47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition
5 - 8 January 2009, Orlando, Florida
AIAA 2009-401

A Unifying Lifting Collocation Penalty Formulation for the Euler Equations on Mixed Grids

Z.J. Wang[1] and Haiyang Gao[2]
Department of Aerospace Engineering, Iowa State University, Ames, IA, 50011

Recently a new high-order formulation for 1D conservation laws was developed by Huynh based on the idea of "flux reconstruction". The formulation was capable of unifying several popular methods including the discontinuous Galerkin, staggered grid multi-domain method, or the spectral difference/spectral volume methods into a single family. The extension of the method to quadrilateral and hexahedral elements is straightforward. In an attempt to extend the method to other element types such as triangular, tetrahedral or prismatic elements, the idea of "flux reconstruction" is generalized into a "lifting collocation penalty" approach. With a judicious selection of solution points and flux points, the approach can be made simple and efficient to implement for mixed grids. In addition, the formulation includes the discontinuous Galerkin, spectral volume and spectral difference methods as special cases. Several test problems are presented to demonstrate the capability of the method.

I. Introduction

THERE has been a surge of recent research activities in the computational fluid dynamics (CFD) community on high-order methods capable of solving the Navier-Stokes equations on unstructured grids. For a review of some of these activities, the readers can refer to [12,44]. This surge comes from the expectation that these methods have the potential of delivering higher accuracy with less CPU time than the 1st or 2nd order methods for problems with both complex physics and geometry, such as helicopter blade vortex interactions, flow over high lift configurations.

For compressible flow computations in aerospace applications, the leader of these high-order methods is arguably the discontinuous Galerkin (DG) method [8-10,3,4,41,31]. There are also significant research activities in other methods, such as the k-exact finite volume method [2,6,11,20], streamline-upwind Petrov-Galerkin (SUPG) method [21,42], spectral and spectral-element type method [14,5,18,24], the residual distribution (RD) or fluctuation splitting method [1], staggered-grid (SG) multi-domain spectral method [25], spectral volume (SV) [43,45-47,28] and spectral difference (SD) [27,29,30] methods. The SD method can be viewed as the extension of the SG method to triangular meshes. Recently, Van den Abeele et al [38,39] and Huynh [22] independently showed that the SG and SD methods are independent of how solution points are selected. Only the flux points determine the characteristics of both methods. Therefore one does not need to use a staggered grid at all. The solution and flux points can coincide to improve efficiency [39]. All the above mentioned methods share a common property, i.e., "upwinding" is incorporated in one way or another into the methods to account for the wave dynamics of hyperbolic conservation laws. In some sense, all the methods have embedded the main characteristics of the underlying wave dynamics into the numerics to achieve stability, consistency and accuracy. On the other hand, one can also distinguish these methods based on different criteria. For example, one can divide all these methods into two categories, those with continuous and those with discontinuous solution spaces. The SUPG and RD methods assume the solutions to be continuous across element interfaces, while the DG, SG, SV and SD methods assume discontinuous solutions. For methods with discontinuous solution spaces, Riemann solvers [34,33,32,26,23] are used to compute common fluxes at element interfaces to incorporate "upwinding", similar to the Godunov finite volume method [13,40]. In this paper, we focus on the methods with discontinuous solution spaces only. These methods of course share many similarities. In fact, the SV and SD methods are identical for one dimensional conservation laws if the flux points in the SD method coincide with the partition boundaries in the SV method [39]. This equivalence, however, does not extend to two or three dimensions. The main difference lies in how degrees-of-freedom (DOFs) are chosen, and updated. Although all approaches based on the DG method are mathematically identical, at least for linear equations,

[1] Professor of Aerospace Engineering, 2271 Howe Hall, Associate Fellow of AIAA.
[2] Graduate Research Assistant, Department of Aerospace Engineering, 2271 Howe Hall, AIAA Member.

different choices of DOFs are used by various researchers resulting in different efficiency and numerical properties. Depending on how DOFs are selected, DG approaches can be further divided into modal and nodal approaches. In the SV method, the DOFs are always the sub-cell or control volume (CV) averages, while in the SG/SD method, the DOFs are the solutions at a pre-defined nodal set, i.e., the solution points (SPs). They are updated using approaches similar to the traditional finite volume or finite difference methods (or collocation method). Figs. 1 and 2 show the SV partition and the distribution of possible solution and flux points for the SD method. Comparisons of the DG, SV and SD methods have been carried out in [49] and [37]. Here are some of the conclusions from these comparisons:

1. All three methods are capable of achieving the optimal order of accuracy, i.e., (k+1)th order of accuracy for degree k polynomial reconstruction;
2. The DG method has lower error magnitude than the SV and SD methods;
3. The SV and SD methods allow larger time steps than the DG method for stability;

Very recently, a new formulation based on the idea of flux reconstruction (FR) was developed by Huynh [22] for 1D conservation laws, which is referred to the FR method. This method is a nodal formulation, with an element-wise discontinuous polynomial solution space. The solution polynomial is interpolated from the solutions at a set of solution points. In addition, a new flux polynomial is reconstructed, which satisfies conservation at element interfaces. This formulation has some remarkable properties. The framework is easy to understand, efficient to implement and recovers several known methods such as the DG, SG or the SV/SD methods in one dimension. The DG approach based on the FR method is probably the simplest and most efficient amongst all DG formulations. Another scheme in the family, named the g2 scheme, is probably the most efficient among all the methods with a discontinuous solution space, and has the largest CFL number for stability. The extension of the method to quadrilateral and hexahedral grids is straightforward. In the present study, we attempt to extend the method to triangular elements.

The paper is organized as follows. For the sake of completeness, we briefly review Huynh's FR method in Section 2. After that, the generalization of the method to triangular elements is described in Section 3 with a lifting collocation penalty formulation, including the formulation for mixed triangular-quadrilateral grids. Section 4 presents the computational results for several benchmark problems, including accuracy studies on mixed unstructured grids. Conclusions for the present study and future work are summarized in Section 5.

II. Review of the Flux Reconstruction Method

This review presents the essential idea of the flux reconstruction method [22]. Consider the following scalar conservation law

$$\frac{\partial Q}{\partial t} + \frac{\partial F(Q)}{\partial x} = 0 \tag{2.1}$$

where Q is the state variable and F is the flux. The computational domain *[a, b]* is discretized into N elements, with the *ith* element defined by $V_i \equiv [x_{i-1/2}, x_{i+1/2}]$. Each element can be transformed into the standard element *[-1, 1]* using a linear transformation. The DOFs at the *ith* element are the nodal values of the state variable $Q_{i,j}$ at *k+1* solution points, $x_{i,j}$, *j = 1, ..., k+1*. Then the solution is approximated by the following degree *k* Lagrange interpolation polynomial

$$Q(x) \approx Q_i^h(x) = \sum_{j=1}^{k+1} L_j(x) Q_{i,j} \tag{2.2}$$

where $L_j(x)$ is the Lagrange polynomial or shape function. Given this numerical solution, the flux at every point is well defined, i.e., $F(Q_i^h(x))$. For non-linear conservation laws, $F(Q_i^h(x))$ may not be a polynomial. Instead, $F(Q_i^h(x))$ is approximated by the following degree *k* flux polynomial

$$F_i(x) = \sum_{j=1}^{k+1} L_j(x) F(Q_i^h(x_{i,j})). \tag{2.3}$$

Obviously, if the flux is a linear function of Q, $F_i(x)$ is identical to $F(Q_i^h(x))$. Since we do not explicitly enforce continuity at element interfaces, the state variable is discontinuous across the interfaces. In order to update the DOFs, a new flux function $\hat{F}_i(x)$ is reconstructed, which must satisfy the following criteria:

- $\hat{F}_i(x)$ is a degree *k+1* polynomial, i.e., one order higher than the solution polynomial;
- $\hat{F}_i(x)$ is close to $F_i(x)$ in some sense. In other words, some norm of the difference $\left\|\hat{F}_i(x) - F_i(x)\right\|$ is minimized;
- At both ends of the element, the flux takes on the value of the Riemann fluxes, i.e.,

$$\hat{F}_i(x_{i-1/2}) = \tilde{F}(Q_{i-1}^h(x_{i-1/2}), Q_i^h(x_{i-1/2})) \equiv \tilde{F}_{i-1/2}$$

$$\hat{F}_i(x_{i+1/2}) = \tilde{F}(Q_i^h(x_{i+1/2}), Q_{i+1}^h(x_{i+1/2})) \equiv \tilde{F}_{i+1/2},$$

where $\tilde{F}(Q^-, Q^+)$ is any Riemann flux given the two discontinuous solutions at the left and right of the interface. Once this flux function is found, the DOFs are updated using the following differential equation

$$\frac{\partial Q_{i,j}^h}{\partial t} + \frac{\partial \hat{F}_i(x_{i,j})}{\partial x} = 0. \tag{2.4}$$

Obviously the above criteria do not uniquely define $\hat{F}_i(x)$. As a matter of fact, only two conditions are given. We need *k* extra conditions to define the degree *k+1* polynomial $\hat{F}_i(x)$. Using special polynomials such as the Radau and Legendre polynomials, Huynh [22] successfully recovered $\hat{F}_i(x)$ for the DG, SG (or SD/SV) methods, at least for linear conservations laws. The reconstructed flux is first re-written as

$$\hat{F}_i(x) = F_i(x) + \sigma_i(x) \tag{2.5}$$

where $\sigma_i(x)$ is a correction flux polynomial, which should be as close as possible to 0. The correction is then further expressed to satisfy the two end conditions

$$\sigma_i(x) = [\tilde{F}_{i-1/2} - F_i(x_{i-1/2})]g_L(x) + [\tilde{F}_{i+1/2} - F_i(x_{i+1/2})]g_R(x), \tag{2.6}$$

where $g_L(x)$ and $g_R(x)$ are both degree *k+1* polynomials called correction functions, and they satisfy

$$\begin{aligned} g_L(x_{i-1/2}) &= 1, \quad g_L(x_{i+1/2}) = 0 \\ g_R(x_{i-1/2}) &= 0, \quad g_R(x_{i+1/2}) = 1. \end{aligned} \tag{2.7}$$

Eq. (2.4) then becomes

$$\frac{\partial Q_{i,j}^h}{\partial t} + \frac{\partial F_i^h(x_{i,j})}{\partial x} + [\tilde{F}_{i-1/2} - F_i(x_{i-1/2})]g_L'(x_{i,j}) + [\tilde{F}_{i+1/2} - F_i(x_{i+1/2})]g_R'(x_{i,j}) = 0. \tag{2.8}$$

Because of symmetry, we only need to consider $g_L(x)$, or simply *g(x)*. It is more convenient to consider the correction function in the standard element $g(\xi)$ on [-1, 1]. Many correction functions were presented in [22], corresponding to different numerical methods. Several schemes are described next.

