18th AIAA Computational Fluid Dynamics Conference
25 - 28 June 2007, Miami, FL

AIAA 2007-4326

# A p-Multigrid Spectral Difference Method with Explicit and Implicit Smoothers on Unstructured Grids

C. Liang[1], R. Kannan[2] and Z.J. Wang[3]
*Department of Aerospace Engineering, Iowa State University, Ames, IA 50014*

**The convergence of high-order methods, such as recently developed Spectral Difference (SD) method, can be accelerated using both implicit temporal advancement and a p-Multigrid approach. A good combination of these two can significantly improve the efficiency of the SD method for steady flow problems. In this paper, we demonstrate a p-Multigrid approach developed for a 2D Euler solver using the SD method. A fast preconditioned Lower-Upper Symmetric Gauss Seidel (LU-SGS) implicit method is developed and tested for both linear scalar, and nonlinear Euler equations. Meanwhile, a five-stage Runge-Kutta explicit method is employed for comparison. We are able to achieve significant speedup (up to two orders) using both p-Multigrid method and the implicit smoother while maintaining very stable convergence property and nearly ideal order of accuracy. The numerical results are very promising, and indicate that the approach has great potential for 3D flow problems.**

## Nomenclature

| | |
|---|---|
| d | = defect of residual for Multigrid approach |
| $D$ | = Left hand Jacobian matrix $\left(\dfrac{I}{\Delta t}-\dfrac{\partial R_c}{\partial \tilde{Q}_c}\right)$ |
| I | = Restriction/prolongation operator |
| $M_{k,i}$ | = Shape functions defined by the flux points |
| $N_p$ | = Number of unknown points of a cell |
| $\tilde{Q}$ | = Solution at unknown points |
| $R_c(\tilde{Q})$ | = Current cell residual |

**Subscripts**

| | |
|---|---|
| c | = current cell |
| i | = index of cell |
| j | = index of solution points |
| k | = index of flux points |
| nb | = neighbouring cells |

## 1. Introduction

The SD method has been recently developed by Liu et al. [16] for wave equations on unstructured triangular grids and further developed by Wang and Liu [30] for 2D Euler equations. It has been shown that the 3rd –order SD produces more accurate results than a 2nd-order Finite-Volume method on a much coarser grid with fewer solution unknowns. Recently, the extension to the Navier-Stokes equations was also presented in [20,31].

With the high-order SD method, it is clearly possible to achieve lower error levels than the traditional first order or second order methods. However, all the above developments of the SD methods employed explicit TVD

[1]Present affiliation: Postdoctoral RA, Department of Mathematics, University of Glasgow, G12 8QW, U.K.
[2]Graduate Research Assistant, 0237 Howe Hall, Iowa State University
[3]Associate Professor of Aerospace Engineering, Iowa State University, Associate Fellow of AIAA

Runge-Kutta schemes for the temporal advancement and those numerical experiments are all performed on single grids. These explicit temporal discretizations can be implemented with ease. In addition, they require limited memory and can be easily vectorized and parallelized. However it is well-known that the explicit scheme is limited by the Courant-Friedrichs-Lewy (CFL) condition. When the polynomial order of the SD method is high or the grid is stretched due to complex geometries, the convergence rate of the explicit scheme slows down rapidly. Solution strategies to remedy this problem include the pre-conditioning methods [28], implicit methods [12] and multigrid methods [22].

The implicit methods are normally formulated by the linearization of a given set of equations. The development of implicit methods for high order methods like Discontinuous Galerkin methods [17,21] has been very noteworthy. Even though these implicit methods offer extremely high speedup, they need huge memory to store the associated matrices. This is greatly felt when the high order methods is applied to 3D problem. One intelligent way to mitigate the above problem is to use the traditional multi stage Runge-Kutta method as the high level smoother and the implicit scheme for the lower levels [17].

The p-multigrid method employs smoothing operators which hasten 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 [25], and extended by Maday and Munoz [19]. The acceleration of the convergence by the p-multigrid algorithm on Euler equations was demonstrated by Bassi and Rebay [2], Fidkowski et al [8], Nastase and Mavriplis [21] and Luo et al. [17] with the Discontinuous Galerkin method.

In this paper, we adopt an efficient preconditioned implicit LU-SGS approach [4,13] as the iterative smoother for the p-multigrid algorithm. The original LU-SGS approach was developed by Yoon and Jameson on structured grids [12,33]. The preconditioned LU-SGS approach further improved the efficiency. For comparison purpose, we also use an explicit TVD Runge-Kutta scheme as a p-multigrid smoother. As explained in the earlier paragraph, the high-order implicit methods require much larger storage of Jacobian matrices than the explicit method. Therefore, in this paper, we also intend to adopt explicit smoother for the high p-levels and simultaneously employing implicit schemes embedded with much smaller size of Jacobian matrix on low p-levels.

The paper is organized as follows. Section 2 presents the governing equations using the corresponding SD method. Temporal relaxation schemes are discussed in Section 3. In Section 4, we focus on the nonlinear FAS p-Multigrid method. Section 5 includes the validations of the implicit scheme over the wave equations on a single grid followed by the results and discussion of the Euler equations. In the end, some conclusions are drawn in Section 6.

## 2. Governing Equations and the SD Formulation

The 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. Figure 1 shows the placement of unknown and flux points for a triangular cell. In this paper, 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 [14,15] and can be viewed as an extension of the multi-domain spectral method to a simplex unstructured grid.

