48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2010, Orlando, Florida
AIAA 2010-540

A High-Order Unifying Discontinuous Formulation for 3D Mixed Grids

Takanori Haga[1], Haiyang Gao[2] and Z.J. Wang[3]
Department of Aerospace Engineering and CFD Center, Iowa State University, Ames, IA 50011

The newly developed unifying discontinuous formulation based on flux reconstruction and lifting collocation penalty (LCP) approaches for conservation laws is extended to solve the Navier-Stokes equations for 3D mixed grids. In the current development, tetrahedrons and triangular prisms are considered. The LCP formulation is an extension of the flux reconstruction (FR) method to arbitrary element types. As with the FR method, it can unify several popular high order methods including the discontinuous Galerkin and the spectral volume methods into a more efficient differential form without any explicit integration. By selecting the solution points to coincide the flux points, solution reconstruction can be completely avoided. Numerical accuracy of the scheme is assessed for grid refinement studies. Several typical test cases are computed by solving the Euler equations and the compressible Navier-Stokes equations to demonstrate its performance.

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. These methods are expected to 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 devices.

For compressible flow computations in aerospace applications, the discontinuous Galerkin (DG) method [27,4-6,1,2,25,36] has attracted intensive interest. One particular feature of the DG method is the discontinuous solution space, which allows the scheme to be very flexible in dealing with complex configuration and in accommodating solution based adaptations. Other methods assuming discontinuous solutions are staggered-grid (SG) multi-domain spectral method [18], spectral volume (SV) [37,40-42,21,33,12,11] and spectral difference (SD) [20,22,24,34] methods. It is also common among these methods with discontinuous solution spaces to employ one of the Riemann solvers [29,28,26,17,19] to compute unique fluxes at element interfaces to incorporate "upwinding", similar to the Godunov type finite volume method [9,35]. These methods of course share many similarities. The main difference lies in how degrees-of-freedom (DOFs) are chose, and updated.

Recently, yet another formulation based on the idea of flux reconstruction (FR) was developed by Huynh [15,16] for 1D conservation laws. The FR 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. The extension of the method to quadrilateral and hexahedral grids is straightforward. In the previous study [39], the idea of "flux reconstruction" was generalized into a "lifting collocation penalty (LCP)" approach for arbitrary mixed grids, and numerical simulation on 2D triangular and quadrilateral grids were performed.

The present work focuses on the 3D extension of the LCP method. For high Reynolds number flows in aerodynamic applications, employing prismatic cells will have advantages in the accuracy and the computational costs to resolve boundary layer near the wall. In the present study, the LCP method is developed for tetrahedral and prismatic mixed grids.

---

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

II. Governing Equations

The 3D compressible Navier-Stokes equations can be written as a system of partial differential equations in conservation form:

$$\frac{\partial Q}{\partial t} + \nabla \cdot \left(F^c(Q) - F^v(Q, \nabla Q) \right) = 0, \tag{2.1}$$

where $Q = (\rho, \rho u, \rho v, \rho w, e)$ is the conservative state vector, $F^c(Q)$ is the convective flux, $F^v(Q)$ is the viscous flux. For a perfect gas, the pressure is related to the total energy e by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right). \tag{2.2}$$

The specific heat ratio is set to 1.4 for are and the Prandtl number in the viscous flux are assumed to be a constant of 0.72 for laminar flows. The computations for solving the Euler equations are performed by omitting the viscous flux.

III. Review of the Lifting Collocation Penalty Formulation

We will first review the LCP formulation for a hyperbolic conservation law, which can be written as

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

with suitable initial and boundary conditions. Q is the vector of conserved variables, and $\vec{F}$ is the flux vector. Assume that the computational domain is discretized into N non-overlapping triangular (in 2D) or tetrahedral (in 3D) elements $\{V_i\}$. The weighted residual form of (3.1) on element V_i can be derived by multiplying (3.1) with an arbitrary weighting or test function W and integrating over V_i,

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

Let Q_i^h be an approximate solution to Q at element V_i. We assume that the solution belongs to the space of polynomials of degree k or less, i.e., $Q_i^h \in P^k$, within each element without continuity requirement across element interfaces. Then, we require that the numerical solution Q_i must satisfy (3.2), i.e.,

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

Because the approximated solution is in general discontinuous across element interfaces, the fluxes at the interfaces are not well defined. To evaluate a unique flux and also to provide element coupling, a common Riemann flux is used to replace the normal flux, i.e.,

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

where Q_{i+}^h is the solution from outside of the current element V_i. Thus, Eq. (3.3) becomes

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

If the space of W is chosen to be one of trial functions Q^h, Eq. (3.5) is equivalent to the DG formulation. For the sake of much simpler formulation, we wish to eliminate the test function. Applying integration by parts to the last term of (3.5), we obtain

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

Note that the surface integral comes from the last term of (3.5) is evaluated by only using the internal solution. The last term of (3.6) can be viewed as a penalty term, i.e., penalizing the normal flux differences $[\tilde{F}] \equiv \tilde{F}^n(Q_i, Q_{i+}, \vec{n}) - F^n(Q_i)$. Introduce a "correction field" $\delta_i \in P^k$, which is determined from the following relation defining so-called "lifting operator" for $[\tilde{F}]$.

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

Substituting (3.7) into (3.6), we obtain

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

In the present study, in order to simplify the derivation we also approximate the flux divergence by polynomials of degree k or less, i.e. $\nabla \bullet \vec{F}(Q_i^h) \in P^k$. If W is selected such that a unique solution exists, (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, we have successfully reduced the weighted residual formulation to an equivalent simple differential form, which does not involve any explicit surface or volume integrals.

To find the approximate solution Q_i^h, let the degrees-of-freedom (DOFs) be the solutions at a set of points $\{r_{i,j}\}$, named solution points (SPs). Then equation (3.9) must hold at the SPs, i.e.,

$$\frac{\partial Q_{i,j}^h}{\partial t} + \nabla \bullet \vec{F}(Q_{i,j}^h) + \delta_{i,j} = 0. \tag{3.10}$$

Let's examine (3.7) more carefully. The correction field δ_i can be expressed in terms of $\delta_{i,j}$ using a Lagrange interpolation on the SPs, i.e.,

$$\delta_i = \sum_j L_j^{\mathrm{SP}}(\vec{r}_{i,j}) \delta_{i,j}, \tag{3.11}$$

where $L^{\mathrm{SP}} \in P^k$ is the Lagrange polynomials based on the SPs. In the case of a non-linear flux vector, $\vec{F}(Q_i^h)$ is not polynomial in general. In the present study, we approximate $\vec{F}(Q_i^h)$ by polynomial of degree k to evaluate RHS of eq. (3.7). Thereby, we assume that the flux difference $[\tilde{F}]$ is a polynomial on planer face f, and can be determined based on values of $[\tilde{F}]_{f,l}$ at a set of flux points (FPs) $\{r_{i,j}\}$ using a Lagrange interpolation, i.e.,

$$[\tilde{F}]_f = \sum_l L_l^{\mathrm{FP}}(\vec{r}_{f,l}) [\tilde{F}]_{f,l}, \tag{3.12}$$

where $L^{\mathrm{FP}} \in P^k$ is the Lagrange polynomials based on the FPs. Then, if the locations of the solution and flux points are specified and the weighting function W is chosen so as to have the same dimension as the correction field δ_i, $\delta_{i,j}$ can be uniquely defined by solving the linear system derived from eq. (3.7). Consequently, it is computed explicitly in the following formula:

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

where $\alpha_{j,f,l}$ are constant coefficients independent of the solution for any triangle or tetrahedron with straight faces. Substituting (3.13) into (3.10) we obtain the following formulation

$$\frac{\partial Q^h_{i,j}}{\partial t} + \nabla \bullet \vec{F}(Q^h_{i,j}) + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_{l} \alpha_{j,f,l} [\tilde{F}]_{f,l} S_f = 0. \tag{3.14}$$

Obviously, this is a collocation-like formulation with penalty-like terms to provide the coupling between elements. It can be shown that the location of SPs does not affect the numerical scheme for linear conservation laws [34,15]. For efficiency, the solution points are always chosen to coincide with the flux points. Therefore, any data interpolation is no longer needed for flux calculation, which dramatically reduces the computational cost. Any convergent nodal sets with enough points at the element interface are good candidates, e.g., those found in [3,13,43].

