46th AIAA Aerospace Sciences Meeting and Exhibit
7 - 10 January 2008, Reno, Nevada

AIAA 2008-779

High-Order Adaptive Quadrature-Free Spectral Volume Method on Unstructured Grids

Rob Harris* and Z. J. Wang†
Department of Aerospace Engineering, Iowa State University, Ames, IA 50011

The high-order quadrature-free spectral volume (SV) method is extended to handle local adaptive hp-refinement (grid and order refinement). Efficient edge-based adaptation utilizing a binary tree search algorithm is employed. An adaptation criteria is selected which focuses computational effort near discontinuities, and effectively reduces the physical area of the domain necessitating data limiting for stability. This makes the method very well suited for capturing and preserving discontinuities with high resolution. Both h- and p- refinement are presented in a general framework where it is possible to perform either or both on any grid cell at any time. Several well-known inviscid flow test cases, subjected to various levels of adaptation, are utilized to demonstrate the effectiveness of the method.

I. Introduction

The spectral volume (SV) method is a recently developed finite volume method for hyperbolic conservation laws on unstructured grids.[1-7] The SV method belongs to a general class of Godunov-type finite volume method[8-9], which has been under development for several decades, and is considered to be the current state-of-the-art for the numerical solution of hyperbolic conservation laws. For a more detailed review of the literature on the Godunov-type method, refer to Wang[1], and the references therein. Many of the most popular numerical methods, such as the k-exact finite volume[10-11], the essentially non-oscillatory (ENO)[12-13], and weighted ENO[14] methods are also Godunov-type methods. A thorough review and comparison of these methods can be found in Wang.[15] The SV method is also closely related to the discontinuous Galerkin (DG)[16-20] method, a popular finite-element method for conservation laws. The DG method has also been extended for use with local adaptive hp-refinement by Remacle et al.[21] and Flaherty et al.[22] Both the SV and DG methods employ multiple degrees of freedom within a single element, but the SV method avoids the volume integral required in the DG method. Each simplex in the SV method utilizes a "structured" set of sub-cells to support a polynomial reconstruction for the conserved variables, and a nodal set to support a polynomial reconstruction for the flux vector. For a more thorough comparison of the SV and DG methods, refer to Wang.[1,15]

The accuracy of the SV method is heavily dependent on both the grid and the degree of polynomial interpolation. Adaptive hp-refinement is used to locally refine or coarsen both the grid and solution polynomial as the flow develops. It is often desirable to refine the grid near discontinuities, and to refine the polynomial near fine smooth features, to avoid the high computational cost of global refinement. This is the focus of the current work.The paper is organized as follows. In Section 2, we review the basic formulation of the quadrature-free SV method. After that, the adaptive hp-refinement procedure is described in detail in Section 3. Numerical results for several inviscid flow test cases, including flow in the presence of strong shock waves and flow over curved bodies, are presented in Section 4. Finally, conclusions and some possibilities for future work are summarized in Section 5.

II. Review of the Quadrature-Free Spectral Volume Method

Consider the multidimensional conservation law

$$\frac{\partial Q}{\partial t}+\frac{\partial f(Q)}{\partial x}+\frac{\partial g(Q)}{\partial y}+\frac{\partial h(Q)}{\partial z}=0, \tag{1a}$$

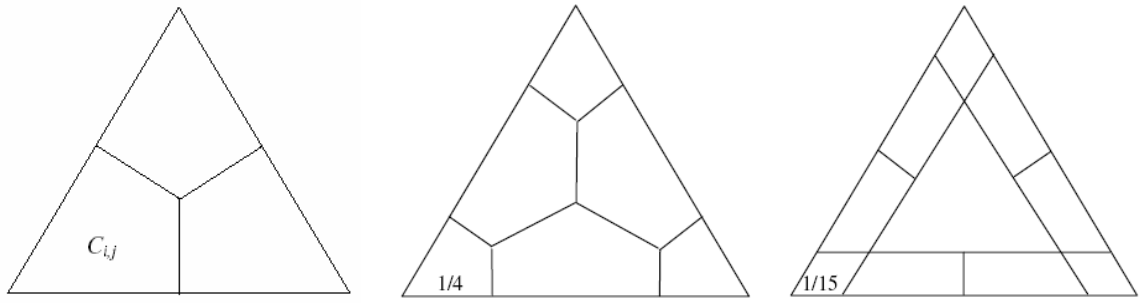
on domain $\Omega \times [0,T]$ and $\Omega \subset R^3$ with the initial condition

$$Q(x,y,z,0)=Q_0(x,y,z), \tag{1b}$$

and appropriate boundary conditions on $\partial\Omega$. In (1), x,y, and z are the Cartesian coordinates and $(x,y,z)\in \Omega$, $t \in [0,T]$ denotes time, Q is the vector of conserved variables, and f, g and h are the fluxes in the x, y and z directions, respectively. Domain Ω is discretized into I nonoverlapping triangular (2D), or tetrahedral (3D) cells. In the SV

*Graduate Research Assistant, 0237 Howe Hall, rharris@iastate.edu, AIAA Member.
†Associate Professor of Aerospace Engineering, 2271 Howe Hall, zjw@iastate.edu, Associate Fellow of AIAA.

method, the simplex grid cells are called SVs, denoted S_i, which are further partitioned into CVs, denoted $C_{i,j}$, which depend on the degree of the polynomial reconstruction. Examples of partitions supporting linear, quadratic and cubic reconstructions are shown in Figure 1.



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

Figure 1. Partitions of a triangular SV supporting linear, quadratic and cubic data reconstructions, shown in (a), (b) and (c), respectively.

Volume-averaged conserved variables on the CVs are then used to reconstruct a high-order polynomial inside the SV. To represent the solution as a polynomial of degree m, we need N pieces of independent information, or degrees of freedom (DOFs). Where N is calculated as follows:

$$N = \frac{(m+1)(m+2)\cdots(m+d)}{d!}, \tag{2}$$

where d is the spatial dimension of the problem. The DOFs in the SV method are the volume-averaged conserved variables at the N CVs. Define the CV-averaged conserved variable for $C_{i,j}$ as

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q dV, \quad j=1,\ldots,N, \quad i=1,\ldots,I, \tag{3}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. Given the CV-averaged conserved variables for all CVs in S_i, a polynomial $p_i(x,y,z) \in P^m$ (the space of polynomials of at most degree m) can be reconstructed such that it is a $(m+1)^{\text{th}}$ order accurate approximation to $Q(x,y,z)$ inside S_i.

$$p_i(x,y,z) = Q(x,y,z) + O\left(h^{m+1}\right), \quad (x,y,z) \in S_i, \quad i=1,\ldots,I, \tag{4}$$

where h is the maximum edge length of all the CVs. This reconstruction can be solved analytically by satisfying the following conditions:

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} p_i(x,y,z) dV = \overline{Q}_{i,j}, \quad j=1,\ldots..,N. \tag{5}$$

This polynomial $p_i(x,y,z)$ is the $(m+1)^{th}$ order approximation we are looking for as long as the solution $Q(x,y,z)$ is smooth in the region covered by S_i. The reconstruction is expressed more conveniently as

$$p_i(x,y,z) = \sum_{j=1}^{N} L_j(x,y,z) \overline{Q}_{i,j}, \tag{6}$$

where $L_j(x,y,z) \in P^m$ are the shape functions which satisfy

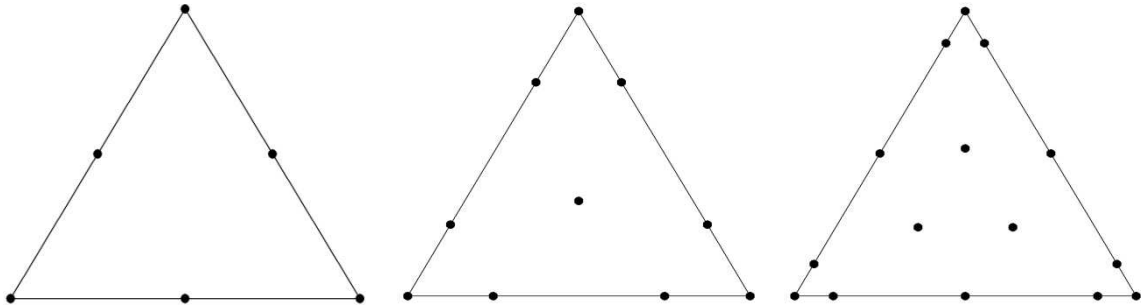
$$\frac{1}{V_{i,j}} \int_{C_{i,j}} L_n(x,y,z) dV = \delta_{jn}. \tag{7}$$

Integrating (1) in $C_{i,j}$, we obtain

$$\frac{d\overline{Q}_{i,j}}{dt} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} (F \cdot \hat{n}) dA = 0, \qquad j=1,\ldots,N, \qquad i=1,\ldots,I, \tag{8}$$

where $F=(f,g,h)$, A_r represents the r^{th} face of $C_{i,j}$, $\hat{n}$ is the outward unit normal vector of A_r, and K is the number of faces in $C_{i,j}$. More details of this, including representative plots of the shape functions can be found in Wang and Liu.[2]

A nodal set, such as those shown in Figure 2, is selected from Hesthaven[23] and used to support a degree m+1 polynomial reconstruction for the flux vector.