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

$$\frac{\partial \tilde{Q}}{\partial t}+\frac{\partial \tilde{f}}{\partial X}+\frac{\partial \tilde{g}}{\partial Y}=0 \quad \text{and} \quad R_c(\tilde{Q})=-\frac{\partial \tilde{f}}{\partial X}-\frac{\partial \tilde{g}}{\partial Y} \tag{2.1}$$

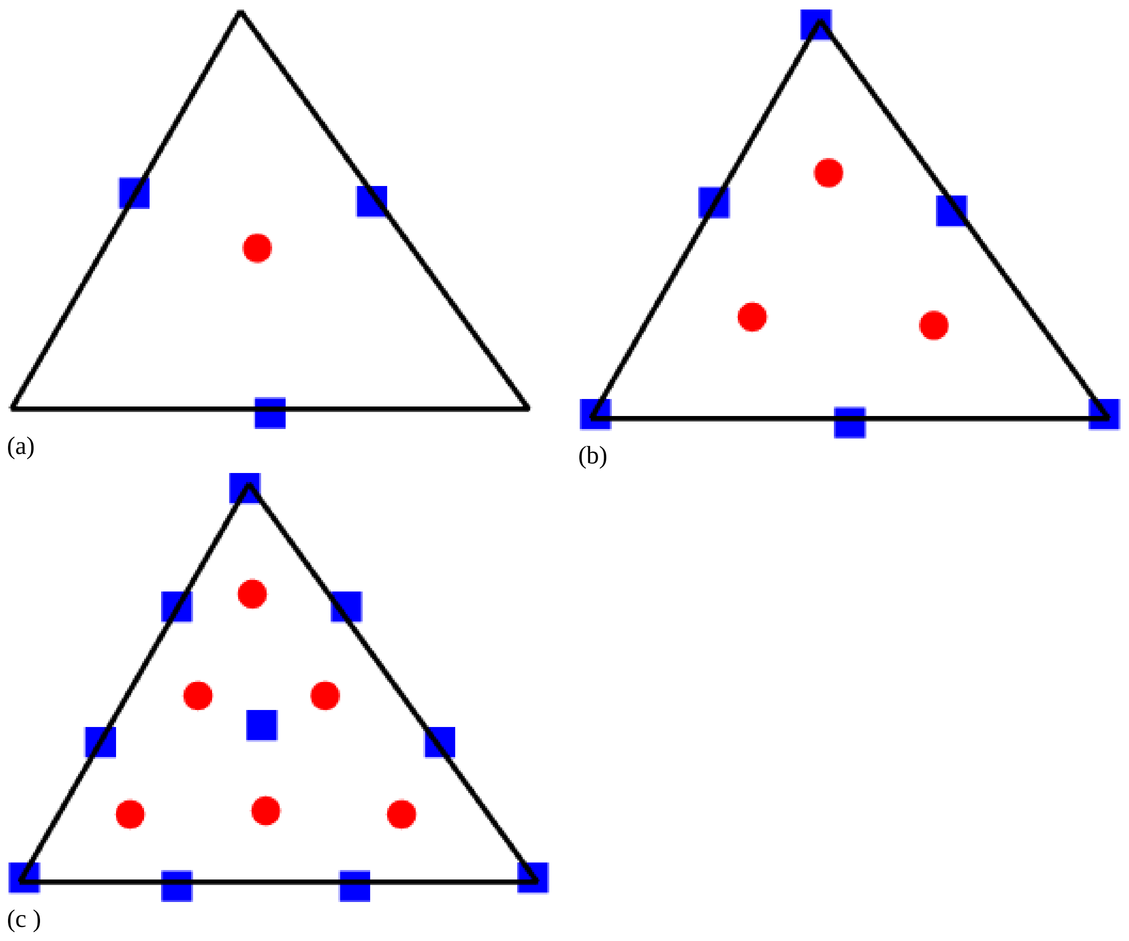


Figure 1: Placement of unknown (●) and flux (■) points for a triangular element. (a): First order; (b): Second order; (c): Third order.

The current cell residual term $R_c(\tilde{Q})$ can be evaluated once the neighbouring 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 $N_p$ is the number of unknown points required to support a degree p polynomial construction as already illustrated in figure 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. We employ one-dimensional Riemann solvers, namely, the Rusanov [26] or Roe [24] 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, 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 [30]. 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 $F_{n2}$. A flux vector $F(r_{k,i})$ at this corner point for this particular cell can be constructed using $F_{n1}$ and $F_{n2}$. 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

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

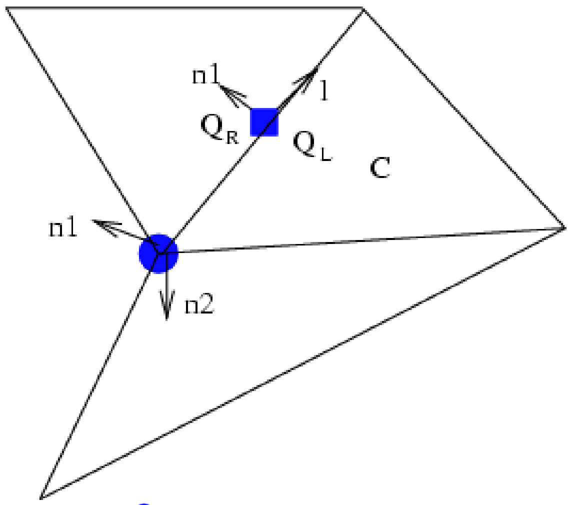


Figure 2: Flux computation for a corner (●) and an edge point (■) using one-dimensional Riemann solvers.