Finally, we remark on the relationship between the LCP formulation and other methods including DG, SV and SD methods. Starting from the weighted residual form of the governing equations, different formulations can be derived depending on the weighting function. For example, a nodal DG formulation is obtained by choosing weighting functions to be Lagrange polynomials, and a SV formulation is obtained by defining weighting functions as piecewise constant at the sub-cells. As a result, the only difference between those schemes appears in the correction coefficients. In the original work [39], it was shown that the resulting LCP scheme is basically conservative by using the correction coefficients for the DG, SV and SD scheme. In this study, we choose the weighting function to be one of the Lagrange polynomials based on the SPs, i.e., eq. (3.9) is identical to the DG formulation.

A. Inviscid Flux Calculation

In the eq. (3.14), we need to discretize the internal flux divergence and the common flux at the interface. Here we consider the inviscid flux. Instead of approximating the inviscid flux by the Lagrange interpolation on the SPs, the flux divergence is calculated "exactly" at the solution points by the chain rule (CR) approach

$$\nabla \bullet \vec{F}_{\mathrm{c}}(Q^h_{i,j}) = \frac{\partial \vec{F}_{\mathrm{c}}(Q^h_{i,j})}{\partial Q} \bullet \nabla Q^h_{i,j}, \tag{3.15}$$

where $\frac{\partial \vec{F}_{\mathrm{c}}}{\partial Q}$ is the inviscid flux Jacobian matrix. Note that $\nabla \bullet \vec{F}_{\mathrm{c}}(Q^h_i)$ is generally not a degree k polynomial, but it can be approximated by the Lagrange polynomial of degree k using the flux divergence at the solution points, i.e.,

$$\nabla \bullet \vec{F}_{\mathrm{c}}(Q^h_i) \approx \nabla \bullet \vec{F}^{\mathrm{CR}}_{\mathrm{c}}(Q^h_i) = \sum_j L_j(\vec{r}_{i,j}) \nabla \bullet \vec{F}_{\mathrm{c}}(Q^h_{i,j}), \tag{3.16}$$

where $L \in P^k$ is the Lagrange polynomials based on the SPs. This implies the flux vector $\vec{F}^{\mathrm{CR}}_{\mathrm{c}}(Q^h_i)$ belongs to P^{k+1} which is one degree higher than the approximation of $\vec{F}(Q^h_i) \in P^k$ used in the correction term. The chain rule approach is known to be more accurate, though the resulting scheme is not strictly conservative due to the inconsistency between the approximated flux vectors in the flux divergence term and the correction term. However, it is also known that the mass conservation error is still very small [39], and so this approach will be beneficial for many problems.

The common inviscid flux at the interfaces can be obtained with any Riemann solver. In this paper, The Roe flux [28] is used for all the cases.

B. Viscous Flux Calculation

In the recent study [8], the LCP method is extended to the Navier-Stokes equations on 2D mixed meshes. Here we follow the viscous flux discretization using the BR2 formulation [2]. In the DG methods, a mixed formulation introducing a new variable $R = \nabla Q$ in degree *k* polynomial is commonly used to discretize the viscous flux, which is a function of the state variables and their gradients. The approximations of *R* by polynomials of degree *k* also can be found by the LCP formulation, this will result in a collocation formulation

$$R_{i,j} = (\nabla Q_i^h)_j + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [\tilde{Q}]_{f,l} \vec{n}_f S_f, \tag{3.17}$$

where $[\tilde{Q}]_{f,l} \equiv Q_{f,l}^{com} - Q_{i;f,l}$, $Q_{f,l}^{com}$ is the common solution on the interface *f*, and $Q_{i;f,l}$ is the solution on *f* evaluated in the element *i*. In the BR2 method, the common solution is simply the average of the solutions at two sides of elements on *f*. The viscous fluxes at the solution points are evaluated by $F_{i,j}^v = F^v(Q_{i,j}^h, R_{i,j})$, then the viscous flux divergence is obtained through the Lagrange interpolation.

In the correction term, the common viscous flux is also need to be determined. Besides the common solution, we also define a common gradient on the interfaces *f*. The common gradient is evaluated as

$$\nabla Q_{f,l}^{com} = \frac{1}{2}\left(\nabla Q_{f,l}^- + r_{f,l}^- + \nabla Q_{f,l}^+ + r_{f,l}^+\right), \tag{3.18}$$

where $\nabla Q_{f,l}^-$ and $\nabla Q_{f,l}^+$ are the gradients of the solution from the left and right cells, while $r_{f,l}^-$ and $r_{f,l}^+$ are the local lifting correction to the gradients only due to the common solution on face *f*

$$r_{f,l}^{\pm} = \frac{1}{|V^{\pm}|} \sum_m \alpha_{l,f,m} [\tilde{Q}]_{f,m}^{\pm} \left(\mp \vec{n}_f\right) S_f, \tag{3.19}$$

where *m* is the index for the flux points on *f* and $\vec{n}_f$ is the unit normal vector directing from left to right.

IV. Discretization on mixed grids with curved boundary

It is obvious that (3.9) is valid for arbitrary types of elements besides triangles and tetrahedrons. The current development for 3D hybrid meshes accommodates two kinds of element shapes, i.e., tetrahedron and triangular prism. Other types of element such as hexahedron and pyramid will be developed in the near future. The use of prismatic cells in addition to tetrahedral cells has the advantage in both accuracy and computational costs to resolve boundary layers near solid walls. In order to achieve an efficient implementation, all elements are transformed from the physical domain (*x*, *y*, *z*) into a corresponding standard element in the computational domain (ξ, η, ζ) as shown in Fig. 1. Here we consider the transformations for the elements with curved sides (faces and edges). The discretization for the curved elements is conducted in the same way as the straight sided elements by applying the LCP formulation in the standard elements. In the present study, the quadratic triangular face is employed to represent curved wall boundaries. For the sake of computational efficiency, the quadratic representation is adopted for only one of the faces of tetrahedron which will be attached to the wall in inviscid flows, and for only two triangular faces of prism which will be used in the thin layers of prism cells to assure the quality of the element shape especially in high Reynolds number flows.