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

Figure 2. Nodal sets in a triangular SV supporting quadratic, cubic and quartic data reconstructions for the flux vector, shown in (a), (b) and (c), respectively.

The flux vector F can be computed at any point (x, y, z) by the following

$$F(x,y,z) = \sum_{i=1}^{N_S} M_i(x,y,z) F_i \ , \tag{9}$$

where N_s is the number of nodes in the nodal set, calculated from (2), F_i is the flux vector evaluated at node i , and $M_i(x,y,z)$ are the shape functions defined by the nodal set which satisfy:

$$M_n(x_j, y_j, z_j) = \delta_{jn}. \tag{10}$$

Some representative examples of the shape functions are shown in Harris et al.[7]

The average of (9) over a particular face is given by

$$\overline{F} = \sum_{i=1}^{N_S} \overline{M}_i F_i \ , \tag{11}$$

where $\overline{M}_i$ are the face-averaged node-based shape functions for that face evaluated in the standard element. This can be done either analytically using Mathematica[24] or the like, or numerically using Gauss quadrature formulas.

The face integral in (8) is then given as

$$\int_{A_r} (F \cdot \hat{n}) dA = A_r \overline{F}_n \ , \tag{12}$$

where $\overline{F}_n$ is the dot product of (11) with $\hat{n}$. This expression is exact for internal faces. For faces on SV boundaries, we use

$$\int_{A_r} (F \cdot \hat{n}) dA \cong \frac{A_r}{2} \left[\overline{F}_{n,L} + \overline{F}_{n,R} - \alpha_c \left(\overline{Q}_R - \overline{Q}_L \right) \right], \tag{13}$$

where $\overline{F}_{n,L}$ and $\overline{F}_{n,R}$ denote the face-averaged normal component of the flux vector due to the SV to the left and right of the interface, respectively, and α_c is taken as the maximum absolute eigenvalue as in the Rusanov flux[25], or the dissipation matrix as in the Roe flux[26], which is evaluated at the face center. $\overline{Q}_R$ and $\overline{Q}_L$ are the face-averaged conserved variables due to the SV to the right and left of the interface, respectively.

IV. Adaptive hp-Refinement

Local adaptive grid refinement is used to focus computational effort near discontinuities and fine smooth features to reduce the overall computational effort in the entire domain. h-refinement involves modification of cell

sizes while p-refinement involves modification of polynomial orders. We wish to utilize either or both on-the-fly as the flow develops. Both h- and p-refinement are carried out using only local operations to maximize the efficiency and accuracy of the procedure.

1. H-refinement

In this study, since we are only dealing with triangular SVs, h-refinement is basically a matter of grid regeneration, with no required modification of the solver itself. An efficient hierarchical edge-based adaptation algorithm is employed, which allows the grid to be adapted any number of levels from the base (coarsest) grid at any time. Let ε_i be an error indicator for edge i, and let ε_{max} be the maximum allowable value of ε_i. If for any edge i, $\varepsilon_i > \alpha\varepsilon_{max}$, then edge i is split into two edges, otherwise edge i is maintained. Here $\alpha>0$ is a user specified constant. This procedure begins with edges in the base grid (root edges), and continues until the maximum number of adaptation levels is reached. The error indicator ε_i is always computed using the solution from the previously adapted (finest) grid. When the above procedure completes, new SVs are added to the grid as a result of the split edges. There are essentially four different situations that can occur when the grid is adapted, as shown in Figure 3.

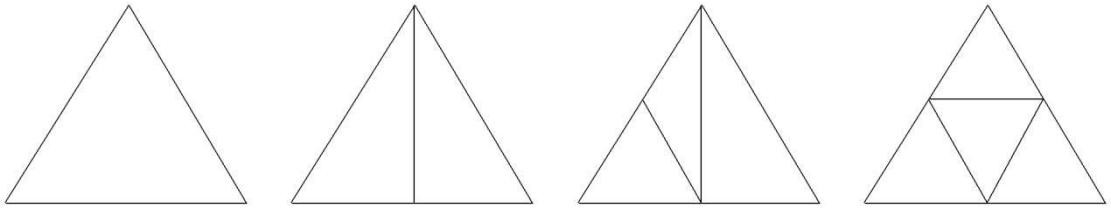


Figure 3. Four situations that can occur when a SV is refined; (a) No edges are split so the SV is unchanged; (b) Two new SVs are generated due to one split edge; (c) Three new SVs are generated due to two split edges; (d) Four new SVs are generated due to three split edges.

For each SV in the grid, the difference in adaptation level for that SV's edges is allowed to be no greater than one. This is done to ensure that all grids are comparable in quality to the base grid. When the creation of new SVs is complete, new cell-averages are then computed using

$$\overline{Q}_{i,j} = \sum_{k=1}^{N_S} \overline{M}_{j,k} Q_{i,k} \ , \tag{14}$$

where $\overline{M}_{j,k}$ are the node-based shape functions for node k averaged over CV j, and $Q_{i,k}$ are the conserved variables evaluated at node k of SV i. If node k exists within a SV in the previously adapted grid, then $Q_{i,k}$ are obtained from (6) using CV-averaged solutions $\overline{Q}_{i,j}$ from that SV. Otherwise, if node k exists at the junction between two or more SVs in the previously adapted grid, then $Q_{i,k}$ are obtained from an average of (6) among all SVs which have the physical location of node k in common. The above interpolation gives rise to an inherent loss of precision associated with coarsening of the solution, which is an unavoidable consequence of the h-refinement procedure.

The error indicator ε_i is computed using

$$\varepsilon_i = |\Delta\psi_i| A_i^u \ , \tag{15}$$

where $\Delta\psi_i$ is the difference of ψ (a variable of interest) between the two endpoints of edge i, A_i is the area of edge i, and $u>0$ is a user specified parameter. For all cases considered here, $u=1/2$ and ε_{max} is taken to be the L_2 norm of ε over all edges.

2. P-refinement

p-refinement, or order refinement, allows for a distribution of SVs where the degree of the polynomial reconstruction may vary from one SV to another. Unlike h-refinement, p-refinement does require significant modification of the solver itself. Among other things, the terms N and N_s in (6) and (9) are no longer constant, but

depend on the level of p-refinement of the current cell. In addition, computation of the face-averaged terms in (13) is not as straightforward as before. For example, consider the case where a linear SV is adjacent to a quadratic SV, as shown in Figure 4.

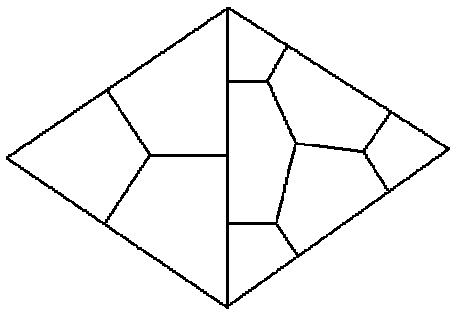


Figure 4. Two adjacent SVs with p-refinement levels differing by one. The left SV contains a linear partition, and the right SV contains a quadratic partition.

Here, the face-averaged shape functions for CV faces on SV boundaries must be computed in parts. For the corner CVs in the quadratic partition (right), the face-averaged shape functions are computed as usual, but the face-averaged shape function for the side CV must be computed in two parts to coincide with the intersection of that CV face with the face of the adjacent CV in the linear SV. With the face-averaged shape functions computed in this manner, the face-averaged terms in (13) can be readily computed.

Let the error indicator ε_i for edge i be the same as is defined in (15), and ε_{max} is again taken to be the L_2 norm of ε over all edges. If for any edge i, $\varepsilon_i > \beta\varepsilon_{max}$, then the degree of polynomial reconstruction, for the cells adjacent to edge i, is increased by 1. Similarly, if $\varepsilon_i < \gamma\varepsilon_{max}$, then the degree of polynomial reconstruction, for the cells adjacent to edge i, is decreased by 1. Here $\gamma,\beta>0$ are user specified constants

3. Hp-refinement

For simplicity, simultaneous h- and p-refinement are carried out in a decoupled manner. H-refinement is first performed to generate a new grid, and p-refinement is then performed to increase or decrease the degree of the polynomial reconstruction for each SV in the new grid. As new SVs are created as a consequence of h-refinement, the polynomial degree is set to minimum (1 in this case), and may not be increased as a result of p-refinement. This is a safeguard to ensure that the lowest possible degree polynomial is used near very high gradient regions. Such a measure should minimize oscillations due to extreme flow phenomena such as shock waves.

