40th Fluid Dynamics Conference and Exhibit
28 June - 1 July 2010, Chicago, Illinois

AIAA 2010-4435

Comparison of h- and p- Adaptations for Spectral Difference Methods

Yi Li, * Sachin Premasuthan * and Antony Jameson †

Aeronautics and Astronautics Department, Stanford University, CA 94305, USA

This paper applies some of the current mesh adaptation approaches to the high-order Spectral Difference (SD) method and investigates their performance on modeling inviscid steady flow. In particular, an entropy-based adjoint approach is used as the adaptation criterion. Both h-, p-refinement and p-coarsening are implemented. A mortar element method is used for treatment of non-conforming interfaces between grids with different sizes or orders. The performance is assessed by the accuracy of certain solution outputs (e.g. the drag coefficient) and the accuracy of capturing local flow features (e.g. shocks). While flow discontinuities are understandably better resolved with h-refinement, it is found that in regions of smooth flow, p-refinement offers a higher accuracy with the same number of degrees of freedom.

I. Introduction

Mesh generation has been an important procedure in computational fluid dynamics (CFD) modeling. A good mesh should be able to capture as many details as possible in the flow, while at the same time not overly resolved in the regions with mostly uniform flow. As a result one can achieve optimal accuracy in the solution with minimal computational cost. In order to generate such a mesh, a lot of prior knowledge about the physics of the modeled problem as well as the assumptions and limitations of the computational scheme is needed. Sometimes the time and cost invested in mesh generation can take up a large percentage of the total modeling effort. Even so, when we encounter complex structural geometries or flows with a wide range of length scales, it is very hard to produce a mesh that is 'optimal'.

For these reasons, adaptive methods in CFD have received much attention over the past thirty years due to their flexibility in resolving complex geometries and other problems such as unsteady flow calculations. The goal of these methods is to evenly distribute error over the whole flow domain in order to minimize global inaccuracy. The adaptation is usually based on some error indicators calculated from solutions given in the past iterations, so that regions with larger errors are refined (and in some cases, those with less errors coarsened). As the process solely depends on numerical solutions obtained during the simulation, it can be considered 'automatic' and no prior knowledge about the problem is required. One example is shock capturing: the adaptive methods will automatically detect the shock region which should be refined, so one does not need to know the shock location *a priori*.

I.A. Error estimation

As the true error is not available, many error estimation algorithms and indicators have been developed, which mainly fall into two categories. The first category is flow feature-based, where regions with special flow characteristics are sought via measurements such as solution gradients or curvatures, in attempt to detect features like shocks, surfaces or vortices etc. Indicators of this type are usually computationally cheap but often provide spurious information. For example, a sudden change in mesh size may lead to a solution gradient that is just as steep as that across the shock. Therefore, while they sometimes work well for simpler cases, they are in general less reliable in calculating complex flows.

The second category which has recently become widely popular is called output-based error estimation. These methods usually target one or more output of interest such as the lift or drag on an airfoil, and

*PhD Candidate, Department of Aeronautics and Astronautics, Student Member AIAA.
†Thomas V. Jones Professor, Department of Aeronautics and Astronautics, AIAA Fellow.

rely on the solution of the adjoint problems for the output. Therefore the adaptation is focused in regions which produce more error in the specific output. In order to obtain an all-purpose solution, usually several adjoint problems have to be solved and the computational cost becomes significant. Fortunately the effort pays off and the results have been sufficiently promising and consistent. On top of that, Fidkowski and Roe proposed an entropy adjoint approach which uses a set of entropy variables that automatically satisfies the entropy adjoint equations[1]. This provides a cheap yet effective error estimation which targets the areas of numerical entropy generation. For these advantages, this work has chosen the entropy adjoint approach as the adaptation criterion.

I.B. *h*-, *p*- and *hp*-version

A mesh is characterized by the local mesh size h and the order of approximation p. Adaptive methods started with h refinement initially, and later on Babuska *et al.* introduced p and hp methods into the field[4]. The h-version refines mesh size while keeping the order of polynomial fixed; the p-version uses a fixed mesh but increases p to improve accuracy. The hp-version combines the two approaches. All three methods have achieved success in commercial finite element codes. The h- and p-version have been extensively implemented in finite element methods and comparisons between the two have indicated a doubled asymptotic convergence rate with respect to the number of degrees of freedom as well as CPU time for p-version[3]. However the suitability of each version should depend on the problem being considered.

Implementation of each version can be very different, too, which depends on the numerical scheme used. In general p-adaptation is much simpler in mesh construction. For the high-order spectral difference method with quadrilateral grids, special attention is needed for interfaces between elements of different sizes or orders. In this work a mortar element method is used, which was initially proposed by Kopriva[5] and later implemented in SD by Premasuthan *et al.*[9]

I.C. Spectral difference method

With the significantly improved computer power, researchers have started to look at high-order numerical schemes which enable us to model more accurately viscous flows and vortex dominated flows etc. The spectral difference method is a relatively new high-order approach based on differential form of the governing equation. It was originally proposed by Liu *et al.* and extended by Wang *et al.* and Sun *et al.*[7] This work is built on the codes developed by Liang *et al.*[8] to solve the Euler equations on a 2D unstructured quadrilateral mesh. The advantage of the SD scheme is its relative simplicity in formulation compared to other high-order schemes such as the discontinuous Galerkin (DG) method and the spectral volume (SV) method, as no test functions or surface integrals are involved.

Part II of this paper outlines the mathematical formulation of the SD method, the implementation of adaptation mechanisms (the mortar element method) and the computation and theory behind the entropy-adjoint error indicator. Part III presents and discusses the results. Part IV draws conclusions and describes the future work.