Based on a set of locations of nodes defining the shape of element, a set of shape functions can be obtained [44]. Once the shape functions $M_i(\xi,\eta,\zeta)$ are given, the transformation can be written as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi,\eta,\varsigma) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \tag{4.1}$$

where K is the number of points used to define the physical element, (x_i, y_i, z_i) are the Cartesian coordinates of those points. For the transformation given in (4.1), the Jacobian matrix J takes the following form

$$J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}. \tag{4.2}$$

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

$$\frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = J^{-1}. \tag{4.3}$$

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

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial F^\xi}{\partial \xi} + \frac{\partial F^\eta}{\partial \eta} + \frac{\partial F^\zeta}{\partial \zeta} 0, \tag{4.4}$$

where

$$\begin{aligned} \tilde{Q} &= |J| \cdot Q, \\ F^\xi &= |J| \cdot \left(\xi_x F^x + \xi_y F^y + \xi_z F^z \right), \\ F^\eta &= |J| \cdot \left(\eta_x F^x + \eta_y F^y + \eta_z F^z \right), \\ F^\zeta &= |J| \cdot \left(\zeta_x F^x + \zeta_y F^y + \zeta_z F^z \right). \end{aligned} \tag{4.5}$$

Let $\vec{S}_\xi = |J|\left(\xi_x,\xi_y,\xi_z\right)$, $\vec{S}_\eta = |J|\left(\eta_x,\eta_y,\eta_z\right)$ and $\vec{S}_\zeta = |J|\left(\zeta_x,\zeta_y,\zeta_z\right)$. Then we have $F^\xi = \vec{F} \bullet \vec{S}_\xi$, $F^\eta = \vec{F} \bullet \vec{S}_\eta$ and $F^\zeta = \vec{F} \bullet \vec{S}_\zeta$. In our implementation, $|J|$, $\vec{S}_\xi$, $\vec{S}_\eta$ and $\vec{S}_\zeta$ are stored at the solution points.

A. Discretization on Standard Tetrahedron

On a standard tetrahedron, the LCP formulation in eq. (3.14) can be rewritten as

$$\frac{\partial \tilde{Q}_{i,j}^h}{\partial t} + \nabla^\xi \bullet \vec{F}^\xi(\tilde{Q}_{i,j}^h) + \frac{1}{|V^\xi|} \sum_{f \in \partial V} \sum_l \alpha_{j,f,l} [\tilde{F}^\xi]_{f,l} S_f^\xi = 0, \tag{4.6}$$

where superscript ξ means the variables or operations evaluated on the computational domain. For example, $[\tilde{F}^\xi]$ are the normal jumps of the transformed fluxes across the faces of the standard element. The transformed normal flux can be expressed in terms of the flux in the physical space as

$$\begin{aligned}\vec{F}_n^{\xi}\Big|_{f,l} &= \vec{F}^{\xi}\Big|_{f,l} \bullet \vec{n}^{\xi}\Big|_f \\ &= \vec{F}\Big|_{f,l} \bullet \vec{S}_{\xi}\Big|_{f,l} n_{\xi}\Big|_f + \vec{F}\Big|_{f,l} \bullet \vec{S}_{\eta}\Big|_{f,l} n_{\eta}\Big|_f + \vec{F}\Big|_{f,l} \bullet \vec{S}_{\xi}\Big|_{f,l} n_{\xi}\Big|_f \\ &= \vec{F}\Big|_{f,l} \bullet \vec{S}_n\Big|_{f,l} = \vec{F}_n\Big|_{f,l} \bullet \left|\vec{S}_n\right|_{f,l},\end{aligned} \tag{4.7}$$

where $\vec{n}^{\xi} = \left\{n_{\xi}, n_{\eta}, n_{\zeta}\right\}$ is a unit normal vector on a straight face of the standard element, and $\vec{S}_n$ is a normal vector on a face in the physical space defined as $\vec{S}_n = \vec{S}_{\xi} n_{\xi} + \vec{S}_{\eta} n_{\eta} + \vec{S}_{\zeta} n_{\zeta}$.

Note that solving Eq. (4.4), $\tilde{Q} = |J| \cdot Q$ is the solution unknowns, and is assumed to be degree k polynomials in the computational domain instead of Q. As a result, the derivatives of Q should be calculated in the following way,

$$\frac{\partial Q}{\partial \xi} = \frac{1}{|J|}\left[\frac{\partial(|J|Q)}{\partial \xi} - \frac{\partial |J|}{\partial \xi} Q\right], \quad \frac{\partial Q}{\partial \eta} = \frac{1}{|J|}\left[\frac{\partial(|J|Q)}{\partial \eta} - \frac{\partial |J|}{\partial \eta} Q\right], \quad \frac{\partial Q}{\partial \zeta} = \frac{1}{|J|}\left[\frac{\partial(|J|Q)}{\partial \zeta} - \frac{\partial |J|}{\partial \zeta} Q\right]. \tag{4.8}$$

B. Discretization on Standard Prism

For a standard triangular prism, the solution polynomial can be expressed as a tensor product of a 1D and 2D Lagrange polynomial, i.e.,

$$\tilde{Q}_i^h(\xi,\eta,\zeta) = \sum_{k=1} \sum_{j=1} \tilde{Q}_{i;j,k}^h L_j(\xi,\eta) L_k(\zeta), \tag{4.9}$$

where $\tilde{Q}_i^h$ are the state variables at the solution point (j,k), with j the index in ξ-η plane and k the index in ζ direction, $L_j(\xi,\eta)$ is a 2D Lagrange polynomial in a triangle and $L_k(\zeta)$ is a 1D Lagrange polynomial in a segment. Figure 2 shows the possible locations of the solution points for k=3.

For the extension of the FR method to 2D quadratic elements [15], the solution procedure reduces to a series of one-dimensional operations. That is, the solution polynomial is represented as a tensor product of 1D Lagrange polynomials and the correction due to flux difference is performed in a one-dimensional manner. The LCP formulation for a standard prism can be derived in an analogous fashion to the FR method for a quadrilateral as

$$\begin{aligned}&\frac{\partial \tilde{Q}_{i;j,k}^h}{\partial t} + \nabla^{\xi} \bullet \vec{F}^{\xi}(\tilde{Q}_{i;j,k}^h) + \frac{1}{|V_{Tri}|} \sum_{f \in \partial V_{Tri}} \sum_{l} \alpha_{j,f,l} [\tilde{F}(\xi_{f,l}, \eta_{f,l}, \zeta_k)] S_f \\ &+ \left[\tilde{F}^{\xi}(\xi_j, \eta_j, -1) - F^{\xi}(\xi_j, \eta_j, -1)\right] g_L'(\zeta_k) + \left[\tilde{F}^{\xi}(\xi_j, \eta_j, 1) - F^{\xi}(\xi_j, \eta_j, 1)\right] g_R'(\zeta_k) = 0.\end{aligned} \tag{4.10}$$

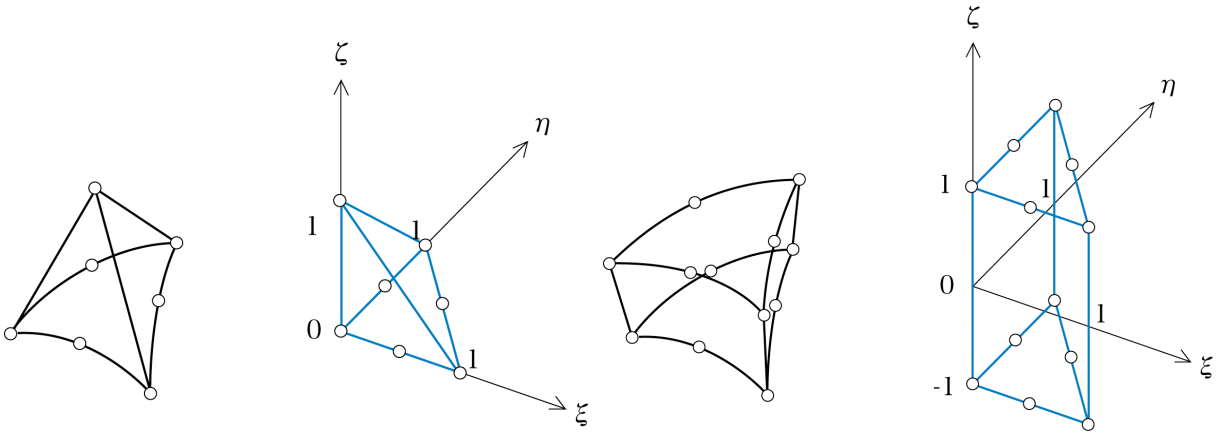


Figure 1. Transformations of a curve boundary tetrahedral and prismatic cell to the standard elements.