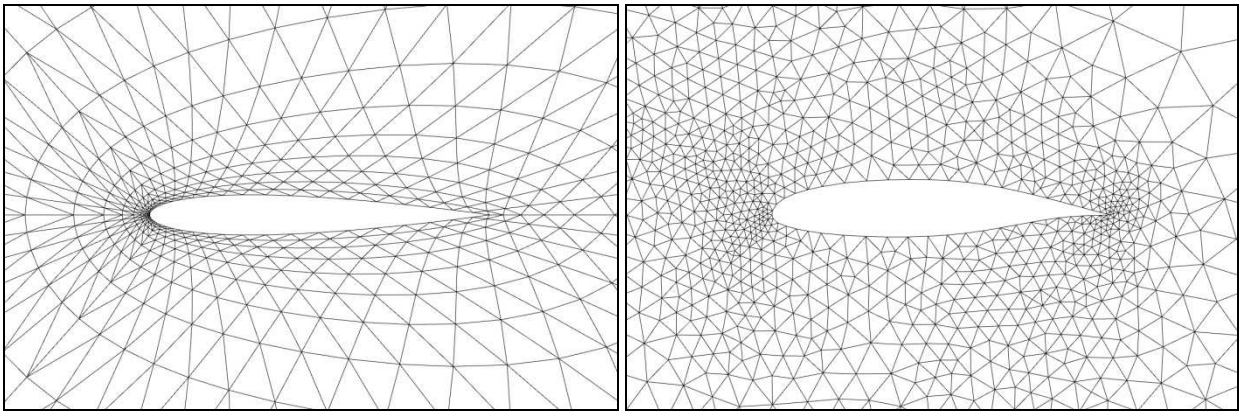
III. Numerical Tests

In this section, the SV method with local adaptive hp-refinement is evaluated for the 2D Euler equations. Several well known inviscid flow test cases are utilized to demonstrate the effectiveness of local hp-refinement. In all cases involving shock waves, the TVD limiter presented in Harris et al.[7] is employed to maintain a stable numerical scheme. In all cases involving curved-wall boundaries, the approach of Krivodonova and Berger[27] is utilized to maintain low computational cost. This approach was successfully implemented for the SV method in Harris et al.[7] All of the following cases employ the Rusanov[25] flux, and for time integration we use either the 2nd or 3rd order Strong Stability-Preserving[28] (SSP) Runge-Kutta scheme.

1. Subsonic flow over NACA 0012 and NASA GA(W)-1 airfoils

As a demonstration of the p-refinement technique, subsonic flow at Mach=0.4, and angle of attack of 5^o around both a NACA 0012 and a NASA GA(W)-1 airfoil is considered. The grid used for the NACA 0012 case is semi-

structured, while the grid for the NASA GA(W)-1 case is fully unstructured. Both Grids are shown in Figure 5. The outer boundary is 20 chord lengths away from the center of the airfoil in both cases.

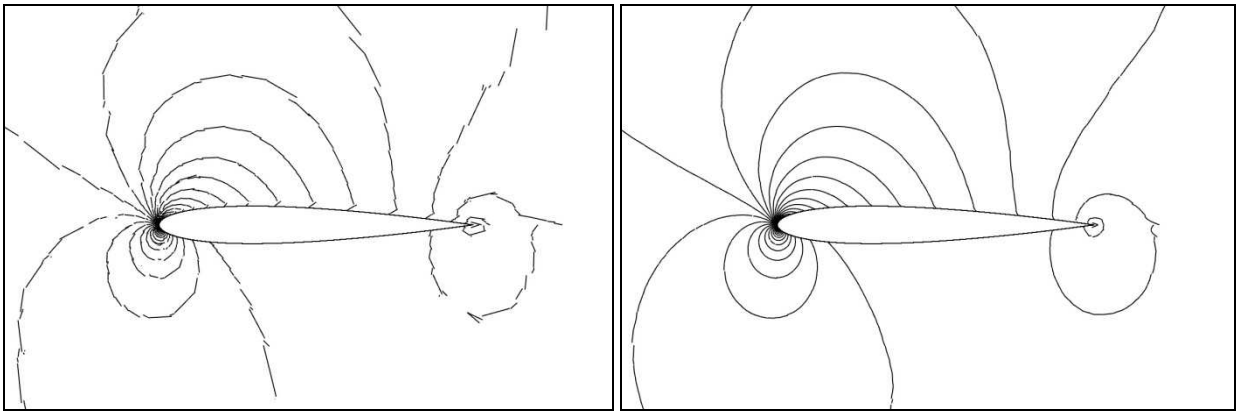


(a) **(b)**

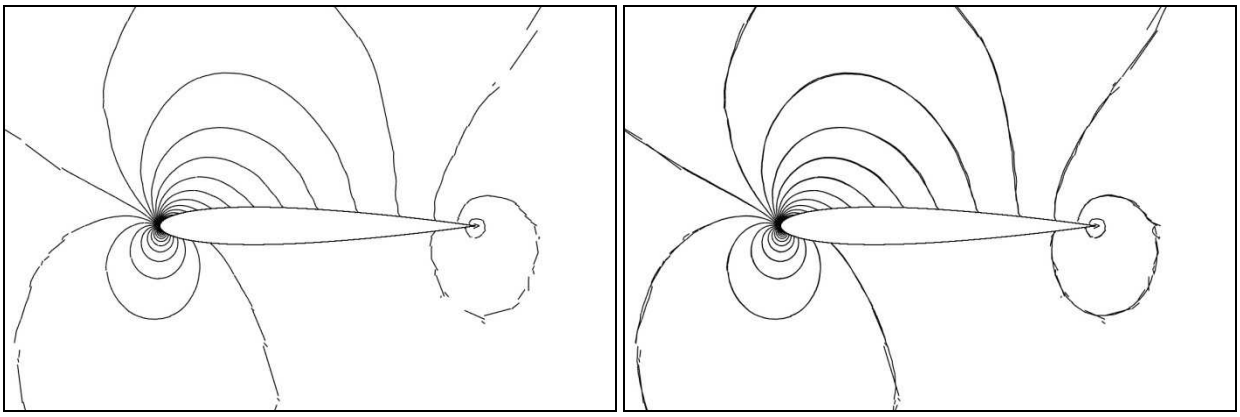
Figure 5. Grids for subsonic flow over airfoils; (a) semi-structured grid for NACA 0012 airfoil (48x16x2 triangles); (b) irregular grid for NASA GA(W)-1 airfoil (2,722 triangles).

For this case, the SVs near the farfield are orders-of-magnitude larger than the SVs near the airfoil surface. Thus, to prohibit adaptation near the farfield, it is necessary to remove the area weighting in (15) for this case. Also, ψ in (15) is taken to be the Mach number.

As a test of the p-refinement technique, a converged solution from a 2nd order simulation is subjected to 1 level of p-refinement and run until convergence. This case will be subsequently denoted as the 2-3 case. Thus, the resulting solution will contain some 2nd order SVs and some 3rd order SVs. Mach contours for this simulation, as well as uniform 2nd and 3rd order simulations for comparison, are shown in Figure 6. Figure 6d shows Mach contours for the 2-3 case and for a uniform 3rd order case for comparison. It is evident that Mach contours for the 2-3 case agree reasonably well with the 3rd order contours, and the large errors present near the airfoil in the 2nd order case are eliminated in the 2-3 case.



(a) **(b)**



(c) **(d)**

Figure 6. Contours of Mach number for subsonic flow over a NACA 0012 airfoil; (a) 2nd order (4,608 DOFs); (b) 3rd order (9,216 DOFs); (c) 1 level of p-adaptation starting from the converged 2nd order solution shown in (a) (6,519 DOFs); (d) The 1 level case shown with the 3rd order case to illustrate differences.

The convergence history for this case is shown in Figure 7. It is apparent that the 2-3 case costs slightly more than the 2nd order case in terms of required time steps, but it costs much less than the 3rd order case. This is encouraging, as the 2-3 case agrees with the 3rd order case extremely well at the airfoil surface (which is where a lift/drag calculation would take place), for significantly less computational cost than that required for a full 3rd order simulation. Figure 8 clarifies which SVs are increased to 3rd order for the 2-3 case. It is clear that the majority of SVs in the domain are still 2nd order, and 3rd order SVs are only used in regions of largest change in Mach number.