1. If *g* is the right Radau polynomial, the resulting scheme is actually the DG method!
2. If *g* has Chebyshev-Lobatto points as its interior roots, the resulting scheme is the SG method or the SD/SV method in 1D. The scheme, however, is mildly unstable, which was also found by Van den Abeele et al [39].

3. If g has a vanishing derivative at the right boundary as well as at the interior Legendre-Lobatto points, the method results in a remarkably simple, yet stable scheme if one chooses the Legendre-Lobatto points as the solution points since the corrections for the interior solution points vanish! Huynh named this scheme the g_2 scheme.
4. If g has the Legendre-Gauss points as its interior roots, the scheme is stable. This suggests that using the Legendre-Gauss points and the two end points as flux points results in a stable SG method in 1D.

As mentioned earlier, the FR method has some remarkable properties. Its DG formulation looks like a pseudo-DG. However, it is identical to the "real" DG formulation! The natural question next is how about simplex elements and other cell types? This is the focus of the present paper.

Let's first try to duplicate Huynh's flux reconstruction idea in 2D for triangular cells. Consider the following 2D conservation laws

$$\frac{\partial Q}{\partial t}+\frac{\partial F^x(Q)}{\partial x}+\frac{\partial F^y(Q)}{\partial y}=0 \tag{2.9}$$

where F^x and F^y are the fluxes in x and y direction respectively. Define a set of solution points which support a degree k polynomial $Q_i(\vec{r})$, such as those shown in Figure 2 for $k = 2$. Fluxes at the solution points can also be computed, and their Lagrange interpolations result in two degree k flux polynomials $F_i^x(\vec{r})$ and $F_i^y(\vec{r})$. Along the element interfaces, the solution is discontinuous. A Riemann flux in the interface normal direction can be computed at each interface point,

$$\tilde{F}^n(\vec{r})=\tilde{F}^n(Q_i(\vec{r}),Q_{i+}(\vec{r}),\vec{n}),\quad \vec{r}\in\partial V_i \tag{2.10}$$

where subscript $i+$ represent a neighboring element of cell i, and $\vec{n}$ is unit normal vector at the interface pointing out of cell i. Now let's try to reconstruct a set of new fluxes $\hat{F}^x$ and $\hat{F}^y$ which satisfies the following criteria;

- Both $\hat{F}^x$ and $\hat{F}^y$ are degree $k+1$ polynomials, i.e., one degree higher than $Q_i(\vec{r})$.
- At the element interfaces, the normal flux is equal to the Riemann flux computed in (2.10), i.e.,

$$(\hat{F}^x(\vec{r}),\hat{F}^y(\vec{r}))\cdot\vec{n}=\tilde{F}^n(\vec{r}),\quad \vec{r}\in\partial V_i \tag{2.11}$$

Comparing with the 1D formulation, the most difficult part is that the interface Riemann flux has to be approximated along the tangential direction because otherwise the flux can be a complex non-linear function of the solutions at cell i and its neighbors. One possible idea is to define another nodal set as the flux points, which can support a degree k+1 polynomial, as shown in Figure 2. If we impose (2.11) at all the interface flux points, we obtain 12 conditions. We still need 8 more conditions to determine the two fluxes $\hat{F}^x$ and $\hat{F}^y$ (we are assuming $k = 2$). We can impose the following conditions at 4 more points inside the cell

$$(\hat{F}^x,\hat{F}^y)=(F_i^x,F_i^y). \tag{2.12}$$

It appears the approach should work, but would be quite cumbersome and expensive. Instead, we pursue a more efficient approach noting that it is actually not necessary to reconstruct the flux. All we need to do is to find a degree k correction polynomial, i.e., the last two terms in the left-hand-side of (2.8). The following section describes the basic idea.

III. Lifting Collocation Penalty Formulation

3.1 Basic Idea

Rewrite the hyperbolic conservation law as

$$\frac{\partial Q}{\partial t} + \nabla \bullet \vec{F}(Q) = 0, \tag{3.1}$$

where $\vec{F} = (F^x, F^y)$ is the flux vector. Assume that the computational domain is discretized into *N* non-overlapping triangular elements $\{V_i\}$. Let *W* be an arbitrary weighting function. The weighted residual form of (3.1) on element V_i can be easily derived by multiplying (3.1) with *W* and integrating over V_i to obtain

$$\int_{V_i} \left(\frac{\partial Q}{\partial t} + \nabla \bullet \vec{F}(Q) \right) W dV = \int_{V_i} \frac{\partial Q}{\partial t} W dV + \int_{\partial V_i} W \vec{F}(Q) \bullet \vec{n} dS - \int_{V_i} \nabla W \bullet \vec{F}(Q) dV = 0. \tag{3.2}$$

Let Q_i^h be an approximate solution to *Q* at element *i*, and $Q_i^h \in P^k$ (the space of degree k or less polynomials, whose dimension is *m*), within each element without continuity requirement across element interfaces. In addition, we require that the numerical solution Q_i^h must also satisfy (3.2), i.e.,

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{\partial V_i} W \vec{F}(Q_i^h) \bullet \vec{n} dS - \int_{V_i} \nabla W \bullet \vec{F}(Q_i^h) dV = 0. \tag{3.3}$$

A common Riemann flux is used to replace the normal flux to provide element coupling, i.e.,

$$F^n(Q_i^h) \equiv \vec{F}(Q_i^h) \bullet \vec{n} \approx \widetilde{F}^n(Q_i^h, Q_{i+}^h, \vec{n}), \tag{3.4}$$

where Q_{i+}^h is the solution outside the current element V_i. Then (3.3) becomes

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{\partial V_i} W \widetilde{F}^n(Q_i^h, Q_{i+}^h, \vec{n}) dS - \int_{V_i} \nabla W \bullet \vec{F}(Q_i^h) dV = 0. \tag{3.5}$$

Applying integration by parts to the last term on the LHS of (3.5), we obtain

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{V_i} W \nabla \bullet \vec{F}(Q_i^h) dV + \int_{\partial V_i} W \left[\widetilde{F}^n(Q_i^h, Q_{i+}^h, \vec{n}) - F^n(Q_i^h) \right] dS = 0. \tag{3.6}$$

The last term on the left side of (3.6) can be viewed as a penalty term, i.e., penalizing the normal flux differences. Introduce a "correction field" $\delta_i \in P^k$, which is determined from a "lifting operator"

$$\int_{V_i} W \delta_i dV = \int_{\partial V_i} W [\tilde{F}] dS, \tag{3.7}$$

where $[\tilde{F}] = \tilde{F}^n(Q_i^h, Q_{i+}^h, \vec{n}) - F^n(Q^h)$ is the normal flux difference. Substituting (3.7) into (3.6), we obtain

$$\int_{V_i} \left[\frac{\partial Q_i^h}{\partial t} + \nabla \bullet \vec{F}(Q_i^h) + \delta_i \right] W dV = 0. \tag{3.8}$$

Because *W* is arbitrary, (3.8) is equivalent to

$$\frac{\partial Q_i^h}{\partial t} + \nabla \bullet \vec{F}(Q_i^h) + \delta_i = 0, \tag{3.9}$$

i.e., (3.9) is satisfied everywhere in element V_i. With the definition of a correction field δ_i, we have successfully reduced the weighted residual formulation to an equivalent simple differential form, which does not involve any explicit surface or volume integrals. The lifting operator obviously depends on the choice of weighting function. If $W \in P^k$, (3.9) is equivalent to the DG formulation.

The performance of this formulation of course hinges on how efficiently the correction field δ can be computed. To get a sense of its form, letting $W = 1$ in (3.7), we obtain

$$\int_{V_i} \delta_i dV = \int_{\partial V_i} [\widetilde{F}] dS, \tag{3.10}$$

or

$$\overline{\delta}_i = \frac{1}{|V_i|} \sum_{f \in \partial V_i} \int_f [\widetilde{F}] dS, \tag{3.11}$$

where $\overline{\delta}$ is the volume averaged δ. We can assume that the flux difference $[\widetilde{F}]$ is a degree k polynomial on face *f*, and can be determined based on values of $[\widetilde{F}]_{f,l}$ at a set of "flux points" using a Lagrange interpolation, as shown in Figure 3. Then the face integral can be computed exactly using a quadrature formula such as

$$\overline{\delta}_i = \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l w_l [\hat{F}]_{f,l} S_f, \tag{3.12}$$

where w_l is the quadrature weight for the surface integral, and S_f is the area of face *f*. Next let the degrees-of-freedom (DOFs) be the solutions at a set of points $\{\vec{r}_{i,j}\}$, named solution points (SPs), as shown in Figure 3. Then equation (3.9) must be true at the SPs, i.e.,

$$\frac{\partial Q^h_{i,j}}{\partial t} + \nabla \bullet \vec{F}(Q^h_{i,j}) + \delta_{i,j} = 0, \tag{3.13}$$

where $\nabla \bullet \vec{F}(Q^h_{i,j}) = \left[\nabla \bullet \vec{F}(Q^h_i)\right]_{\vec{r}_{i,j}}$.

Let's examine (3.7) more carefully. If $[\widetilde{F}]$ is assumed to be degree k polynomials along the cell faces, and that the triangle has straight faces, the correction field δ_i can be computed explicitly in the following formula

$$\delta_{i,j} = \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f, \tag{3.14}$$

where $\alpha_{j,f,l}$ are constant coefficients independent of the solution, but dependent on the weighting function *W*. Substituting (3.14) into (3.13) we obtain the following formulation

$$\frac{\partial Q^h_{i,j}}{\partial t} + \nabla \bullet \vec{F}(Q^h_{i,j}) + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f = 0 \cdot \tag{3.15}$$

Obviously, this is a collocation-penalty (CP) formulation. In addition, a lifting operator is used to completely remove the weighting functions, which are used to compute the constant "lifting coefficients" $\alpha_{j,f,l}$. We will therefore name this formulation the lifting CP or LCP formulation. If the flux vector is linear in *Q*, (3.15) is completely defined, and identical to any DG formulation. For non-linear flux vectors, one can approximate $\vec{F}(Q^h_{i,j})$

with polynomials, either degree k or k+1. For example, in a nodal DG formulation [19], and in Huynh's FR formulation [22], a degree *k* Lagrange interpolation polynomial is used

$$\vec{F}(Q_i^h) \approx \sum_k L_k(\vec{r}_{i,k})\vec{F}(Q_{i,k}^h), \tag{3.16}$$

where L_k is the Lagrange interpolation polynomial based on the flux points $\{\vec{r}_{i,k}\}$, and $Q_{i,k}^h = Q_i^h(\vec{r}_{i,k})$. Numerical tests for non-linear conservation laws indicated that there is a slight accuracy loss (half an order to one order) with this approach, which is named the **LP** (Lagrange polynomial) approach to compute the divergence of the flux vector. In a quadrature-free implementation of the SV method, the flux vector was approximated by a degree k+1 polynomial [16]. However this approach will significantly increase the cost of the LCP method. Therefore it is not pursued in the present study further. Instead, we come to realize that it is not necessary to make any approximations in computing $\nabla \bullet \vec{F}(Q_i^h)$, which can be exactly evaluated using the chain rule, i.e.,

$$\nabla \bullet \vec{F}(Q_i^h) = \frac{\partial F^x(Q_i^h)}{\partial x} + \frac{\partial F^y(Q_i^h)}{\partial y} = \frac{\partial F^x}{\partial Q}\frac{\partial Q_i^h}{\partial x} + \frac{\partial F^y}{\partial Q}\frac{\partial Q_i^h}{\partial y} = \frac{\partial \vec{F}}{\partial Q} \bullet \nabla Q_i^h . \tag{3.17}$$

where $\frac{\partial \vec{F}}{\partial Q}$ is the flux Jacobian matrix. This approach is called the **CR** (chain-rule) approach in evaluating the flux divergence. Note that this "exact" evaluation of flux divergence is only possible in a "collocation" type formulation, which may provide extra accuracy benefits comparing with the more traditional quadrature-based formulations.