The correction process is done in a decoupled manner. The third term is the correction of the flux components in ξ and η direction, which is computed on a plane with fixed $\zeta = \zeta_k$. This is nothing but the correction used in the 2D LCP method for a triangle element. In eq. (4.10), V_{Tri} is the area of triangle, S_f the length of the edge f and l the index for flux points on f. Note that, $[\tilde{F}(\xi_{f,l},\eta_{f,l},\zeta_k)]$ corrects only the solution points on the triangle with fixed k instead of all solution points in the element. The last two terms denote the correction in the ζ direction, which is evaluated by the 1D FR method. g_L and g_R are both the correction functions for the left and right end points of a segment. The flux difference at a end point corrects only the solution points on the segment with fixed j. In the present study, we choose correction functions for the DG scheme from the FR family. For prism cells, the number of solution points corrected by a flux point is smaller than the one for tetrahedral cells due to the decoupled correction procedure. Hence, the method for prisms will be more efficient per DOF than for tetrahedrons. This decoupled procedure also facilitates the implementation employing different degrees of polynomials in ξ-η and ζ directions to adapt flow features. An attempt employing higher order polynomials in the wall normal direction to resolve the boundary layer with coarser prism cells is shown in the later section.

In order to simplify the implementation for mixed grids, we assume the polynomial degree k to be the same for both the tetrahedral and prismatic elements. Furthermore, the flux points along the element interfaces are required to match each other. In the present implementation, the flux points are selected to be the Legendre-Lobatto points at each edge for all tetrahedral and prismatic elements.

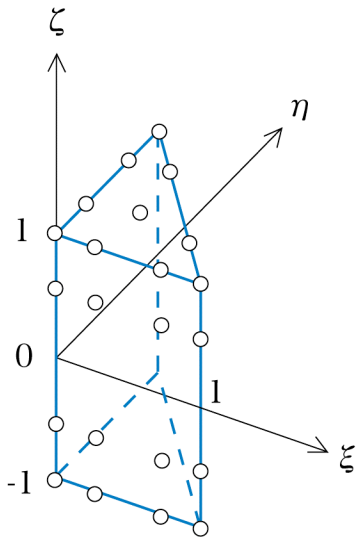


Figure 2. Solution points in the standard prism cell for degree k=3 polynomial (only points on the apparent faces are shown).

V. Numerical Results

A. Test cases for the Euler Equations

1. Accuracy Study with Vortex Evolution Problem

To assess the order of accuracy of the developed method, the propagation of an isentropic vortex in inviscid flow is computed with successive grid refinement. This is an idealized problem for the Euler equations in 2D used by Shu [30]. Here we consider simple extension of this problem to the 3D domain [0, 10]x[0, 10]x[0, 10]. The mean flow is $\{\rho, u, v, w, p\}=\{1, 1, 1, 0, 1\}$. 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 $S = p/\rho^{\gamma}$:

$$\delta u = -\bar{y}\frac{\varepsilon}{2\pi}e^{0.5(1-\bar{r}^2)},\ \delta v = \bar{x}\frac{\varepsilon}{2\pi}e^{0.5(1-\bar{r}^2)},\ \delta w = 0$$
$$\delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{1-\bar{r}^2},\ \delta S = 0, \tag{5.1}$$

where $\bar{r}^2 = \bar{x}^2 + \bar{y}^2$, $\bar{x} = x - 5$, $\bar{y} = y - 5$, and the vortex strength $\varepsilon = 5$. If the computational domain is infinitely big, the exact solution of the Euler equations with the above initial condition is just the passive convection of the isentropic vortex with the mean velocity (1, 1, 0). In the numerical simulation, we impose the exact solution on the boundaries.

The computations are carried out until t=2 on two different types of grids, tetrahedral meshes and prismatic meshes. In generating computational grids, first an equidistant Cartesian grid of NxNxN cells is assumed for the cubic domain and the each cell is further divided into six tetrahedrons or two prisms. Three different grids are employed with N=10, 20 and 40 for each type of cell. For the time integration, a 3rd-order Runge-Kutta explicit scheme is used. The L_1 and L_∞ norms of density error at the solution points are presented for tetrahedral grids and prismatic grids in Table 1 and 2, respectively. The LCP-DG method performs very well on both types of grid, achieving the nearly optimal order of accuracy up to 6th-order in tetrahedral meshes and 4th-order in prismatic meshes.

2. Subsonic Inviscid Flow over a Sphere

In order to verify the developed Euler solver on a mixed mesh with curved wall boundary, a typical steady test case of a subsonic flow around a sphere is considered. The freestream Mach number is M=0.3. Two computational grids are employed. One is a purely prismatic grid and the other is a mixed grid shown in Figures 3 (a) and 4 (a). The mixed grid is composed of five layers of prismatic cells around the quarter sphere and isotropic tetrahedral cells for the remaining region. To preserve the geometry of the sphere well with a relatively coarse mesh, the curved wall boundaries are represented by quadratic polynomials.

The computed density contours obtained with the 2nd- to 4th-order schemes are shown at Figure 3 (b)-(d) and Figure 4 (b)-(d). In both grids, the trends of improvement in the solution by increasing the order of discretization are similar. The computed density contours using the 4th order scheme appear to be perfectly symmetric without visible numerical dissipation and also quite smooth across the interface between prismatic and tetrahedral cells. In this case, a block LU-SGS implicit scheme [32,10] was used to obtain steady solutions efficiently, and all the cases converged to machine zero.

B. Test cases for the Navier-Stokes Equations

1. Accuracy Study with Couette Flow Problem

A laminar flow between two parallel walls is considered here to verify the discretization of viscous effects. The distance between the walls is set to H=10 and the computational domain is chosen to be the cube of [0, 10]x[0, 10]x[0, 10]. The speed of the moving upper wall (y=10) in the x direction is U=0.3. The temperatures of the lower wall (y=0) and the upper one are T_0=0.8 and T_1=0.85 respectively. The analytical solution for this case is

$$u = \frac{y}{H}U,\ v = 0,\ w = 0,$$
$$T = T_0 + \frac{y}{H}(T_1 - T_0) + \frac{\mu U^2}{2k}\frac{y}{H}\left(1 - \frac{y}{H}\right), \tag{5.2}$$
$$p = p_0,\ \rho = \frac{\gamma p}{T},$$

where γ is specific heat ratio and k is thermal conductivity. The static pressure is set to p_0=1/γ and the viscosity of the fluid is assumed to be μ=0.01. The flow variables at boundary faces are simply fixed to the exact solution.

Three successively refined prism grids are generated with N=2, 4 and 8 by the similar way in the vortex propagation case. Each cube is split in to two prisms by the plane which is perpendicular to the y=0 plane. The error norms for the BR2 formulation are presented in Table 3. The density is used to evaluate the error. It is shown that nearly the formal order of accuracy is achieved for the 2nd- to 4th-order schemes.

