

# A High-Order Unifying Discontinuous Formulation for the Navier-Stokes Equations on 3D Mixed Grids

T. Haga, H. Gao and Z. J. Wang\*

Department of Aerospace Engineering and CFD Center, Iowa State University, 50011 Ames, USA

**Abstract.** The newly developed unifying discontinuous formulation named the correction procedure via reconstruction (CPR) 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 CPR method can unify several popular high order methods including the discontinuous Galerkin and the spectral volume methods into a more efficient differential form. By selecting the solution points to coincide with the flux points, solution reconstruction can be completely avoided. Accuracy studies confirmed that the optimal order of accuracy can be achieved with the method. Several benchmark test cases are computed by solving the Euler and compressible Navier-Stokes equations to demonstrate its performance.

**Key words:** high-order, mixed unstructured grids, Navier-Stokes equations, discontinuous Galerkin, spectral collocation, finite difference

**AMS subject classification:** 65M70, 76M20, 76M22

## 1. Introduction

Advantages of high-order methods are well recognized in the computational fluid dynamics (CFD) community especially for aeroacoustic noise predictions, vortex dominated flows, large eddy simulation and direct numerical simulation (DNS) of turbulent flows. Since the truncation error of a high-order method decreases more rapidly than that of a lower order method when the solution is

---

\*Corresponding author. E-mail: zjw@iastate.edu

sufficiently smooth, the more stringent the accuracy requirement is, the more efficient a high-order method becomes in computational cost. For the practical use in industries, lower order (1st or 2nd) unstructured grid methods are usually employed for the reason of superior geometrical flexibility and robustness. However, these methods are likely too dissipative to capture small vortex structures in turbulent flows. Also, as is reported in the Drag Prediction Workshop (DPW), it will be prohibitively expensive to reach the grid converged solution even in the steady RANS simulation for a relatively simple configuration due to the slow convergence rate. Increased prediction accuracy is often required for many aerodynamic problems with both complex physics and geometry, such as helicopter blade vortex interactions, flow over high lift devices, and aero-acoustic noise generated by the landing gear.

In the past decades, there has been significant progress in developing high-order methods capable of solving the Navier-Stokes (NS) equations on unstructured grids. For compressible flow computations in aerospace applications, the discontinuous Galerkin (DG) method [1, 2, 4, 5, 7, 28, 30, 42] has attracted intensive interest. One particular feature of the DG method is the discontinuous solution space of high-order approximations for each element, which allows the scheme to be very flexible in dealing with complex configuration and in accommodating solution based adaptations. Other methods assuming element-wise discontinuous solution are staggered-grid (SG) multi-domain spectral method [20], spectral volume (SV) [12, 14, 24, 39, 43, 46, 47, 48] and spectral difference (SD) [22, 23, 27, 40] methods. Another notable feature that is common among these methods is the use of one of the Riemann solvers [18, 21, 29, 31, 32] to compute unique fluxes at element interfaces to incorporate “upwinding” characteristics of wave propagation, similar to the Godunov type finite volume method [11, 41]. The main difference among these methods lies in how the governing equations are discretized and the degrees-of-freedom (DOFs) are chosen and updated. The DG method is based on the weighted residual form. Various types of DG schemes are derived with different choice of DOFs. Depending on how the DOFs are defined, DG schemes can be further divided into modal and nodal approaches. The SV method is discretized in the integral form similar to the finite volume method and the DOFs are always the sub-cell or control volume (CV) averages. The SG/SD method is based on the differential form without any integration and the DOFs are chosen as the nodal values within each element. More comprehensive reviews of these methods are given in [44].

Recently, a novel formulation named correction procedure via reconstruction (CPR) was developed by Huynh [16, 17] for 1D conservation laws, and extended to simplex and hybrid meshes by Wang and Gao [45]. The CPR method is based on a nodal differential form, with an element-wise discontinuous polynomial solution space. The solution polynomial is interpolated from the solutions at a set of solution points. 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. Furthermore, by choosing the solution points to coincide with the flux points, the reconstruction of solution polynomials to calculate the residual can be completely avoided. The DG scheme derived through the CPR framework is probably the simplest and most efficient amongst all DG formulations since explicit integrations are avoided. In a recent study [9], the CPR method has been extended to the Navier-Stokes equations on 2D mixed meshes. These successful developments laid a solid foundation for its efficient implementation and demonstration

on arbitrary grids in 3D.

