	1440	3.7883E-06	4.008
360	5760	2.3651E-07	4.002
1440	23040	1.4743E-08	4.004

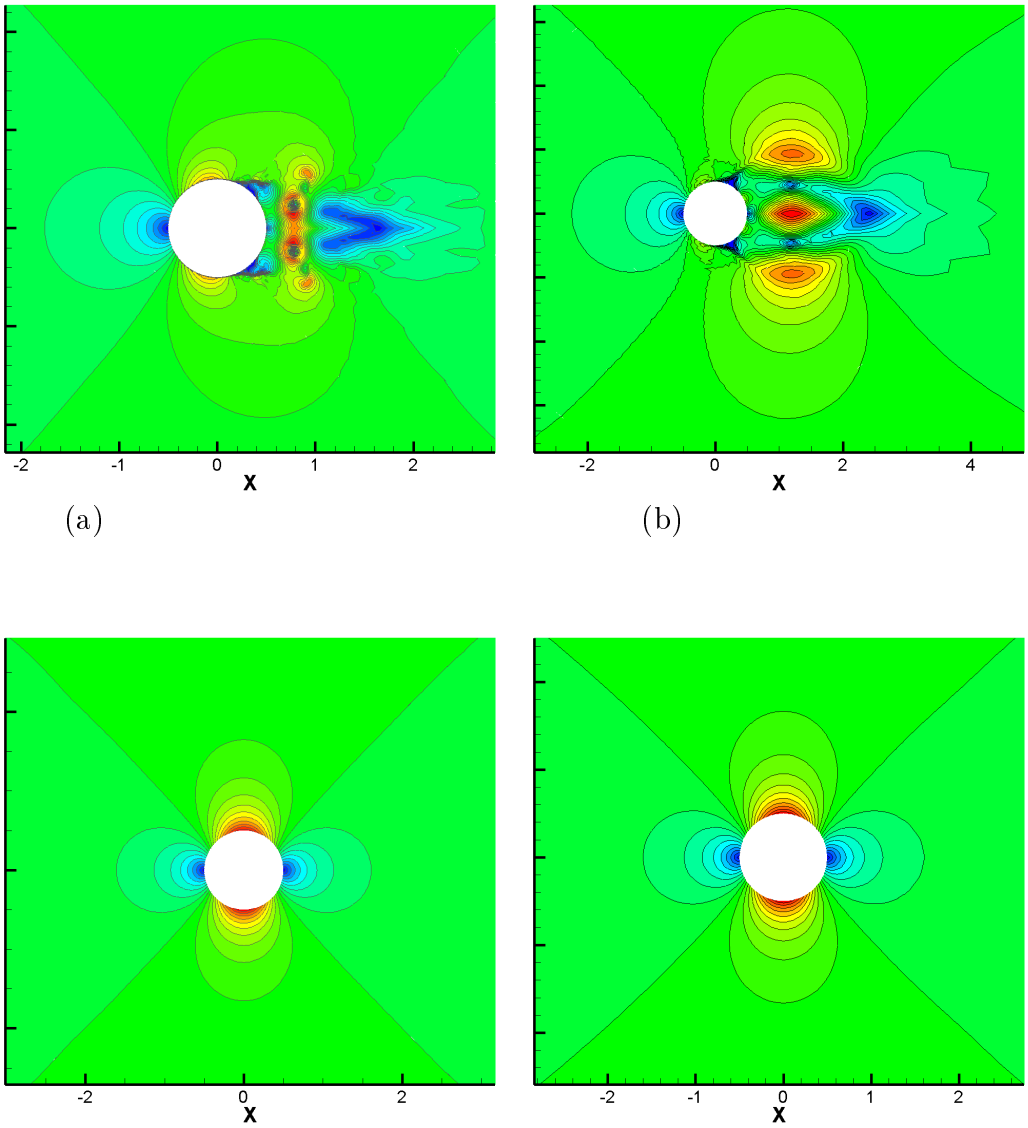


Figure 4.3: Mach contours for inviscid flow past a circle (a) 3rd order SD with linear boundary (b) 4th order SD with linear boundary (c) 3rd order SD with quadratic boundary (d) 4th order SD with cubic boundary

