Journal of Computational Physics 228 (2009) 8161–8186

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids

Z.J. Wang *, Haiyang Gao

Department of Aerospace Engineering and CFD Center, Iowa State University, 2271 Howe Hall, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 13 January 2009
Received in revised form 27 July 2009
Accepted 31 July 2009
Available online 12 August 2009

Keywords:
High-order methods
Euler equations
Unstructured grids
Discontinuous Galerkin
Spectral volume
Spectral difference
Collocation
Finite difference

ABSTRACT

Recently a new high-order formulation for 1D conservation laws was developed by Huynh using 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.

1. 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 [13,48]. This surge stems 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 and flow over high lift configurations.

For compressible flow simulations in aerospace applications, the leader of these high-order methods is arguably the discontinuous Galerkin (DG) method [34,9–11,3,4,31,45]. There are also significant research activities in other methods, such as the *k*-exact finite volume method [2,6,12,32], streamline-upwind Petrov–Galerkin (SUPG) method [21,46], spectral and spectral-element type methods [15,5,19,24], the residual distribution (RD) or fluctuation splitting method [1], staggered-grid (SG) multi-domain spectral method [25], spectral volume (SV) [47,49–51,28] and spectral difference (SD) [27,29,30] methods. The SD method can be viewed as the extension (or generalization) of the SG method to triangular meshes. In fact, Van den Abeele et al. [42] 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 the methods. Therefore one does not need to use a staggered grid at all, and the solution and flux points can coincide to improve efficiency [42]. All the above mentioned

* Corresponding author. Tel.: +1 515 294 1614; fax: +1 515 294 3262.
E-mail address: zjw@iastate.edu (Z.J. Wang).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.036

methods share a common property in that "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 physics into the numerics to achieve stability, consistency and accuracy. On the other hand, one can also distinguish these methods based on certain criteria. For example, one can divide all these methods into two categories, those with continuous and 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 or flux splitting approaches [36,35,33,23,26] are used to compute the common fluxes at element interfaces to incorporate "upwinding", similar to the Godunov finite volume method [14,44]. In this paper, we focus on 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 [43]. 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, different choices of DOFs are used by various researchers resulting in different efficiency and numerical properties. Depending on how DOFs are selected, DG schemes 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 (or collocation) methods. 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 [54,39]. 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 for 1D conservation laws [22], 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 effi-

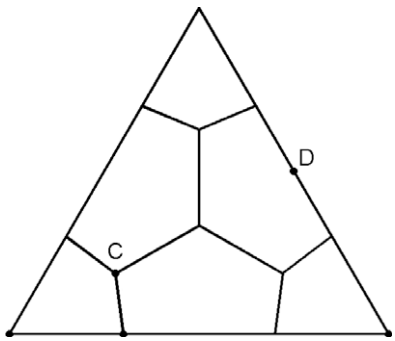


Fig. 1. The partition of a triangular spectral volume into six control volumes, which support a unique quadratic data reconstruction in the spectral volume.

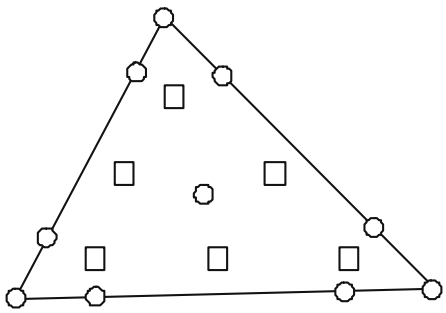


Fig. 2. Solution (squares) and flux points (circles) for the 3rd order spectral difference method.

cient amongst all DG formulations. Another scheme in the family, named the g_2 scheme [22], is probably the most efficient among all the methods with a discontinuous solution space, and has the largest (known) 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 the 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 possible future work are summarized in Section 5.

2. 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 $Q=Q(x,t)$ is the state variable defined on the space-time domain $[a,b]\times[0,T]$ 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 $Q^h_{i,j}=Q^h_{i,j}(t)$ at $k+1$ solution points, $x_{i,j}$, $j=1,\ldots,k+1$. Then the solution is approximated by the following degree k Lagrange interpolation polynomial

$$Q(x,t)\approx Q^h_i(x,t)=\sum_{j=1}^{k+1}L_j(x)Q^h_{i,j}(t), \tag{2.2}$$

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

$$F\left(Q^h_i\right)\approx\overline{F}_i(x,t)=\sum_{j=1}^{k+1}L_j(x)F(Q^h_{i,j}(t)). \tag{2.3}$$

Obviously, if the flux is a linear function of Q, $\overline{F}_i$ is identical to $F\left(Q^h_i\right)$. In order to simplify the notation, we will drop the dependence on time t, which is assumed implicitly. 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 $\widehat{F}_i(x)$ is reconstructed, which must satisfy the following conditions:

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

$$\widehat{F}_i(x_{i-1/2})=\widetilde{F}\left(Q^h_{i-1}(x_{i-1/2}),Q^h_i(x_{i-1/2})\right)\equiv\widetilde{F}_{i-1/2},$$
$$\widehat{F}_i(x_{i+1/2})=\widetilde{F}\left(Q^h_i(x_{i+1/2}),Q^h_{i+1}(x_{i+1/2})\right)\equiv\widetilde{F}_{i+1/2},$$

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

$$\frac{\partial Q^h_{i,j}}{\partial t}+\frac{\partial\widehat{F}_i(x_{i,j})}{\partial x}=0. \tag{2.4}$$

Obviously the above conditions do not uniquely define $\widehat{F}_i(x)$. As a matter of fact, only two conditions are given. We need k extra conditions to define the degree $k+1$ polynomial $\widehat{F}_i(x)$. Using special polynomials such as the Radau and Legendre polynomials, the DG, SG (or SD/SV) methods can be successfully recovered. In order to fix the idea, the reconstructed flux is first re-written as

$$\widehat{F}_i(x)=\overline{F}_i(x)+\sigma_i(x), \tag{2.5}$$

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

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

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

$$\frac{\partial Q^h_{i,j}}{\partial t} + \frac{\partial \overline{F}_i(x_{i,j})}{\partial x} + [\widetilde{F}_{i-1/2} - \overline{F}_i(x_{i-1/2})]g'_L(x_{i,j}) + [\widetilde{F}_{i+1/2} - \overline{F}_i(x_{i+1/2})]g'_R(x_{i,j}) = 0. \tag{2.8}$$

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 schemes. Several of them 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. [40,42].
3. 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.
4. 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! This scheme was called the g_2 scheme in [22].

As mentioned earlier, the FR method has some remarkable properties. Its DG formulation looks like a pseudo-DG method. 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 the 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 $Q^h_i(\vec{r})$, such as those shown in Fig. 3 for $k=2$. Fluxes at the solution points can also be computed, and their Lagrange interpolations result in two degree k flux polynomials $\overline{F}^x_i(\vec{r})$ and $\overline{F}^y_i(\vec{r})$. Along the element interfaces, the solution is discontinuous. A Riemann flux in the interface normal direction can be computed at each interface point,

$$\widetilde{F}^n(\vec{r}) = \widetilde{F}^n\left(Q^h_i(\vec{r}), Q^h_{i+}(\vec{r}), \vec{n}\right), \quad \vec{r} \in \partial V_i, \tag{2.10}$$

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

1. Both $\widehat{F}^x_i$ and $\widehat{F}^y_i$ are degree $k+1$ polynomials, i.e., one degree higher than $Q^h_i(\vec{r})$.
2. At the element interfaces, the normal flux is equal to the Riemann flux computed in (2.10), i.e.,

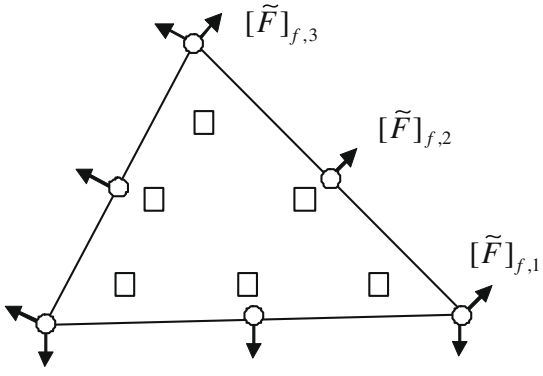


Fig. 3. Solution points (squares) and flux points (circles) for $k=2$.

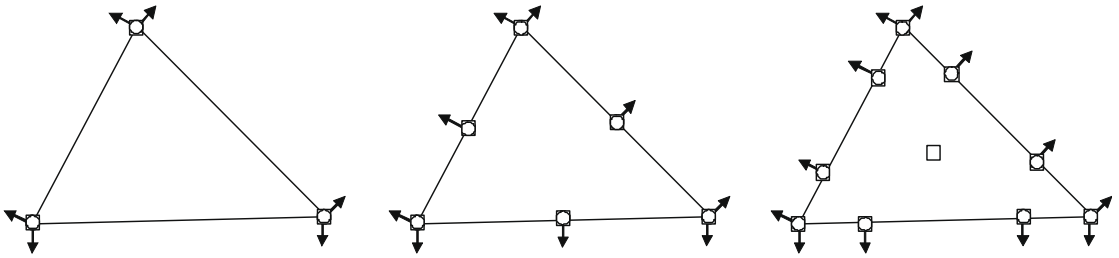


Fig. 4. Efficient arrangement of solution (squares) and flux points (circles) in a simplex for $k = 1$, 2, and 3.

$$\left(\widehat{F}_i^x(\vec{r}), \widehat{F}_i^y(\vec{r})\right) \cdot \vec{n} = \widetilde{F}^n(\vec{r}), \quad \vec{r} \in \partial V_i. \tag{2.11}$$

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 Fig. 4(c) for $k = 2$. 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 $\widehat{F}_i^x$ and $\widehat{F}_i^y$ (we are assuming $k = 2$). We can impose the following conditions at 4 more points inside the cell

$$\left(\widehat{F}_i^x, \widehat{F}_i^y\right) = \left(\overline{F}_i^x, \overline{F}_i^y\right). \tag{2.12}$$

It appears that 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 on the left-hand-side of (2.8). The following section describes the basic idea.