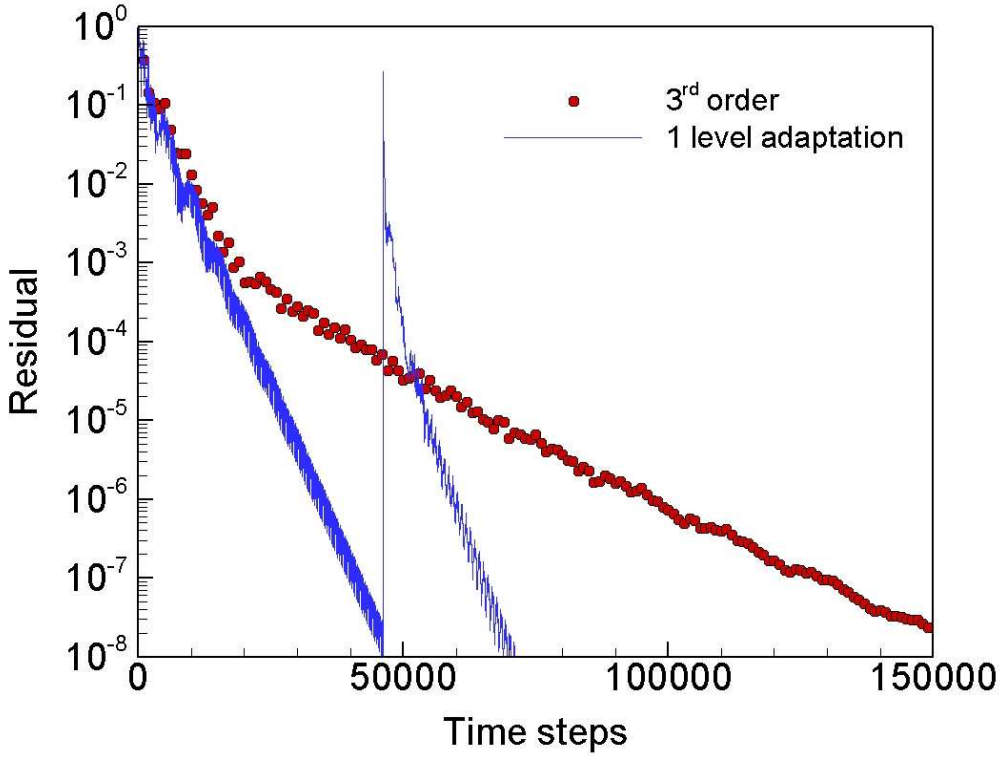
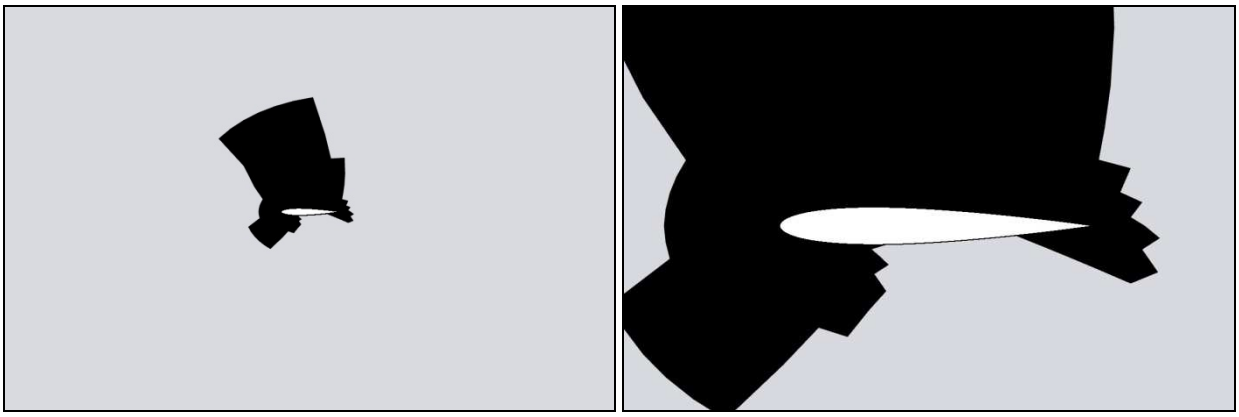


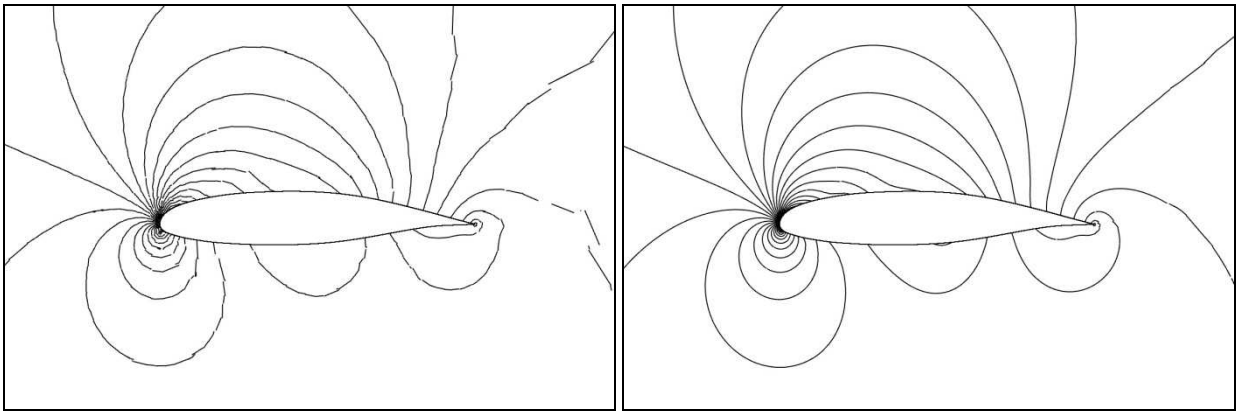
Figure 7. Convergence history for subsonic flow over NACA 0012 airfoil. The red circles are for a uniform 3rd order case, while the blue line is adapted 1 level from a converged 2nd order solution (i.e. the portion below around 50,000 time steps is the 2nd order scheme by itself, and after that is the 1 level adapted case with some 2nd order SVs and some 3rd order SVs).



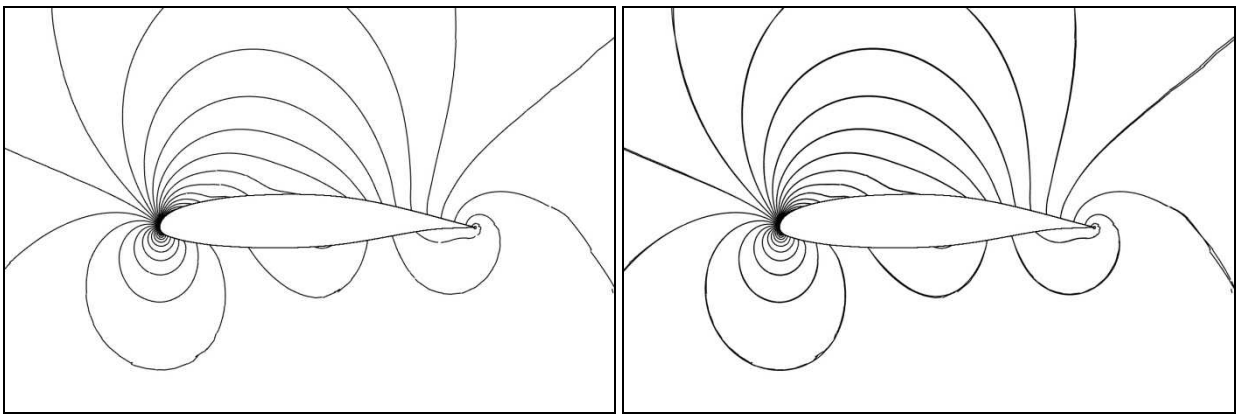
(a) **(b)**

Figure 8. Schematics showing the status of p-refinement for subsonic flow over NACA 0012 after refining from a converged 2nd order solution. Black regions represent 3rd order SVs, while gray regions represent 2nd order SVs; (a) zoomed-out view; (b) zoomed-in view.

The above study is repeated for the NASA GA(W)-1 airfoil shown in Figure 5b. Figure 9 shows Mach contours for this case, and Figures 10 and 11 show the convergence history and refinement status, respectively. This case uses a finer grid, so the results are not quite as discernible as with the NACA 0012 case, but none-the-less, a smoother and more accurate solution is attained for a slightly higher cost than a 2nd order simulation.



(a) **(b)**



(c) **(d)**

Figure 9. Contours of Mach number for subsonic flow over a NASA GA(W)-1 airfoil; (a) 2nd order (8,166 DOFs); (b) 3rd order (16,332 DOFs); (c) 1 level of p-adaptation starting from the converged 2nd order solution shown in (a) (11,958 DOFs); (d) The 1 level case shown with the 3rd order case to illustrate differences.