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, attention now is turned to temporal relaxation schemes.

## 3. Time relaxation schemes

In order to solve the flow to a steady state from a nearly arbitrary initial guess, a relaxation scheme is needed. The unsteady equation is considered in the current paper and the time integration schemes (either explicit or implicit schemes) can be used as a smoother.

$$\frac{\partial \tilde{Q}}{\partial t} = R_i^c(\tilde{Q}) = -\nabla \bullet F_i(r_{j,i}) \tag{3.1}$$

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 high order (several RK steps) scheme, it is seldom used as it required lots of storage and thus adversely affects its simplistic nature. Therefore, a primary effort in this paper is devoted to a new LU-SGS implicit scheme and it is explained in the following.

At each current cell *c*, using the backward Euler difference, (3.1) can be written as

$$\frac{\tilde{Q}_c^{n+1} - \tilde{Q}_c^n}{\Delta t} - \left[R_c(\tilde{Q}^{n+1}) - R_c(\tilde{Q}^n)\right] = R_c(\tilde{Q}^n) \tag{3.2}$$

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

$$R_c(\tilde{Q}^{n+1}) - R_c(\tilde{Q}^n) \approx \frac{\partial R_c}{\partial \tilde{Q}_c}\Delta\tilde{Q}_c + \sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb}\,, \tag{3.3}$$

where *nb* indicates all the neighbouring cells contributing to the residual of cell *c*. Therefore, the fully linearized equations for (3.2) can be written as

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c}\right)\Delta\tilde{Q}_c - \sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb} = R_c(\tilde{Q}^n)\,. \tag{3.4}$$

However, it costs too much memory to store the left-hand side implicit Jacobian matrices. Therefore, we employ a LU-SGS scheme to solve (3.4), and it utilizes the most recent solution for the neighbouring cells,

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c}\right)\Delta\tilde{Q}_c^{(k+1)} = R_c(\tilde{Q}^n) + \sum_{nb\neq c}\frac{\partial R_c}{\partial Q_{nb}}\Delta\tilde{Q}_{nb}^*\,. \tag{3.5}$$

The Jacobian matrix

$$D = \left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c}\right), \tag{3.6}$$

is the element (or cell) matrix, which is not very big for 2nd to 3rd order SD schemes. For instance, D is [6x6] for the 3rd order SD wave equation. Neither do we want to store the matrices $\frac{\partial R_c}{\partial Q_{nb}}$ , thus (3.5) is further manipulated as follows

$$\begin{aligned}
R_c(\tilde{Q}^n) + \sum_{nb\neq c}\frac{\partial R_c}{\partial \tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^* &= R_c(\tilde{Q}_c^n, \{\tilde{Q}_{nb}^n\}) + \sum_{nb\neq c}\frac{\partial R_c}{\partial \tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^* \\
&\approx R_c(\tilde{Q}_c^n, \{\tilde{Q}_{nb}^*\}) \approx R_c(\tilde{Q}_c^*, \{\tilde{Q}_{nb}^*\}) - \frac{\partial R_c}{\partial \tilde{Q}_c}\Delta Q_c^* = R_c(\tilde{Q}^*) - \frac{\partial R_c}{\partial \tilde{Q}_c}\Delta Q_c^*.
\end{aligned} \tag{3.7}$$

Let $\Delta^* Q_c^{(k+1)} = \Delta\, Q_c^{(k+1)} - \Delta\, Q_c^*$. We can combine (3.5) and (3.7) together to obtain

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c}\right)\Delta^*\tilde{Q}_c^{(k+1)} = R_c(\tilde{Q}^*) - \frac{\Delta Q_c^*}{\Delta t}\,. \tag{3.8}$$

Equation (3.8) is then solved with a direct LU decomposition solver for an element and the solver is swept symmetrically forward and backward through all the computational grid elements. Once (3.8) 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 (3.8) can be neglected and leading to quicker convergence. Note that $\Delta Q_c^*$ 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, $\Delta Q_c^*$ 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 (3.8) as implicit normal approach.

If $\frac{\Delta Q_c^*}{\Delta t}$ term is dropped, the iterative solver is defined as implicit simplified approach. Some comparison between these approaches will be made in section 5.

## 4. p-Multigrid Method

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 [7, 9, 19, 30]. 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.

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.

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.

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

$$R_p(Q_p) = r_p \, , \tag{4.1}$$

and the rhs $r_p$ is zero for the highest level polynomial;

$$R_{p-1}(Q_{p-1}) = r_{p-1} \; ; \tag{4.2}$$

$$R_{p-2}(Q_{p-2}) = r_{p-2} \, . \tag{4.3}$$

We want to employ the implicit LU-SGS schemes as the smoothers for all three levels. For simplicity, the following steps summarise a standard two-grid scheme with a V cycle at p and p-1 levels.