3. 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 a weighting or test 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. We assume that the solution belongs to the space of polynomials of degree k or less, i.e., $Q_i^h \in P^k$(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 satisfy the following equation

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

Because the solution is discontinuous across element interfaces, a common Riemann flux has been used in (3.3) to replace the normal flux in (3.2) to provide element coupling, i.e.,

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

where Q_{i+}^h is the solution outside the current element V_i. Applying integration by parts to the last term on the LHS of (3.3), we obtain

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

In the present study, we require that the space of the test functions have the same dimension as the solution space. In addition, the test space is so selected such that a unique solution exists for (3.5). The last term on the left side of (3.5) 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"

$$\int_{V_i} W\delta_i dV = \int_{\partial V_i} W[\widetilde{F}]dS, \tag{3.6}$$

where $[\widetilde{F}] = \widetilde{F}^n\left(Q^h, Q^h_+, \vec{n}\right) - F^n(Q^h)$ is the normal flux difference. Substituting (3.6) into (3.5), we obtain

$$\int_{V_i} \left[\frac{\partial Q_i^h}{\partial t} + \nabla \bullet \vec{F}\left(Q_i^h\right) + \delta_i\right] W dV = 0. \tag{3.7}$$

In order to simplify the derivation for now, we assume that the flux vector is linear in Q. Therefore, the term inside the square bracket in (3.7) belongs to P^k. Because the test space is selected to ensure a unique solution, (3.7) is equivalent to

$$\frac{\partial Q_i^h}{\partial t} + \nabla \bullet \vec{F}\left(Q_i^h\right) + \delta_i = 0, \tag{3.8}$$

i.e., (3.8) 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 $W \in P^k$, (3.8) 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 (3.6), we obtain

$$\int_{V_i} \delta_i dV = \int_{\partial V_i} [\widetilde{F}]dS, \tag{3.9}$$

or

$$\bar{\delta}_i = \frac{1}{|V_i|}\sum_{f\in\partial V_i}\int_f [\widetilde{F}]dS, \tag{3.10}$$

where $\bar{\delta}$ is the volume averaged δ. Since we assume that $\vec{F}$ is linear, the flux difference $[\widetilde{F}]$ should be a degree k polynomial along 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 Fig. 3. Then the face integral can be computed exactly using a quadrature formula such as

$$\bar{\delta}_i = \frac{1}{|V_i|}\sum_{f\in\partial V_i}\sum_l w_l [\widehat{F}]_{f,l} S_f, \tag{3.11}$$

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 Fig. 3. Then Eq. (3.8) must be true at the SPs, i.e.,

$$\frac{\partial Q_{i,j}^h}{\partial t} + \nabla \bullet \vec{F}\left(Q_{i,j}^h\right) + \delta_{i,j} = 0, \tag{3.12}$$

where $\nabla \bullet \vec{F}\left(Q_{i,j}^h\right) = \left[\nabla \bullet \vec{F}\left(Q_i^h\right)\right]_{\vec{r}_{i,j}}$.

Let's examine (3.6) more carefully. If $\vec{F}$ is linear, and that the triangle has straight faces, the correction field δ_i can be computed explicitly in the following form

$$\delta_{i,j} = \frac{1}{|V_i|}\sum_{f\in\partial V_i}\sum_l \alpha_{j,f,l}[\widetilde{F}]_{f,l} S_f, \tag{3.13}$$

where $\alpha_{j,f,l}$ are ***constant*** lifting coefficients independent of the solution and geometry, but dependent on the weighting function W. Substituting (3.13) into (3.12) we obtain the following formulation

$$\frac{\partial Q_{i,j}^h}{\partial t} + \nabla \bullet \vec{F}\left(Q_{i,j}^h\right) + \frac{1}{|V_i|}\sum_{f\in\partial V_i}\sum_l \alpha_{j,f,l}[\widetilde{F}]_{f,l} S_f = 0. \tag{3.14}$$

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.

Next let's address the case of general non-linear flux vector. For non-linear flux vectors, one can approximate $\vec{F}\left(Q_i^h\right)$ with polynomials, either degree k or $k+1$. For example, in a nodal DG formulation [20], and in the FR formulation [22], a degree k Lagrange interpolation polynomial is used

$$\vec{F}\left(Q_i^h\right) \approx \overrightarrow{\overline{F}}_i\left(Q_i^h\right) = \sum_j L_j(\vec{r}_{i,j})\,\vec{F}\left(Q_{i,j}^h\right), \tag{3.15}$$

where L_j is the Lagrange interpolation polynomial based on the solution points $\{\vec{r}_{i,j}\}$, and $Q_{i,j}^h = Q_i^h(\vec{r}_{i,j})$. In this case, $\nabla \cdot \overrightarrow{\overline{F}}_i$ is a degree $k-1$ polynomial, which also belongs to P^k. Therefore, (3.8) is still true for non-linear fluxes. 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 [17]. In that case, the divergence of the flux vector is a degree k polynomial. Again (3.8) is valid. However this approach will significantly increase the cost of the LCP method because of the need to compute the fluxes at the extra set of points. Therefore it is not pursued in the present study further. Instead, we compute $\nabla \bullet \vec{F}\left(Q_i^h\right)$ at the solution points exactly, using the chain-rule, i.e.,

$$\nabla \bullet \vec{F}\left(Q_{i,j}^h\right) = \frac{\partial F^x(Q_{i,j}^h)}{\partial x} + \frac{\partial F^y(Q_{i,j}^h)}{\partial y} = \frac{\partial F^x\left(Q_{i,j}^h\right)}{\partial Q}\frac{\partial Q_{i,j}^h}{\partial x} + \frac{\partial F^y\left(Q_{i,j}^h\right)}{\partial Q}\frac{\partial Q_{i,j}^h}{\partial y} = \frac{\partial \vec{F}\left(Q_{i,j}^h\right)}{\partial Q} \bullet \nabla Q_{i,j}^h. \tag{3.16}$$

where $\frac{\partial \vec{F}}{\partial Q}$ is the flux Jacobian matrix. Obviously, $\nabla \bullet \vec{F}\left(Q_i^h\right)$ is generally not a degree k polynomial. However, it can be approximated by the Lagrange interpolation polynomial of the flux vector divergence at the solution points, i.e.,

$$\nabla \bullet \vec{F}\left(Q_i^h\right) \approx \nabla \bullet \vec{F}^{CR}\left(Q_i^h\right) = \sum_j L_j(\vec{r}_{i,j}) \nabla \bullet \vec{F}\left(Q_{i,j}^h\right). \tag{3.17}$$

With this approximation, (3.8) is again valid for general non-linear flux vectors. 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 (3.14) a bit further. It is obviously a collocation-like formulation, with penalty-like terms to provide the coupling between elements. However, with a special choice of scheme coefficients $\alpha_{j,f,l}$, we can show that (3.14) is equivalent to the DG method, at least in the linear case. From (3.6), it is easy to show that δ_i is uniquely defined once W and the flux points are given. In the linear case, 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. 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 [7,18,52]. Fig. 4 shows the location 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.,

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

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

In the DG method, the weighting function W is set to be one of the Lagrange polynomials L_j. Substituting W into (3.6), we obtain the following equations

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

The unknowns in (3.20) $\delta_{i,j}$ can be easily solved in terms of the normal flux jumps at the flux points $[\widehat{F}]_{f,l}$, and the coefficients $\alpha_{j,f,l}$ be determined, which are constant for any straight-sided triangle. The coefficients for various schemes are included in Appendix A. 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

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

Although all the flux points coincide with the solution points, as shown in Fig. 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.

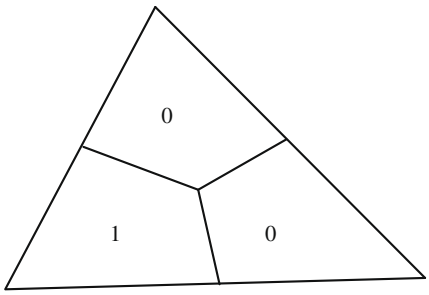


Fig. 5. One of the weighting functions for the spectral volume method, $k = 1$.

In the SV method, the weighting functions are piece-wise constant at the sub-cells. The number of partitions is m. One weighting function is shown in Fig. 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 several SV schemes are also included in Appendix A. The coefficients for the first solution point of the 2nd order SV method are {2 0.2 −0.7 −0.7 0.2 2}, corresponding to the following formula

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

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

$$\delta_i = \nabla \bullet [\widehat{\vec{F}} - \vec{F}^I\left(Q_i^h\right)], \tag{3.23}$$

where $\widehat{\vec{F}}$ is a reconstructed flux vector from the Riemann fluxes at the element interface and fluxes in the interior of the element, and $\vec{F}^I\left(Q_i^h\right)$ is the interpolated flux vector Lagrange polynomial based on values of the fluxes computed using Q_i^h at the flux points. $\widehat{\vec{F}}$ and $\vec{F}^I\left(Q_i^h\right)$ 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 form 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

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

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 cost, for 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 always the identity matrix 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 terms of both memory and CPU time than the DG method with surface and volume integral quadratures.

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 a 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 satisfied, i.e.,

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

which corresponds to (3.3) with $W = 1$. 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

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

If we use the LP approach to compute the divergence of the interior flux vector, i.e., $\nabla \bullet \vec{F}\left(Q_i^h\right) \approx \nabla \bullet \overline{\vec{F}}_i\left(Q_i^h\right)$, the divergence of the flux vector belongs to P^k. Since the quadrature rule is exact for degree k (or less) polynomials, we have

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

In order to prove (3.25), we need to show

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

where w_l is the surface integral quadrature weight. 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

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

In the case of $k = 1$, $w_l = 1/2$ 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.

If the CR approach is used in the computation of the flux vector divergence, the LCP method is not strictly conservative. This is due to the fact that the flux vector $\vec{F}^{CR}\left(Q_i^h\right)$ belongs to P^{k+1} since $\nabla \bullet \vec{F}^{CR}\left(Q_i^h\right)$ belongs to P^k. Therefore the added accuracy of the CR approach is achieved at the cost of full conservation. A test case will be presented later to assess the conservation error of the CR approach.