2. Boundary Layer Flow over a Flat Plate

The laminar boundary layer flow over a plate is then computed using the LCP method. The Reynolds number based on the plate length is Re_x =10,000 and the freestream Mach number is M=0.2. The plate length L is set to 1. The boundary layer thickness at the trailing edge is estimated by the approximate relation $\delta = 5L/\sqrt{Re_x}$. The computational domain is selected to be ($-2 \le x \le 1$, $0 \le y \le 100\delta$, $0 \le z \le \delta$). Note that the domain size in the y-direction is chosen to be large enough not to affect the results especially in the v-velocity profiles. The freestream values are specified at the inflow boundary at x=-2 and the top boundary at $y=100\delta$. For the lower boundary at y=0, the symmetry conditions are used on the upwind side to the wall ($-2 \le x \le 0$) and the adiabatic wall conditions are imposed on the wall ($0 \le x \le 1$). At the outflow boundary at x=0, only static pressure is prescribed. On the side boundaries at z=0 and δ, the symmetric conditions are assumed. First, we generated a three dimensional Cartesian mesh. The grid cells are clustered near the leading edge and the cell sizes are increased geometrically in both x- and y-directions. In the spanwise z-direction, we generate only one cell. Then we divide each hexahedral cell into two prisms to obtain a purely prismatic grid.

The computed u and v velocity profiles are compared with the Blasius's solution in Fig. 5. The computational grid used for the computations is generated to have 4 cells in the boundary layer at x=1.0 and 13 cells along the plate. The solution is apparently getting more accurate with the increasing of the order of polynomial approximation, and it is more clearly shown in the comparison of v-profiles. The computed skin friction coefficients on the wall are also plotted at Figure 6. The agreement with the Blasius's solution also becomes better with p-order refinement.

One of the concerning issues when we apply an CFD solver to engineering problems will be the stiffness arising from using high aspect ratio cells that is clustered near the solid wall to resolve the boundary layer especially in high Reynolds number flows. Reynolds numbers appeared in aerospace flow problems usually become ~10^6 or more, and so even if we make use of an implicit time integration scheme for numerical simulations, we will likely encounter still small time step restriction or deteriorated convergence rate. A possible remedy for this problem is employing a line solver [23,7]. Here we consider another approach to alleviate the stiffness issue by employing higher-order prism elements rather than having large number of lower order elements in the boundary layer. Since we use a tensor basis polynomial in prisms, we can use higher order polynomial only in the normal direction to the wall while using lower order one in the tangential directions to the wall so as to prevent the unnecessary increase of the computational cost.

Figure 7 shows the computed Mach number by using polynomials of degree 5 in the y-direction and polynomials of degree 2 in x- and z- directions. The grid has only two cells in the boundary layer at x=1.0 and 17 cells along the plate. The numbers of prism cells and DOFs are 728 and 26208 respectively. For the comparison, we generated another grid that has more cells in the boundary (8 cells at x=1.0) but the same resolution in the x- and z-directions and employed degree 2 polynomials in all directions, resulting 1736 prisms and 31248 DOFs. In Fig. 8, the computed v-velocity profiles are shown. The computed profiles agree well each other and also with the Blasius's solution. The convergence histories are compared in Fig. 9. The computations were performed using the LU-SGS scheme with the same time step. Compared to the computation using the lower order scheme with the finer grid, employing the higher order scheme with less grid cells gave the reductions of about 38% and 30 % in terms of Time steps and CPU times to reach machine zero residual, although the DOFs is about 16% less than the other's.

3. Subsonic Viscous Flow over a Sphere

A steady viscous flow around a sphere is computed to validate the developed NS solver on a full 3D mixed mesh. The Reynolds number based on the diameter was chosen to be 118 so that we can compare the obtained results with experimental data [31] and numerical results using the SD scheme [32,38]. The Mach number is 0.2535 that is the same value in the reference computations. The mesh is generated to have five layers of prism cells and isotropic tetrahedral cells for the remaining region. We plot the cut of the grid on a plane with y=0 and surface mesh on the sphere in Fig. 10. The total number of cells is 24,334.

The computations were performed using the 2nd- to 4th-order schemes. The computed Mach number contours and streamlines near wake using the 4th-order LCP scheme are shown in Fig. 11 and Fig. 12, respectively. We confirmed that the computed streamlines and the size of separation region agree well with both of the experimental picture and the numerical results in the references. Here we only show a comparison of the computed skin friction profiles at the cross section (y=0) of the sphere in Fig. 13. The skin friction coefficients computed by the 4th-order LCP scheme and the 6th-order SD scheme are right on top of each other. The 3rd-order LCP result also agrees well with other results, though one can see only miner differences between those profiles. The predicted separation angle using the 4th-order LCP scheme is 123.6 deg (the wind side stagnation point has an angle of 0), which is identical to

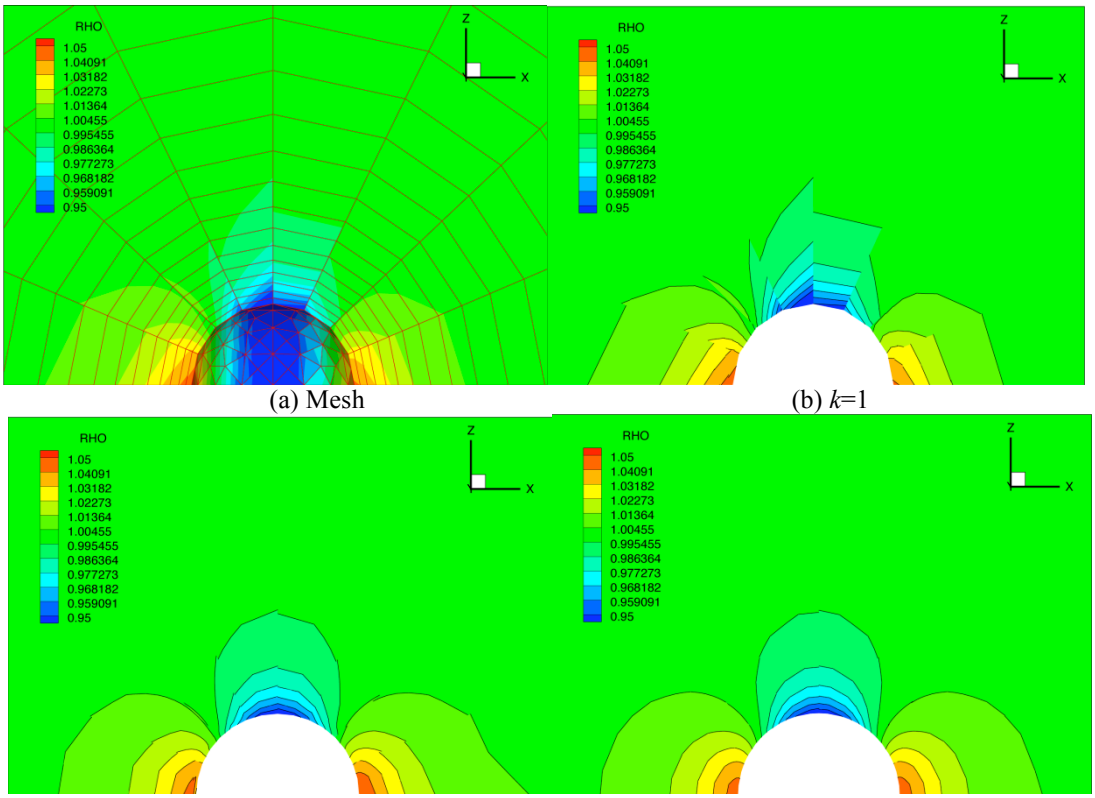
the value predicted by the 6th-order SD scheme. In Fig. 14, the computed drag coefficient by 4th-order LCP is compared to available experimental data. The agreement is also very good.