Now that the LCP formulation is complete, let's examine (3.15) a bit further. With a special choice of lifting coefficients $\alpha_{j,f,l}$, (3.15) is equivalent to the DG method, at least in the linear case. From (3.7), it is easy to show that δ_i is uniquely defined once *W* and the flux points are given. Therefore, the location of the solution points does not affect the solution polynomial $Q_i^h(\vec{r})$ (although $Q_{i,j}^h$ obviously depends on the location of $\vec{r}_{i,j}$) or the correction polynomial for linear conservation laws. Therefore, the most efficient choice is to make the solution points coincide with the flux points because no data interpolations are then needed for flux computations. Any convergent nodal sets with enough points at the element interface are good candidates, e.g., those found in [7,17,48]. Figure 4 shows possible locations of the DOFs for k = 1, 2 and 3.

3.2 Connection between the LCP and DG, SV and SD Methods

Let's first express the solution and the correction in terms of the DOFs, i.e.,

$$Q_i^h = \sum_j L_j Q_{i,j}^h , \tag{3.18}$$

$$\delta_i = \sum_j L_j \delta_{i,j} , \tag{3.19}$$

where L_j is the Lagrange polynomial based on the solution points. In the DG method, the weighting function *W* is set to be one of the Lagrange polynomial L_j. Substituting *W* into (3.7), we obtain the following equations

$$\int_{V_i} L_k \sum_j L_j \delta_{i,j} dV = \sum_{f \in \partial V_i} \int_f L_k \sum_l L_l [\hat{F}]_{f,l} dS , \quad k = 1,..., m. \tag{3.20}$$

The unknowns in (3.20) $\delta_{i,j}$ can be easily solved in terms of the normal flux jumps at the flux points $[\hat{F}]_{f,l}$, and the coefficients $\alpha_{j,f,l}$ be determined, which are constant for any straight-sided triangles. In the case of $k = 1$, the coefficients for the first solution point are {2.5 0.5 -1.5 -1.5 0.5 2.5}. Therefore, the formula for the correction is

$$\delta_{i,1} = \frac{1}{|V_i|}\left[(2.5[\tilde{F}]_{1,1} + 0.5[\tilde{F}]_{1,2})S_1 + (-1.5[\tilde{F}]_{2,1} - 1.5[\tilde{F}]_{2,2})S_2 + (0.5[\tilde{F}]_{3,1} + 2.5[\tilde{F}]_{3,2})S_3\right]. \tag{3.21}$$

Although all the flux points coincide with the solution points, as shown in Figure 4, it is necessary to distinguish flux points according to which face they are located on because each face has a different normal direction. In addition, the flux points on each face are numbered independently for easy identification and implementation.

In the SV method, the weighting function is 1 within a partition of the element, and 0 elsewhere, for example, as shown in Figure 5 in the case of $k = 1$. Repeating the same with all the partitions, we obtain again m equations for m unknowns $\delta_{i,j}$, which can be uniquely solved. The coefficients for the first solution point of the second order SV method are {2 0.2 -0.7 -0.7 0.2 2}, corresponding to the following formula

$$\delta_{i,1} = \frac{1}{|V_i|}\left[(2[\tilde{F}]_{1,1} + 0.2[\tilde{F}]_{1,2})S_1 + (-0.7[\tilde{F}]_{2,1} - 0.7[\tilde{F}]_{2,2})S_2 + (0.2[\tilde{F}]_{3,1} + 2[\tilde{F}]_{3,2})S_3\right]. \tag{3.22}$$

In the SD method, the correction field is computed based on the direct differential of a reconstructed flux vector, i.e.,

$$\delta_i = \nabla \bullet [\hat{\tilde{F}} - \vec{F}(Q_i^h)], \tag{3.23}$$

where $\hat{\tilde{F}}$ is a reconstructed flux vector from the Riemann fluxes at the element interface and fluxes in the interior of the element. The components of $\hat{\tilde{F}}$ are polynomials of degree $k+1$, one degree higher than the solution polynomial, so that δ_i is of degree k. The derivation is a little more involved than those for the DG and SV method. We found that only on an equilateral triangular grid can the SD method degenerate into the LCP formulation given in (3.14). This is not surprising because the SD method is generally not only dependent on the normal fluxes at element interfaces, but also on the tangential fluxes. The $k = 1$ linear case has the following coefficients $\alpha_{j,f,l}$ at the first solution point {2 0 -0.5 -0.5 0 2}, resulting in the following formula

$$\delta_{i,1} = \frac{1}{|V_i|}\left[2[\tilde{F}]_{1,1}S_1 + (-0.5[\tilde{F}]_{2,1} - 0.5[\tilde{F}]_{2,2})S_2 + 2[\tilde{F}]_{3,2}S_3\right]. \tag{3.24}$$

Note that the coefficients are quite different for the DG, SV and SD methods. These schemes have been numerically confirmed to be equivalent to the DG, SV and SD methods for linear conservation laws.

Since the DG method is the most popular among the high-order methods for compressible flow, let's compare the LCP formulation with the DG method in terms of operations, in the case of k = 2.

Here are the main operations in the conventional DG methods with surface and volume quadratures:

- *Reconstruct the state variables at 6 volume quadrature points and 9 surface quadrature points.*
- *Compute the analytical fluxes at the 6 volume quadrature points, and Riemann fluxes at the 9 surface quadrature points.*
- *Multiply the fluxes with the quadrature weights, and the weighting functions or their gradients to form the residual;*
- *Multiply the residual by the inverse of the mass matrix.*

The main operations in the LCP method include:

- *Compute the analytical fluxes at the 6 solution points, and 9 Riemann fluxes at surface flux points.*
- *Multiply the fluxes with the differential quadrature weights to form the divergence of the flux vector, and add the penalty terms to form the residual.*

Due to the special choice of DOFs, the reconstruction cost is completely avoided, and the mass matrix is avoided in the LCP method. In addition, it is not necessary to store the weighting functions or their gradients. Therefore, the LCP formulation is more efficient in both memory and CPU time.

3.3 Conservation Constraints for Lifting Coefficients

It should not be a surprise that more choices of coefficients are possible, and may even be beneficial, for example, to allow the largest possible time step, or preserve certain range of frequency contents better than the standard methods. However, since we are dealing with conservation laws, the schemes should be conservative with any coefficients. In order to guarantee conservation, the integral conservation laws must be satisfies, i.e.,

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} dV + \int_{\partial V_i} \widetilde{F}^n(Q_i^h, Q_{i+}^h, \vec{n}) dS = 0, \tag{3.25}$$

which corresponds to (3.6) with $W = 1$. This means the LCP method is fundamentally conservative at the element level. Next let's derive the conservative constraints which must be satisfied by the lifting coefficients. Assume the following quadrature is used for the volume integral

$$\int_{V_i} Q_i^h dV = |V_i| \sum_j v_j Q_{i,j}^h, \tag{3.26}$$

where v_j are the quadrature weights for the volume integral. Then the first term in (3.25) becomes

$$\int_{V_i} \frac{\partial Q^h}{\partial t} dV = |V_i| \sum_j v_j \frac{\partial Q_{i,j}^h}{\partial t} = -|V_i| \sum_j v_j \left[\nabla \bullet \vec{F}(Q_{i,j}^h) + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f \right]. \tag{3.27}$$

Since the quadrature rule is exact for degree k (or less) polynomials, we have

$$|V_i| \sum_j v_j \nabla \bullet \vec{F}(Q_{i,j}^h) = \int_{V_i} \nabla \bullet \vec{F}(Q_i^h) dV = \int_{\partial V_i} F^n(Q_i^h) dS. \tag{3.28}$$

According to (3.25), we obtain

$$|V_i| \sum_j v_j \left[\frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f \right] = \int_{\partial V_i} \left[\widetilde{F}^n - F^n(Q^h) \right] dS = \sum_{f \in \partial V_i} \sum_l w_l [\widetilde{F}]_{f,l} S_f. \tag{3.29}$$

The LHS of (3.29) can be simplified into

$$\sum_j v_j \left[\sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f \right] = \sum_{f \in \partial V_i} \sum_l \sum_j v_j \alpha_{j,f,l} [\widetilde{F}]_{f,l} S_f. \tag{3.30}$$

Comparing (3.30) and (3.29), we obtain the following condition

$$w_l = \sum_j v_j \alpha_{j,f,l}. \tag{3.31}$$

This is also the necessary and sufficient condition for the LCP formulation to be conservative. In the linear case, $w_l = 1/2$ and $v_j = 1/3$, the conservation condition is then

$$\frac{3}{2} = \sum_{j=1}^{3} \alpha_{j,f,l}, \tag{3.32}$$

which is satisfied by the DG, SV and SD schemes.