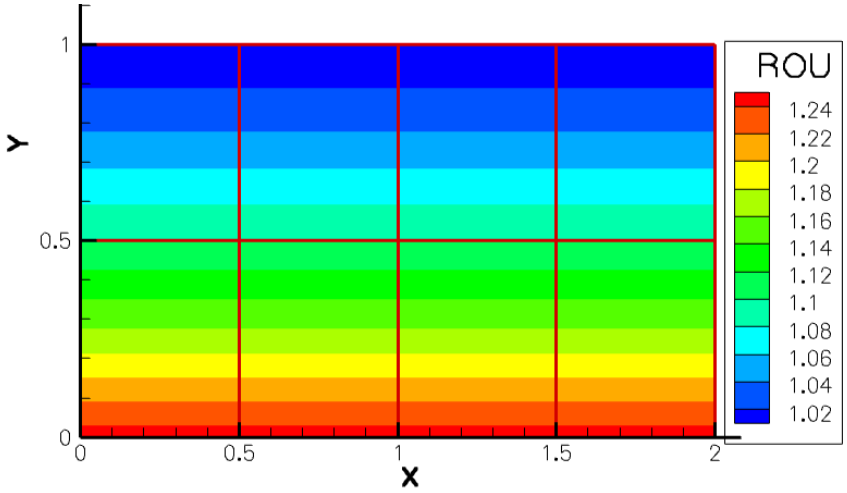
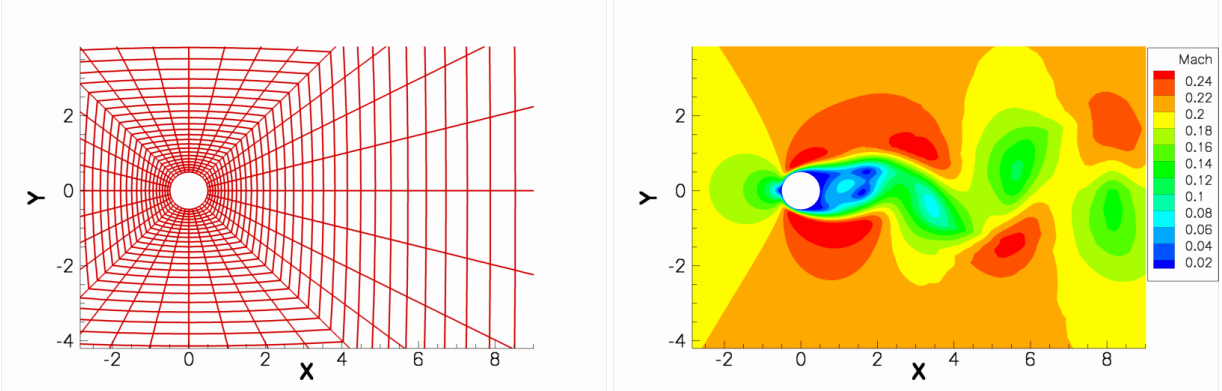


Figure 4.4: Density contours for Couette flow problem using 3rd order SD (4×2 grid)



(a) (b)

Figure 4.5: Viscous flow around circle (a) Computational grid; (b) Mach number contours using 4th order SD.

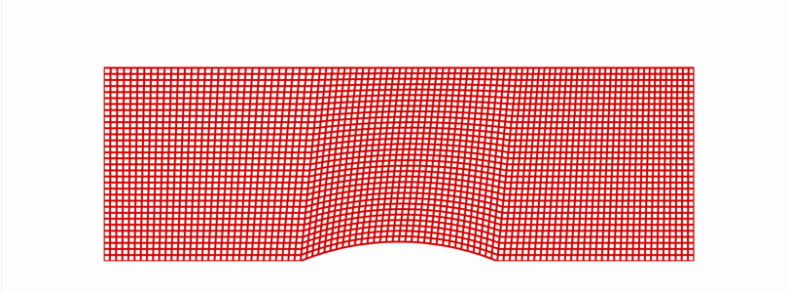


Figure 4.6: Grid used for subsonic bump case

Table 4.2: L^2 errors and orders of accuracy for Couette flow case

No. of elements	No. of DOFs	L2-error	Order
2nd order SD			
2	8	1.4180E-02	-
8	32	3.3520E-03	2.081
32	128	9.1210E-04	1.878
128	512	2.4350E-04	1.905
3rd order SD			
2	18	1.4783E-03	-
8	72	1.5199E-04	3.282
32	288	1.6525E-05	3.201
128	1152	1.7991E-06	3.199
4th order SD			
2	32	1.9784E-04	-
8	128	1.1827E-05	4.064
32	512	7.9780E-07	3.980
128	2048	4.7330E-08	4.075