II. Formulation

II.A. 2D SD formulation

The 2D compressible Euler equation can be written in the conservative form as

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{1}$$

where Q is the vector of conservative variables and F and G are the flux vectors. The unstructured quadrilateral elements in the physical domain are then transformed into a compuational domain of standard square elements $(\xi, \eta) \in [0,1] \times [0,1]$. Then the equation becomes

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0 \tag{2}$$

with

$$\tilde{Q} = |J| \cdot Q, \qquad J = \frac{\partial(x,y)}{\partial(\xi,\eta)},$$

$$\begin{pmatrix} \tilde{F} \\ \tilde{G} \end{pmatrix} = |J| \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \cdot \begin{pmatrix} F \\ G \end{pmatrix}.$$

In each standard element, the solution points are defined at Gauss-Lobatto points and the flux points at Legendre-Gauss-Quadrature points plus the two end points. If the order of the element is N, then in each dimension there are N solution points to build a degree $N-1$ polynomial, and $N+1$ flux points to build a degree N polynomial:

$$h_i(X) = \prod_{s=1, s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s} \right), \qquad l_{i+\frac{1}{2}}(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_{s+\frac{1}{2}}}{X_{i+\frac{1}{2}} - X_{s+\frac{1}{2}}} \right). \tag{3}$$

The reconstructed solution and flux polynomials are therefore just the tensor products of two corresponding 1D polynomials:

$$\begin{aligned} Q(\xi,\eta) &= \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\tilde{Q}_{i,j}}{|J_{i,j}|} h_i(\xi) h_j(\eta), \\ \tilde{F}(\xi,\eta) &= \sum_{i=0}^{N} \sum_{j=1}^{N} \tilde{F}_{i+\frac{1}{2},j} l_{i+\frac{1}{2}}(\xi) h_j(\eta), \\ \tilde{G}(\xi,\eta) &= \sum_{i=1}^{N} \sum_{j=0}^{N} \tilde{G}_{i,j+\frac{1}{2}} h_i(\xi) h_{j+\frac{1}{2}}(\eta). \end{aligned} \tag{4}$$

As the reconstructed flux values are discontinuous at element interfaces, a Riemann solver is needed to compute a common flux to ensure conservation and stability. In this work the Rusanov flux solver is used. The algorithm can be summarized in the following steps:

1. Given the conserved variables at solution points, compute the conserved variables at flux points;
2. Compute the interior fluxes at flux points from the conserved variables;
3. Use the Rusanov solver to compute the interface fluxes;
4. Compute the derivatives of fluxes at solution points and then the new conserved variables values from equation (2).

II.B. Adaptation mechanism

Both h-refinement and p-refinement and coarsening have been used. Dynamic memory allocation is employed via a linked-list structure for easier implementation of h-refinement. Newly created cells upon refinement are padded to the tail of the linked list, while the original cells are also kept in the list for possible future process of reversing the refinement.

II.B.1. h-refinement

The most direct intuitive way of refinement is to calculate an error indicator for each cell and refine the cells with indicator values above a threshold. Each quadrilateral is divided into four sub-elements by joining the midpoints of opposite faces. This results in a hanging node between coarse and refined cells, as shown in Figure 1. The flux points on both sides of such an interface are no longer at the same locations, and the normal way of solving the Riemann problem between each pair of flux points would not work unless special treatment is used. Here a mortar element method described by Kopriva[5] is implemented in 2D SD and outlined below (Figure 2):

i. Choose "mortar" faces (mi, $i = 1, 2$) of the same length as the child faces (Ci, $i = 1, 2$). Each mortar has a local coordinate $z \in [0, 1]$.

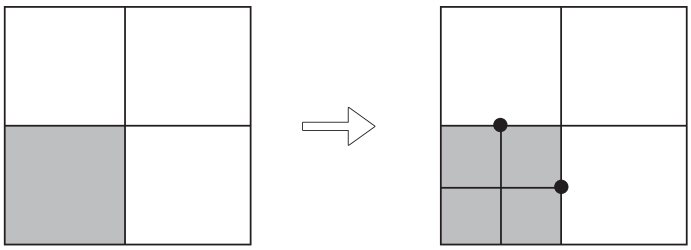


Figure 1. Refinement of one cell results in hanging-nodes.

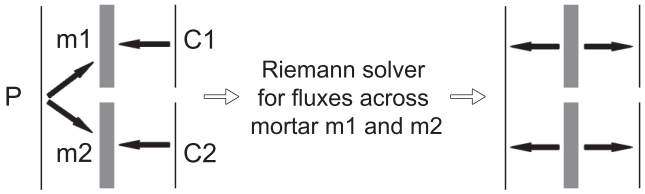


Figure 2. Mortar configuration for h refinement. We have assumed there are two child faces for simplicity.

ii. Project the conserved variables from the parent face (Q_P) onto the mortars (Q_{mi_l}). On the other side, $Q_{mi_r} = Q_{Ci}$. We assume all faces have the order N, so there are N flux points on each face at the 1D solution point locations. The mapping is then based on least square projections, which result in the formula:

$$Q_{mi_l} = M^{-1} S_i Q_P$$

where M and S_i are matrices with elements

$$M_{jk} = \int_0^1 h_j(z) h_k(z) dz$$

$$(S_i)_{jk} = \int_0^1 h_j(o_i + s_i z) h_k(z) dz$$

$$j, k = 1...N$$

where o_i and s_i are the offset and scale of mortar mi with respect to the parent face P. Function $h_j(z)$ is given by equation (3).

iii. Use the Rusanov solver to compute the interface fluxes across each mortar (F_{mi_l} and F_{mi_r}).

iv. Project the computed fluxes back to the parent face (F_P). On the other side, $F_{Ci} = F_{mi_r}$. The projection is given by:

$$F_P = \sum_i M^{-1} S_i F_{mi_l}$$

where M is the same matrix as above, and S_i are matrices with elements

$$(S_i)_{jk} = s_i \int_0^1 h_j(z) h_k(o_i + s_i z) dz.$$

Figure 3. Mortar configuration for p configuration. Here we assume the left face has order N, and the right face has order $J > N$. The mortar is chosen to have the higher order J.