3.4. Extension to mixed grids

It is obvious that (3.8) is valid for arbitrary types of elements besides triangles. For viscous flow problems, it is often advantageous to have quadrilateral cells or prismatic cells 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 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 Fig. 6. In order to achieve an efficient implementation, all elements are transformed from the physical domain (x,y) into a standard square element $(\xi,\eta) \in [-1,1] \times [-1,1]$ as shown in Fig. 7. The transformation can be written as

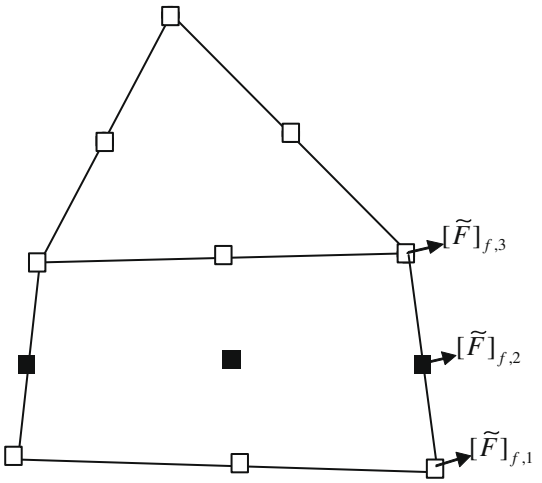


Fig. 6. Solution and flux points for the 3rd order LCP scheme on hybrid meshes.

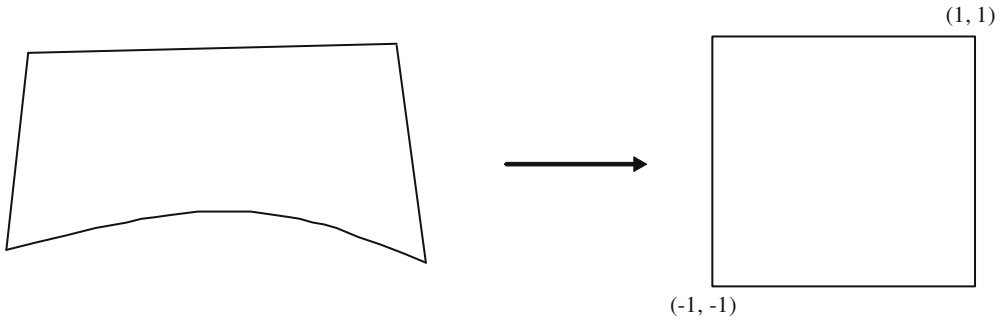


Fig. 7. Transformation of a general quadrilateral element to a standard element.

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

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. The Jacobian matrix of the transformation 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 exist, and the Jacobian 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.,

$$\widetilde{Q}_i^h(\xi, \eta) = \sum_{l=1}^{k+1} \sum_{j=1}^{k+1} \widetilde{Q}_{i,j,l}^h L_j(\xi) \cdot L_l(\eta), \tag{3.40}$$

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

$$F^{\xi}\left(\widetilde{Q}_i^h\right) \approx \overline{F}_i^{\xi}(\xi, \eta) = \sum_{l=1}^{k+1} \sum_{j=1}^{k+1} F_{i,j,l}^{\xi} L_j(\xi) \cdot L_l(\eta), \tag{3.41a}$$

$$F^{\eta}\left(\widetilde{Q}_i^h\right) = \overline{F}_i^{\eta}(\xi, \eta) = \sum_{l=1}^{k+1} \sum_{j=1}^{k+1} F_{i,j,l}^{\eta} L_j(\xi) \cdot L_l(\eta). \tag{3.41b}$$

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

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

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

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

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

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

Finally the DOFs are updated using the following equation

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

Note that the correction is done in a "one-dimensional" manner. For example, $[\widetilde{F}]_{f,2}$ in Fig. 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 (3.44) can also be computed using the CR approach. Numerical tests will be presented to compare the LP and CR approaches.

In the present implementation, we choose the DG scheme from the FR family. The solution points for all triangular and quadrilateral elements at each edge are the 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 2nd 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 2nd 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 curved boundary cell shown in Fig. 8, a 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. (3.14) for the transformed equation on a standard triangle becomes

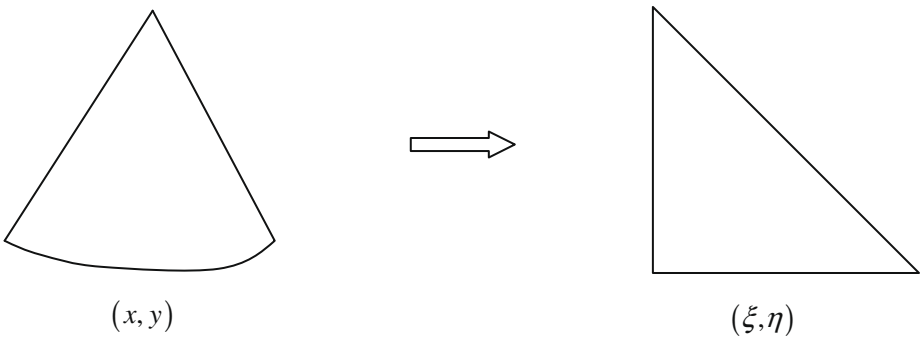


Fig. 8. Transformation of a curve boundary triangle to a standard element.

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

where superscript ξ means the variables or operations 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 Eq. (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, as described in (3.44).

4. Numerical results

4.1. Accuracy study with 2D scalar conservation laws

4.1.1. Linear wave equation

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

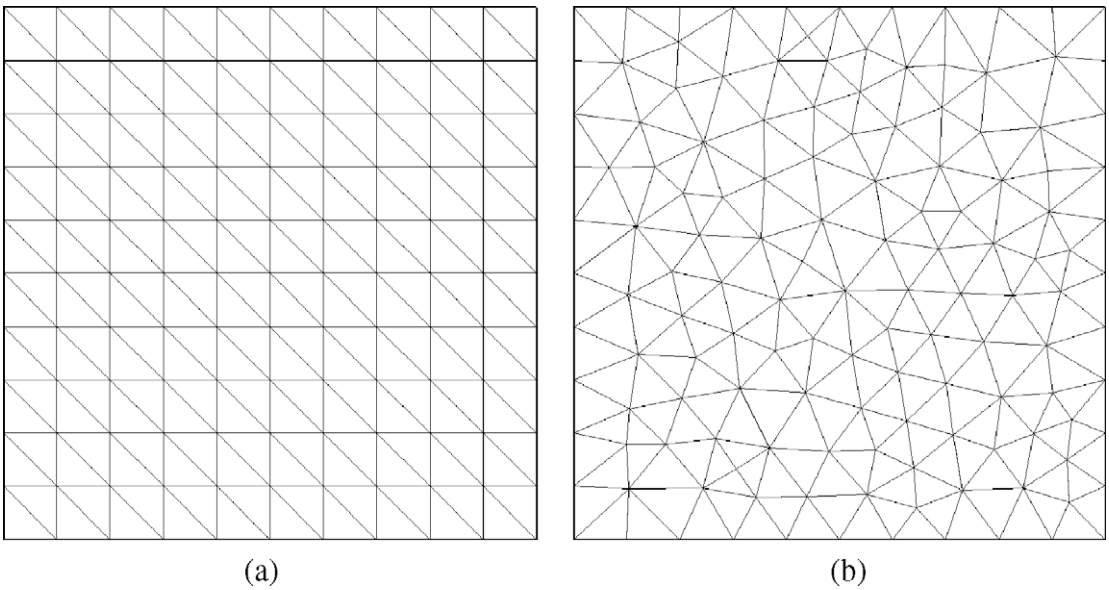


Fig. 9. Regular and irregular "10 × 10 × 2" triangular computational grids.

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

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

Table 2
Test of the LCP-SV scheme for $u_t + u_x + u_y = 0$, with $u_0(x,y) = \sin\pi(x+y)$, at $t = 1$ on two triangular meshes.

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

Table 3
Test of the LCP-DG scheme 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 the irregular triangular mesh using the LP approach in computing the interior flux divergence.

Polynomial degree k	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
1	$10 \times 10 \times 2$	2.12e−2	–	2.65e−2	–	8.44e−2	–
	$20 \times 20 \times 2$	6.96e−3	1.61	9.96e−3	1.41	4.29e−2	0.98
	$40 \times 40 \times 2$	2.19e−3	1.67	3.75e−3	1.41	2.29e−2	0.91
	$80 \times 80 \times 2$	6.79e−4	1.69	1.38e−3	1.44	1.02e−2	1.17
2	$10 \times 10 \times 2$	4.36e−3	–	6.40e−3	–	3.01e−2	–
	$20 \times 20 \times 2$	7.76e−4	2.49	1.37e−3	2.20	9.91e−3	1.60
	$40 \times 40 \times 2$	1.26e−4	2.62	2.81e−4	2.29	3.01e−3	1.72
	$80 \times 80 \times 2$	1.93e−5	2.71	5.43e−5	2.37	8.17e−4	1.88
3	$10 \times 10 \times 2$	5.41e−4	–	9.59e−4	–	7.93e−3	–
	$20 \times 20 \times 2$	4.70e−5	3.52	1.05e−4	3.19	1.75e−3	2.18
	$40 \times 40 \times 2$	3.56e−6	3.72	9.86e−6	3.41	2.73e−4	2.68
	$80 \times 80 \times 2$	2.65e−7	3.75	8.48e−7	3.54	2.62e−5	3.38
5	$10 \times 10 \times 2$	1.15e−5	–	3.51e−5	–	6.20e−4	–
	$20 \times 20 \times 2$	2.94e−7	5.28	1.16e−6	4.92	3.26e−5	4.25
	$40 \times 40 \times 2$	6.37e−9	5.53	3.14e−8	5.21	1.57e−6	4.38
	$80 \times 80 \times 2$	1.23e−10	5.69	6.95–10	5.50	3.87e−8	5.34

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

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 Fig. 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 3rd or 4th order accuracy [16,38]. The results are made time step independent so that the spatial error is the dominant error source. The numerical simulation is carried out 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

$$Error_{L2}=\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{3}\left(Q_{i,j}^{h}-Q_{i,j}^{e}\right)^2}{3N}}, \tag{4.2}$$

where Q^e denotes the exact solution. For the linear wave equation, all DG formulations are identical, so are all SV formulations. In the present LCP family, the DG and SV formulations are denoted by LCP-DG and LCP-SV respectively. Tables 1 and 2 present the computed errors with both formulations on the regular and irregular meshes. Note that optimal accuracy has

Table 4
Test of the LCP-DG scheme 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 the irregular triangular mesh using the CR approach in computing the interior flux divergence.