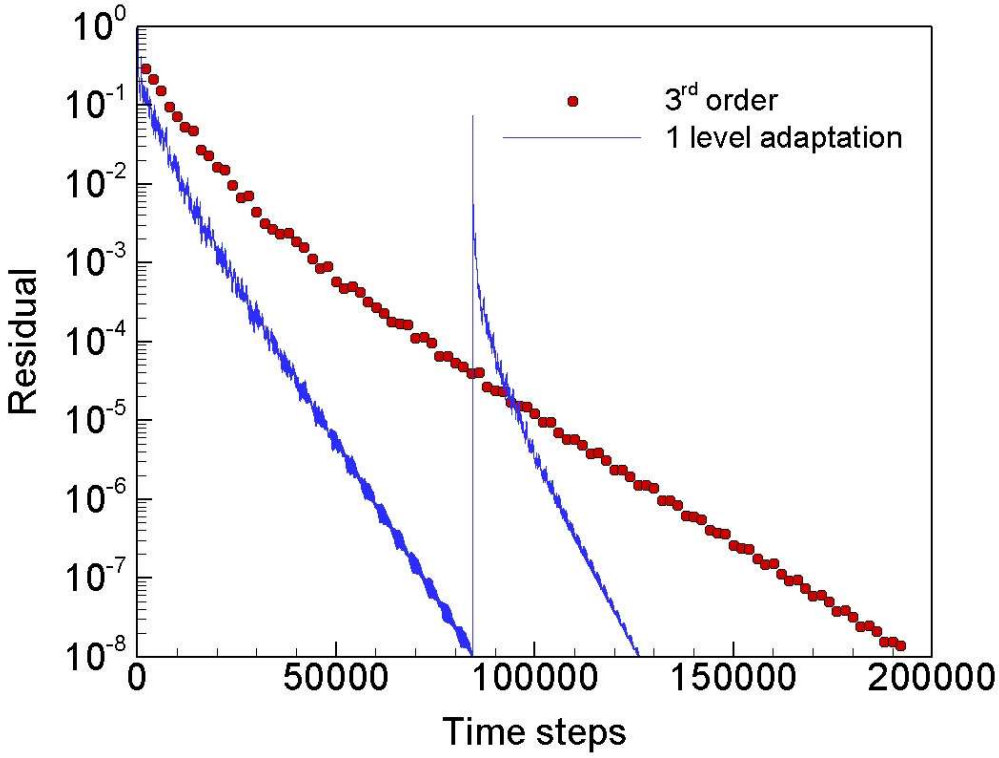
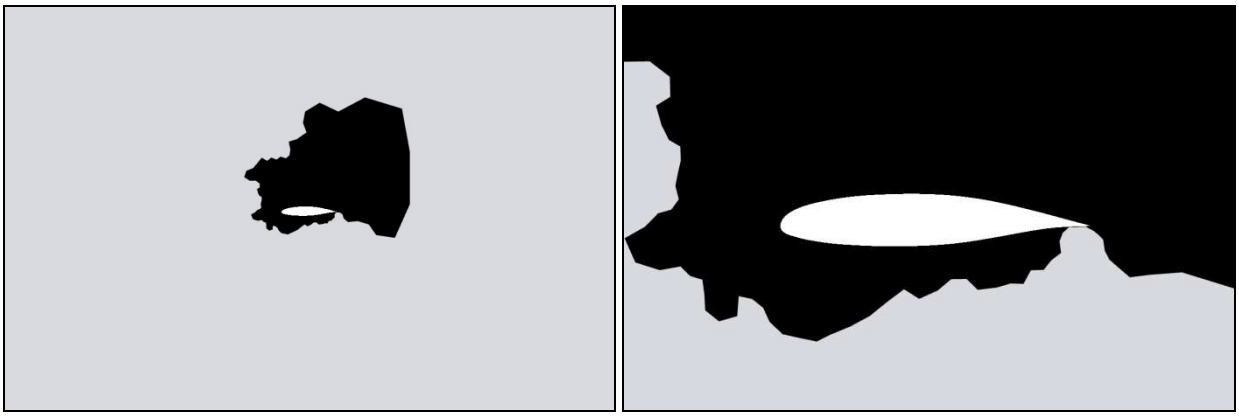


Figure 10. Convergence history for subsonic flow over a NASA GA(W)-1 airfoil. The red circles are for a uniform 3rd order case, while the blue line is adapted 1 level from a converged 2nd order solution (i.e. the portion below around 50,000 time steps is the 2nd order scheme by itself, and after that is the 1 level adapted case with some 2nd order SVs and some 3rd order SVs).



(a) **(b)**

Figure 11. Schematics showing the status of p-refinement for subsonic flow over a NASA GA(W)-1 airfoil after refining from a converged 2nd order solution. Black regions represent 3rd order SVs, while gray regions represent 2nd order SVs; (a) zoomed-out view; (b) zoomed-in view.

2.Transonic flow over a NACA 0012 airfoil

As a demonstration of the h-refinement technique, transonic flow at Mach=0.9, and angle of attack of 1^o around a NACA 0012 airfoil is considered. The base grid used for this simulation is the same as that used in the previous section. Here the error indicator (15) is computed based on density and total energy, and again the area weighting is removed to avoid unnecessary refinement in the farfield. This case involves shock waves on both the upper and lower surface of the airfoil, so the aforementioned TVD limiter is utilized to maintain stability. A converged 2^{nd} order solution is again taken as the initial condition, and grid is then re-adapted 3 times and then frozen for the remainder of the simulation.

Figures 12 and 13 show the computational grids and Mach contours, respectively, for 1, 2, 3 and 4 levels of adaptive h-refinement. It is evident that without refinement, the shock waves are smeared over several grid cells and the solution is of low quality. As the adaptation level is increased, the grid density in the vicinity of both shock waves and expansions is increased markedly. This produces a much higher quality solution with more precisely captured and finely resolved shock waves.