II.B.2. p-adaptation

Order elevation involves prolongating the solution at order N to the solution points used for order $N+1$, which is relatively straightforward knowing the reconstructed solution polynomials. The interface between two cells with different orders again needs to be treated with the mortar scheme. The algorithm is summarized below (Figure 3):

i. Choose a "mortar" face of the same order as the face with higher order. Here we assume the left face has order N and the right face has order $J > N$, therefore the mortar face has order J.

ii. Project the conserved variables from the left face (Q_L) onto the mortar (Q_{m_l}). On the other side, we have $Q_{m_r} = Q_R$.

$$Q_{m_l} = M^{-1} S Q_L$$

where M and S are matrices with elements

$$M_{jk} = \int_0^1 h_j^m(z) h_k^m(z) dz \qquad j,k = 1...J$$

$$S_{jk} = \int_0^1 h_j^m(z) h_k^L(z) dz \qquad j = 1...J, k = 1...N$$

and h_j^m and h_j^L are the 1D polynomials interpolated on the mortar face and the left face respectively.

iii. Use the Rusanov solver to compute the interface fluxes across the mortar (F_{m_l}, F_{m_r}).

iv. Project the computed fluxes back to the left face (F_L). On the right side, we have $F_R = F_{m_r}$.

$$F_L = M^{-1} S^T F_{m_l}$$

where S is the same matrix as above, and M is a matrix with elements

$$M_{jk} = \int_0^1 h_j^L(z) h_k^L(z) dz \qquad j,k = 1...N.$$

II.C. Entropy ajoint approach

In output-based error estimation, one tries to minimize the error in the output of interest $J(Q)$, and the error can be measured by $(J(Q_H) - J(Q_h))$, where the subscript H and h represent coarse and fine mesh respectively. This error converges to the true error as $h \to 0$. For a partial differential equation $\mathcal{R}(Q) = 0$, it can be shown that

$$J(Q_H) - J(Q_h) = -\mathcal{R}_h(Q_H, \psi_h) = -\psi_h^T \mathcal{R}_h(Q_H) \tag{5}$$

where ψ_h is the adjoint solution for output J on fine mesh. Therefore an error indicator could be

$$\eta_{\kappa_H} = \left| \psi_h^T \mathcal{R}_h(Q_H) \right|, \tag{6}$$

which is the residual on the coarse grid weighted by an adjoint on the fine grid. The entropy approach proposed by Fidkowski and Roe uses entropy variables given by (in 2D)

$$\mathbf{v} = \left[\frac{\gamma - S}{\gamma - 1} - \frac{\rho V^2}{2p}, \frac{\rho u_1}{p}, \frac{\rho u_2}{p}, -\frac{\rho}{p}\right]^T \tag{7}$$

with $S = \log p - \gamma \log \rho$. It is shown that $\mathbf{v}$ automatically satisfies the entropy adjoint equation and therefore this method targets the area of spurious entropy generation. The error indicator is given by equation (6) with ψ_h replaced by $\mathbf{v}_h$. Also, while Q_H and $\mathbf{v}_H$ is computed at order N, Q_h and $\mathbf{v}_h$ is computed on the same mesh at order $N + 1$.

The adaptation procedure is outlined as follows:

i. Solve the primal problem on the current mesh at current order N to obtain Q_H;

ii. Project Q_H into an order $(N + 1)$ space and iterate a few times to get Q_h;

iii. Compute $\mathbf{v}_h$ from Q_h using equation (7);

iv. Calculate the error η_{κ_H} using equation (6);

v. Refine/coarsen a fraction of the cells with the largest/smallest indicator;

vi. Inject the solution to the adapted mesh.

In this work, adaptation is carried out at every drop of 10^{-2} in the residual.

III. Results

The proposed method has been applied to four test cases, all of which feature 2-D inviscid steady flow. Both subsonic and supersonic flows are examined and discussed in this section.

III.A. Subsonic flow over a cylinder

The first test case is a 2-D cylinder in subsonic flow at $M = 0.3$. The initial mesh and density contour are shown in Figure 4. Three regimes are tested: (1) h-refinement, (2) p-refinement and (3) p-adaptation (both refining and coarsening). Three adaptations are performed in each case. For (1) and (2), 10% of total cells are refined at each adaptation. For (3), 30% of cells are refined and 10% are coarsened at each adaptation. Figure 5(a) shows the error in log scale calculated from the solution on the initial mesh. It indicates that the spurious entropy generation occurs mostly near the surface around the cylinder as well as in the downstream. Figure 5(b) shows the mesh after three h-refinements and Figure 5(c) shows the order distribution after three p-refinements. Figure 6 shows the order distribution after three p-adaptations (in region around the cylinder and in the whole domain respectively). It can be seen that all three regimes show a similar pattern of distribution of degrees of freedom. This is expected as they use the same adaptation criterion.

Since there should be no drag on a stationary cylinder in inviscid uniform flow, the error in drag coefficient can be calculated. Figure 7 plots the C_D error against the number of degrees of freedom and the C_D error against the CPU time. To achieve the same accuracy, p-refinement appears to require the least number of degrees of freedom and also the least CPU time, followed by p-adaptation and h-refinement. This suggests that p-methods are more efficient than h-methods for smooth flows. It is worth noting that p-refinement on its own can be better than a combination of both refinement and coarsening. Figure 5 and 6 provide some explanations for this observation: after a few adaptations of both refining and coarsening, the regions of different orders are more fragmented than that after only refinements. This results in a lot of interfaces with order-mismatch, and therefore a lot of mortar elements which is likely to reduce both the accuracy and the speed. This could potentially be improved with some methods of refining or coarsening blocks of cells each time, to minimize the number of interfaces with order-mismatch or hanging nodes. It is also worth noting that compared to the solution on the initial mesh, all adaptation regimes provide much improved accuracy with respect to both the number of degrees of freedom and the CPU time.

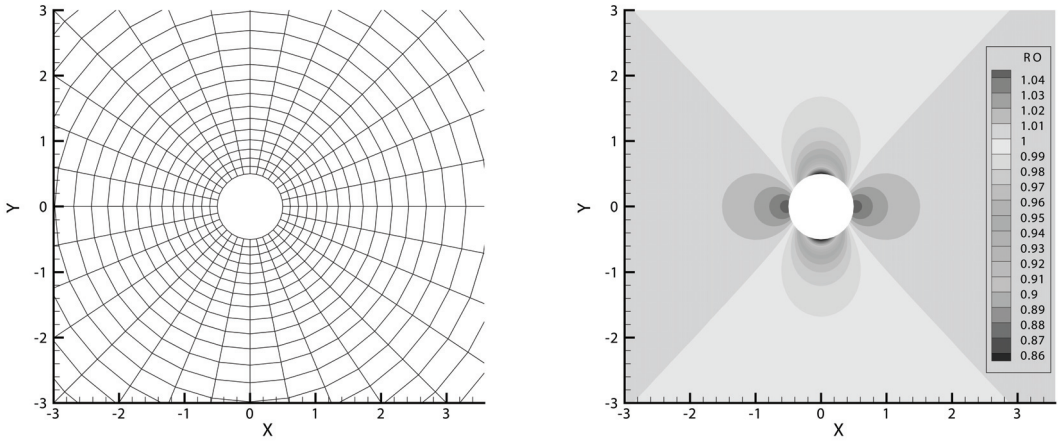


Figure 4. Initial mesh and density contour of inviscid flow past a 2-D cylinder, $M = 0.3$.