3.4 Extension to Mixed Grids

It is obvious that (3.9) is valid for arbitrary types of elements besides triangles. For viscous flow problems, it is often advantageous to have quadrilateral cells or prismatic cell near solid walls to resolve viscous boundary layers. In the present study focusing on 2D problems, we address mixed grids including both triangular and quadrilateral meshes. As mentioned earlier, it is straightforward to extend the FR approach [22] to quadrilateral meshes. In order to simplify the implementation, we assume the polynomial degree k to be the same for both the triangular and quadrilateral meshes. Furthermore, the flux points along the element interfaces are required to match each other, as shown in Figure 6. In order to achieve an efficient implementation, all elements are transformed from the physical domain (x,y) into a standard square element $(\xi,\eta) \in$ [-1,1]x[-1,1] as shown in Figure 7. The transformation can be written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi,\eta) \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \tag{3.33}$$

where K is the number of points used to define the physical element, (x_i, y_i) are the Cartesian coordinates of those points, and $M_i(\xi,\eta)$ are the shape functions. For the transformation given in (3.33), the Jacobian matrix J takes the following form

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}. \tag{3.34}$$

For a non-singular transformation, its inverse transformation must also exist, and the Jocobian matrices are related to each other according to

$$\frac{\partial(\xi,\eta)}{\partial(x,y)} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = J^{-1}.$$

Therefore the metrics can be computed according to

$$\xi_x = y_\eta / |J|, \quad \xi_y = -x_\eta / |J|, \quad \eta_x = -y_\xi / |J|, \quad \eta_y = x_\xi / |J|. \tag{3.35}$$

The governing equations in the physical domain are then transformed into the computational domain (standard element), and the transformed equations take the following form

$$\frac{\partial \widetilde{Q}}{\partial t} + \frac{\partial F^\xi}{\partial \xi} + \frac{\partial F^\eta}{\partial \eta} = 0. \tag{3.36}$$

where

$$\widetilde{Q} = |J| \cdot Q \tag{3.37}$$

$$F^\xi = |J|(\xi_x F^x + \xi_y F^y) \tag{3.38}$$

$$F^\eta = |J|(\eta_x F^x + \eta_y F^y). \tag{3.39}$$

Let $\vec{S}_\xi = |J|(\xi_x, \xi_y)$, $\vec{S}_\eta = |J|(\eta_x, \eta_y)$. Then we have $F^\xi = \vec{F} \bullet \vec{S}_\xi, F^\eta = \vec{F} \bullet \vec{S}_\eta$. In our implementation, $|J|$ and $\vec{S}_\xi$, $\vec{S}_\eta$ are stored at the solution points. Within the *ith* element, the solution polynomial is a tensor product of 1D Lagrange polynomials, i.e.,

$$Q_i^h(\xi,\eta)=\sum_{l=1}^{k+1}\sum_{j=1}^{k+1}Q_{i;j,l}^h L_j(\xi)\cdot L_l(\eta)\,, \tag{3.40}$$

where $Q_{i;j,l}^h$ are the state variables at the solution point (*j,l*), with *j* the index in ξ direction and *l* the index in η direction, $L_j(\xi)$ and $L_l(\eta)$ are 1D Lagrange polynomials in ξ and η directions. Based on the reconstructed solution $Q_i^h(\xi,\eta)$, the fluxes can be defined using $F^\xi(Q_i^h), F^\eta(Q_i^h)$. Again, one can also choose to represent the fluxes with Lagrange interpolation polynomials in the following form:

$$F_i^\xi(\xi,\eta)=\sum_{l=1}^{k+1}\sum_{j=1}^{k+1}F_{i;j,l}^\xi L_j(\xi)\cdot L_l(\eta)\,, \tag{3.41a}$$

$$F_i^\eta(\xi,\eta)=\sum_{l=1}^{k+1}\sum_{j=1}^{k+1}F_{i;j,l}^\eta L_j(\xi)\cdot L_l(\eta)\,. \tag{3.41b}$$

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. Again Riemann fluxes are computed at all four element interfaces in the normal directions, which are the same or opposite directions of $\vec{S}_\xi$ or $\vec{S}_\eta$. For example, at interfaces $\xi=-1$, and $\xi=1$, the outgoing normals are

$$\vec{n}\big|_{\xi=-1}=-\vec{S}_\xi/\left|\vec{S}_\xi\right|. \tag{3.42a}$$

$$\vec{n}\big|_{\xi=1}=\vec{S}_\xi/\left|\vec{S}_\xi\right|. \tag{3.42b}$$

Therefore the Riemann flux corresponding to F^ξ is computed according to

$$\widetilde{F}^\xi(-1,\eta)=-\widetilde{F}^n(Q_i(-1,\eta),Q_{i+}(-1,\eta),\vec{n})\left|\vec{S}_\xi\right|, \tag{3.43a}$$

$$\widetilde{F}^\xi(1,\eta)=\widetilde{F}^n(Q_i(1,\eta),Q_{i+}(1,\eta),\vec{n})\left|\vec{S}_\xi\right|. \tag{3.43b}$$

Finally the DOFs are updated using the following equation

$$\begin{aligned}&\frac{\partial\widetilde{Q}_{i;j,l}}{\partial t}+\frac{\partial F_i^\xi(\xi_{j,l},\eta_{j,l})}{\partial\xi}+\frac{\partial F_i^\eta(\xi_{j,l},\eta_{j,l})}{\partial\eta}\\&+[\widetilde{F}^\xi(-1,\eta_{j,l})-F_i^\xi(-1,\eta_{j,l})]g_L'(\xi_{j,l})+[\widetilde{F}^\xi(1,\eta_{j,l})-F_i^\xi(1,\eta_{j,l})]g_R'(\xi_{j,l})\\&+[\widetilde{F}^\eta(\xi_{j,l},-1)-F_i^\eta(\xi_{j,l},-1)]g_L'(\eta_{j,l})+[\widetilde{F}^\eta(\xi_{j,l},1)-F_i^\eta(\xi_{j,l},1)]g_R'(\eta_{j,l})=0.\end{aligned} \tag{3.44}$$

Note that, the correction is done in a "one dimensional" manner. For example, $[\widetilde{F}]_{f,2}$ in Figure 6 only corrects the DOFs at the solid squares. In other words, for quadrilateral cells, the operations are actually one-dimensional, making the method more efficient per DOF than for triangular cells. The flux divergence in (3.41) can be obtained using either the LP or the CR approach. Numerical tests show that the CR approach gives more accurate and consistent results.

In the present implementation for quadrilateral cells, we choose either the DG or the g_2 schemes from the FR family [22]. The g_2 scheme is the simplest possible. For all interior solution point, the correction vanishes. In the case shown in Figure 6, the correction for $[\widetilde{F}]_{f,2}$ needs only to apply to the solution point coinciding with the flux location! Recall that the solution points for the g_2 approach are the Legendre-Lobatto points. Therefore, the solution

points for all triangular elements at each edge are also Legendre-Lobatto points along the edge tangential direction to make the interface treatment as simple as possible.

3.5 Extension to Curved Boundary Cells

In second-order CFD solvers, curved boundaries are represented with linear line segments or planar facets. However, for high order solvers, this simple representation may not be able to preserve the geometry well enough since the computational grid is normally much coarser, and may even make the solvers unstable. The approach to solve the problem is to represent the curved boundary with higher order polynomials instead of the linear representation for second order solvers.

The change of boundary representations actually does not alter the solution technique, since all we need to do is to transform the curved boundary cell into a standard cell. For example, for a boundary element shown in Figure 8, a quadratic transformation is used to transform it to a standard element. After the transformation, the governing equation becomes (3.36) on the standard element, which has straight edges. The LCP formulation is then applied to the transformed equation on the standard right triangle. Eq. (3.15) for the transformed equation on a standard triangle becomes

$$\frac{\partial \widetilde{Q}_{i,j}^h}{\partial t} + \nabla^\xi \bullet \vec{F}^\xi(\widetilde{Q}_{i,j}^h) + 2\sum_{f\in\partial V_i}\sum_l \alpha_{j,f,l}[\widetilde{F}^\xi]_{f,l} S_f^\xi = 0, \tag{3.45}$$

where superscript ξ means the variables or operations are evaluated on the computational domain. For example, $[\widetilde{F}^\xi]_{f,l}$ are the normal jumps of the transformed fluxes across the faces of the standard triangle.

Note that when solving Equation (3.36), $\widetilde{Q} = |J| \cdot Q$ is the state vector, and is assumed to be a degree k polynomial in the computational domain instead of Q. As a result, the derivative of Q should be calculated in the following way

$$\frac{\partial Q}{\partial \xi} = \frac{1}{|J|}\left[\frac{\partial(|J|Q)}{\partial \xi} - \frac{\partial |J|}{\partial \xi} Q\right], \tag{3.46a}$$

$$\frac{\partial Q}{\partial \eta} = \frac{1}{|J|}\left[\frac{\partial(|J|Q)}{\partial \eta} - \frac{\partial |J|}{\partial \eta} Q\right]. \tag{3.46b}$$

Since the transformation is quadratic, $\frac{\partial |J|}{\partial \xi}$ and $\frac{\partial |J|}{\partial \eta}$ are non-zero. It is obvious that when the cell has straight faces, $|J|$ is a constant in the element. Then (3.46) reduces to the usual derivative evaluation.

For a quadrilateral cell, the curved boundary cells are treated in a similar manner. Each cell is transformed to a standard square with a high-order transformation, and then (3.36) is solved in the computational domain on a standard element.

IV. Numerical Results

4.1 Accuracy Study with 2D Scalar Conservation Laws

Linear Wave Equation

In this case, we test the accuracy of the L*CP* method using the two-dimensional linear equation:

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \qquad -1 \le x \le 1, \quad -1 \le y \le 1, \\ &u(x, y, 0) = u_0(x, y), \qquad \textit{periodic boundary condition.} \end{aligned} \tag{4.1}$$

The initial condition is $u_0(x, y) = \sin \pi(x + y)$. Two types of computational meshes are used, one regular and the other irregular as shown in Figure 9. The finer meshes are generated recursively by cutting each coarser grid cell into four finer grid cells. The time integration schemes used are the TVD or SSP Runge-Kutta schemes of third or

4th order accuracy [35,15,36]. The results are made time step independent so that the spatial error is the dominant error source. The numerical simulation is carried until t = 1. The simulation was initialized by injecting the exact initial solution into the solution points. The numerical solutions at the three vertices of all the triangular elements are then used to compute the L_2 error norm to measure the global solution quality

$$Error_{L2} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{3}(Q_{i,j}^{h} - Q(\vec{r}_{i,j}))^2}{3N}}, \tag{4.2}$$

where $Q(\vec{r})$ denotes the exact solution. For linear wave equations, all DG formulations are identical, so are all SV formulations. In the present LCP family, the DG formulation is denoted by LCP-DG. Table 1 and Table 2 present the computed errors with both formulations on the regular and irregular meshes. Note that optimal accuracy has been achieved for all the test cases, on both the regular and irregular meshes, and the DG method has produced lower error magnitude.

2D Burgers Equation

In this case, we test the accuracy of the *LCP* method on the two-dimensional non-linear wave equation:

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial u^2/2}{\partial x} + \frac{\partial u^2/2}{\partial y} = 0, \qquad -1 \le x \le 1, \quad -1 \le y \le 1, \\ &u(x,y,0) = \frac{1}{4} + \frac{1}{2}\sin\pi(x+y), \qquad \textit{periodic boundary condition.} \end{aligned} \tag{4.3}$$

The initial solution is smooth. Due to the non-linearity of the Burgers equation, discontinuities will develop in the solution. At t = 0.1, the exact solution is still smooth. The numerical simulation is therefore carried out until t = 0.1 on the irregular grids. In Table 3, we present the L_2 errors produced using the LCP-DG formulation. Both the LP (Lagrange polynomial) approach and CR (chain-rule) approach are used in evaluating the interior flux derivatives and the accuracy of these two approaches is compared. Note in Table 3 that the CR approach is better in every case, since it not only produces smaller error, but also shows more consistent and better accuracy in grid refinement studies.