Polynomial degree k	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
1	$10\times10\times2$	1.34e−2	–	1.84e−2	–	7.33e−2	–
	$20\times20\times2$	3.64e−3	1.88	5.06e−3	1.86	2.57e−2	1.51
	$40\times40\times2$	9.39e−4	1.95	1.35e−3	1.91	8.63e−3	1.57
	$80\times80\times2$	2.37e−4	1.99	3.50e−4	1.95	2.60e−3	1.73
2	$10\times10\times2$	1.57e−3	–	2.75e−3	–	2.12e−2	–
	$20\times20\times2$	2.10e−4	2.90	4.04e−4	2.77	4.86e−3	2.13
	$40\times40\times2$	2.62e−5	3.00	5.50e−5	2.88	1.03e−3	2.24
	$80\times80\times2$	3.15e−6	3.06	7.27e−6	2.92	1.68e−4	2.62
3	$10\times10\times2$	1.31e−4	–	3.68e−4	–	4.13e−3	–
	$20\times20\times2$	8.64e−6	3.92	2.58e−5	3.83	4.50e−4	3.20
	$40\times40\times2$	5.70e−7	3.89	1.82e−6	3.83	3.62e−5	3.64
	$80\times80\times2$	3.63e−8	3.97	1.27e−7	3.84	2.77e−6	3.71
5	$10\times10\times2$	2.42e−6	–	1.07e−5	–	2.55e−4	–
	$20\times20\times2$	5.08e−8	5.57	2.61e−7	5.35	9.73e−6	4.71
	$40\times40\times2$	8.17e−10	5.96	4.45e−9	5.87	2.17e−7	5.49
	$80\times80\times2$	1.43e−11	5.84	7.99e−11	5.75	4.91e−9	5.47

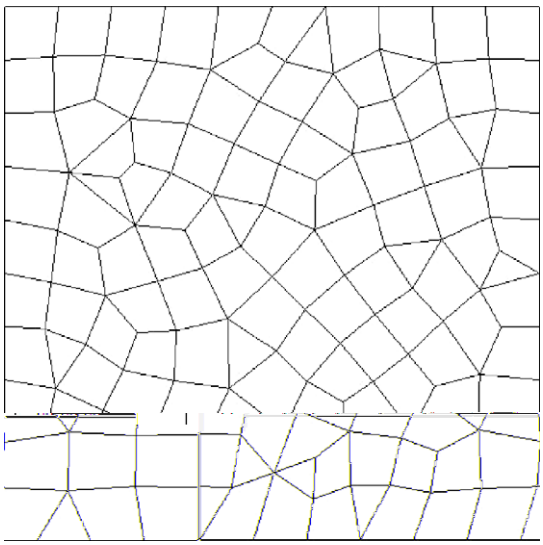


Fig. 10. Coarse mixed mesh for the accuracy study.

been achieved for all the test cases, on both the regular and irregular meshes. The SD formulation cannot completely degenerate into the LCP method, and therefore is not considered in the accuracy study.

4.1.2. 2D Burgers equation

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

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

Table 5
Test of the LCP-DG scheme for the isotropic vortex propagation problem, on an irregular triangular mesh and a mixed mesh.

Polynomial degree k	Grid size	Triangular mesh LP		Triangular mesh CR		Mixed mesh CR	
		L_2 error	Order	L_2 error	Order	L_2 error	Order
1	$10\times10\times2$	2.01e−2	–	1.39e−2	–	1.58e−2	–
	$20\times20\times2$	6.67e−3	1.59	4.41e−3	1.66	5.32e−3	1.57
	$40\times40\times2$	1.73e−3	1.95	1.08e−3	2.03	1.50e−3	1.83
	$80\times80\times2$	4.84e−4	1.84	2.54e−4	2.09	3.54e−4	2.08
2	$10\times10\times2$	7.14e−3	–	4.41e−3	–	2.95e−3	–
	$20\times20\times2$	1.07e−3	2.74	5.19e−4	3.09	5.62e−4	2.39
	$40\times40\times2$	1.60e−4	2.74	5.84e−5	3.15	7.42e−5	2.92
	$80\times80\times2$	2.29e−5	2.80	6.94e−6	3.07	8.63e−6	3.10
3	$10\times10\times2$	1.79e−3	–	6.70e−4	–	5.79e−4	–
	$20\times20\times2$	1.40e−4	3.68	4.79e−5	3.81	5.05e−5	3.52
	$40\times40\times2$	9.75e−6	3.84	2.96e−6	4.02	3.51e−6	3.85
	$80\times80\times2$	6.96e−7	3.81	1.71e−7	4.11	1.89e−7	4.22

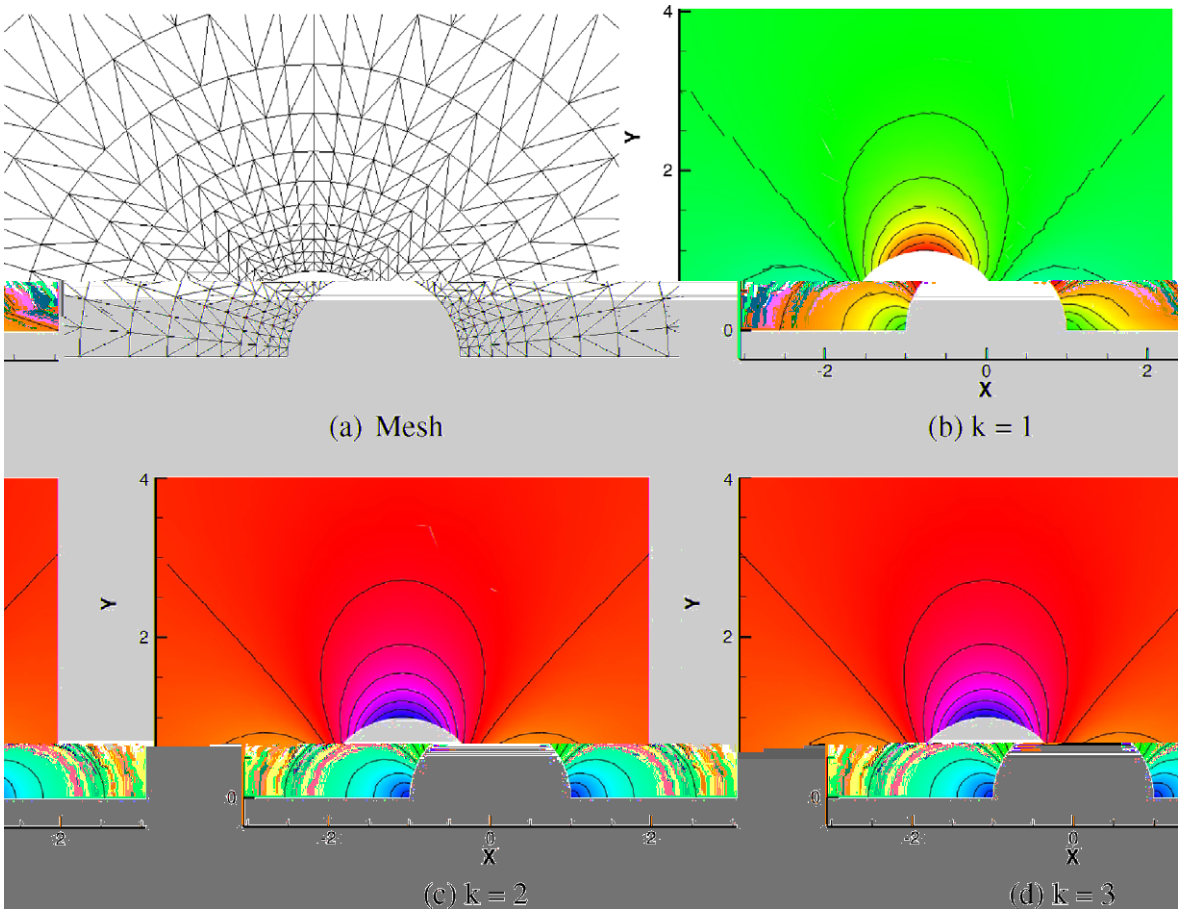


Fig. 11. Computational mesh and Mach number contours computed with the 2nd, 3rd and 4th order schemes for subsonic flow around a cylinder.

The initial solution is smooth. Due to the non-linearity of the Burgers equation, discontinuities will eventually develop in the solution. At $t = 0.1$, the exact solution is still smooth. The numerical simulations are therefore carried out until $t = 0.1$ on both the regular and irregular grids. In Tables 3 and 4, we present the L_1, L_2 and L_∞ errors produced using the LCP-DG formulation on the irregular grids, with both the LP (Lagrange polynomial) and the CR (chain-rule) approaches in evaluating the interior flux divergence. Comparing Tables 3 and 4, we find that the CR approach gives better accuracy in every case and in every norm of error. It not only produces smaller errors, but also shows more consistent numerical orders of accuracy, especially in the L_∞ norm. For example, on the finest mesh with $k = 5$, the CR approach produced a L_1 error which is nearly an order lower than the LP approach.

4.2. Accuracy study with vortex propagation problem

This is an idealized problem with an exact solution for the Euler equations in 2D used by Shu [37]. 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 $S = p/\rho^\gamma$:

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

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 $(1, 1)$.