III.B. Subsonic flow over an airfoil

The second test case is a NACA 0012 airfoil in inviscid subsonic flow, at $M = 0.4$ and an angle of attack $\alpha = 4°$. Figure 8 shows the initial mesh, an O-mesh. Two sets of tests are conducted with an initial solution of order 2 and 3 respectively. The error map of the solution on the original mesh, the mesh after three h-refinements and the order distribution after three p-refinements are shown in Figure 9 and 10. At each adaptation, 10% of the cells are h-refined or 20% are p-refined. Comparing the error maps for $N = 2$ and $N = 3$, we can see that the error throughout the domain is smaller for $N = 3$.

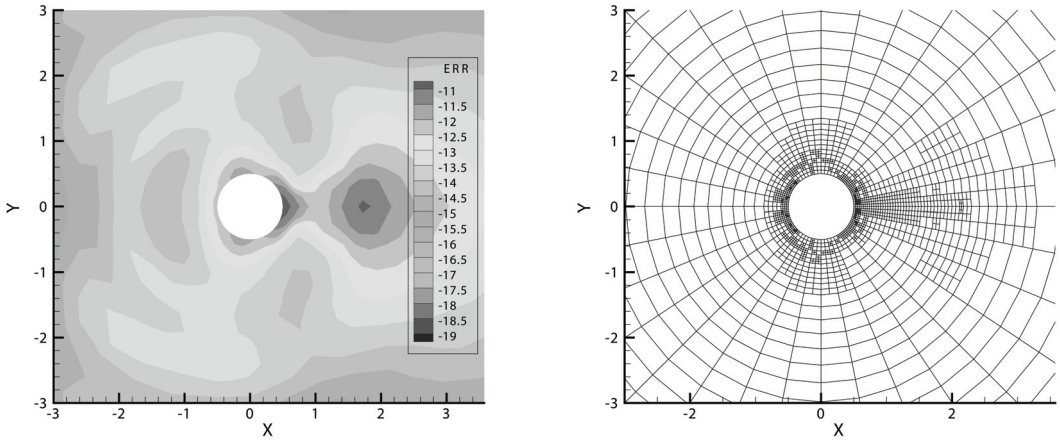
It is also worth noting that the final adapted mesh much depends on the order of accuracy. For $N = 3$, the refined grids are more concentrated around the leading and trailing edges, where the flow changes most rapidly. For $N = 2$, however, the refinement occurs more spread out. This observation agrees with that of Barter.[11] The reason is that in regions with smooth flow, a higher order solution can provide an accuracy high enough to overweigh the benefit of h-refinement. (This also agrees with the observation in the previous test case that p-refinement is more effective than h-refinement for smooth flows.) The error contour is also compared for h- and p-refinements, as shown in Figure 11. The error in the flow domain has decreased after refinement. p-refinement provides greater accuracy around the airfoil than h-refinement but has larger errors around the trailing edge (therefore the order distribution pattern in Figure 10(c)).

The h-refined mesh with $N = 2$ is also compared with Fidkowski and Roe's, as shown in Figure 12. They adapted an initial C-mesh with $N = 2$ eight times, each time refining 10% of the cells, and used the same entropy adjoint error indicator. The refined meshes from both cases show a similar pattern. The refinement with SD, however, results in a cleaner mesh which also shows the trace of the stagnation line.

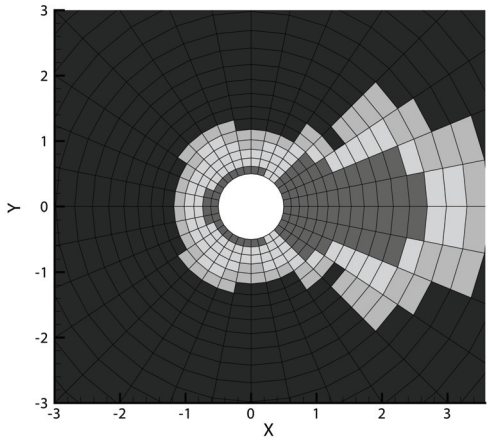
A 'true' solution is obtained by uniformly refining the most refined mesh (Figure 10(b)) and then solving it with $N = 4$. The errors in C_D and C_L for different meshes are plotted in Figure 13. As seen, p-refinement still give higher accuracy than the h-refinement counterpart. Solutions started with $N = 3$ are also much more accurate. This shows that for more complex flows, higher order schemes are preferable even with the use of adaptation techniques.

III.C. Supersonic flow over a bump

This test case is a bump in a supersonic flow at $M = 1.4$. It features an oblique shock generated at the leading edge of the bump which is reflected off the ceiling of the duct and interacts with another oblique shock at the trailing edge. Figure 14(a) shows the initial mesh and density contour of the solution. Note that artificial viscosity is implemented for SD to deal with shocks.[9] Because the flow involves discontinuities, the suitable adaptation regime would be h-refinement, as a higher order would only introduce more oscillations in the solution. Figure 14(b) shows the mesh after three h-refinements and its solution density contour. It can be seen that the entropy adjoint error indicator accurately captures the shocks and the refined mesh gives much thinner shocks than the initial mesh does. Figure 15 shows the initial error map and the error



(a) (b)



(c)

Figure 5. (a) Error map of the initial solution ($N = 3$); (b) mesh after three h-refinements; (c) order distribution after three p-refinements (N ranges from 3 to 6 in the domain).

after two refinements. After refinement, the error is concentrated in a thinner region around the shock.