Hybrid elements such as hex, prism, pyramid and tetrahedron will provide great geometrical flexibility for practical problems in 3D. In particular, for high Reynolds number flows in aerodynamic applications, prismatic cells have the advantages in accuracy and computational cost to resolve boundary layers near the wall. There have been several attempts to develop the DG method on arbitrary grid elements. In [33] different types of elements such as hex, prism and pyramid are projected onto a reference cube using collapsed Cartesian coordinates and hierarchical basis functions over the cube are used. Luo et.al. [25] presented a different approach based on the Taylor series expansion at the center of arbitrary element. Gassner et.al. [10] used polymorphic nodal element in the modal based formulation to reduce the cost of numerical integrations. However one obvious shortcoming of these formulations is the high computational cost of the surface and volume integrations coming from the weighted residual formulation. Another difficulty is the treatment of curvilinear boundary elements. In the CPR method, curvilinear boundary elements are dealt with the same mapping technique as finite element method (FEM). The Jacobian and metrics of the transformation matrix are stored only at the solution points and there is no additional implementation for a surface or volume integration. The simple formulation of the CPR method is expected to alleviate the computational costs and facilitate the efficient treatment of curved wall boundaries.

In the present study, we develop the CPR for solving the Euler and Navier-Stokes equations on 3D mixed meshes. Tetrahedral and prismatic elements are considered in the present study with the intention to resolve viscous boundary layer flows efficiently. The remainder of this article is organized as follows. The basic formulation of the CPR method is described in the next section. In Section 3, The discretization of the compressible Navier-Stokes equations is derived in the CPR framework. Subsequently, we discuss how to implement the CPR method efficiently in each particular element with curvilinear geometry in Section 4. Section 5 presents the computational results for several benchmark problems, including accuracy studies on mixed unstructured grids. Conclusions for the present study and possible future works are summarized in Section 6.

## 2. Review of the Correction Procedure via Reconstruction Formulation

We first review the CPR formulation for a hyperbolic conservation law, which is written as

$$\frac{\partial Q}{\partial t} + \vec{\nabla} \cdot \vec{F}(Q) = 0, \quad (2.1)$$

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 elements  $\{V_i\}$ . The weighted residual form of Eq. (2.1) on element  $V_i$  can be derived through multiplying

Eq. (2.1) by an arbitrary weighting or test function  $W$  and integrating over  $V_i$ ,

$$\int_{V_i} \frac{\partial Q}{\partial t} W dV + \int_{\partial V_i} W \vec{F}(Q) \cdot \vec{n} dS - \int_{V_i} \vec{\nabla} W \cdot \vec{F}(Q) dV = 0. \quad (2.2)$$

Let  $Q_i^h$  be an approximate solution to  $Q$  on element  $V_i$ . We assume that the solution belongs to the space of polynomials of degree  $k$  or less, i.e.,  $Q_i^h \in P^k(V_i)$ , within each element without continuity requirement across element interfaces. Then, we require that the numerical solution  $Q_i^h$  must satisfy Eq. (2.2), i.e.,

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{\partial V_i} W \vec{F}(Q_i^h) \cdot \vec{n} dS - \int_{V_i} \vec{\nabla} W \cdot \vec{F}(Q_i^h) dV = 0. \quad (2.3)$$

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.,

$$F^n(Q_i^h) \equiv \vec{F}(Q_i^h) \cdot \vec{n} \approx F_{com}^n(Q_i^h, Q_{i+}^h, \vec{n}), \quad (2.4)$$

where  $Q_{i+}^h$  is the solution from outside of the current element  $V_i$ . Thus, Eq. (2.3) becomes

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{\partial V_i} W F_{com}^n dS - \int_{V_i} \vec{\nabla} W \cdot \vec{F}(Q_i^h) dV = 0. \quad (2.5)$$

If the space of  $W$  is chosen to be the same as the solution space, Eq. (2.5) is equivalent to the DG formulation. For the sake of a simpler formulation, we wish to eliminate the test function. Applying integration by parts to the last term of Eq. (2.5), we obtain

$$\int_{V_i} \frac{\partial Q_i^h}{\partial t} W dV + \int_{V_i} W \vec{\nabla} \cdot \vec{F}(Q_i^h) dV + \int_{\partial V_i} W [F_{com}^n - F^n(Q_i^h)] dS = 0. \quad (2.6)$$

The last term of Eq. (2.6) can be viewed as a penalty term, i.e., penalizing the normal flux differences  $[F^n] = F_{com}^n - F^n(Q_i^h)$ . Let us introduce a ‘‘correction field’’  $\delta_i \in P^k(V_i)$ , which is determined from the following relation defining the so-called ‘‘lifting operator’’ for  $[F^n]$ .