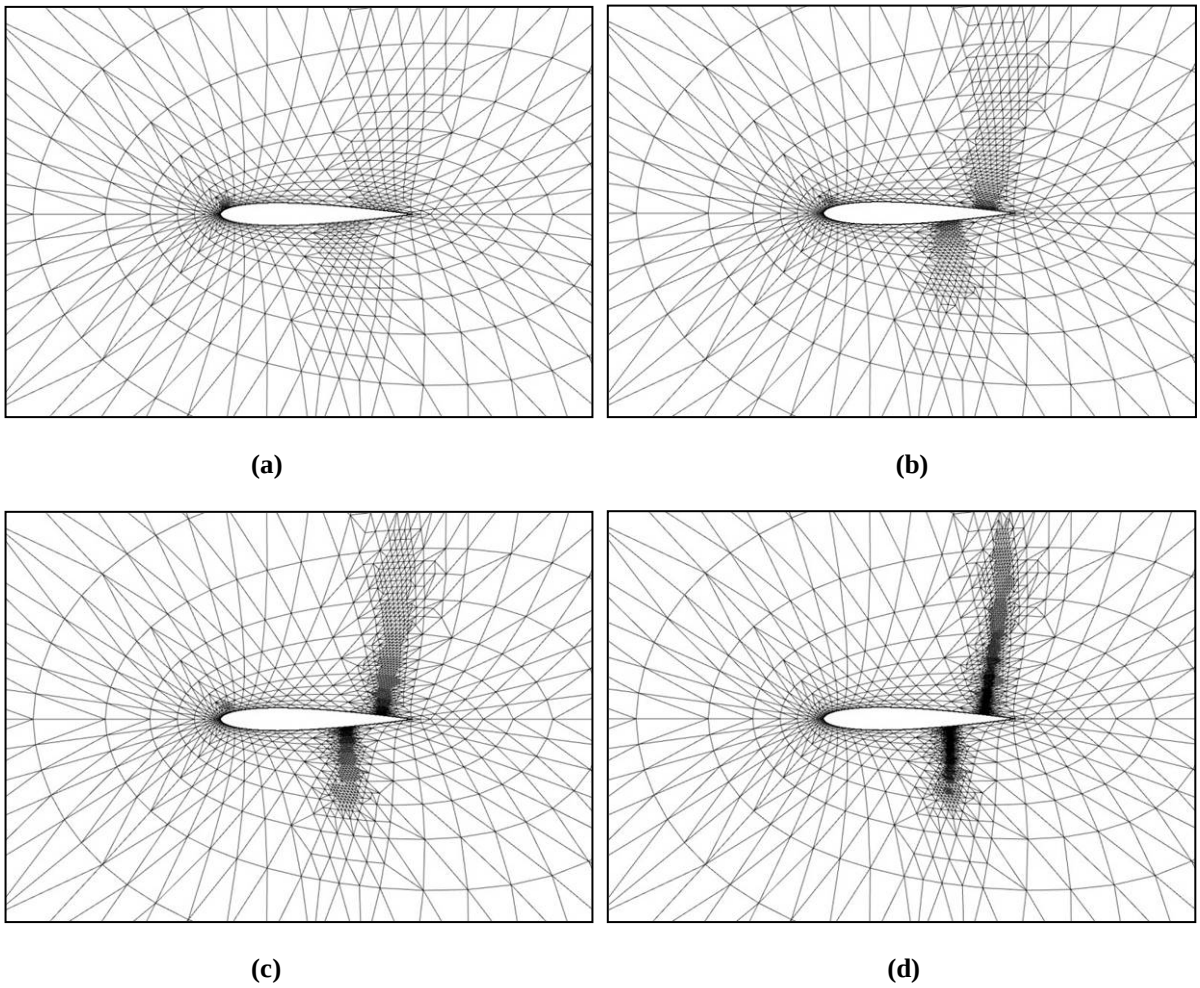


Figure 12. Grids for 2^{nd} order solution of transonic flow over NACA 0012 airfoil with adaptive h-refinement; (a) 1 level (3,349 triangles); (b) 2 levels (9,337 triangles); (c) 3 levels (30,498 triangles); (d) 4 levels (92,551 triangles); A converged 2^{nd} order solution on the base grid (Figure 5a) is used as the initial condition for all cases. The grid is re-adapted 3 times (once every 100 time steps for the first 300 time steps), and then frozen for the remainder of the simulation.

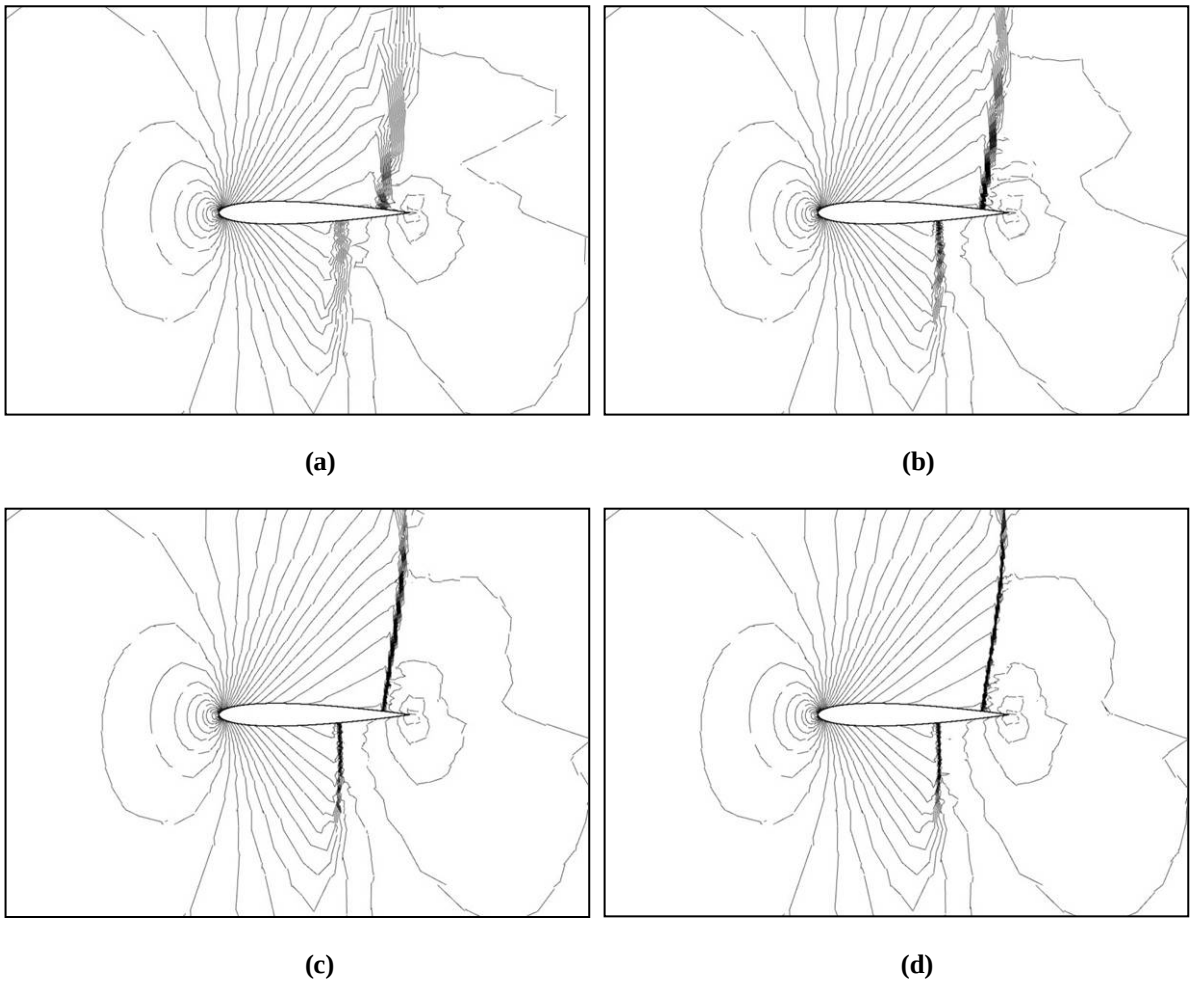


Figure 13. Mach contours for 2nd order solution of transonic flow over NACA 0012 airfoil with adaptive h-refinement; (a) 1 level (10,047 DOFs); (b) 2 levels (28,011 DOFs); (c) 3 levels (91,494 DOFs); (d) 4 levels (277,653 DOFs); A converged 2nd order solution on the base grid (Figure 5a) is used as the initial condition for all cases.

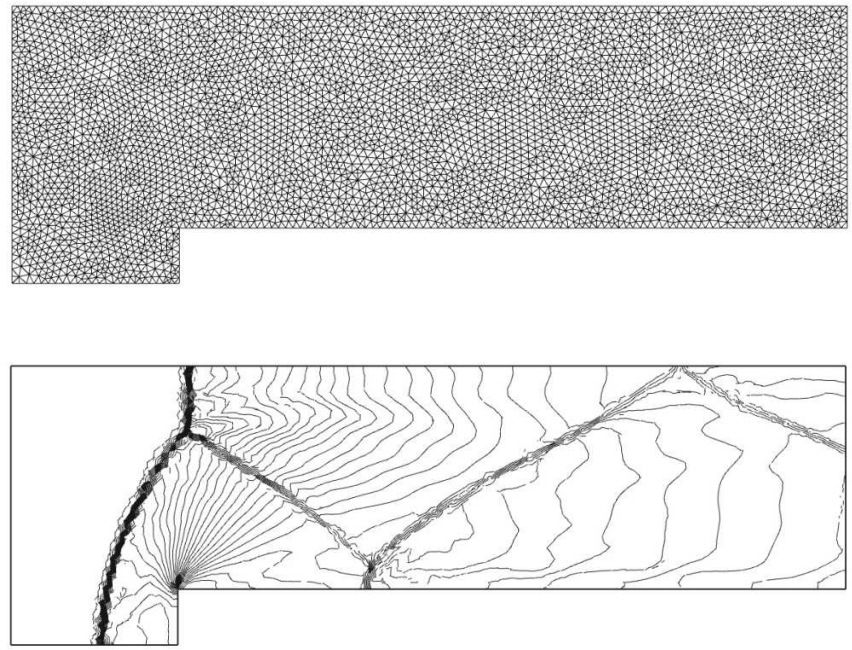
3. Mach 3 wind tunnel with a step

This problem was studied extensively by Woodward and Colella[29], and has been widely used to assess the performance of shock-capturing methods. The 2D wind tunnel is 3 units long and 1 unit wide, with a step of 0.2 units high located at 0.6 units from the tunnel inlet. The initial condition is a Mach 3 right-going uniform flow. Inviscid wall boundary conditions (reflective) are used for tunnel wall boundaries, while inflow and outflow boundary conditions are used at the inlet and exit of the wind tunnel. It is well known that the corner of the step is a singularity, and often leads to a spurious Mach stem at the downstream bottom wall, and an erroneous entropy layer at the bottom wall. In Woodward and Colella[29], various numerical treatments were used to remedy these artifacts. In the present study, no special treatments were used for the singularity to see how the singularity affects the numerical solutions.