III.D. Transonic flow over an airfoil

The final test case is a NACA 0012 airfoil in transonic flow at $M = 0.8$, at an angle of attack of 1.25°. It uses the same initial mesh as the second test case. Figure 16(a) and 16(b) show the Mach and error contour on the initial mesh respectively. There are oscillations in Mach upstream of the shock, and the shock is smeared across a few cells. Figure 16(c) shows the mesh after three h-refinements. The regions around the leading and trailing edges as well as around the shock are refined. Figure 16(d) shows the Mach contour of the new solution, in which the shock is much better resolved.

It is worth noting that the mortar element method, when used together with artificial viscosity, further limits the CFL constraints and the rate of convergence.

IV. Conclusion

Both h- and p-refinements and p-coarsening are studied for subsonic flows. In the absence of flow discontinuities, it is found that p-refinement gives the best accuracy with respect to the number of degrees of

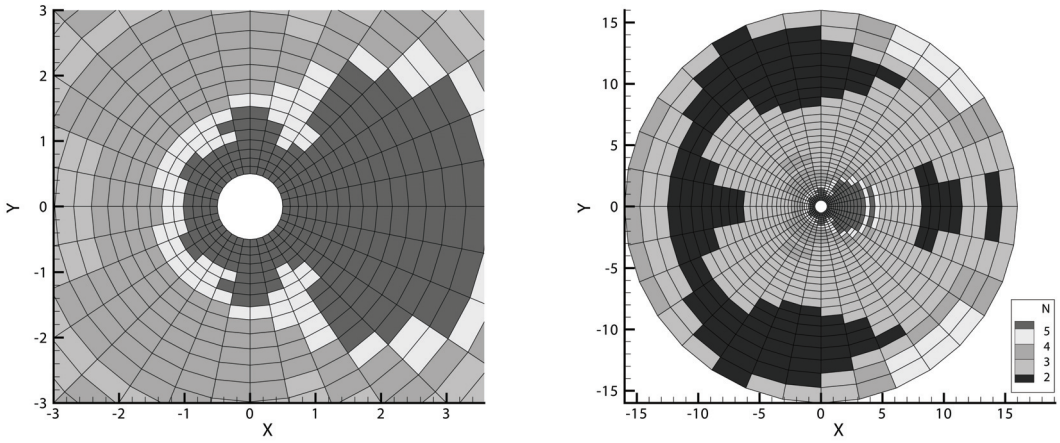
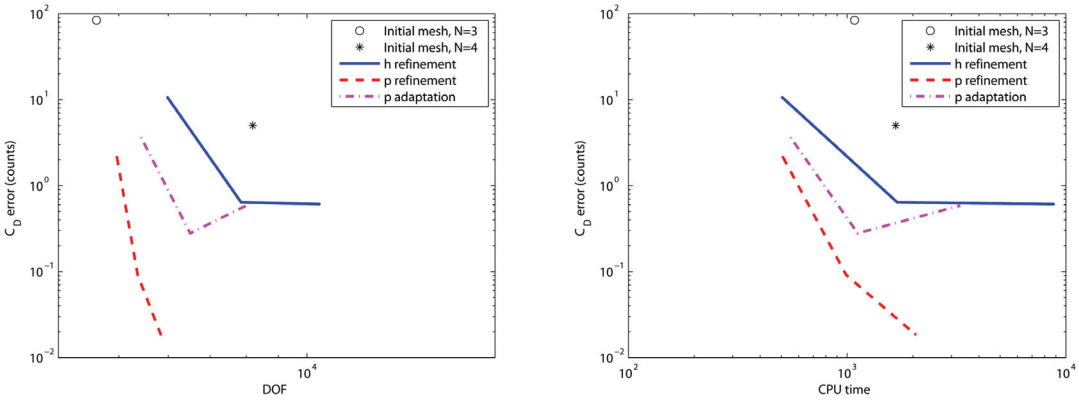


Figure 6. Order distribution after three p-adaptations (N ranges from 2 to 6 in the domain). The figure on the left shows the region around the cylinder while the one on the right shows the whole computed domain.



(a) (b)

Figure 7. (a) Error in C_D (counts) against number of degrees of freedom; (b) error in C_D (counts) against CPU time.

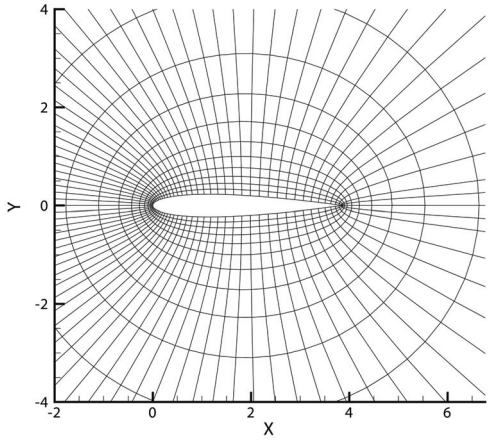
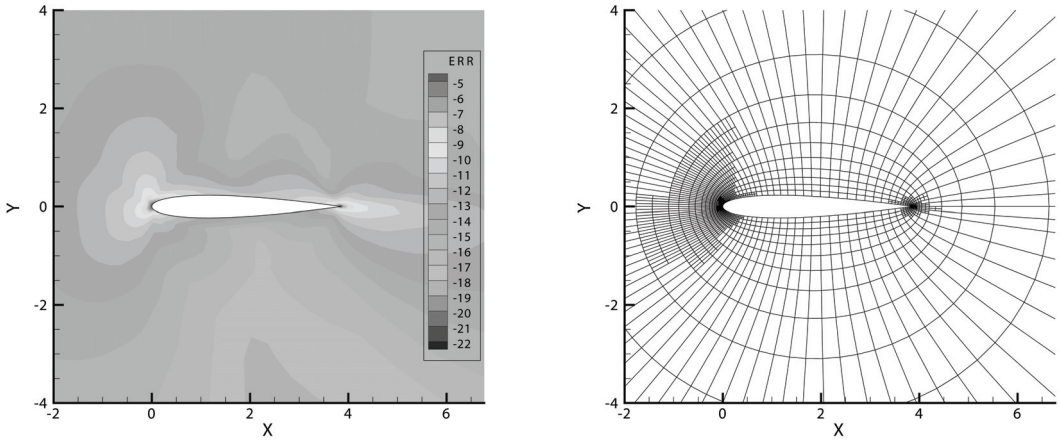
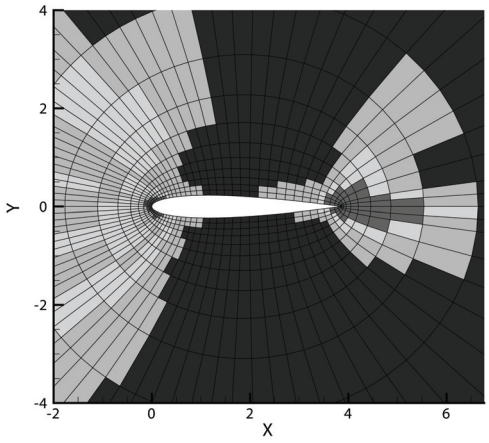


Figure 8. Initial mesh for a NACA 0012 airfoil.