4.2 Accuracy Study with Vortex Evolution Problem

This is an idealized problem for the Euler equations in 2D used by Shu [20]. The mean flow is $\{\rho, u, v, p\} = \{1, 1, 1, 1\}$. An isotropic vortex is then added to the mean flow, i.e., with perturbations in u, v, and temperature $T = p/\rho$, and no perturbation in entropy $S = p/\rho^{\gamma}$:

$$\begin{aligned} &(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-y, x), \\ &\delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \\ &\delta S = 0, \end{aligned}$$

where $r^2 = x^2 + y^2$, and the vortex strength $\varepsilon = 5$. If the computational domain is infinitely big, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with the mean velocity (1, 1). In the numerical simulation, the computational domain is taken to be [-5, 5]x[-5, 5], with characteristic inflow and outflow boundary conditions imposed on the boundaries.

The numerical simulations were carried out until t = 2 on two different grids, one irregular triangular mesh and one mixed mesh as shown in Figure 9 and Figure 10. The finer irregular grids are generated recursively by cutting each coarser grid cell into four finer grid cells, while all mixed meshes are generated independently, which is mainly composed of irregular quadrilateral cell, with a small number of triangular cells. For irregular triangular mesh, two tests are performed. In Test 1, the interior flux derivatives are evaluated using the LP (Lagrange polynomial)

approach, while in Test 2, these derivatives are evaluated by the CR (chain rule) approach described in Section 3.1. For the mixed mesh, only the latter approach is used. The LCP-DG scheme is used for triangular cells while the FR-DG scheme is used for quadrilateral cells. For time integration, a 3-stage Runge-Kutta explicit scheme is used for time marching in all the cases.

In Table 4, the L_2 norms of density error at the solution points are presented for all three tests from 2nd order to 4th order. Comparing the results in Table 4, we find that the CR approach is much more accurate in every case. The CR approach not only produced the smaller errors, but also demonstrated more consistent order of accuracy in the grid refinement studies. In addition, the LCP method performs very well on the mixed grids, achieving the optimal order of accuracy on relatively poor quality meshes.

Because the CR approach in evaluating the interior flux derivatives is shown superior to the LP approach, it is used in all other test cases presented next.

4.3 Flow over a Cylinder

To test the new method for curved wall boundaries, the case of a subsonic flow at Mach 0.3 around a half-cylinder is simulated. For the curve wall, a quadratic wall representation is used. Three grids are used, one triangular mesh, and two mixed mesh as shown in Figure 11(a), 12(a) and 13(a). The first mixed mesh is composed of regular quadrilateral cells near the wall boundary and irregular triangular cells elsewhere, while the second mixed mesh is composed of fully mixed irregular quadrilateral and triangular cells, and has poor grid quality near the wall.

The 4th order LCP-DG scheme is used for triangular cells and the 4th order FR-DG scheme used for the quadrilateral cells. For time integration, a BLU-SGS implicit solver is used with 1st order time accuracy, and all the cases converged to machine zero.

The 4th order results for the 3 grids are shown in Figures 11, 12 and 13. Note that smooth solutions are obtained with all the grids, even on Mixed Mesh #2, and all the solutions agree well with each other. The results show that the LCP method performs well for mixed mesh with curve boundary elements, since the solutions are smooth across triangular and quadrilateral cells. Furthermore, the solution is also smooth near the curved boundary.

4.4 Flow over NACA 0012 Airfoil

As the final test case for a more realistic geometry, the subsonic flow around a NACA 0012 airfoil is simulated with the LCP method, with 2nd, 3rd and 4th order accuracy. The flow conditions are Mach = 0.3 and angle of attack α = 5 degrees. Again curved boundaries are represented with quadratic polynomials. The mesh, shown in Figure 14, is composed of regular quadrilateral mesh near the airfoil while irregular triangular and quadrilateral mesh elsewhere, with a total of 1,938 cells. The outer boundary of the computational domain is about 20 chords from the center of the airfoil. Again, the LCP-DG scheme is used for triangular cells and FR-DG is used for quadrilateral cells. A BLU-SGS solver is used for time integration, and all the cases converge to machine zero.

The pressure and Mach number solutions from the 2nd, 3rd and 4th order simulation are shown in Figure 15-17. The computed solution for the 2nd order case is not smooth, with visible jump across cell boundaries, due to the coarse mesh, while in the 3rd and 4th order cases, the solution is quite smooth. In addition, the 3rd and 4th order results are almost the same with only minor differences near the trailing edge. In Figure 18, the entropy errors along the upper surface of the airfoil is plotted for the 2nd, 3rd and 4th order results as an indicator of the accuracy. Although the entropy error does show a decrease with p-refinement, the convergence rate is not exponential probably due to the non-smoothness of the geometry and solution near the trailing edge.

V. Conclusions

In the present study, a remarkably simple and efficient flux reconstruction formulation developed by Huynh for conservation laws is extended and generalized to simplex cells. Instead of reconstructing the flux polynomial, the correction polynomial is reconstructed using a lifting operator. The new formulation is therefore called a *lifting collocation penalty* method. The final formulation is free of weighting functions, although the lifting coefficients are dependent on the weighting functions. Through a judicious selection of solution and flux points, solution reconstructions can be completely avoided. In addition, the mass matrix is always identity for arbitrary meshes. Furthermore, like the FR formulation, the discontinuous Galerkin, spectral volume, and in a special case the spectral difference method can all be recovered by the present formulation with different choices of lifting coefficients. The present formulation is more efficient than the original DG formulation based on volume and surface integral quadratures. For non-linear flux functions, the LCP method allows the interior flux derivatives to be computed "exactly" using the chain rule approach, which appears to be beneficial in solution accuracy. The extension to curved boundaries can be conducted in a straightforward manner because no surface or volume integrals are

required. Numerical tests have been carried out for linear, nonlinear scalar conservation laws, and the 2D Euler equations, for mixed triangular and quadrilateral meshes. These tests fully demonstrate the capability of the LCP formulation. The extension to the Navier-Stokes equations, and to 3D mixed meshes including tetrahedral, prismatic, pyramidal, and hexahedral cells is currently carried out, and will be reported in future publications.

Acknowledgements

The study was funded by AFOSR grant FA9550-06-1-0146, and 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, DOE, or the U.S. Government.

References

[1] R. Abgrall and P.L. Roe, High-order fluctuation splitting schemes on triangular mesh. J Sci Comput 2003;19:3–36.

[2]T.J. Barth and P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction," AIAA Paper No. 90-0013, 1990.

[3]F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138, 251-285 (1997).

[4]F. Bassi and S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. J Comp Phys 1997;131(1):267–79.

[5]M.H. Carpenter, and D. Gottlieb, Spectral methods on arbitrary grids. J Comp Phys 1996;129(1):74–86.

[6]J. Casper and H.L. Atkins, A finite volume high-order ENO scheme for two-dimensional hyperbolic systems, J. Comput. Phys. 106, 62-76 (1993).

[7]Q. Chen and I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Engrg, 128, 405-417 (1995).

[8]B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation 52, 411-435 (1989).

[9]B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84, 90-113 (1989).

[10]B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Garlerkin method for conservation laws V: multidimensional systems, J. Comput. Phys., 141, 199 - 224, (1998).

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

[12] J.A. Ekaterinaris, High-order accurate, lownumerical diffusion methods for aerodynamics, Progress in Aerospace Sciences 41 (2005) 192–300.

[13]S.K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47, 271 (1959).

[14]D. Gottlieb, S. A. Orszag. Numerical Analysis of Spectral Method: Theory and Applications. Philadelphia: SIAM, 1977

[15] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes. Math Comput 1998;67:73–85.

[16]R. Harris, Z.J. Wang and Y. Liu, "Efficient Quadrature-Free High-Order Spectral Volume Method on Unstructured Grids: Theory and 2D Implementation," Journal of Computational Physics, Volume 227, No. 3, pp. 1620-1642 (2008).

[17] 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.

[18]J.S. Hesthaven and D. Gottlieb, Stable spectral methods for conservation laws on triangles with unstructured grids, Comput. Methods Appl. Mech. Engin. Vol. 175, pp. 361-381 (1999).

[19]J.S. Hesthaven and Tim Warburton, Nodal Discontinuous Galerkin Methods, Springer, 2008.

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

[21]T.J.R. Hughes, 1987. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. Int. J. Numer. Methods Fluids 7:1261–75.

[22]H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper 2007-4079.

[23]A. Jameson, 1994. Analysis and design of numerical schemes for gas dynamics. I. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. Int. J. Comput. Fluid Dyn. 4:171–218.

[24]G.E. Karniadakis, and S.J. Sherwin, Spectral-hp element methods. Oxford: Oxford University Press; 1999.

[25]D.A. Kopriva and J.H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys. 125, 244 (1996).

[26] M-S. Liu, Mass flux schemes and connection to shock instability, J. Comput. Phys. 160, 623-648 (2000).

[27]Y. Liu, Vinokur M, and Wang ZJ. "Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids," in Proceedings of the 3rd International Conference on Computational Fluid Dynamics, Toronto, Canada, July 12-16 2004.

[28]Y. Liu, Vinokur M, and Wang ZJ. "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems," Journal of Computational Physics Vol. 212, pp. 454-472 (2006).

[29]Y. Liu, Vinokur M, and Wang ZJ. "Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids," Journal of Computational Physics Vol. 216, pp. 780-801 (2006).
[30]G. May and A. Jameson, "A spectral difference method for the Euler and Navier-Stokes equations", AIAA paper No. 2006-304, 2006.
[31]C.R. Nastase and D.J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, Journal of Computational Physics 213 (2006) 330–357.
[32]S. Osher, Riemann solvers, the entropy condition, and difference approximations, SIAM J. on Numerical Analysis 21, 217-235 (1984).
[33]P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 357-372 (1981).
[34]V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1, (1961), pp. 261-279.
[35]C.-W. Shu, Total-Variation-Diminishing time discretizations, SIAM Journal on Scientific and Statistical Computing 9, 1073-1084 (1988).
[36]R.J. Spiteri and S.J. Ruuth, A new class of optimal high-order strong-stabilitypreserving time discretization methods. SIAM J. Numer. Anal. 40, 469–491(2002).
[37]Y. Sun and Z.J. Wang, Evaluation of discontinuous Galerkin and spectral volume methods for 2d Euler equations on unstructured grids, AIAA-2003-3680.
[38]K. Van den Abeele and C. Lacor, An accuracy and stability study of the 2D spectral volume method, Journal of Computational Physics, **226** (1), 1007–1026, 2007.
[39]K Van den Abeele, C. Lacor and Z.J. Wang, "On the Stability and Accuracy of the Spectral Difference Method," Journal of Scientific Computing, Volume 37, pp. 162–188, 2008.
[40]B. van Leer, Towards the ultimate conservative difference scheme V. a second order sequel to Godunov's method, J. Comput. Phys. 32, 101-136 (1979).
[41]B. Van Leer and S. Nomura, Discontinuous Galerkin for diffusion, AIAA Paper No. 2005-5108, 2005.
[42]V. Venkatakrishnan, S.R. Allmaras, D.S. Kamenetskii and F.T. Johnson, Higher order schemes for the compressible Navier-Stokes equations, AIAA-2003-3987.
[43]Z.J. Wang, Spectral (Finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. Vol. 178, pp. 210-251, 2002.
[44]Z.J. Wang, High-order methods for the Euler and Navier-Stokes equations on unstructured grids, Journal of Progress in Aerospace Sciences, Vol. 43, pp. 1-47, 2007.
[45]Z.J. Wang, and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids ii: extension to two-dimensional scalar equation, J. Computational Physics, Vol. 179, pp. 665-697, 2002.
[46]Z.J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids iii: one-dimensional systems and partition optimization, Journal of Scientific Computing, Vol. 20, No. 1, pp. 137-157 (2004).
[47]Z.J. Wang, L. Zhang and Y. Liu, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids IV: Extension to Two-Dimensional Euler Equations, Journal of Computational Physics Vol. 194, No. 2, pp. 716-741 (2004).
[48]T. Warburton, An explicit construction of interpolation nodes on the simplex, J. of Engineering Mathematics, Vol. 56, pp. 247-262, 2006.
[49]M Zhang, C-W. Shu, An analysis and a comparison between the discontinuous Galerkin method and the spectral finite volume methods. Comput Fluids 2005;34(4–5):581–92.