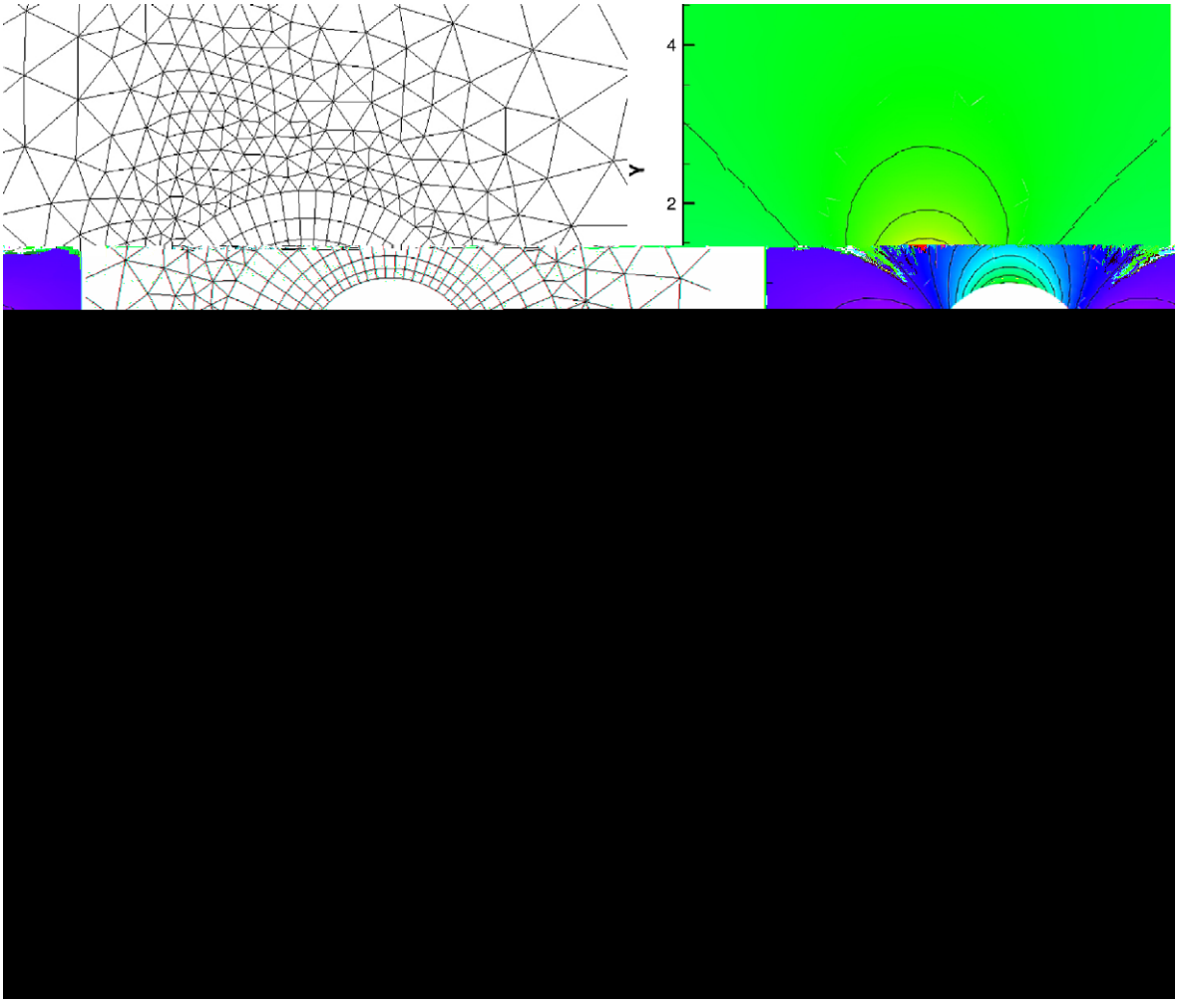


Fig. 12. Mixed computational mesh #1 and Mach number contours computed with the 2nd, 3rd and 4th order schemes for subsonic flow around a cylinder.

In the numerical simulation, the computational domain is taken to be $[-5,5] \times [-5,5]$, with characteristic inflow and outflow boundary conditions imposed on the boundaries.

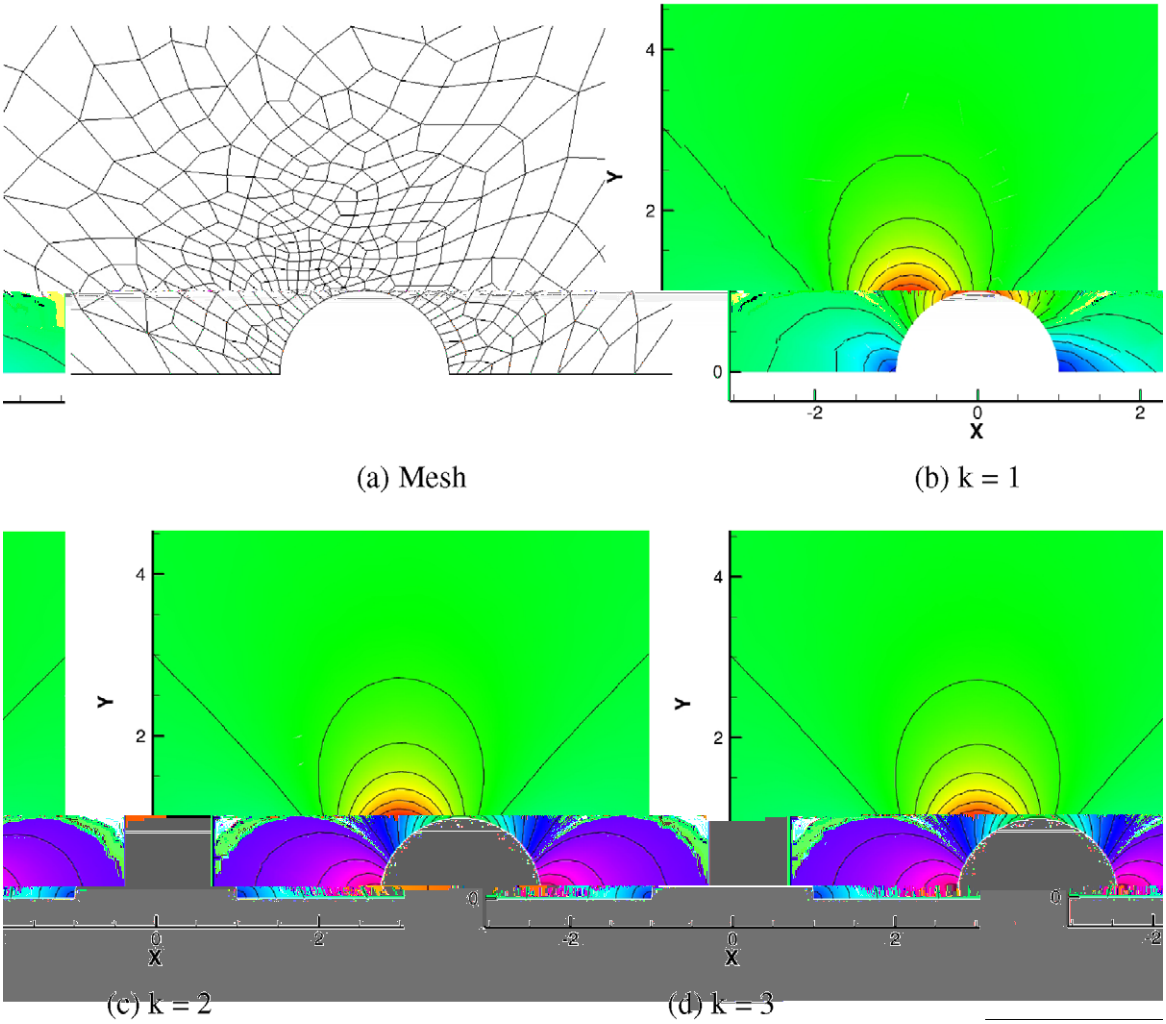


Fig. 13. Mixed computational mesh #2 and Mach number contours computed with the 2nd, 3rd and 4th order schemes for subsonic flow around a cylinder.

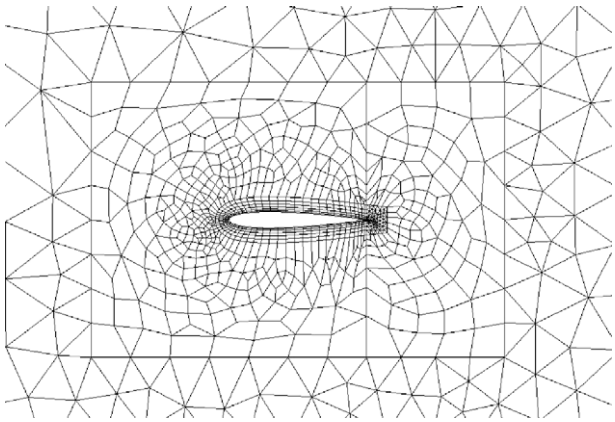


Fig. 14. A mixed mesh around the NACA0012 airfoil.

The numerical simulations are carried out until $t = 2$ on two different grids, one irregular triangular mesh and one mixed mesh as shown in Figs. 9 and 10. The finer irregular grids are generated recursively by cutting each coarser grid cell into four finer grid cells, while all mixed meshes, which are mainly composed of irregular quadrilateral cell with a small number of triangular cells, are generated independently. On the irregular triangular mesh, we again test both the LP and CR approaches in evaluating the interior flux divergence, while on the mixed mesh, we employ the CR approach. On all meshes the LCP-DG or FR-DG formulation is used. For time integration, a 3-stage Runge–Kutta explicit scheme is used for time marching in all the cases.

In Table 5, the L_2 density errors at the solution points are presented on both set of meshes for $k = 1$–3. Note that the CR approach is more accurate than the LP approach on the triangular meshes for every polynomial degree and on every mesh. The CR approach not only produces the smaller errors, but also demonstrates more consistent numerical orders of accuracy in the grid refinement study. In addition, the LCP method performs very well on the mixed grids, achieving the optimal order of accuracy on relatively poor quality meshes.

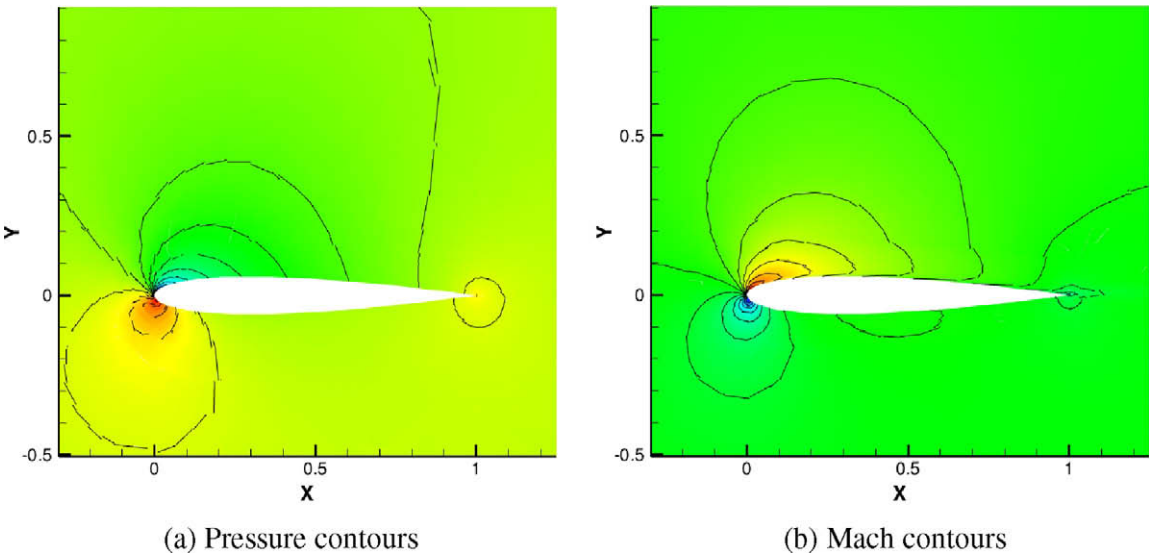


Fig. 15. Pressure and Mach contours computed with a 2nd order scheme, subsonic flow around a NACA0012 airfoil.