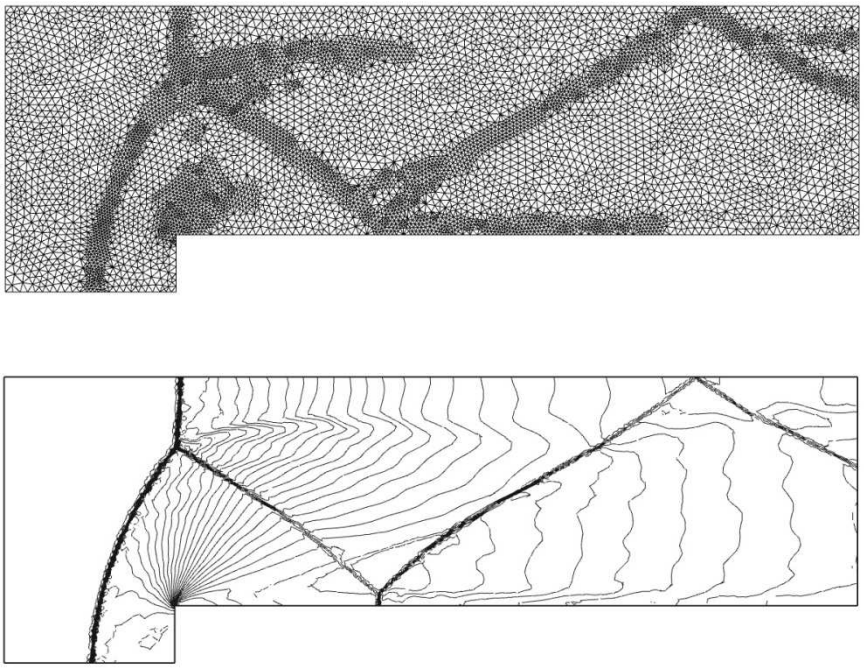
Both 2nd and 3rd order simulations are carried out, using 1-3 levels of h-refinement. Figure 14 shows the grid and density contours for a 2nd order simulation under 0, 1, 2, and 3 levels of h-refinement, respectively. All plots show 30 even contours of density between 0.054957 and 4.7699. It is clear that as the adaptation level is increased, the grid becomes exceedingly dense in the vicinity of the shock wave, near the corner of the step, and downstream of the spurious Mach stem. Also as the adaptation level is increased, the shock is captured more accurately with less

smearing, and the spurious Mach stem downstream of the step has been reduced in size to the extent that it is barely discernible. Figure 15 shows similar results for a 3rd order simulation. In both 2nd and 3rd order cases, the spurious Mach stem is completely eliminated for 2 or more levels of h-refinement.

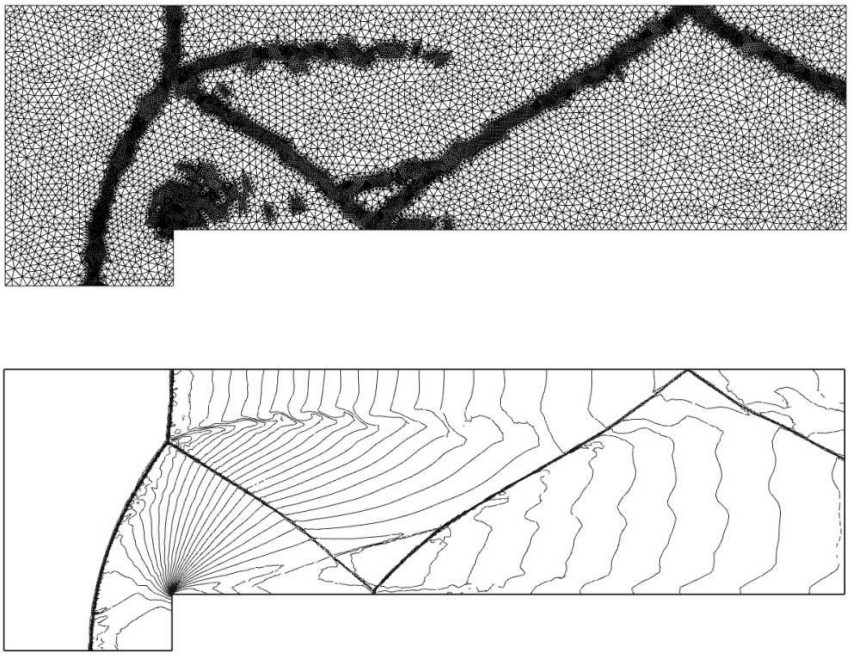
For comparison, Figure 16 shows 2nd and 3rd order results for a uniform grid that is much finer than the base grid used for the above h-refinement cases. It is evident that even using only 1 level of h-refinement produces a much better solution with far fewer degrees of freedom than that on the uniform grid.



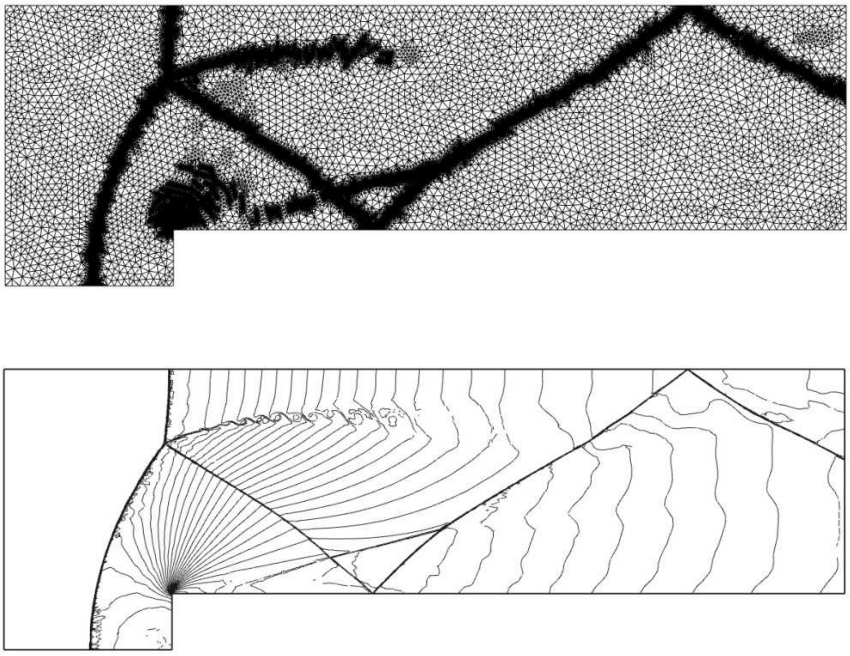
(a)



(b)

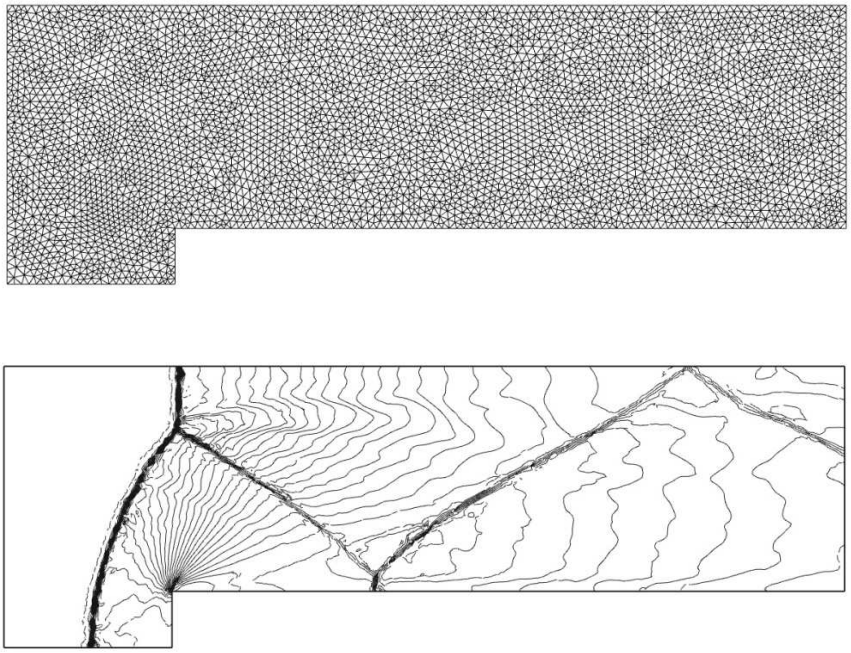


(c)