Table 1. Test of LCP-DG for vortex propagation problem (tetrahedral grids).

Polynomial degree k	*Grid size*	L_1 error	L_1 order	L_∞ error	L_∞ order
1	10x10x10x6	5.23e-3	-	9.56e-2	-
	20x20x20x6	1.42e-3	1.88	3.57e-2	1.42
	40x40x40x6	3.43e-4	2.05	9.76e-3	1.87
2	10x10x10x6	1.68e-3	-	6.06e-2	-
	20x20x20x6	2.61e-4	2.69	1.19e-2	2.35
	40x40x40x6	3.77e-5	2.79	1.51e-3	2.98
3	10x10x10x6	4.00e-4	-	2.05e-2	-
	20x20x20x6	2.44e-5	4.04	1.67e-3	3.62
	40x40x40x6	1.33e-6	4.20	1.00e-4	4.06
5	10x10x10x6	5.66e-5		2.34e-3	
	20x20x20x6	9.70e-7	5.87	7.78e-5	4.91

Table 2. Test of LCP-DG for vortex propagation problem (prismatic grids).

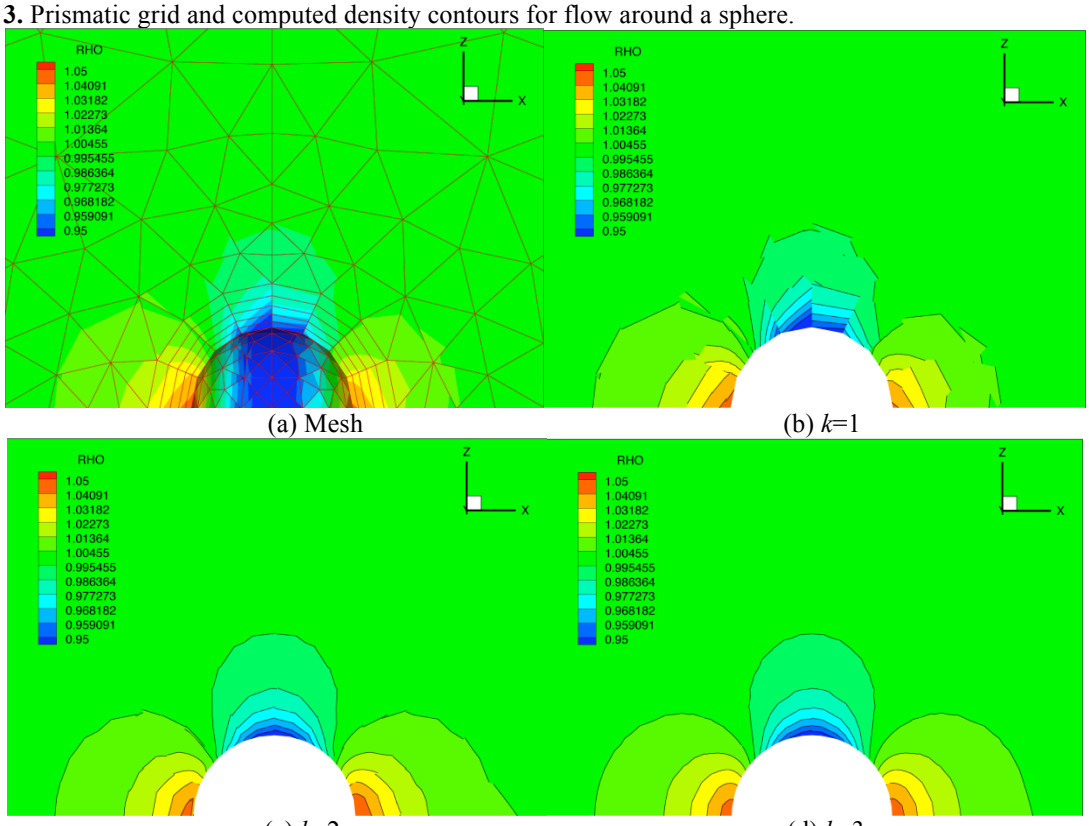
Polynomial degree k	*Grid*	L_1 error	L_1 order	L_∞ error	L_∞ order
1	10x10x10x2	7.37e-3	-	1.34e-1	-
	20x20x20x2	2.12e-3	1.80	4.85e-2	1.47
	40x40x40x2	5.19e-4	2.03	1.19e-2	2.03
2	10x10x10x2	2.17e-3	-	4.77e-2	-
	20x20x20x2	2.67e-4	3.02	8.65e-3	2.46
	40x40x40x2	2.88e-5	3.21	1.04e-3	3.06
3	10x10x10x2	4.36e-4	-	1.54e-2	-
	20x20x20x2	2.70e-5	4.01	1.43e-3	3.43
	40x40x40x2	1.64e-6	4.04	9.38e-5	3.93



(a) Mesh (b) k=1

(c) k=2 (d) k=3

Figure 3. Prismatic grid and computed density contours for flow around a sphere.



(a) Mesh (b) k=1

(c) k=2 (d) k=3

Figure 4. Mixed grid (tetrahedrons and prisms) and computed density contours for flow around a sphere.

Table 3. Test of LCP-DG (BR2) for Couette flow problem (prismatic grids).

Polynomial degree k	*Grid*	L_1 error	L_1 order	L_∞ error	L_∞ order
1	2x2x2x2	5.5525e-4	-	2.4030e-3	-
	4x4x4x2	1.1909e-4	2.221	3.9968e-4	2.588
	8x8x8x2	3.1063e-5	1.939	1.1574e-4	1.788
2	2x2x2x2	8.1732e-6	-	2.0928e-5	-
	4x4x4x2	1.2867e-6	2.667	3.3742e-6	2.633
	8x8x8x2	1.6758e-7	2.941	5.4916e-7	2.619
3	2x2x2x2	2.6248e-7	-	8.1984e-7	-
	4x4x4x2	2.0331e-8	3.690	5.7014e-8	3.846
	8x8x8x2	1.3907e-9	3.870	4.2087e-9	3.760

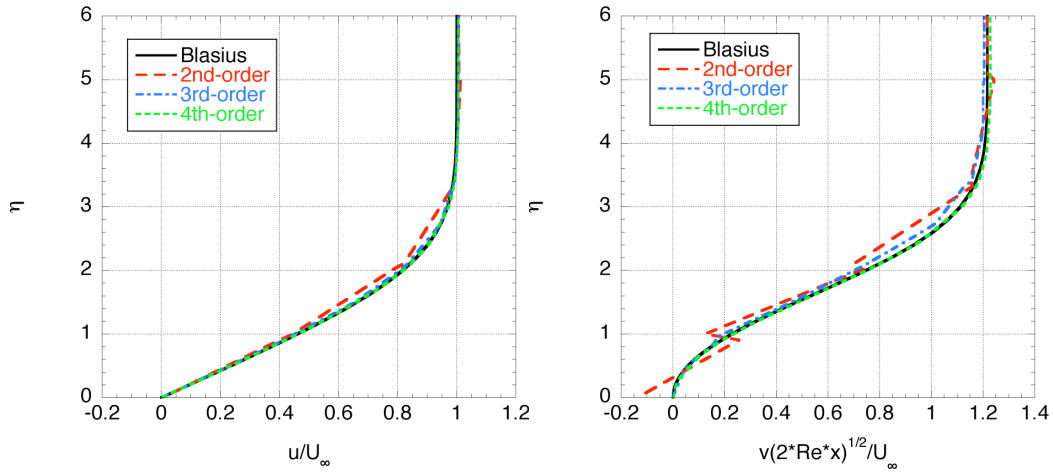


Figure 5. Comparisons of velocity profiles in the boundary layer at x=0.5. u- and v-profiles in left and right.