- Improve $Q_p$ by application of a few times the smoother similar as equation (3.7)

$$\left[\frac{I}{\Delta t} - \frac{\partial R_c^p}{\partial Q_c^p}\right]^n \Delta^2 Q_c^p = R_p(Q_p^*) - r_p \tag{4.4}$$

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

$$Q_{p-1}^0 = I_p^{p-1}\left(Q_p\right) \tag{4.5}$$

- Compute the defect on the finest level

$$d_p = r_p - R_p\left(Q_p\right) = -R_p\left(Q_p\right) \tag{4.6}$$

- Compute the right hand side of equation (4.2) as

$$r_{p-1} = R_{p-1}\left(Q_{p-1}^0\right) + I_p^{p-1} d_p \tag{4.7}$$

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

$$\left[\frac{I}{\Delta t} - \frac{\partial R_c^{p-1}}{\partial Q_c^{p-1}}\right]^n \Delta^2 Q_c^{p-1} = R_{p-1}(Q_{p-1}^*) - r_{p-1} \tag{4.8}$$

- Correct the current solution on the finest level by

$$\tilde{Q}_p = Q_p + I_{p-1}^p(\overline{Q}_{p-1} - Q_{p-1}^0) \tag{4.9}$$

- Improve $\tilde{Q}_p$ by application of a few times the iterative smoother (the same as equation 4.4) to get $\overline{Q}_p$

We can call the above two-grid V cycle recursively for p-1 and p-2 levels just behind the smoother (equation 4.8) to accomplish a three-level V cycle. In fact, if it is called twice, a W cycle is constructed.

Note that only implicit LU-SGS smoothers are described in the above procedure for simplicity. In practice, we can replace any of the implicit smoothers (4.4 or 4.8) 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.

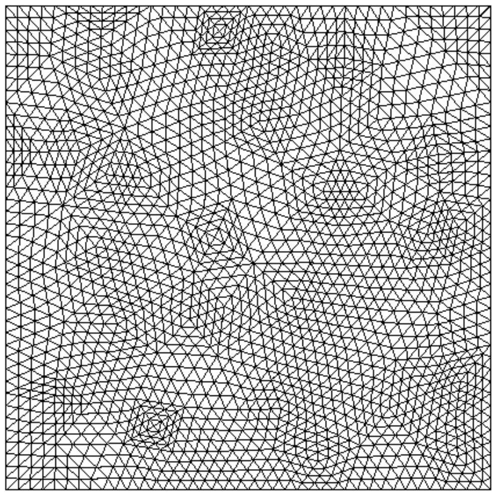
## 5. Numerical results and discussion

### 5.1 Validation using 2D linear scalar conservation law

We consider the linear straight wave over a square domain. It is also considered in [5] using the explicit scheme. The equation can be described as

$$\frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} + \frac{\partial q}{\partial y} = 0\,,\ (x, y) \in [0,1]^2 \text{ and } t > 0 \tag{5.1}$$

The computational unstructured grid with 40x40x2 elements is used as shown in figure 3(a). Analytical solutions are specified at the left and bottom boundaries. Figure 3(b) shows the converged solution obtained using the normal implicit scheme on the same grid. Figure 4 demonstrates that the converging speed of the simplified implicit scheme without $\frac{\Delta Q_c^*}{\Delta t}$ term in equation (3.8) is about seven 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.


(a)

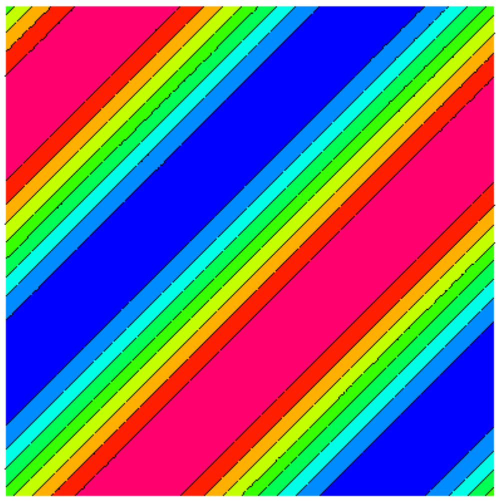

(b)

Figure 3: 2D Linear wave equation case. (a): Grid 40x40x2; (b): Contour Plot

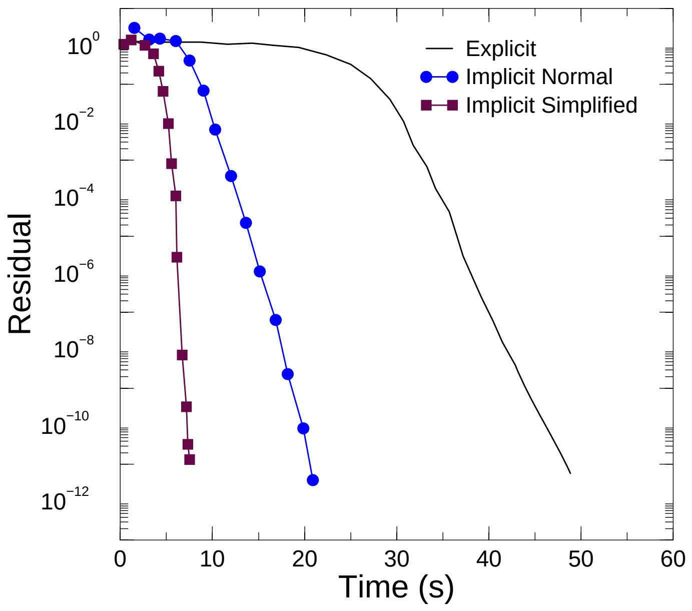


Figure 4. Comparison of convergence history with respect to time using explicit R-K and implicit LU-SGS schemes for the 2D linear wave case