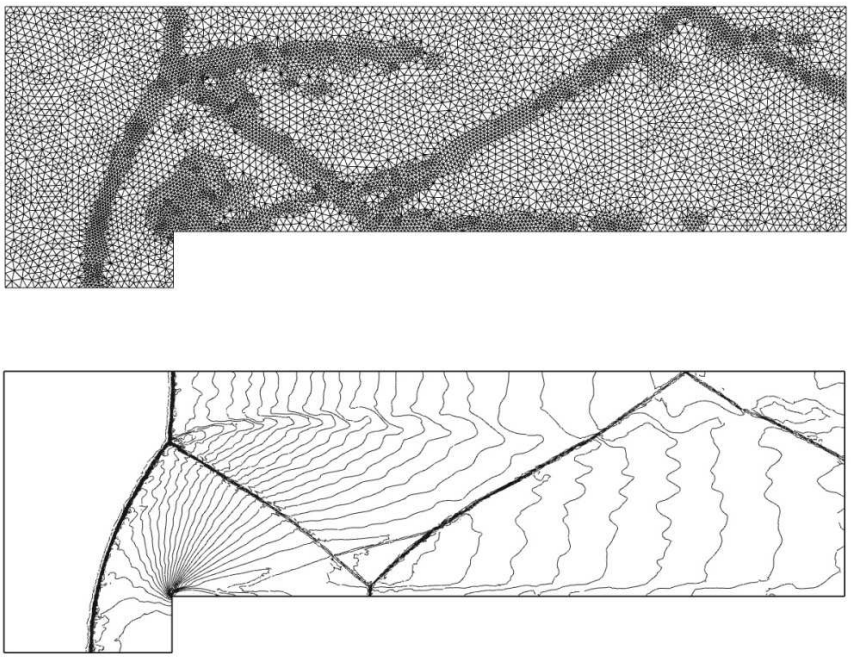


(d)

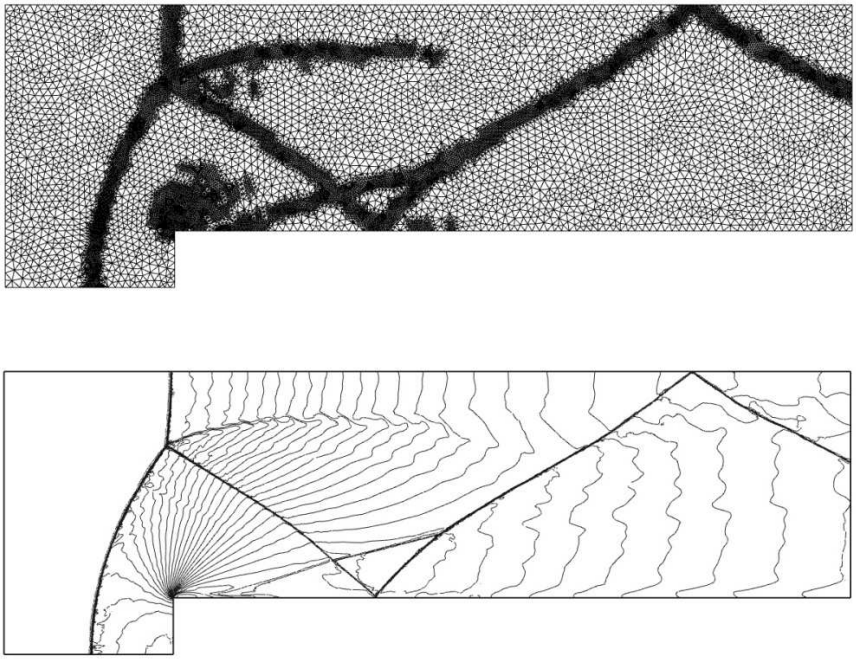
Figure 14. Density contours and grid for 2nd order SV scheme under adaptive h-refinement at time=4.0; (a) No adaptation (base grid, 26,238 DOFs); (b) 1 level of adaptation (43,179 DOFs at last time step); (c) 2 levels of adaptation (84,678 DOFs at last time step); (d) 3 levels of adaptation (190,017 DOFs at last time step); Grid is refined from base grid every 100 time steps.



(a)

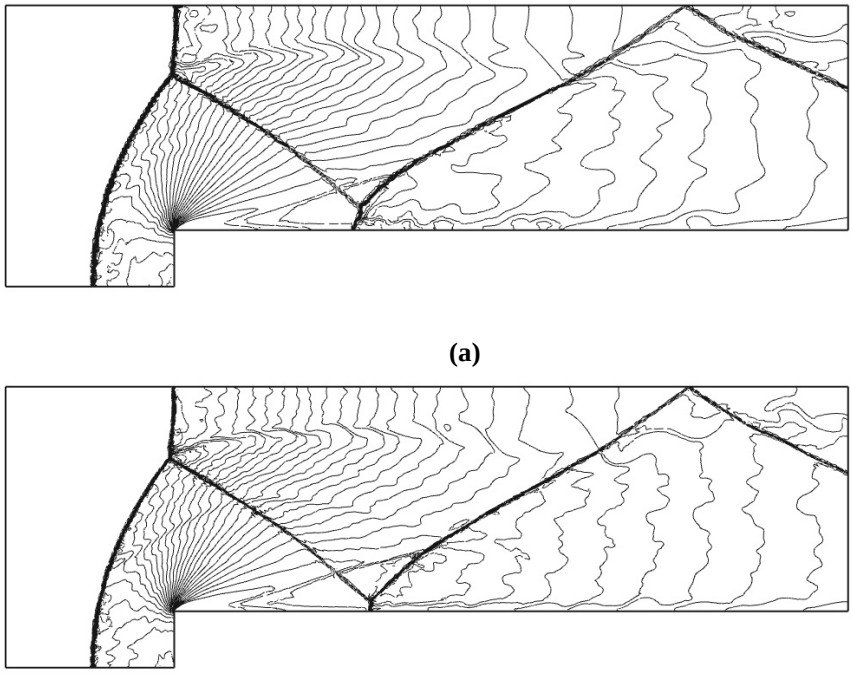


(b)



(c)

Figure 15. Density contours and grid for 3rd order SV scheme under adaptive h-refinement at time=4.0; (a) No adaptation (base grid, 52,476 DOFs); (b) 1 level of adaptation (88,800 DOFs at last time step);(c) 2 levels of adaptation (175,200 DOFs at last time step); Grid is refined from base grid every 100 time steps.



(b)

Figure 16. Density contours and grid for SV schemes on uniform grid at time=4.0; (a) 2nd order, 111,438 DOFs; (b) 3rd order, 222,876 DOFs;

V. Conclusions

The high-order quadrature-free spectral volume method has been successfully extended for use with local adaptive hp-refinement. A hierarchical edge-based adaptation algorithm was employed for high efficiency. The p-refinement methodology was effectively utilized for the case of subsonic flow over both a NACA 0012 airfoil and a NASA GA(W)-1 airfoil. The h-refinement technique was employed with success for transonic flow over a NACA 0012 airfoil, and for an unsteady supersonic problem with strong shocks. The extension of the adaptive quadrature-free SV method for use with implicit solvers for the Euler and Navier-Stokes equations will be the subject of future research.

Acknowledgements

This study has been supported by the Department of Energy (DOE) and the Air Force Office of Scientific Research (AFOSR). 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 the DOE and AFOSR.

References

1. Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.
2. 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.
3. 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.
4. 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.
5. Y. Liu, M. Vinokur, Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, J. Comput. Phys. 212 (2006) 454-472.
6. Y. Sun, Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow, J. Comput. Phys. 215 (2006) 41-58.
7. R. Harris, Z. J. Wang, Y. Liu, Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation, J. Comput. Phys. 227 (2008) 1620-1642.
8. S. K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271.
9. B. van Leer, Towards the ultimate conservative difference scheme V. a second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101-136.
10. T. J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.
11. M. Delanaye, Yen Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.
12. A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.
13. R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45-58.
14. C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97-127.
15. Z.J. Wang, High-Order Methods for the Euler and Navier-Stokes Equations on Unstructured Grids, Journal of Progress in Aerospace Sciences, Vol. 43 No. 1-3 (2007).
16. B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411-435.
17. B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90-113.
18. B. Cockburn, S. Hou, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545-581.
19. B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199-224.
20. H. L. Atkins, Chi-Wang Shu, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations, AIAA J. 96 (1996) 1683.
21. J-F Remacle, J.E. Flaherty, M.S. Shephard, An Adaptive Discontinuous Galerkin Technique with an Orthogonal Basis Applied to Compressible Flow Problems, SIAM Review, Vol. 45 No. 1 (2003) 55-73.

22. J. Flaherty, L. Krivodonova, J.-F. Remacle and M. Shephard, Aspects of discontinuous Galerkin methods for hyperbolic conservation laws. Finite Elements Anal. Design 38 (2002) 889–908.
23. J. S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, SIAM J. Numer. Anal. Vol. 35 No. 2 (1998) 655-676.
24. S. Wolfram, *Mathematica* Book (Wolfram Media and Cambridge Univ. Press, New York, 1999), 4th Ed.
25. V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267-279.
26. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357-372.
27. L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, J. Comput. Phys. 211 (2006) 492-512.
28. S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89-112.
29. P. Woodward, P. Colella, The Numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115-173.