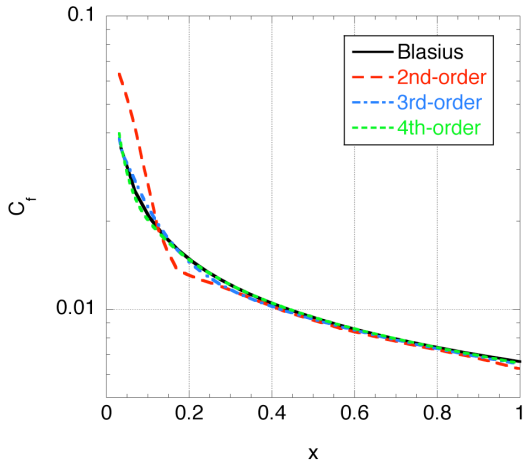


Figure 6. Comparison of the skin friction coefficient along the plate.

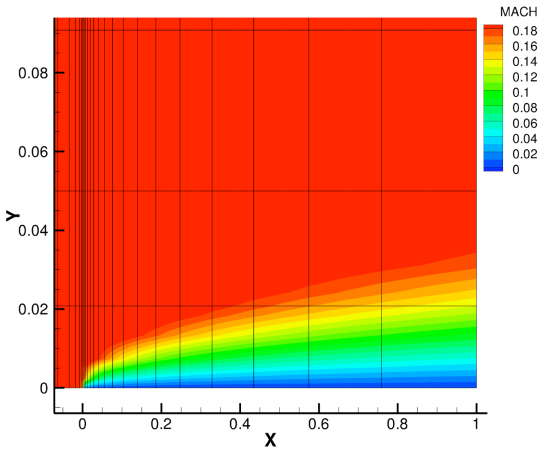


Figure 7. Grid and Mach number contours using the LCP scheme with polynomials of degree 5 in the *y*-direction (*y* direction stretched by factor 10).

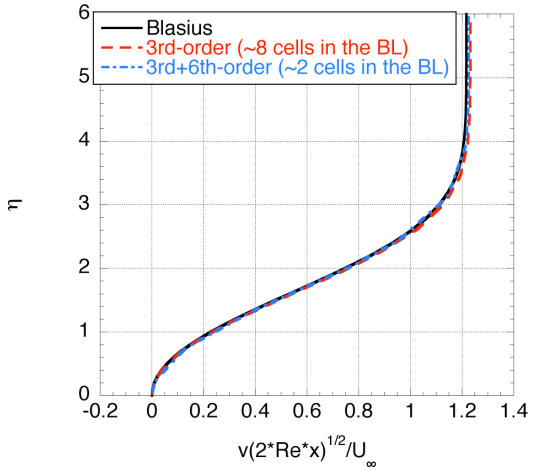


Figure 8. Comparison of *v*-velocity profiles using different degrees of polynomial and grids.

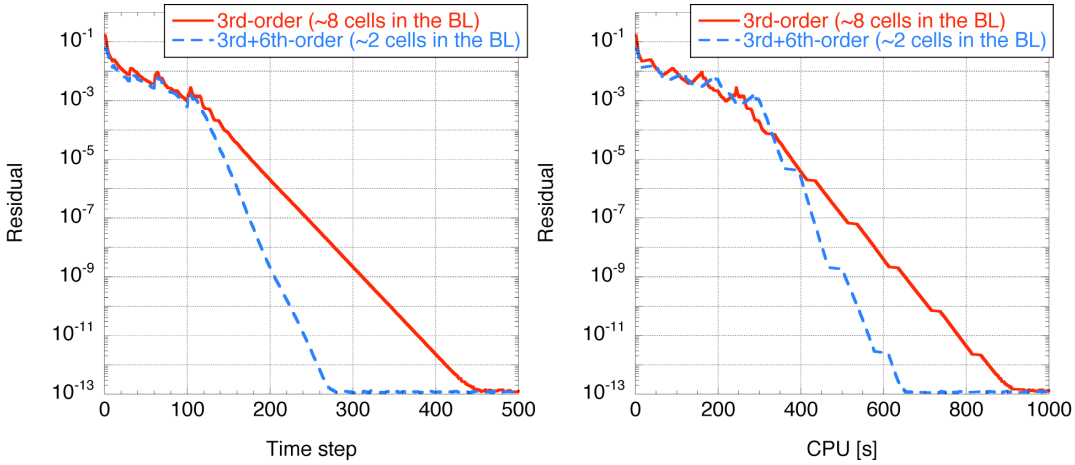


Figure 9. Comparisons of the convergence histories using different degrees of polynomial and grids.

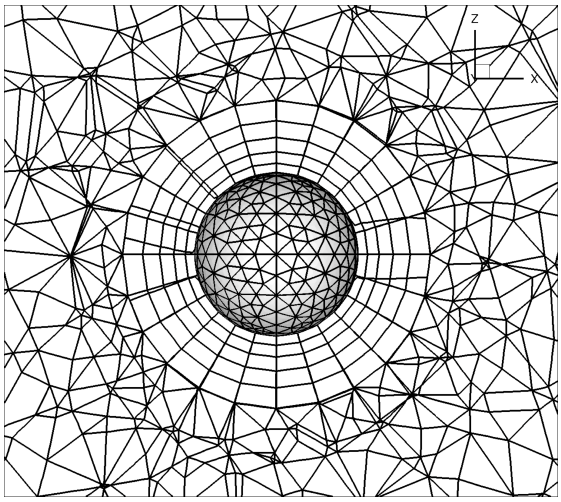


Figure 10. Computational grid around a sphere for viscous flow simulation.

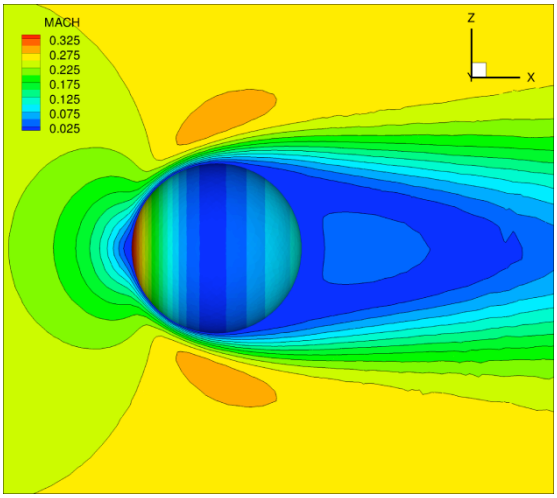


Figure 11. Computed pressure (on the sphere) and Mach number (on y=0 plane) distributions using the 4th-order LCP scheme.