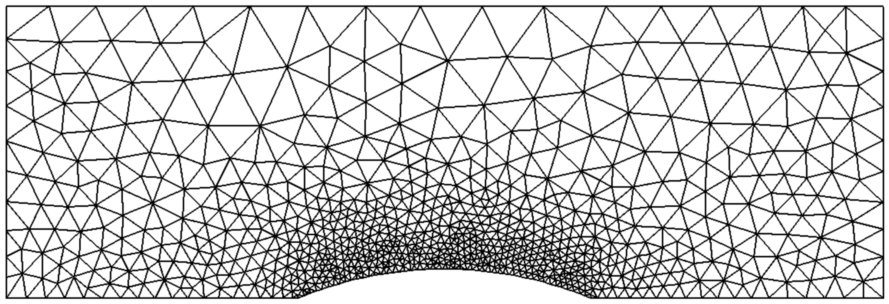
5.2 Results for the Euler equations

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 below $5\times10^{-4}$. The Roe flux is used for the flux vector computation at the element boundary interfaces. A quadratic curved boundary condition is adopted [27] for non-straight wall boundary surfaces.

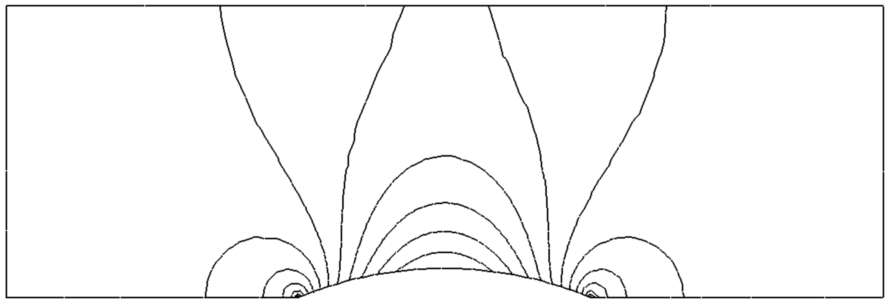
A. 2D subsonic flow over a bump

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 [19, 15]. 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 5(a). The circular surface of the bump needs a higher-order boundary treatment and a quadratic boundary as described in [27] is adopted.

Figure 5(b) shows the pressure contour obtained by the three-level p-Multigrid method. It is approximately identical to the pressure contour shown in [15]. The maximum CFL number used for all the implicit computations is around 8. Figure 6 shows the residual convergence history of the implicit schemes and the explicit scheme on a single grid is also shown. Driving normalized residuals to $1\times10^{-8}$, 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 of two order compared to the explicit scheme on a single grid. Note that all the Multigrid methods mentioned so far is based on V cycles for this particular case.



(a)



(b)

Figure 5. Subsonic flow over a bump confined in a channel. (a): Computational grid. (b): Computed pressure contours.

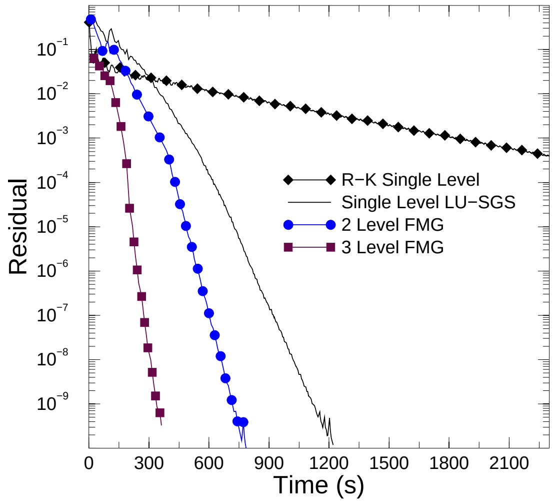


Figure 6. Comparison of convergence history with respect to time for the bump testing case using explicit R-K and implicit LU-SGS schemes.

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 7(a). However the total time consumed is lesser when the V cycles were employed. This is shown in fig 7(b). Note that, the number of iterative smoothers performed for V cycles is 1-1-20-1-1 for p2-p1-p0-p1-p2 levels respectively. A bigger number of total p1- and p0- level iterative smoothers is used in the W cycles, i.e., one, one and 16 iterations for p2, p1 and p0 levels respectively.