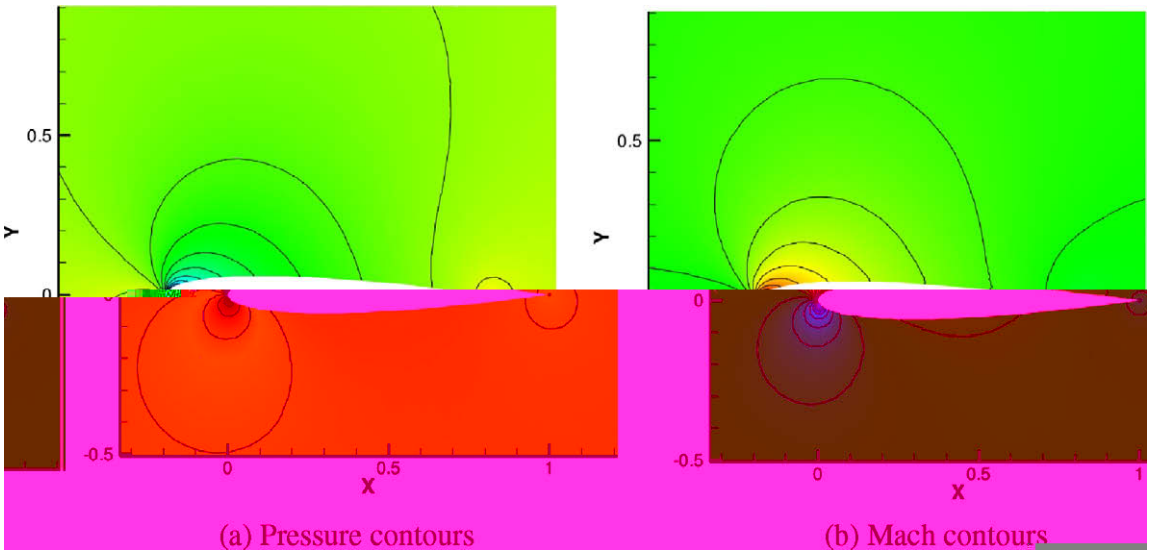


Fig. 16. Pressure and Mach contours computed with a 3rd order scheme, subsonic flow around a NACA0012 airfoil.

4.3. Flow over a circular 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 curved wall, a quadratic representation is used in all the simulations. Three grids are used, one triangular

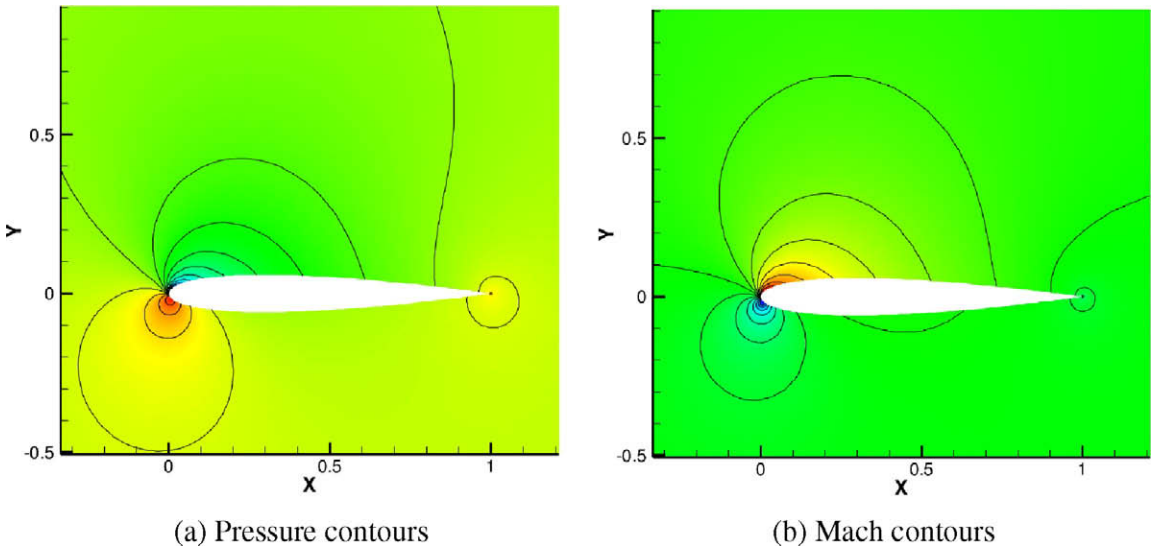


Fig. 17. Pressure and Mach contours computed with a 4th order scheme, subsonic flow around a NACA0012 airfoil.

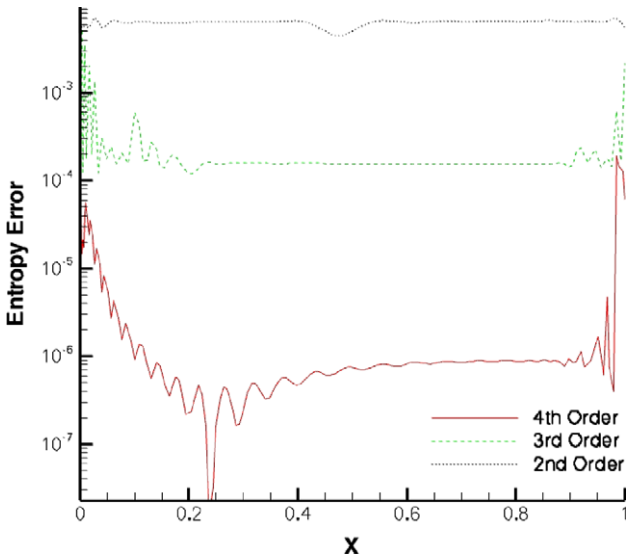


Fig. 18. Computed entropy errors on the upper surface of a NACA0012 airfoil.

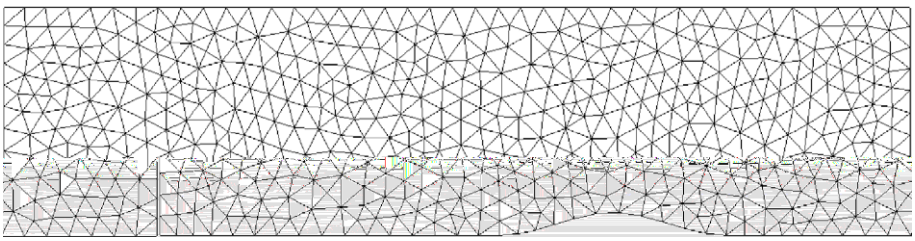
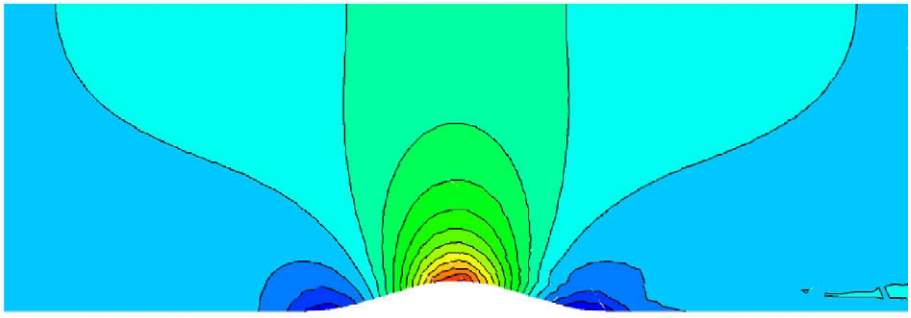


Fig. 19. Computational mesh for flow through a channel with a sine bump.

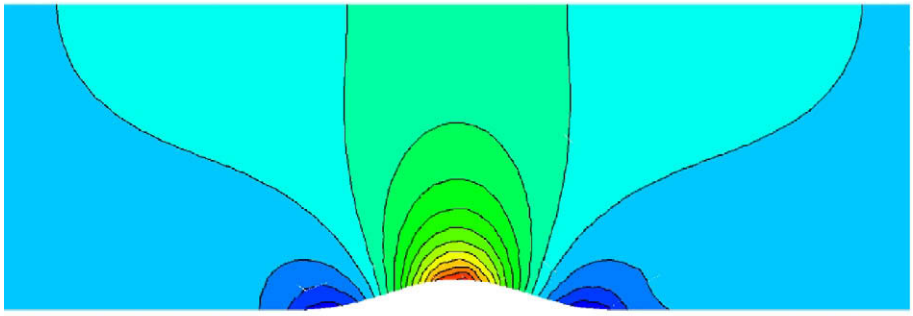
mesh, and two mixed meshes as shown in Figs. 11a, 12a and 13a. The first mixed mesh is composed of regular quadrilateral cells near the wall and irregular triangular cells elsewhere, while the second mixed mesh is composed of fully mixed irregular quadrilateral and triangular cells, and has a poor grid quality near the wall.

The 2nd–4th order LCP-DG scheme is used for triangular cells and the 2nd– 4th order FR-DG scheme is used for the quadrilateral cells. For time integration, a LU-SGS implicit solver [53,8] is used, and all the cases converged to machine zero.

The results corresponding to $k = 1,2,3$ on the 3 grids are shown in Figs. 11–13. Note that while the 2nd order solutions show some discontinuities across cell interfaces, smooth solutions are obtained on all the grids with the 3rd and 4th order schemes, even on the relatively skewed mixed mesh #2. All the 4th order solutions agree well with each other. The results



(a) LP approach, k = 2



(b) CR approach, k = 2

Fig. 20. Computed mach contours with both the LP and CR approaches with a 3rd order LCP-DG scheme.