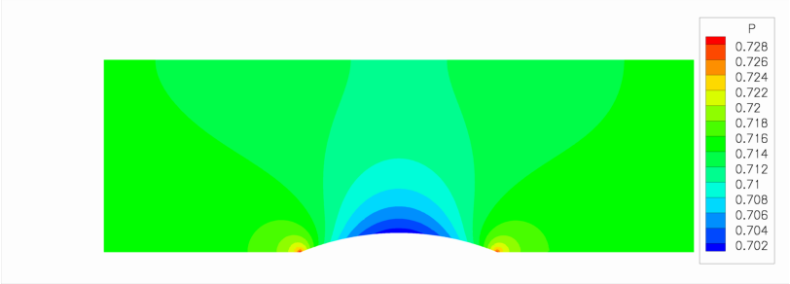
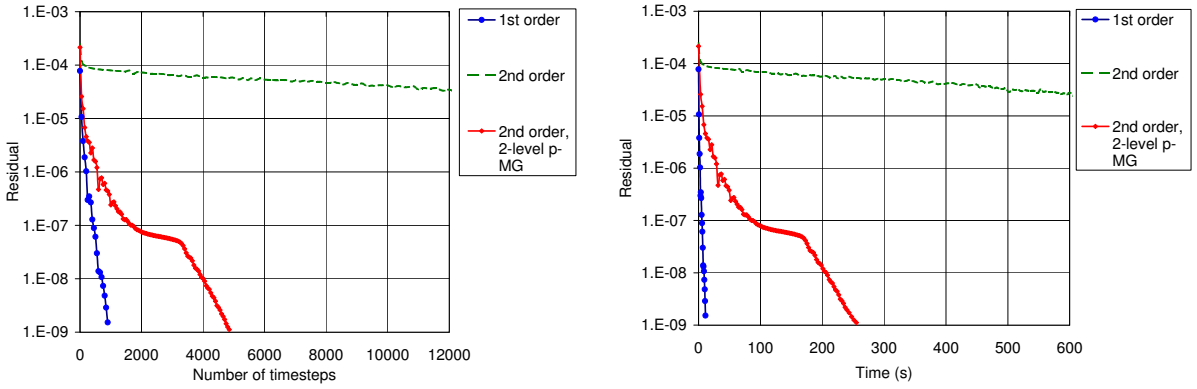
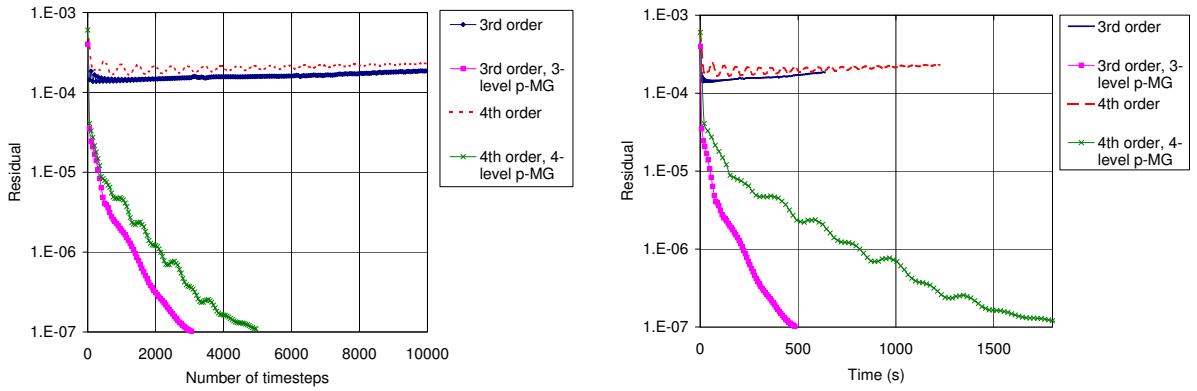


Figure 4.7: Pressure Contours using 4th order SD



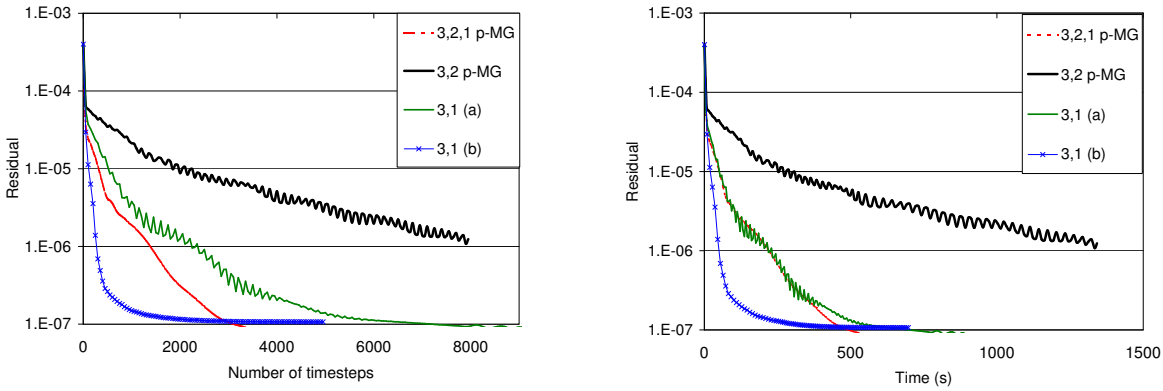
(a) (b)