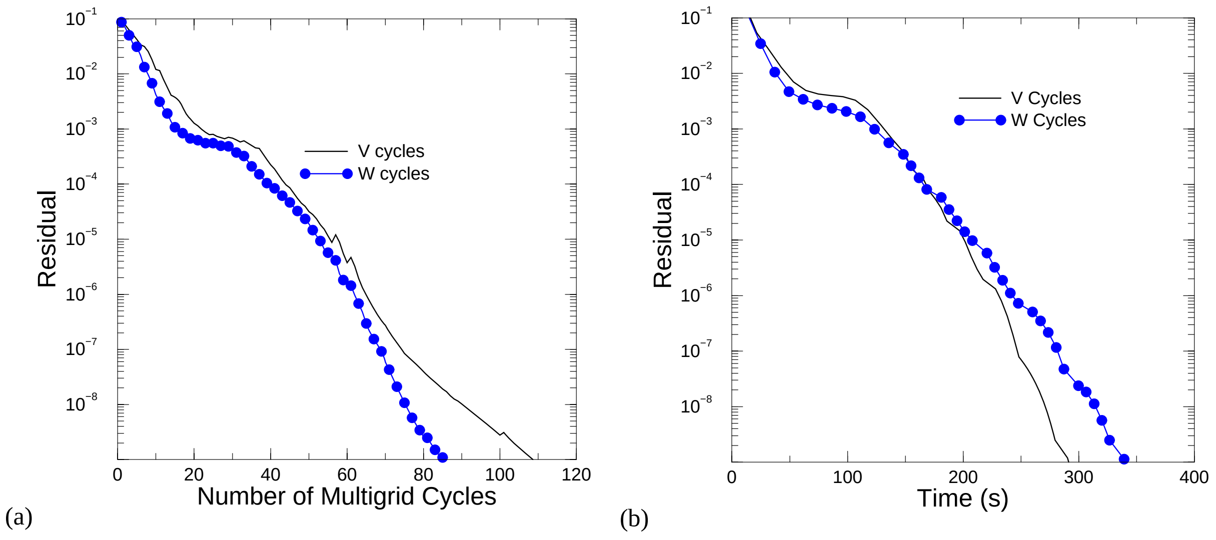


Figure 7. 2D subsonic flow over a bump case using implicit LU-SGS smoother. (a): Convergence history as a function of Multigrid cycles. (b): Convergence History as a function of time.

It is known that Full Multigrid scheme (starting from the coarsest grid level) is a good choice when the initial guess is very bad. Figure 8 shows the convergence history with respect to time for a FMG implemented using V cycles. The scheme starts the computation from the p0 level and the normalized residual drops nearly two orders. It takes about 50 iterations to realize the above situation. The solution is then extrapolated to the p1 level using the standard prolongation operator. This prolongation results in a jump, which can be seen in fig 8. A two-level scheme is used to drop the residual by an order. It takes about 25 iterations to realize the above. The solution is then extrapolated to the p2 level. As expected, a second jump occurs due to the extrapolation. It should be noted that the CPU time consumed by the above mentioned pre-processing computations on p0 and p1 levels is pretty short (70 seconds for this particular case). The three-level solver is now used to further drive the errors out of the domain. It can be seen that the time taken for the pre-processor takes only about a tenth of the total time. It is noted that all the subsequent figures for the Full Multigrid schemes (either implicit or explicit) show convergence histories directly from the p2 level because the CPU time spent on the pre-processing computations is short.

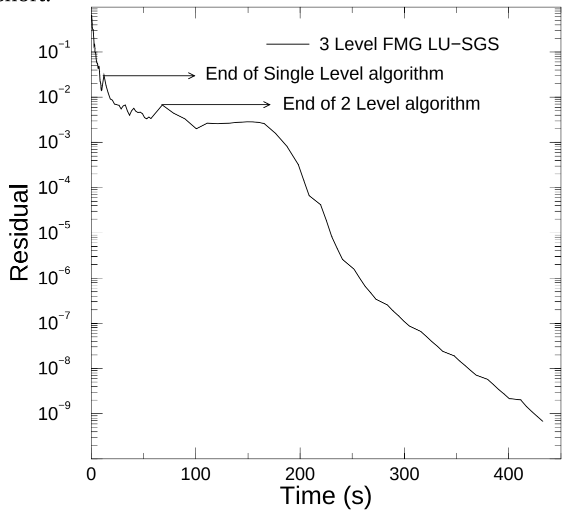


Figure 8. Plot of convergence history with respect to time, showing the time taken for the single, two and three levels of the multigrid for the bump case

All the above Multigrid calculations for the bump case are using implicit smoothers. If the explicit smoothers are used for all three p-levels (figure 9) in which we use one-one-six-one-one-iteration explicit smoothers at the p2-p1-p0-p1-p2 levels to form a standard V cycle, a speedup of 8 is attained over the explicit single level case. Note that the three-level implicit scheme is only around 3.5 times as fast as the implicit scheme on a single grid. As expected, the 3-level explicit scheme is slower than the mixed scheme (1 R-K + 2 LU-SGS). The curve with the second fastest convergence rate in figure 9 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 it is much faster than any of the above discussed methods.

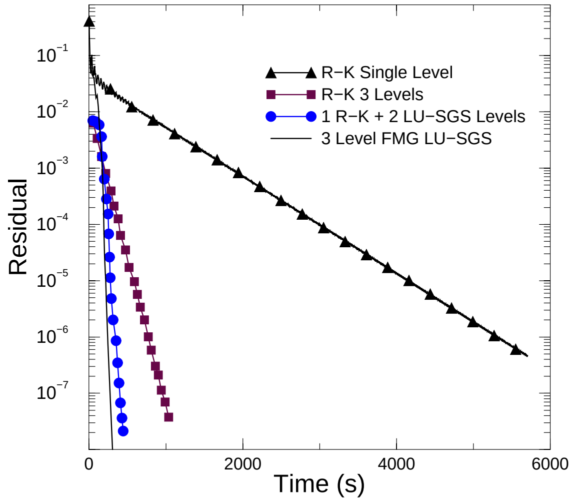


Figure 9. Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the bump testing case

B. Subsonic flow over a NACA0012 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 10. The outer boundary is 20 chords away from the airfoil centre. Figure 11 shows pressure contours obtained using two-level p-Multigrid scheme on p2 and p1 levels. The maximum CFL number used for the implicit computations is around 6.5. The explicit scheme limits the maximum CFL number to 0.06. From figure 12, we can see that the highest speedup factor obtained using the three-level p-Multigrid is again around 100 compared with the explicit scheme on the p2 grid. The three-level p-Multigrid method with implicit smoothers is about 4 times as fast as the single p2-level implicit method. The two-grid scheme using the implicit smoothers for p2 and p1 levels shows a fast convergence too. As also shown in figure 12, its speedup is around 70 compared to the single level explicit scheme.