(a) (b)



(c)

Figure 9. (a) Error map of the initial solution ($N = 2$); (b) mesh after three h-refinements; (c) order distribution after three p-refinements (N ranges from 2 to 5 in the domain).

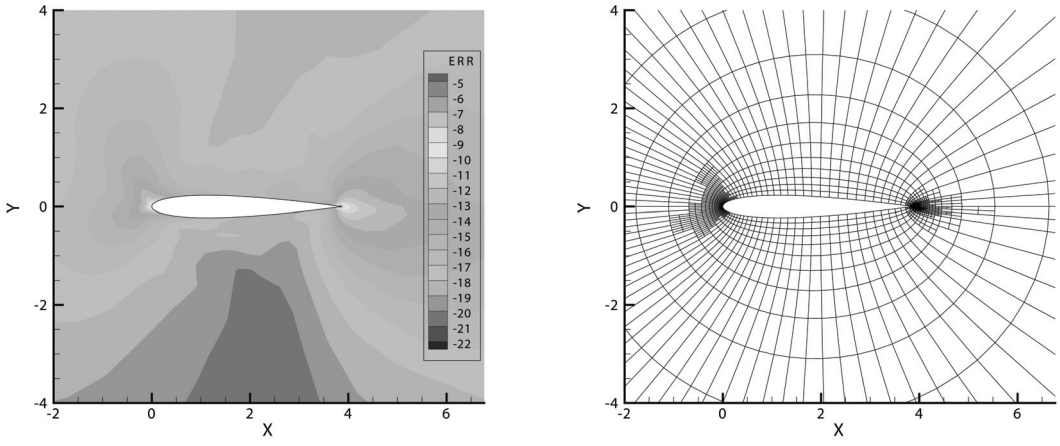
freedom and CPU time. h-refinement is also tested on flows with shocks and is proved to be an effective way of resolving shocks with the high order SD scheme (with artificial viscosity). Mortar elements are used to deal with interfaces with hanging nodes or order mismatch. The method appears to be stable although it does lead to further constraints of CFL numbers and slower convergence, especially when used in conjunction with artificial viscosity. The entropy adjoint variables provide an effective adaptation criterion which track spurious entropy generation. Both shocks and regions of large flow gradients are identified by the entropy adjoint error indicator.

With the conclusion above, it is not hard to see that one sensible way of adaptation is to h-refine regions around flow discontinuities and p-refine regions of smooth flow. This would need a further identifier for flow discontinuities, e.g. a shock indicator. It is also plausible to h-refine while p-coarsen around shocks.

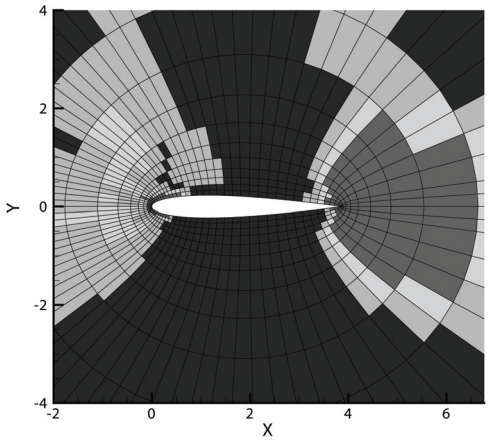
Adaptation will also be useful in unsteady and viscous flows. These can all become topics of future research.

Acknowledgments

Yi Li is supported by a Stanford Graduate Fellowship. The authors would like to thank the grant support from NSF monitored by Dr Leland Jameson (with award number 0708071), and AFOSR monitored by Dr



(a) (b)



(c)

Figure 10. (a) Error map of the initial solution ($N = 3$); (b) mesh after three h-refinements; (c) order distribution after three p-refinements (N ranges from 3 to 6 in the domain).

Fariba Fahroo (with award number FA9550-07-1-0195).

References

[1]K.J. Fidkowski, P.L. Roe, *Entropy-based Mesh Refinement, I: The Entropy Adjoint Approach*, AIAA 2009-3790, 19th AIAA Computational Fluid Dynamics, San Antonio, Texas, 22-25 June 2009.

[2]M.J. Berger, A. Jameson, *Automatic Adaptive Grid Refinement for the Euler Equations*, MAE Report 1633, AIAA Journal, 23, 1985, pp. 561-568.

[3]I. Babuska, M. Suri, *The P and H-P Versions of the Finite Element Method, Basic Principles and Properties*, SIAM Review, Vol. 36, No. 4 (Dec., 1994), pp. 578-632.

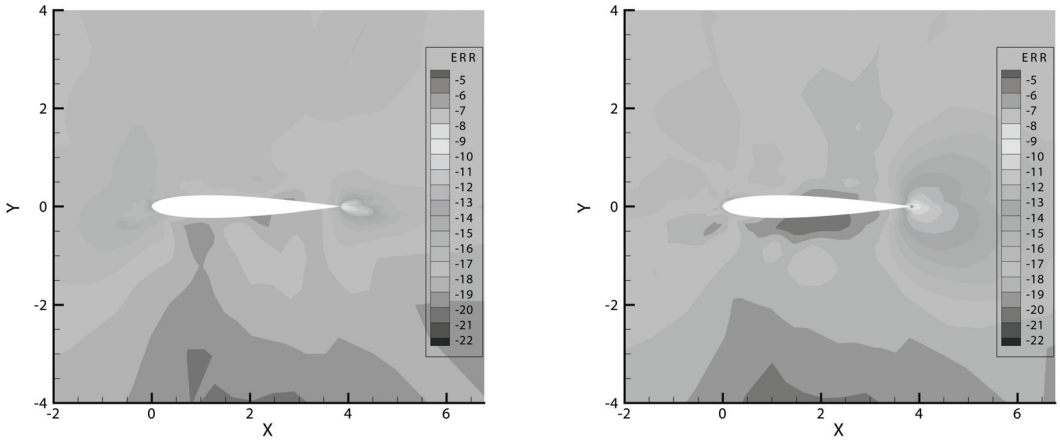
[4]I. Babuska, B. A. Szabo, I. N. Katz, *The p-Version of the Finite Element Method*, SIAM Journal on Numerical Analysis, Vol. 18, No. 3 (Jun., 1981), pp. 515-545.