Table 1. Test of LCP-DG for $u_t + u_x + u_y = 0, with\ u_0(x,y) = \sin\pi(x+y), at\ t = 1$, on triangular mesh.

Polynomial degree k	*Grid size*	Regular Mesh		Irregular Mesh	
		L_2 *error*	Order	L_2 *error*	Order
1	10x10x2	2.44e-2	-	4.45e-2	-
	20x20x2	5.89e-3	2.05	1.05e-2	2.08
	40x40x2	1.46e-3	2.01	2.57e-3	2.03
	80x80x2	3.64e-4	2.00	6.41e-4	2.00
2	10x10x2	1.88e-3	-	3.99e-3	-
	20x20x2	2.38e-4	2.98	5.14e-4	2.96
	40x40x2	2.98e-5	3.00	6.47e-5	2.99
	80x80x2	3.73e-6	3.00	8.10e-6	3.00
3	10x10x2	7.55e-5	-	2.59e-4	-
	20x20x2	4.94e-6	3.93	1.59e-5	4.03
	40x40x2	3.08e-7	4.00	9.91e-7	4.00
	80x80x2	1.93e-8	4.00	6.19e-8	4.00
5	10x10x2	7.53e-8	-	5.87e-7	-
	20x20x2	1.18e-9	6.00	9.22e-9	5.99
	40x40x2	1.85e-11	6.00	1.43e-10	6.01

Table 2. Test of CP-SV for $u_t + u_x + u_y = 0, with\ u_0(x,y) = \sin\pi(x+y), at\ t = 1$ for triangular mesh

Polynomial degree k	*Grid size*	Regular Mesh		Irregular Mesh	
		L_2 *error*	Order	L_2 *error*	Order
1	10x10x2	5.94e-2	-	1.01e-1	-
	20x20x2	1.45e-2	2.03	2.62e-2	1.95
	40x40x2	3.72e-3	1.96	6.55e-3	2.00
	80x80x2	9.23e-4	2.01	1.63e-3	2.01
2	10x10x2	2.84e-3	-	7.47e-3	-
	20x20x2	3.71e-4	2.94	9.09e-4	3.04
	40x40x2	4.73e-5	2.97	1.13e-4	3.01
	80x80x2	5.97e-6	2.99	1.42e-5	2.99
3	10x10x2	1.04e-4	-	4.37e-4	-
	20x20x2	6.53e-6	3.99	2.58e-5	4.08
	40x40x2	4.11e-7	3.99	1.56e-6	4.05
	80x80x2	2.57e-8	4.00	9.61e-8	4.02

Table 3. Test of LCP-DG for $u_t + uu_x + uu_y = 0, with\ u_0(x,y) = 0.25 + 0.5\sin\pi(x+y), at\ t = .1$, on triangular mesh. The LP and CR approaches to compute the interior flux divergence are compared.

Polynomial degree k	*Grid size*	Irregular Mesh (LP)		Irregular Mesh (CR)	
		L_2 *error*	Order	L_2 *error*	Order
1	10x10x2	2.65e-2	-	1.84e-2	-
	20x20x2	9.96e-3	1.41	5.06e-3	1.86
	40x40x2	3.75e-3	1.41	1.35e-3	1.91
	80x80x2	1.38e-3	1.44	3.50e-4	1.95
2	10x10x2	6.40e-3	-	2.75e-3	-
	20x20x2	1.37e-3	2.20	4.04e-4	2.77
	40x40x2	2.81e-4	2.29	5.50e-5	2.88
	80x80x2	5.43e-5	2.37	7.27e-6	2.92
3	10x10x2	9.59e-4	-	3.68e-4	-
	20x20x2	1.05e-4	3.19	2.58e-5	3.83
	40x40x2	9.86e-6	3.41	1.82e-6	3.83
	80x80x2	8.48e-7	3.54	1.27e-7	3.84
5	10x10x2	3.46e-5		1.07e-5	-
	20x20x2	1.15e-6	4.91	2.61e-7	5.35
	40x40x2	3.15e-8	5.19	4.45e-9	5.87
	80x80x2	7.08e-10	5.48	8.27e-11	5.75

Table 4. Test of LCP-DG for Euler equations with the vortex propagation case, for triangular and mixed meshes

Polynomial degree k	*Grid size*	Irregular Triangular Mesh - Test 1 (LP)		Irregular Triangular Mesh - Test 2 (CR)		Mixed Mesh (CR)	
		L_2 *error*	Order	L_2 *error*	Order	L_2 *error*	Order
1	10x10x2	2.01e-2	-	1.39e-2	-	1.58e-2	-
	20x20x2	6.67e-3	1.59	4.41e-3	1.66	5.32e-3	1.57
	40x40x2	1.73e-3	1.95	1.08e-3	2.03	1.50e-3	1.83
	80x80x2	4.84e-4	1.84	2.54e-4	2.09	3.54e-4	2.08
2	10x10x2	7.14e-3	-	4.41e-3	-	2.95e-3	-
	20x20x2	1.07e-3	2.74	5.19e-4	3.09	5.62e-4	2.39
	40x40x2	1.60e-4	2.74	5.84e-5	3.15	7.42e-5	2.92
	80x80x2	2.29e-5	2.80	6.94e-6	3.07	8.63e-6	3.10
3	10x10x2	1.79e-3	-	6.70e-4	-	5.79e-4	-
	20x20x2	1.40e-4	3.68	4.79e-5	3.81	5.05e-5	3.52
	40x40x2	9.75e-6	3.84	2.96e-6	4.02	3.51e-6	3.85
	80x80x2	6.96e-7	3.81	1.71e-7	4.11	1.89e-7	4.22

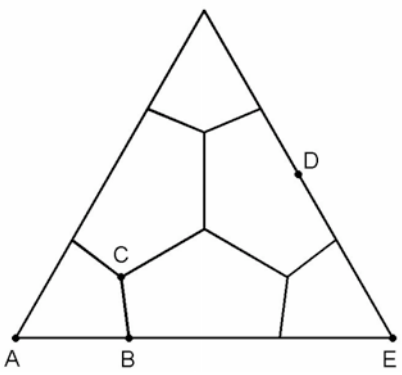


Figure 1. The partition of a triangular spectral volume into six control volumes, which support a unique quadratic data reconstruction in the spectral volume

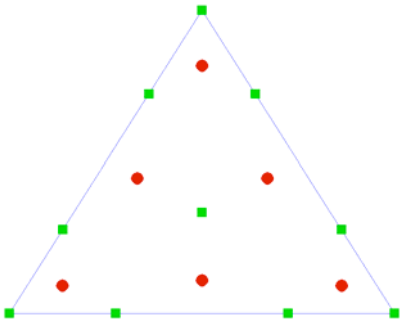


Figure 2. Solution (solid circles) and flux points (solid squares) for the 3rd order spectral difference method

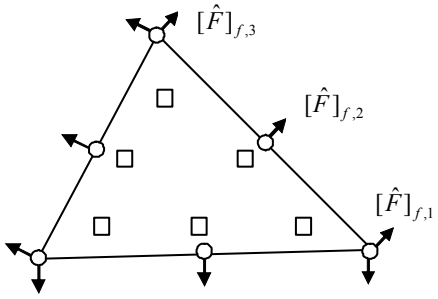
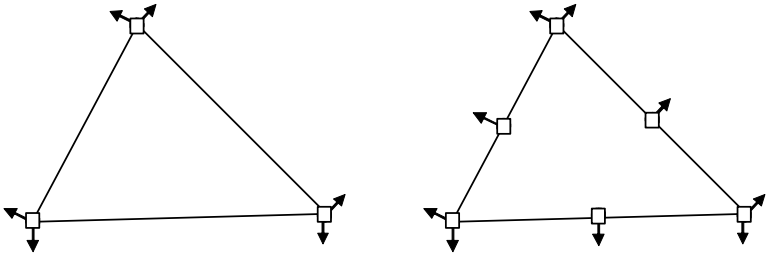
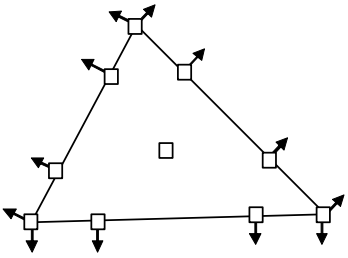


Figure 3. Solution points (squares) and flux points (circles) for k = 2



(a) k = 1 (b) k = 2



(c) k = 3

Figure 4. Solution and flux points in a simplex for k = 1, 2, and 3

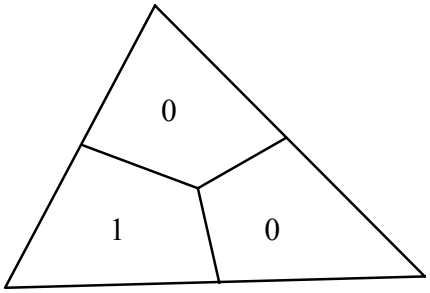


Figure 5. One of the weighting functions for the spectral volume method, k = 1.