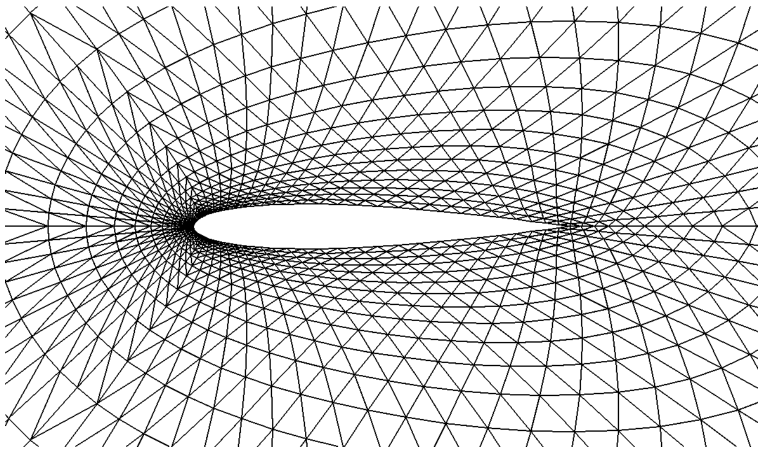


Figure 10: Grid(72*24*2) used for the subsonic flow over the NACA 0012 airfoil

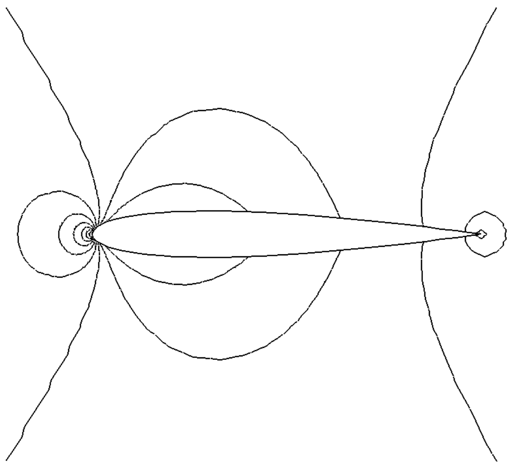


Figure 11: Pressure Contours obtained for the subsonic flow over the NACA 0012 airfoil

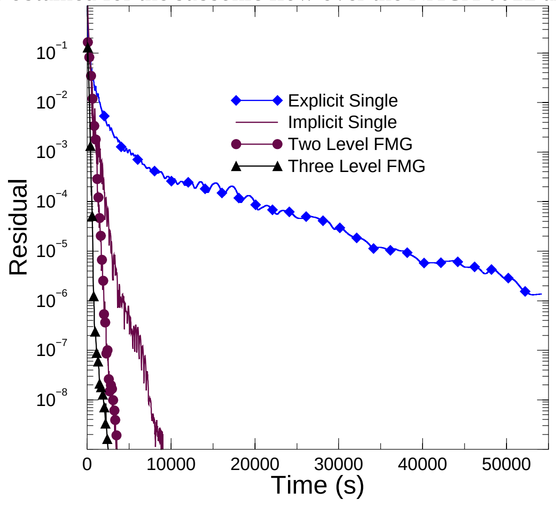


Figure 12: Comparison of convergence history with respect to time for the airfoil testing case using explicit R-K and implicit LU-SGS schemes

The effect of the explicit smoothers is also studied as shown in figure 13. 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 big V cycle. The second calculation is defined as a mixed three-level approach since 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 3-level approach is about 1.4 times as fast as the approach of 3-level R-K. The 3-level R-K approach is around 8 times as fast as the single level explicit R-K method to reach residual level of $1\times10^{-6}$. As expected, the 3-level p-Multigrid method using implicit smoothers is much faster than any of the above discussed schemes.

For the above computations performed for the airfoil case, the iteration numbers for the V cycles shall be pointed out as follows. For three-level p-Multigrid method using implicit smoothers, two iterations are sufficient for the p2 and p1 levels and as many as 20 iterations can be used on the p0 level during every V cycle. In short, 2-2-20-2-2 iterations of smoothers are employed on p2-p1-p0-p1-p2 levels. For the two-level p-Multigrid method, a smaller V-cycle is constructed using one relaxation iteration at the p2 level, two relaxation iterations at the p1 level and two more smoothing iterations at the p2 level to remove the prolongation noise.