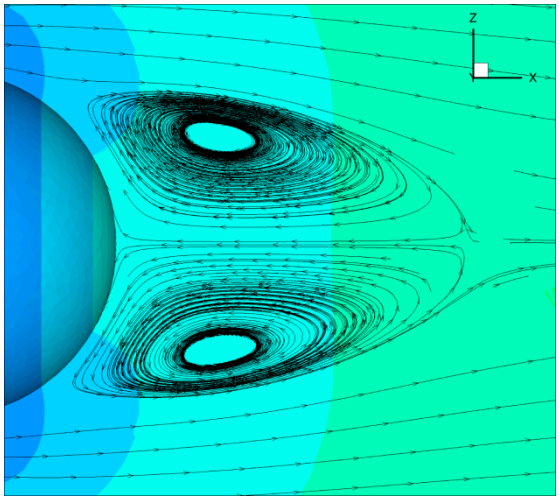


Figure 12. Computed streamlines using the 4th-order LCP scheme near the wake region behind the sphere.

VI. Conclusion

The LCP method is successfully extended to 3D hybrid unstructured meshes using tetrahedral and prismatic elements. The LCP formulation for tetrahedral elements is directly derived by the same manner for 2D triangular elements and the one for prism is obtained by just a combination of the 1D FR scheme and the 2D triangle LCP. The resulting scheme needs no explicit integrations and no data reconstructions. This numerical efficiency will be more significant in 3D simulations in comparison to 2D simulations because numerical complexities involved in high-order quadratures and reconstructions rapidly increase in 3D.

The developed LCP scheme is verified by grid converging study for an inviscid flow and a viscous flow, indicating that the developed scheme is capable of achieving the nearly optimal order of accuracy. Then, several validation cases are computed for solving the 3D Euler equations and the 3D NS equations. The LCP scheme performs very well to obtain high-order accurate solution for all cases. Future studies include further extension to adopt hexahedral and pyramidal cells for more flexible geometry discretizations and hp-adaptaion technique for realizing practical high accurate CFD simulations.

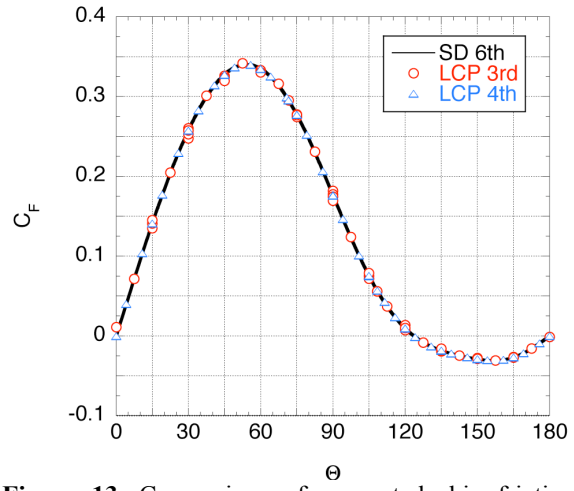


Figure 13. Comparison of computed skin friction coefficients using the 3rd- and 4th-order LCP schemes and 6th-order SD scheme[38].

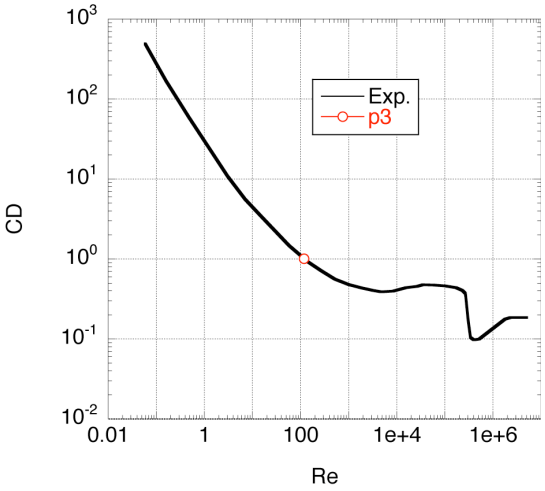


Figure 14. Comparison between the computed drag coefficient using the 4th-order LCP scheme and experimental data for a sphere.

Acknowledgments

This study has been supported by the Department of Energy (DOE) and the Air Force Office of Scientific Research (AFOSR). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DOE and AFOSR.

References

[1] 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.
[2] F. Bassi, S. Rebay, GMRES discontinuous Galerkin solution of the compressible Navier–Stokes equations, in Lecture Note in Computational Science and Engineering 11, Springer Verlag, New York, (2000), 197-208.
[3] 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.
[4] 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.
[5] 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.
[6] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[7] K.J. Fidkowski, T.A. Oliver, J. Lu, D.L. Darmofal, p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. J Comput. Phys. 207 (2005), 92-113.
[8] H. Gao, Z.J.Wang, A high-order lifting collocation penalty formulation for the Navier-Stokes equations on 2D mixed grids, AIAA Paper 2009-3784.
[9] 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.
[10] T. Haga, K. Sawada, Z.J. Wang, An implicit LU-SGS scheme for the spectral volume method on unstructured tetrahedral grids, *Communications in Computational Physics*, Vol. 6, No. 5, pp. 978-996 (2009).
[11] T. Haga, M. Furudate, K. Sawada, RANS simulation using high-order spectral volume method on unstructured tetrahedral grids, AIAA Paper, 2009-404.
[12] 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.
[13] 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.
[14] J.S. Hesthaven, Tim Warburton, Nodal Discontinuous Galerkin Methods, Springer, 2008.
[15] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper

2007-4079.
[16] H.T. Huynh, A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion, AIAA Paper 2009-403.
[17] 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.
[18] D.A. Kopriva, J.H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1996) 244.
[19] M.-S. Liou, A sequel to AUSM, Part II: AUSM+-up for all speeds, J. Comput. Phys. 214 (2006) 137–170.
[20] 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.
[21] 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.
[22] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, J. Comput. Phys. 216 (2006) 780– 801.
[23] D.J. Mavriplis, Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. J Comput. Phys. 145 (1998) 141-165
[24] G. May, A. Jameson, A spectral difference method for the Euler and Navier–Stokes equations, AIAA Paper No. 2006-304, 2006.
[25] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, J. Comput. Phys. 213 (2006) 330–357.
[26] S. Osher, Riemann solvers, the entropy condition, and difference approximations, SIAM J. Numer. Anal. 21 (1984) 217–235.
[27] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Scientific Laboratory Report, LA-UR-73-479, 1973.
[28] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[29] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 261–279.
[30] 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.
[31] S. Taneda, Experimental investigations of the wake behind a sphere at low Reynolds numbers, J. Phys. Soc. Japan, 11 (1956), 1104-1108.
[32] Y. Sun, Z.J. Wang and Y. Liu, High-Order Multidomain Spectral Difference Method for the Navier-Stokes Equations on Unstructured Hexahedral Grids, *Communications in Computational Physics*, Vol. 2, No. 2, pp. 310-333 (2007).
[33] 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.
[34] 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.
[35] 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.
[36] B. Van Leer, S. Nomura, Discontinuous Galerkin for diffusion, AIAA Paper No. 2005-5108, 2005.
[37] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210–251.
[38] Z.J. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, J. Prog. Aerosp. Sci. 43 (2007) 1–47.
[39] Z.J.Wang and H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. 228 (2009) 8161-8186.
[40] 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.
[41] 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.
[42] 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.
[43] T. Warburton, An explicit construction of interpolation nodes on the simplex, J. Eng. Math. 56 (2006) 247–262.
[44] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method The Basics, vol. 1, Butterworth–Heinemann, Oxford, England, 2000.