Figure 4.8: Convergence history for first and second order SD (with and without p-multigrid) for subsonic flow over bump testcase (a) Plot of residual vs timesteps (V-cycles for p-multigrid computation); (b) Plot of residual vs. CPU time.



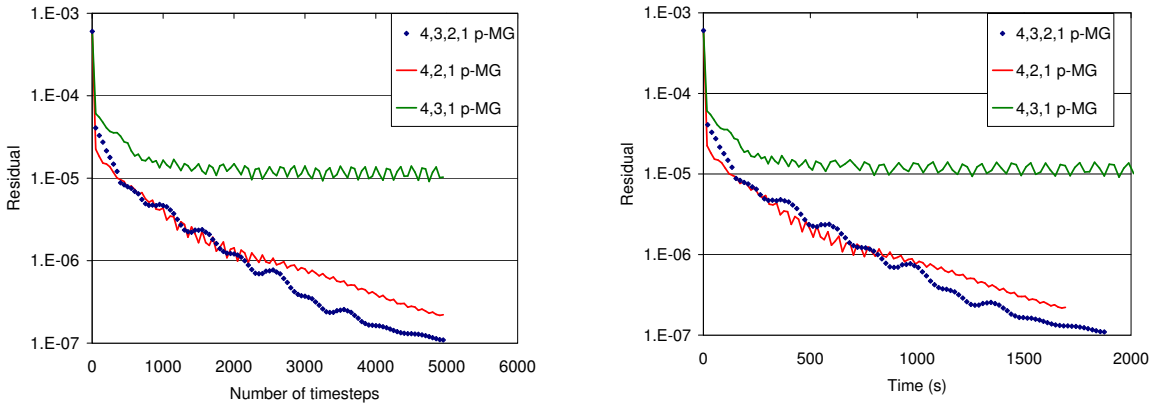
(a) (b)

Figure 4.9: Convergence history for third and fourth order SD (with and without p-multigrid) for subsonic flow over bump testcase (a) Plot of residual vs timesteps (V-cycles); (b) Plot of residual vs. CPU time.



(a) (b)

Figure 4.10: Convergence history for 3rd order SD using different V-cycles for p-multigrid. The curve '3,1(a)' corresponds to 2-level with 2 iterations at the lowest level, while curve '3,1(b)' has 6 iterations at the lowest level (a) Plot of residual vs timesteps (V-cycles); (b) Plot of residual vs. CPU time.



(a) (b)

Figure 4.11: Convergence history for 4th order SD using different V-cycles for p-multigrid. (a) Plot of residual vs timesteps (V-cycles); (b) Plot of residual vs. CPU time.

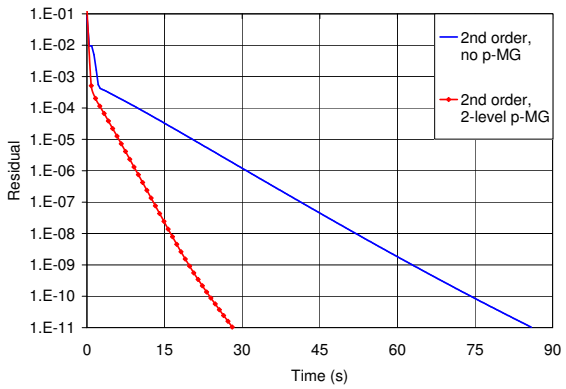


Figure 4.12: Convergence history for 2nd order SD with and without p-multigrid (8×4 grid)