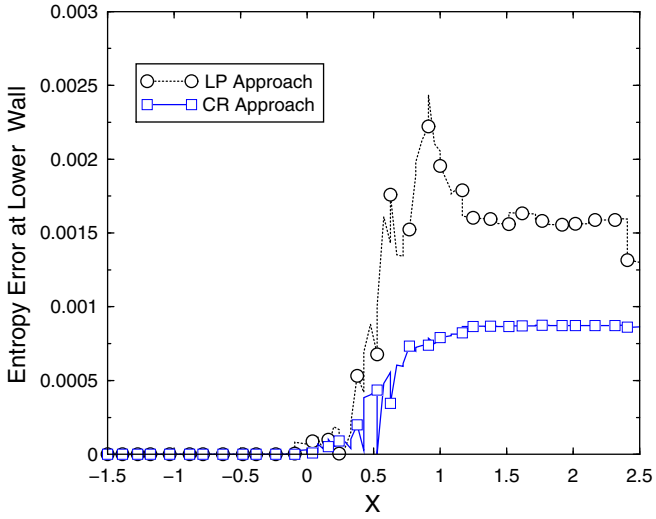


Fig. 21. Computed entropy errors at the lower channel wall with a 3rd order LCP-DG scheme.

show that the LCP method performs well for mixed meshes with curved boundary elements, since the solution becomes smoother across the triangular and quadrilateral cell interfaces as k increases. Furthermore, the high-order solutions are also smooth near the curved boundary.

4.4. Flow over NACA0012 Airfoil

As another test case for a more realistic geometry, the subsonic flow around a NACA0012 airfoil is simulated with the LCP method, for $k = 1$–3. The flow conditions are Mach = 0.3 and angle of attack $\alpha = 5^\circ$. The airfoil surface is represented with piece-wise quadratic polynomials. The mesh, shown in Fig. 14, is composed of regular quadrilateral cells near the airfoil and irregular triangular and quadrilateral cells elsewhere, with a total of 1510 cells. The outer boundary of the computational domain is about 20 chords from the center of the airfoil. All the simulations converged to machine zero.

The pressure and Mach number contours from the 2nd, 3rd and 4th order simulations are shown in Figs. 15–17. The computed solution for the 2nd order case is not smooth, with visible jumps across cell interfaces, while the 3rd and 4th order solutions are quite smooth. In addition, the 3rd and 4th order results are nearly the same with only minor differences near the trailing edge, indicating that the solution is approaching k-independence. In Fig. 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 solution accuracy. The error shows an exponential convergence as the order of accuracy is increased.

4.5. Evaluation of conservation error with flow through a channel with a sine bump

This internal flow problem is selected to assess the conservation error of the LCP formulation with both the LP and CR approaches to compute the interior divergence of the flux vector, presented in Section 3.1. As discussed earlier, the LP approach is fully conservative, but the CR approach is not. However, accuracy studies have indicated that the CR approach is significantly more accurate than the LP approach. Here we investigate how much mass is generated (or lost) due to the non-conservativeness of the CR formulation. The channel has a height of 1 unit and a length of 5 units. The bump height is 0.1 unit, and is defined with

$$y = 0.1\sin^2(\pi x), \quad 0 \leqslant x \leqslant 1.$$

The triangular computational grid is shown in Fig. 19. The grid has a total of 994 cells, and the sine bump is represented with quadratic segments. The inlet Mach number is 0.5. Characteristic boundary conditions are used at both the inlet and exit. The 3rd order LCP-DG scheme is tested in this study. With both the LP and the CR approaches, the simulation converged to machine zero. Then the conservation error for the mass flux is defined as

$$\varepsilon_{mass} = \frac{\left|\int_{exit} \rho v_n dS - \int_{inlet} \rho v_n dS\right|}{\int_{inlet} \rho v_n dS}.$$

The total inlet and exit masses are computed in exactly the same way as in the flow solver using a Riemann solver. The computed Mach contours and entropy errors along the lower channel wall are shown in Figs. 20 and 21. Obviously the CR approach generated significantly less entropy, indicating higher accuracy. The mass conservation error with the LP approach is 3.33×10^{-16}, essentially machine zero as expected, while the CR approach produced a mass conservation error of 3.03×10^{-7}. It is evident that the LP approach is fully conservative, agreeing with the analysis. Note that although the CR approach is not fully conservative, the mass conservation error is still very small. For many problems, this price may be worth paying for the significantly lower solution error produced with the CR approach.

5. Conclusions

In the present study, a remarkably simple and efficient flux reconstruction formulation 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 with 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 divergence to be computed using the chain-rule approach, which appears to be beneficial in solution accuracy, although not strictly conservative. 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 and non-linear scalar conservation laws, and the 2D Euler equations, on 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.

Acknowledgments

The study has been funded by AFOSR Grant FA9550-06-1-0146, and partially by DOE Grant DE-FG02-05ER25677. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR, DOE, or the US Government. We thank the two anonymous reviewers for very constructive comments, which have significantly improved the quality of the paper.

Appendix A. The lifting coefficients for various schemes

According to Eq. (3.13), the penalty corrections at the solution points are computed using

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

When applied to an arbitrary triangular cell in the physical domain with straight sides, $|V_i|$ is the volume of cell i, $[\widetilde{F}]_{f,l}$ is the flux difference computed at the lth flux point along face f, and S_f is simply the area of f. Faces are numbered within each cell according to the order of the vertices of the cell. For example, face 1 is the face formed by vertices 1 and 2, face 2 by vertices 2 and 3, and face 3 by vertices 3 and 1. The start and end points of a face are significant. Complete input files can be provided to interested readers.

A.1. 2nd Order schemes

See Fig. A1 and Tables A1–A4.
For example, according to Table A2

$$\delta_{i,1} = \frac{1}{V_i}(2.5[\widetilde{F}]_{1,1}S_{i,1} + 0.5[\widetilde{F}]_{1,2}S_{i,1} - 1.5[\widetilde{F}]_{2,1}S_{i,2} - 1.5[\widetilde{F}]_{2,2}S_{i,2} + 0.5[\widetilde{F}]_{3,1}S_{i,3} + 2.5[\widetilde{F}]_{3,2}S_{i,3}),$$

and due to symmetry,

$$\delta_{i,2} = \frac{1}{V_i}(0.5[\widetilde{F}]_{1,1}S_{i,1} + 2.5[\widetilde{F}]_{1,2}S_{i,1} + 2.5[\widetilde{F}]_{2,1}S_{i,2} + 0.5[\widetilde{F}]_{2,2}S_{i,2} - 1.5[\widetilde{F}]_{3,1}S_{i,3} - 1.5[\widetilde{F}]_{3,2}S_{i,3}),$$
$$\delta_{i,3} = \frac{1}{V_i}(-1.5[\widetilde{F}]_{1,1}S_{i,1} - 1.5[\widetilde{F}]_{1,2}S_{i,1} + 0.5[\widetilde{F}]_{2,1}S_{i,2} + 2.5[\widetilde{F}]_{2,2}S_{i,2} + 2.5[\widetilde{F}]_{3,1}S_{i,3} + 0.5[\widetilde{F}]_{3,2}S_{i,3}).$$

A.2. 3rd Order schemes

See Fig. A2 and Tables A5–A7.

A.3. The 4th order schemes

See Fig. A3 and Tables A8–A10.

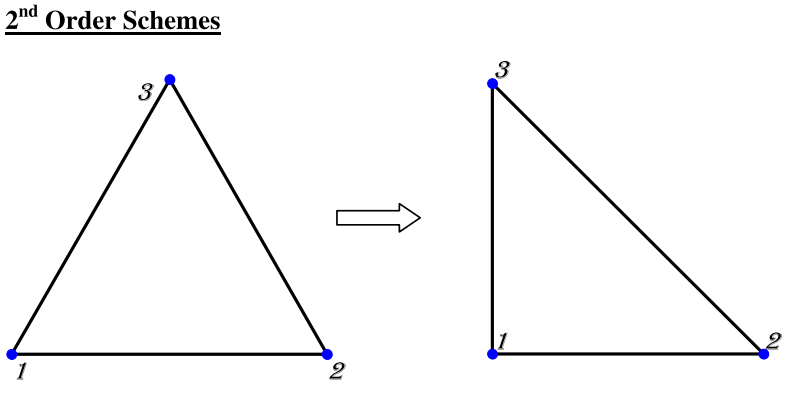


Fig. A1. Solution points for $k = 1$ element.

Table A1
Solution points for $k = 1$.

Point	ξ	η
1	0	0
2	1	0
3	0	1

Table A2
Lifting coefficients $\alpha_{j,f,l}$ for $k = 1$, LCP-DG.

(f,l)	j 1
(1,1)	2.5
(1,2)	0.5
(2,1)	−1.5
(2,2)	−1.5
(3,1)	0.5
(3,2)	2.5

Table A3
Lifting coefficients $\alpha_{j,f,l}$ for $k = 1$, LCP-SV.

(f,l)	j 1
(1,1)	2.0
(1,2)	0.2
(2,1)	−0.7
(2,2)	−0.7
(3,1)	0.2
(3,2)	2.0

Table A4
Lifting coefficients $\alpha_{j,f,l}$ for $k = 1$, LCP-SD.

(f,l)	j 1
(1,1)	2.0
(1,2)	0.0
(2,1)	−0.5
(2,2)	−0.5
(3,1)	0.0
(3,2)	2.0

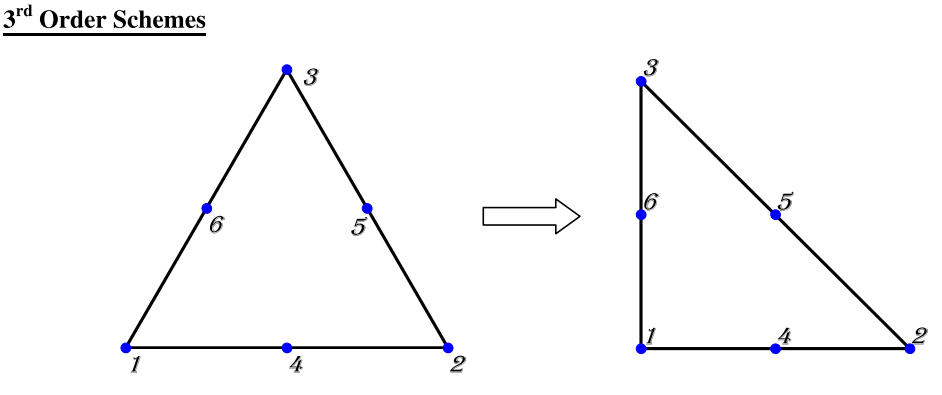


Fig. A2. Solution points for $k = 2$ element.

Table A5
Solution points for $k = 2$.

Point	ξ	η
1	0	0
2	1	0
3	0	1
4	0.5	0
5	0.5	0.5
6	0	0.5

Table A6
Lifting coefficients $\alpha_{j,f,l}$ for $k = 2$, LCP-DG.