[5]D.A. Kopriva, *A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method*, Journal of Computational Physics, Vol. 128, Issue 2 (Oct., 1996) pp. 475-488.

[6]K.J. Fidkowski, D.L. Darmofal, *Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results*, AIAA 2009-1303, 2009.

[7]Y. Sun, Z. J. Wang, Y. Liu, *High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids*, Communication in Computational Physics, Vol. 2, pp. 310-333, 2007.

[8]S. Premasuthan, C. Liang, Z. Wang, A. Jameson, *p-Multigrid Spectral Difference Method For Viscous Compressible Flow Using 2D Quadrilateral Meshes*, AIAA 2009-950, 2009.



(a) (b)

Figure 11. (a) Error map after two h-refinements; (b) error map after two p-refinements. Both starting with $N = 3$.

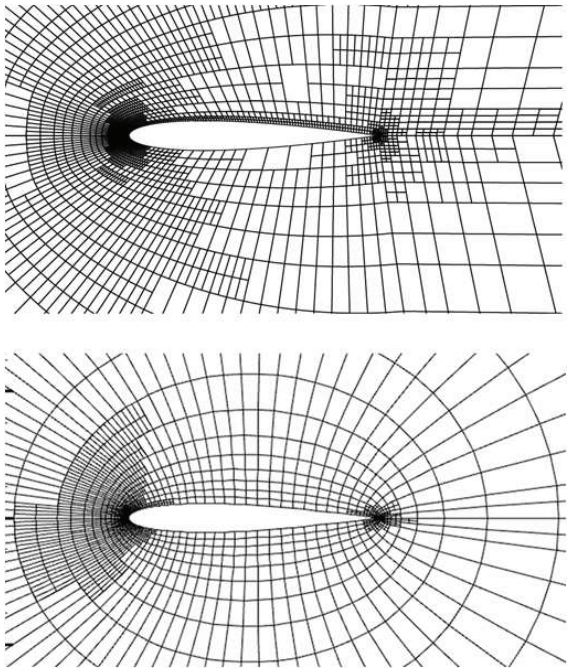


Figure 12. Comparison of adapted SD mesh (bottom) with that of Fidkowski and Roe[1] (top).

[9]S. Premasuthan, C. Liang, A. Jameson, *Computation of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity*, AIAA 2010-1449, 2010.

[10]T.J. Baker, *Mesh adaptation strategies for problems in fluid dynamics*, Finite Elements in Analysis and Design, Vol. 25, Issue 3-4 (Apr., 1997), pp. 243 - 273.

[11]G.E. Barter, *Shock capturing with PDE-based artificial viscosity for an adaptive, higher-order discontinuous Galerkin finite element method*, PhD Thesis, M.I.T., June 2008.

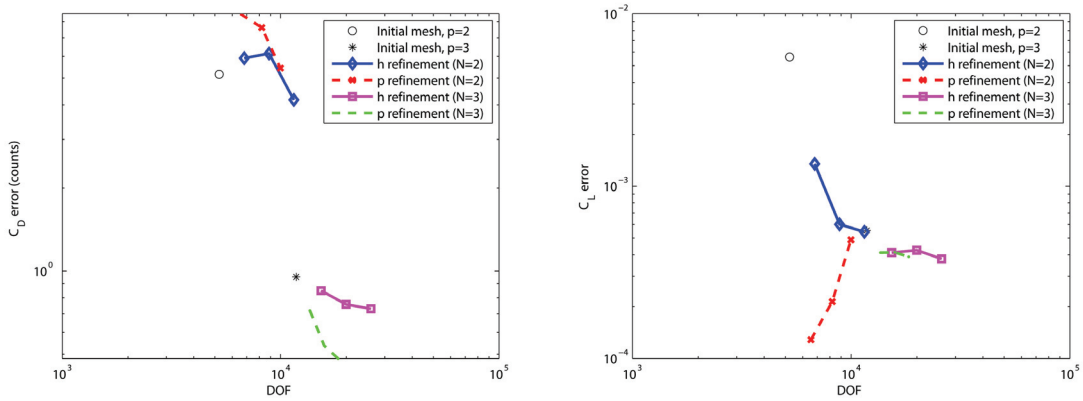


Figure 13. Error in (a) C_D (counts) and (b) C_L against number of degrees of freedom for subsonic airfoil test case.