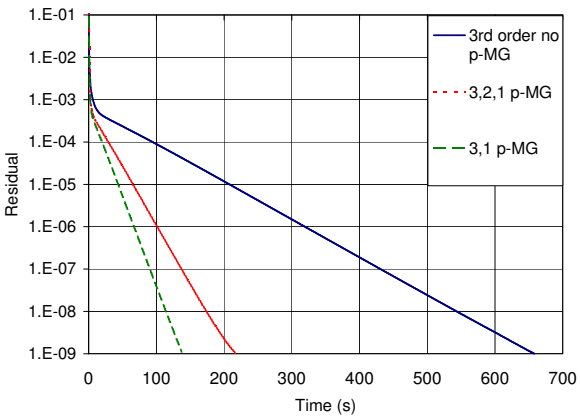


Figure 4.13: Convergence history for 3rd order SD with and without p-multigrid (8×4 grid)

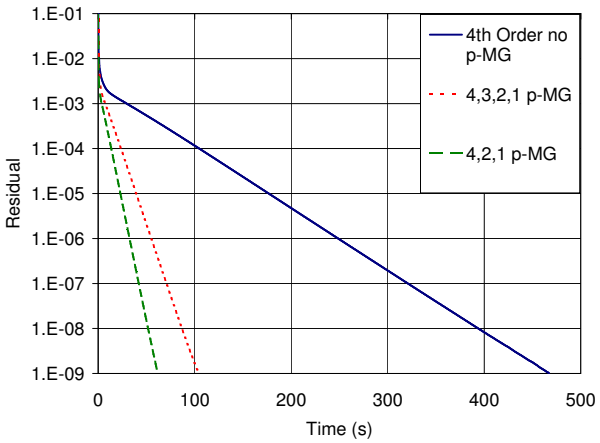


Figure 4.14: Convergence history for 4th order SD with and without p-multigrid (4×2 grid)

Chapter 5

Conclusions and Future work

We have developed a spectral difference solver for inviscid and viscous flows. p-Multigrid has been applied to the inviscid subsonic flow past bump test case and the planar Couette flow test case with upto four multigrid levels, and the results have been promising. The p-multigrid will be applied to other test cases to study its applicability and effectiveness to accelerate convergence. Different multigrid cycles such as W-cycle and Full Multigrid (FMG) will be implemented to find the fastest and most efficient technique.

We plan to extend our p-multigrid spectral difference implementation to 3D. The p-multigrid method can have highly beneficial effects when used for 3D flow problems. This is because the use of implicit methods for 3D flow problems are not very feasible due to the high storage requirements. This memory restriction is aggravated for problems where we seek a high-order solution (> 3). Therefore it is expected that for high order SD schemes for 3D flow problems, p-multigrid could be an efficient and effective technique to accelerate convergence.

Bibliography

Bassi, F., Rebay, S. (1997) High-order accurate discontinuous finite element solution of the 2d euler equations. Journal of Computational Physics 138: 251–285.

Helenbrook, B. T., Atkins, H. L. (2006) Application of p-multigrid to discontinuous galerkin formulations of the poisson equation. AIAA Journal 44: 566–575.

Liang, C., Kannan, R., Wang, Z. J. (2007) A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured grids. AIAA paper AIAA-2007-4326.

Liu, Y., Vinokur, M., Wang, Z. J. (2006*a*) Spectral difference method for unstructured grids i: Basic formulation. J. of Comput. Phys. 216: 780–801.

Liu, Y., Vinokur, M., Wang, Z. J. (2006*b*) Spectral (finite) volume method for conservation laws on unstructured grids v: Extension to three-dimensional systems. Journal of Computational Physics 212: 454–472.

Luo, H., Baum, J. D., Löhner, R. (2006) A p-multigrid discontinuous galerkin method for the euler equations on unstructured grids. Journal of Computational Physics 211: 767–783.

Maday, Y., Munoz, R. (1988) Spectral element multigrid, part 2: Theoretical justification. *Tech. Rep.* Tech. Rep 88-73. ICASE.

Ronquist, E., A.T.Patera (1987) Spectral element multigrid, i. formulation and numerical results. Journal of Scientific Computing 2: 389–406.

Sun, Y., Wang, Z. J., Liu, Y. (2007*a*) Efficient implicit lu-sgs algorithm for high-order spectral difference method on unstructured hexahedral grids. AIAA paper AIAA-2007-313.

Sun, Y., Wang, Z. J., Liu, Y. (2007*b*) High-order multidomain spectral difference method for the navier-stokes equations on unstructured hexahedral grids. Communication in Computational Physics 2: 310–333.

Wang, Z., Liu, Y., May, G., Jameson, A. (2007) Spectral difference method for unstructured grids ii: Extension to the euler equations. Journal of Scientific Computing 32: 45–71.

Wang, Z. J., Liu, Y. (2006) Extension of the spectral volume method to high-order boundary representation. Journal of Computational Physics 211: 154–178.