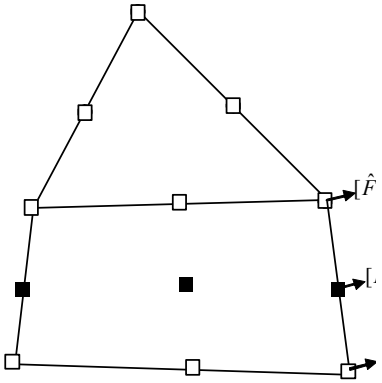


Figure 6. Solution and flux points for the 3rd order CP scheme on hybrid meshes

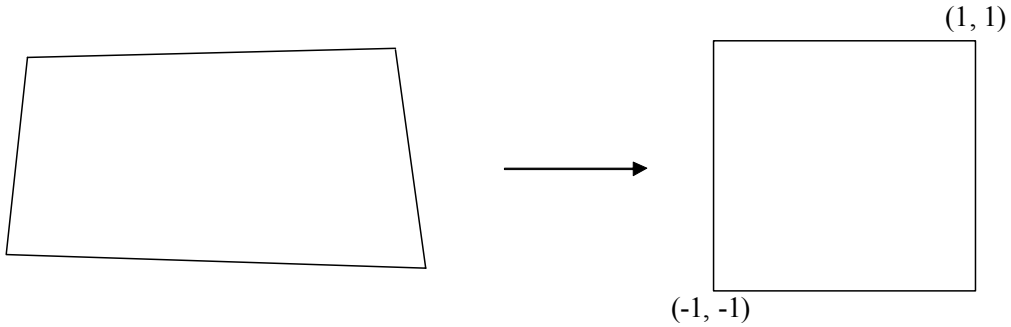


Figure 7. Transformation of a quadrilateral element to a standard element

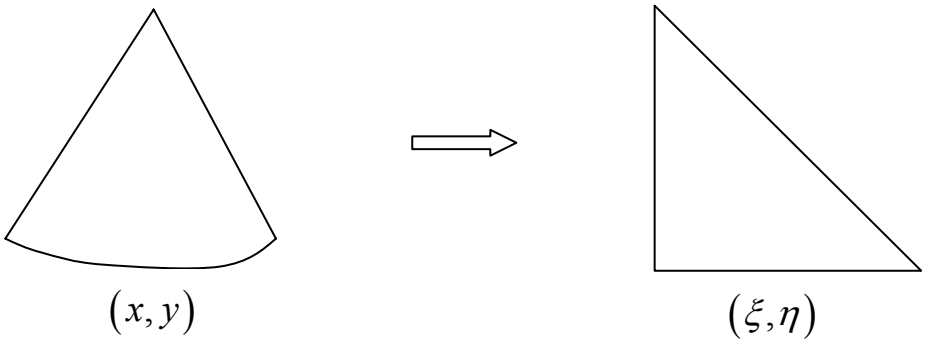
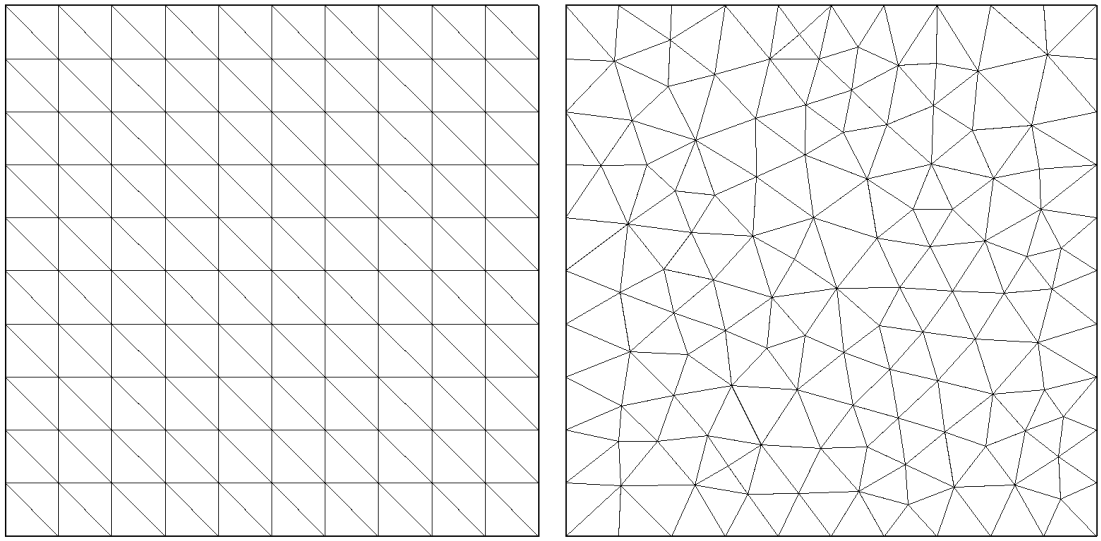


Figure 8. Transformation of a Curve Boundary Triangular Element to a Standard Element



(a) (b)

Figure 9. Regular and Irregular "10x10x2" Computational Grids

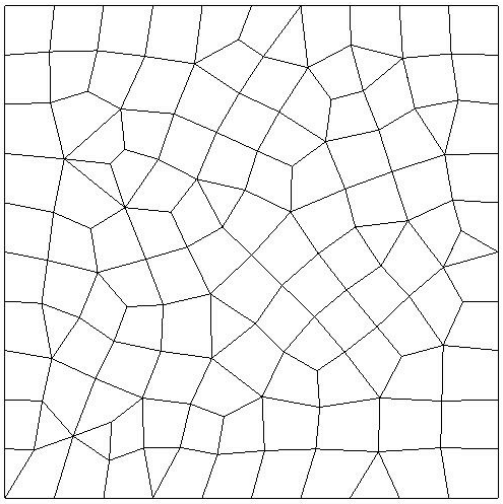
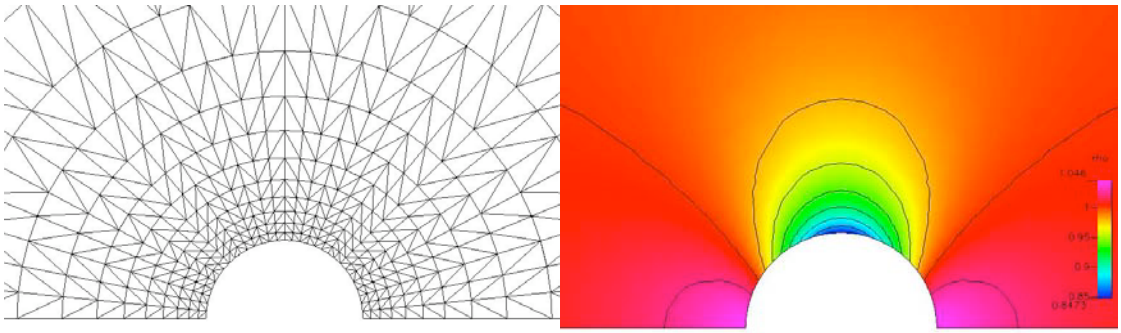
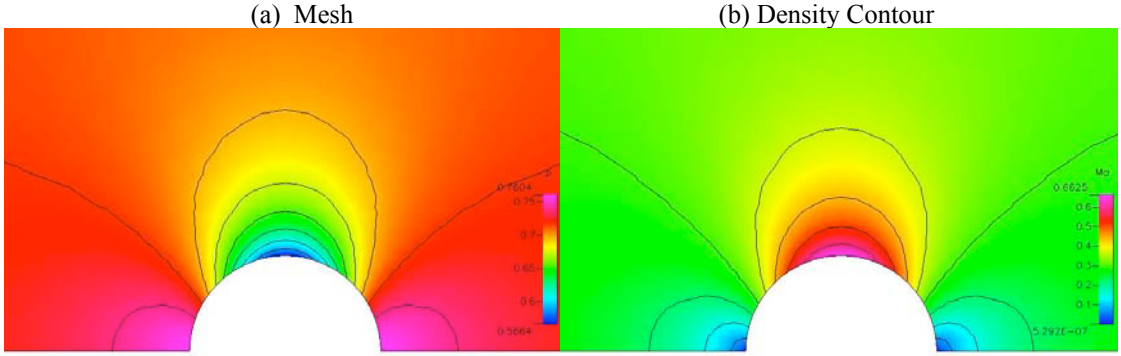