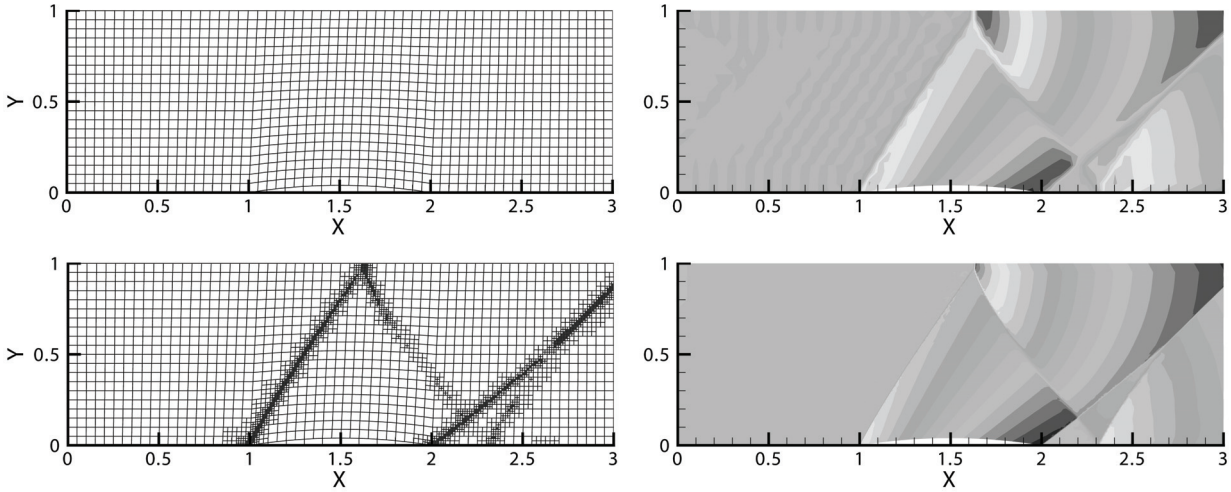
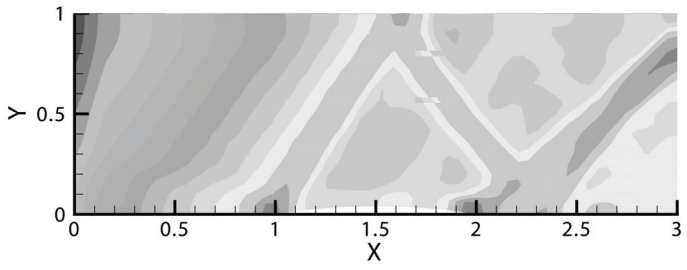
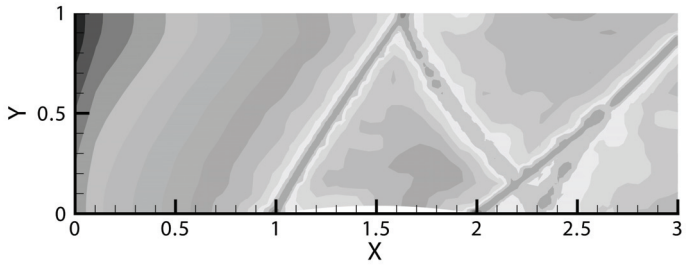


Figure 14. (a) (Top.) The initial mesh and solution density contour with $N = 3$. (b) (Bottom.) The mesh after three h-refinements and the solution density contour ($N = 3$).



(a)



(b)

Figure 15. (a) Error map for the initial solution; (b) error map after two h-refinements.

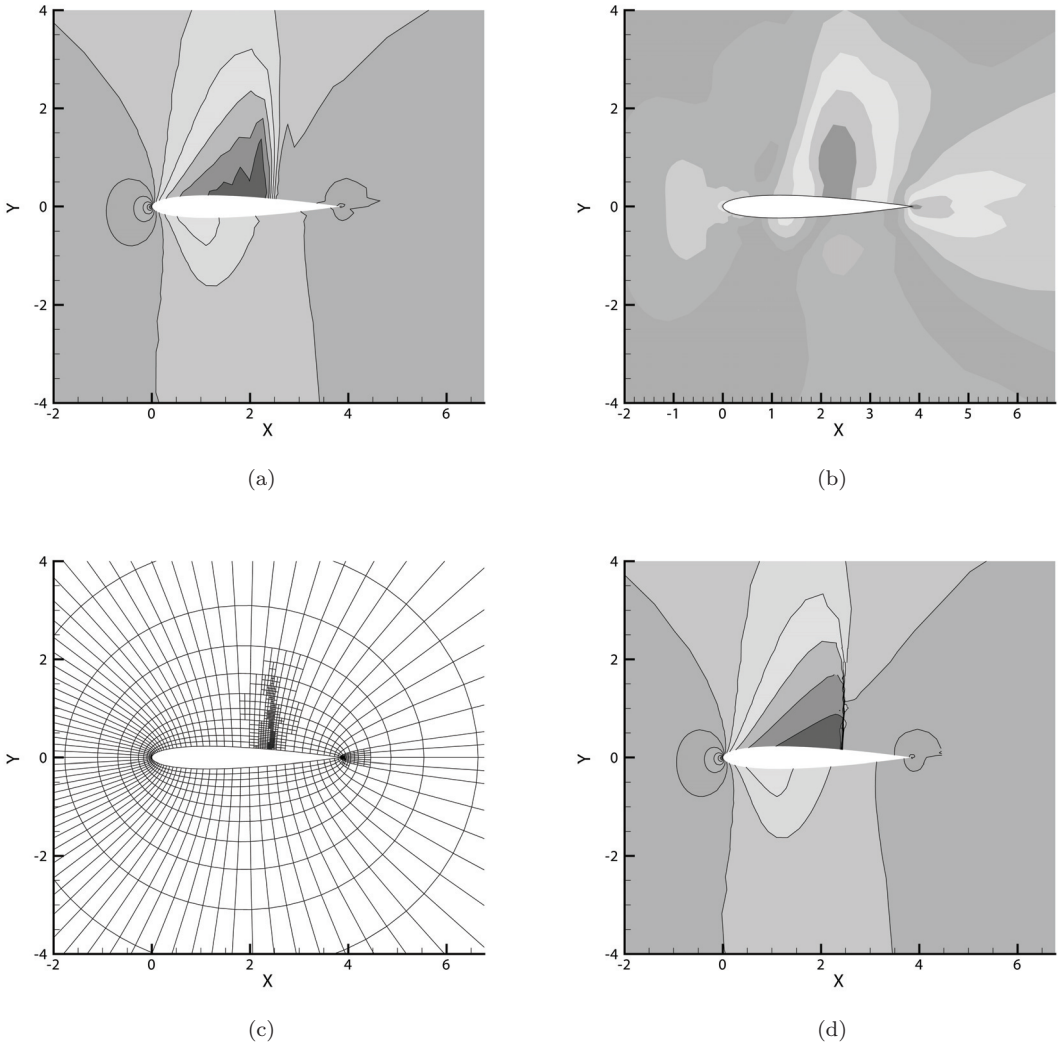


Figure 16. (a) Mach contour of the initial solution ($N = 3$); (b) the initial error map; (c) mesh after three h-refinements; (d) Mach contour of the solution on the adapted mesh.