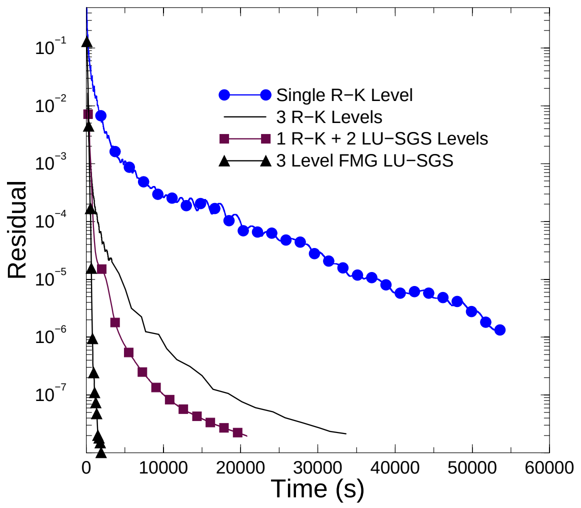


Figure 13: Comparison of convergence history with respect to time using combinations of explicit R-K and implicit LU-SGS schemes for the airfoil testing case

## 6. Conclusions

We have developed a p-multigrid spectral difference solver for the 2D Euler equations with a preconditioned LU-SGS smoother. We found that the simplified implicit scheme is very stable and produces a speedup around one order for the scalar equation compared to the explicit scheme. In fact, the speedup obtained by the implicit scheme with p = 2 is around 20-50 for 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 achieves very good stability and speedup for both 2D wave and 2D Euler equations. The p-Multigrid method with the implicit smoother 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 implicit LU-SGS. However, they can be used at the highest p-level for 3D problems to circumvent the problems associated with large storage of the cell Jacobian matrix in implicit schemes. Our calculations of 2D flows demonstrate 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. Nevertheless, a speedup factor around 8 can be achieved using explicit smoothers for all three p-levels. In the future, we are planning to extend the p-multigrid method with implicit LU-SGS smoothers to 2D and 3D compressible Navier-Stokes equations.

## Acknowledgements

The study was supported by AFOSR grant FA9550-06-1-0146 and partially supported by DOE grant DE-FG02-05ER25677. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR or DOE.

## References

1. M. Aftosmis, D. Gaitonde and T. S. Tavares, Behavior of linear reconstruction tecniques on unstructured meshes, AIAA Journal, 33 (1995), pp 2038-2049.
2. 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.
3. P. Batten, M.A. Leschziner, U.C. Goldberg, Average-state Jacobians and implicit methods for compressible viscous and turbulent flows, Journal of Computational Physics 137 (1997) 38–78.
4. R.F. Chen and Z.J. Wang, Fast, Block Lower-Upper Symmetric Gauss Seidel Scheme for Arbitrary Grids, AIAA Journal, 2000, vol. 38, no. 12: 2238-2245.
5. B. Cockburn, S. Hou and C.W. Shu, TVD Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation 55 (1990), pp. 545–581.

6. B. Cockburn and C.W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional system, Journal of Computational Physics, 141 (1998), pp. 199–224.
7. 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.
8. 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.
9. 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.
10. 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.
11. J. Heys, T.A. Manteufell, S.F. McCormick, L.N. Olson, Algebraic multigrid for higher-order finite elements, J. Comput. Phys. 204 (2005) 520–532.
12. A. Jameson, S. Yoon, Lower–upper implicit schemes with multiples grids for the Euler equations, AIAA Journal 25 (7) (1987) 929–935.
13. A. Jameson and D.A. Caughey, How many steps are required to solve the Euler equations of steady compressible flow: in search of a fast solution algorithm, 15th AIAA Computational Fluid Dynamics Conference, June 2001, Anaheim, CA.
14. 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.
15. 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.
16. 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.
17. 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.
18. H. Luo, D. Sharov, J.D. Baum, R. Lohner, A class of matrix-free implicit methods for compressible flows on unstructured grids, in: Proceedings of the First International Conference on Computational Fluid Dynamics, Kyoto, Japan, 10–14 July 2000.
19. Y. Maday, R. Munoz, Spectral element multigrid, Part 2: Theoretical justification, Tech. Rep. 88-73, ICASE, 1988.
20. G. May and A. Jameson, "A spectral difference method for the Euler and Navier-Stokes equations", AIAA paper No. 2006-304, 2006.
21. 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.
22. R. H. Ni, A multiple grid scheme for solving the Euler equations, AIAA Journal, 20 (1982), pp.1565-1571.
23. P. Rasetarinera, M.Y. Hussaini, An efficient implicit discontinuous spectral Galerkin method, Journal of Computational Physics 172 (2001) 718–738.
24. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J ournal of Computational Physics, 43 (1981), pp. 357-372.
25. E.M. Ronquist, A.T. Patera, Spectral element multigrid, I. Formulation and numerical results, Journal of Scientific Computing 2 (4) (1987) 389–406.
26. V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1, (1961), pp. 261-279.
27. Y. Sun, Z. J. Wang and Y. Liu, High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, Communication in Computational Physics, 2 (2007), pp. 310-333.
28. E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations, Journal of Computational Physics, 72 (1987), pp. 277-298.
29. Z. J. Wang and Y. Liu, Extension of the spectral volume method to high-order boundary representation, Journal of Computational Physics, 211 (2006), pp. 154-178.
30. Z. J. Wang and Y. Liu, The Spectral Difference Method for the 2D Euler Equations on unstructured grids, AIAA paper No. 2005-5112, 2005.
31. 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.
32. T.C. Warburton, I. Lomtev, Y. Du, S.J. Sherwin, G.E. Karniadakis, Galerkin and discontinuous Galerkin spectral/hp methods, Comput. Methods Appl. Mech. Eng. 175 (1999) 343–359.

33. Yoon S and Jameson A. Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations, AIAA Journal, 1988, Vol. 26, pp. 1025-1026.