Figure 10. Coarse mixed mesh for accuracy study

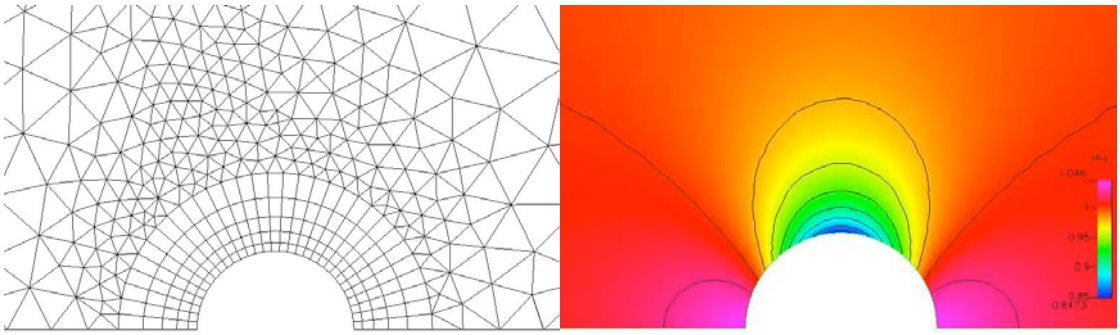


(a) Mesh (b) Density Contour

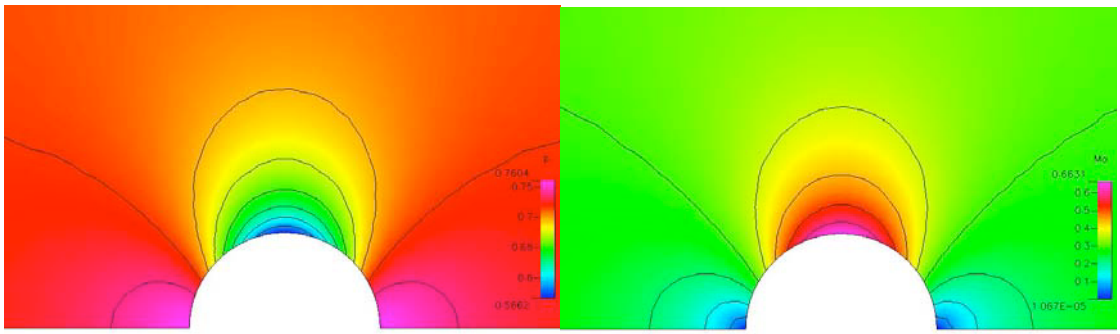


(c) Pressure Contour (d) Mach Number Contour

Figure 11. Triangular Mesh and 4th Order Results for Flow around a Cylinder

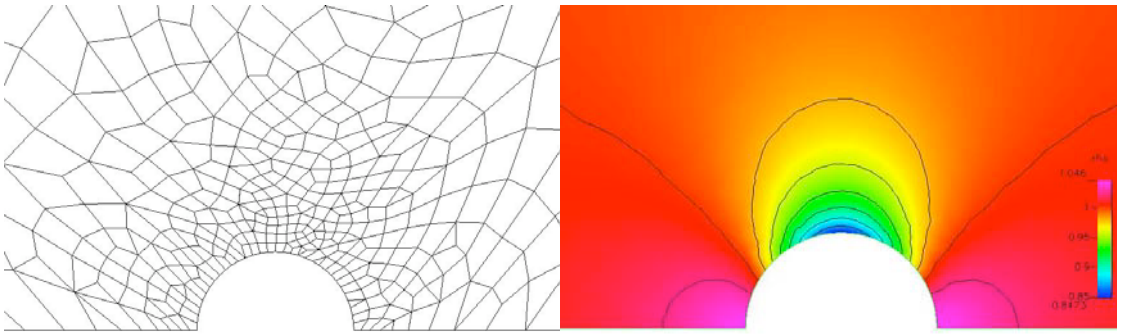


(a) Mesh (b) Density Contour

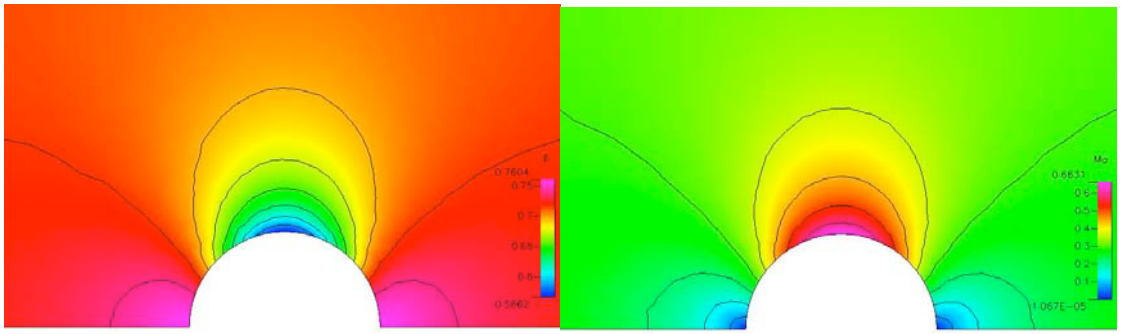


(b) Pressure Contour (d) Mach Number Contour

Figure 12. Mixed Mesh #1 and 4th Order Results for Flow around a Cylinder



(a) Mesh (b) Density Contour



(c) Pressure Contour (d) Mach Number Contour

Figure 13. Mixed Mesh #2 and 4th Order Results for Flow around a Cylinder

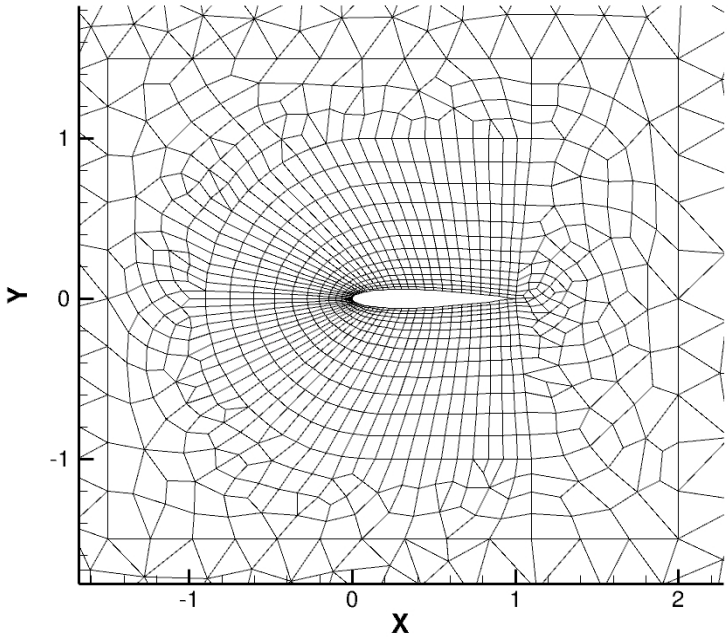
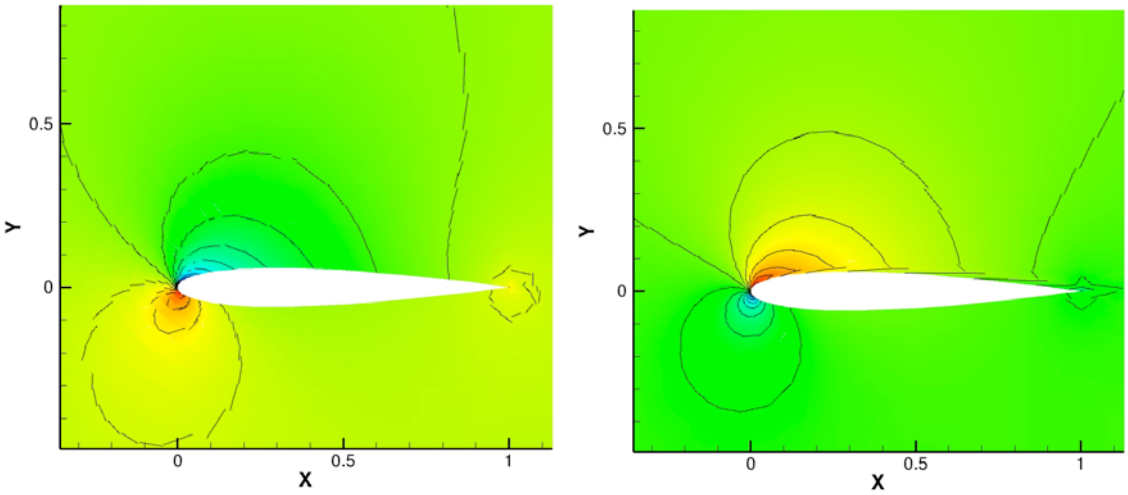
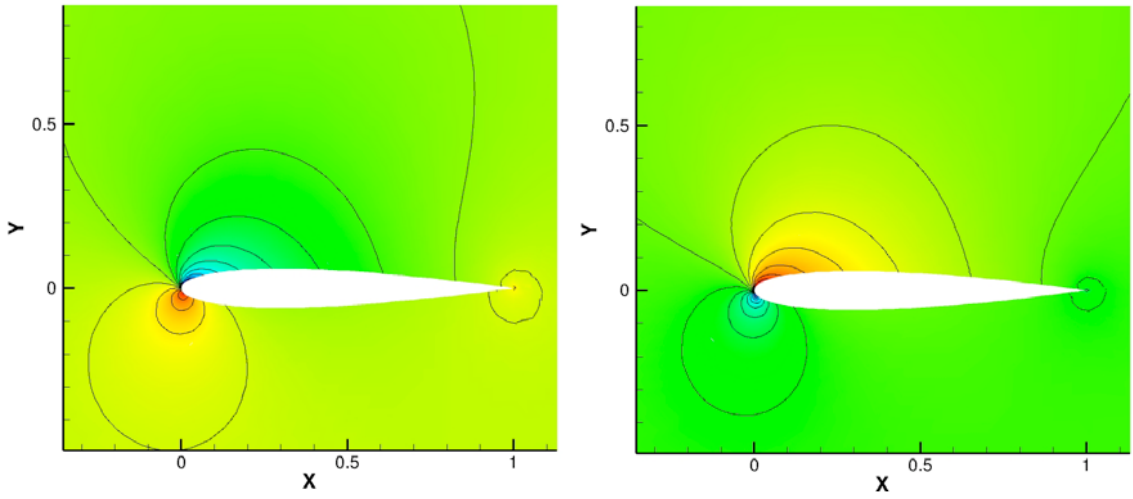


Figure 14. The Mixed Grid in the Simulation of Flow around NACA0012 Airfoil



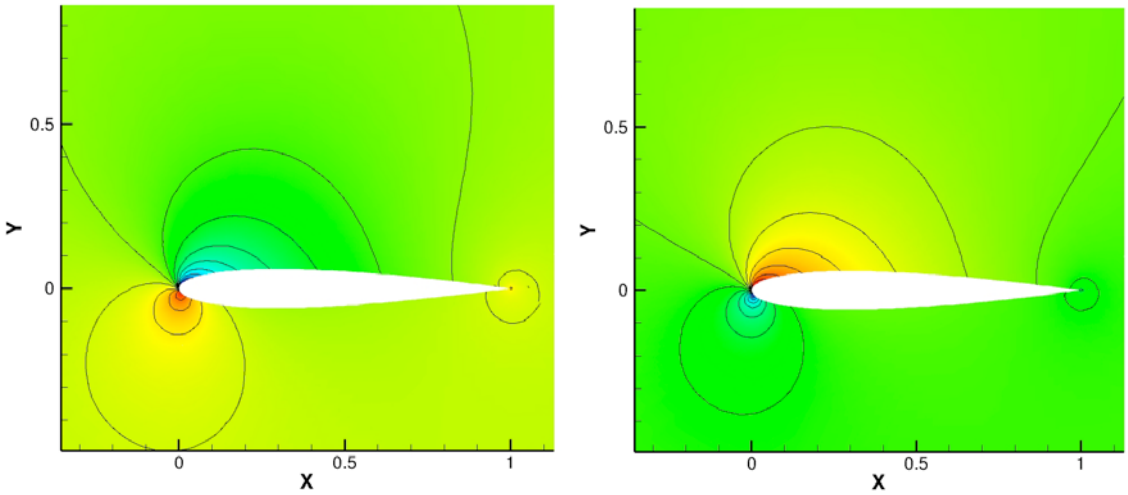
(a) Pressure Contour (b) Mach Number Contour

Figure 15. 2nd Order Solution of the Flow around NACA0012 Airfoil



(a) Pressure Contour (b) Mach Number Contour

Figure 16. 3rd Order Solution of the Flow around NACA0012 Airfoil



(a) Pressure Contour (b) Mach Number Contour

Figure 17. 4th Order Solution of the Flow around NACA0012 Airfoil

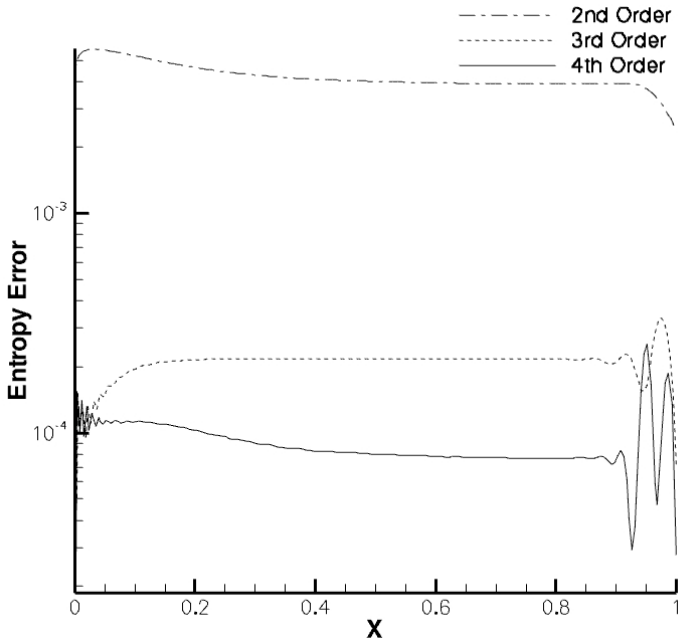


Figure 18. Entropy Error on the Upper Wall of NACA0012 Airfoil