(f,l)	j	
	1	4
(1,1)	4.5	0.5
(1,2)	2.0	5.0
(1,3)	−0.5	0.5
(2,1)	1.0	−0.625
(2,2)	4.0	−1.5
(2,3)	1.0	0.625
(3,1)	−0.5	0.625
(3,2)	2.0	−1.5
(3,3)	4.5	−0.625

Table A7
Lifting coefficients $\alpha_{j,f,l}$ for $k = 2$, LCP-SV (for the partition "SV3P" by Van den Abeele [41]).

(f,l)	j	
	1	4
(1,1)	6.53141002930328315	0.171052933692281709
(1,2)	0.739894614967283900	3.52565238715018730
(1,3)	−0.155939662412153337	0.171052933692281709
(2,1)	0.265607082915502158	−0.545239568903748461
(2,2)	0.369952521317455014	0.762826193575093650
(2,3)	0.265607082915502158	0.874186635211466751
(3,1)	−0.155939662412153337	0.874186635211466751
(3,2)	0.739894614967283900	−0.762826193575093650
(3,3)	6.53141002930328315	−0.545239568903748461

The 4th Order Schemes

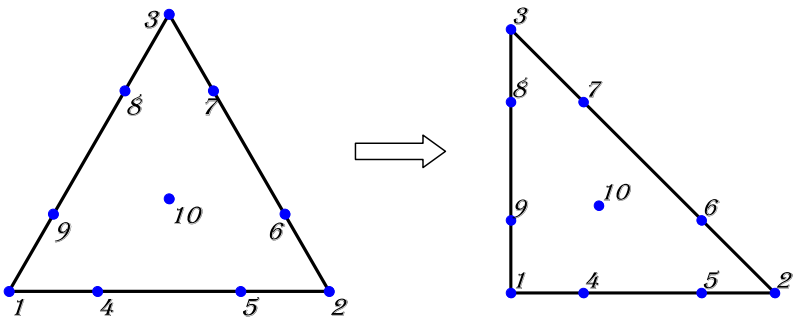


Fig. A3. Solution points for $k = 3$ element.

Table A8
Solution points for $k = 3$.

Point	ξ	η
1	0	0
2	1	0
3	0	1
4	0.27639320225002103	0
5	0.72360679774997897	0
6	0.72360679774997897	0.27639320225002103
7	0.27639320225002103	0.72360679774997897
8	0	0.72360679774997897
9	0	0.27639320225002103
10	0.333333333333333333	0.333333333333333333

Table A9
Lifting coefficients $\alpha_{j,f,l}$ for $k = 3$, LCP-DG.

(f,l)	j		
	1	4	10
(1,1)	7.	0.809016994374947424	0.0617283950617283951
(1,2)	4.04508497187473712	8.	−0.987654320987654321
(1,3)	−1.54508497187473712	1.5	−0.987654320987654321
(1,4)	0.5	−0.309016994374947424	0.0617283950617283951
(2,1)	−0.833333333333333333	0.259115050558307396	0.0617283950617283951
(2,2)	−4.16666666666666667	2.26295146066661059	−0.987654320987654321
(2,3)	−4.16666666666666667	0.0719684550415580098	−0.987654320987654321
(2,4)	−0.833333333333333333	−0.805180584266644234	0.0617283950617283951
(3,1)	0.5	−0.328152749066689099	0.0617283950617283951
(3,2)	−1.54508497187473712	1.07038187266672275	−0.987654320987654321
(3,3)	4.04508497187473712	−2.23863512170822468	−0.987654320987654321
(3,4)	7.	−0.292448383891640729	0.0617283950617283951

Table A10
Lifting coefficients $\alpha_{j,f,l}$ for $k = 3$, LCP-SV (for the partition "SV4P" by Van den Abeele [41]).

(f,l)	j		
	1	4	10
(1,1)	11.203886398585652	0.10515345237391693	0.27526713373649432
(1,2)	2.65511402064370943	6.13135672232345356	−0.389968930474013208
(1,3)	−1.11620906021771755	0.647866338549474290	−0.389968930474013208
(1,4)	0.272083026510325471	−0.356706893899520 49	0.27526713373649432
(2,1)	−0.235339114873132055	−0.225181007760767019	0.27526713373649432
(2,2)	−0.25648796757692409	0.473349519672136746	−0.389968930474013208
(2,3)	−0.25648796757692409	0.138909588123877733	−0.389968930474013208
(2,4)	−0.235339114873132055	−0.913139261303395950	0.27526713373649432
(3,1)	0.272083026510325471	−0.825187098917378380	0.27526713373649432
(3,2)	−1.11620906021771755	0.710864180742611977	−0.389968930474013208
(3,3)	2.65511402064370943	−1.25299752342169654	−0.389968930474013208
(3,4)	11.203886398585652	−0.519507774714493536	0.27526713373649432

References

[1] R. Abgrall, P.L. Roe, High-order fluctuation splitting schemes on triangular mesh, J. Sci. Comput. 19 (2003) 3–36.
[2] T.J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.
[3] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[4] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1) (1997) 267–279.
[5] M.H. Carpenter, D. Gottlieb, Spectral methods on arbitrary grids, J. Comput. Phys. 129 (1) (1996) 74–86.
[6] J. Casper, H.L. Atkins, A finite volume high-order ENO scheme for two-dimensional hyperbolic systems, J. Comput. Phys. 106 (1993) 62–76.
[7] Q. Chen, I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Eng. 128 (1995) 405–417.
[8] R.F. Chen, Z.J. Wang, Fast, block lower–upper symmetric Gauss–Seidel scheme for arbitrary grids, AIAA J. 38 (12) (2000) 2238–2245.
[9] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.
[10] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.

[11] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[12] M. Delanaye, Y. Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.
[13] J.A. Ekaterinaris, High-order accurate, low numerical diffusion methods for aerodynamics, Prog. Aerosp. Sci. 41 (2005) 192–300.
[14] S.K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271.
[15] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Method: Theory and Applications, SIAM, Philadelphia, 1977.
[16] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.
[17] R. Harris, Z.J. Wang, Y. Liu, Efficient quadrature-free high-order spectral volume method on unstructured grids: theory and 2D implementation, J. Comput. Phys. 227 (3) (2008) 1620–1642.
[18] J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, SIAM J. Numer. Anal. 35 (2) (1998) 655–676.
[19] J.S. Hesthaven, D. Gottlieb, Stable spectral methods for conservation laws on triangles with unstructured grids, Comput. Methods Appl. Mech. Eng. 175 (1999) 361–381.
[20] J.S. Hesthaven, Tim Warburton, Nodal Discontinuous Galerkin Methods, Springer, 2008.
[21] T.J.R. Hughes, 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 (1987) 1261–1275.
[22] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper 2007-4079.
[23] A. Jameson, 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 (1994) 171–218.
[24] G.E. Karniadakis, S.J. Sherwin, Spectral-hp Element Methods, Oxford University Press, Oxford, 1999.
[25] D.A. Kopriva, J.H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1996) 244.
[26] M.-S. Liou, A sequel to AUSM, Part II: AUSM+-up for all speeds, J. Comput. Phys. 214 (2006) 137–170.
[27] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, in: Proceedings of the Third International Conference on Computational Fluid Dynamics, Toronto, Canada, July 12–16, 2004.
[28] Y. Liu, M. Vinokur, Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, J. Comput. Phys. 212 (2006) 454–472.
[29] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, J. Comput. Phys. 216 (2006) 780–801.
[30] G. May, A. Jameson, A spectral difference method for the Euler and Navier–Stokes equations, AIAA Paper No. 2006-304, 2006.
[31] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, J. Comput. Phys. 213 (2006) 330–357.
[32] A. Nejata, C. Ollivier-Gooch, A high-order accurate unstructured finite volume Newton–Krylov algorithm for inviscid compressible flows, J. Comput. Phys. 227 (2008) 2582–2609.
[33] S. Osher, Riemann solvers, the entropy condition, and difference approximations, SIAM J. Numer. Anal. 21 (1984) 217–235.
[34] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Scientific Laboratory Report, LA-UR-73-479, 1973.
[35] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[36] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 261–279.
[37] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor (Eds.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, in: A. Quarteroni (Ed.), Lecture Notes in Mathematics, vol. 1697, Springer, 1998, pp. 325–432.
[38] R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong-stability preserving time discretization methods, SIAM J. Numer. Anal. 40 (2002) 469–491.
[39] Y. Sun, Z.J. Wang, Evaluation of discontinuous Galerkin and spectral volume methods for 2d Euler equations on unstructured grids, AIAA-2003-3680.
[40] K. Van den Abeele, T. Broeckhoven, C. Lacor, Dispersion and dissipation properties of the 1D spectral volume method and application to a p-multigrid algorithm, J. Comput. Phys. 224 (2) (2007) 616–636.
[41] K. Van den Abeele, C. Lacor, An accuracy and stability study of the 2D spectral volume method, J. Comput. Phys. 226 (1) (2007) 1007–1026.
[42] K. Van den Abeele, C. Lacor, Z.J. Wang, On the stability and accuracy of the spectral difference method, J. Sci. Comput. 37 (2) (2008) 162–188.
[43] K Van den Abeele, C. Lacor, Z.J. Wang, On the connection between the spectral volume and the spectral difference method, J. Comput. Phys. 227 (2) (2007) 877–885.
[44] B. van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[45] B. Van Leer, S. Nomura, Discontinuous Galerkin for diffusion, AIAA Paper No. 2005-5108, 2005.
[46] V. Venkatakrishnan, S.R. Allmaras, D.S. Kamenetskii, F.T. Johnson, Higher order schemes for the compressible Navier–Stokes equations, AIAA-2003-3987.
[47] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210–251.
[48] Z.J. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, J. Prog. Aerosp. Sci. 43 (2007) 1–47.
[49] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665–697.
[50] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, J. Sci. Comput. 20 (1) (2004) 137–157.
[51] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2) (2004) 716–741.
[52] T. Warburton, An explicit construction of interpolation nodes on the simplex, J. Eng. Math. 56 (2006) 247–262.
[53] S. Yoon, A. Jameson, Lower–upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations, AIAA J. 26 (1988) 1025–1026.
[54] M. Zhang, C-W. Shu, An analysis and a comparison between the discontinuous Galerkin method and the spectral finite volume methods, Comput. Fluids 34 (4–5) (2005) 581–592.