$$\int_{V_i} W \delta_i dV = \int_{\partial V_i} W [F^n] dS. \quad (2.7)$$

Substituting Eq. (2.7) into Eq. (2.6), we obtain

$$\int_{V_i} \left[ \frac{\partial Q_i^h}{\partial t} + \vec{\nabla} \cdot \vec{F}(Q_i^h) + \delta_i \right] W dV = 0. \quad (2.8)$$

In the present study, in order to simplify the derivation we also approximate the flux divergence by polynomials of degree  $k$  or less. Denote  $\Pi(\vec{\nabla} \cdot \vec{F}(Q_i^h))$  a projection of  $\vec{\nabla} \cdot \vec{F}(Q_i^h)$  to  $P^k$ . If  $W$  is selected such that a unique solution exists, Eq. (2.8) is equivalent to

$$\frac{\partial Q_i^h}{\partial t} + \Pi \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right) + \delta_i = 0. \quad (2.9)$$

i.e., Eq. (2.9) 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. Even though we need to solve Eq. (2.7) through a volume and surface integral to define the correction  $\delta_i$ , we can circumvent the numerical quadratures in the DG algorithm which must be exact for degree  $2k + 1$  and  $2k$  polynomials for the surface and volume integral.

To find the approximate solution  $Q_i^h$ , let the DOFs be the solution values at a set of points  $\{\vec{r}_{i,j}\}$ , named solution points (SPs). Then Eq. (2.9) must hold at the SPs, i.e.,

$$\frac{\partial Q_{i,j}^h}{\partial t} + \Pi_j \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right) + \delta_{i,j} = 0, \quad (2.10)$$

where  $\Pi_j \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right)$  denotes the values of  $\Pi \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right)$  at SP  $j$ . Once the location of SPs is defined,  $Q_i^h$  and  $\delta_i$  can be expressed in terms of values at SPs using a Lagrange interpolation, i.e.,

$$Q_i^h = \sum_j L_j^{SP}(\vec{r}_{i,j}) Q_{i,j}^h, \quad (2.11)$$

$$\delta_i = \sum_j L_j^{SP}(\vec{r}_{i,j}) \delta_{i,j}, \quad (2.12)$$

where  $L^{SP} \in P^k$  are the Lagrange polynomials based on the location of SPs.

To compute  $\delta_{i,j}$ , we approximate (for nonlinear conservation laws) the normal flux difference  $[F^n]$  in the RHS of Eq. (2.7) with a degree  $k$  polynomial on each interface. The interpolation can be determined by defining flux points (FPs) as,

$$[F^n]_f \approx \sum_l L_l^{FP}(\vec{r}_{f,l}) [F^n]_{f,l}, \quad (2.13)$$

where  $f$  is an face index, and  $l$  is the FP index, and  $L_l^{FP}$  is the Lagrange interpolation polynomial based on the FPs in a local interface coordinate. Then, if the locations of the solution and flux points are specified,  $\delta_{i,j}$  can be uniquely defined by solving the linear system derived from Eq. (2.7). For simplex elements with straight faces, it can be expressed in the following formula

$$\delta_{i,j} = \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [F^n]_{f,l} S_f, \quad (2.14)$$

where  $\alpha_{j,f,l}$  are constant coefficients independent of the solution and the shape of the simplex.

Next, we focus on how to compute  $\Pi_j \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right)$ . Two approaches are developed in [45]. One is the Lagrange polynomial (LP) approach which approximate the (nonlinear) flux vector with degree  $k$  interpolation polynomials

$$\vec{F}(Q_i^h) \approx \sum_j L_j^{SP}(\vec{r}) \vec{F}(Q_{i,j}^h), \quad (2.15)$$

Then, the projection of the flux divergence is computed as

$$\Pi \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right) = \sum_j \vec{\nabla} L_j^{SP} \cdot \vec{F}(Q_{i,j}^h). \quad (2.16)$$

In this case,  $\Pi \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right)$  is a degree  $k - 1$  polynomial, which also belongs to  $P^k$ . Numerical experiments indicate that there is a slight loss of accuracy with the LP approach, but it is fully conservative [45]. Another is the chain rule (CR) approach. The divergence of the flux vector can be computed analytically given the approximate solution using the chain rule, i.e.,

$$\vec{\nabla} \cdot \vec{F}(Q_{i,j}^h) = \frac{\partial \vec{F}(Q_{i,j}^h)}{\partial Q} \cdot \nabla Q_{i,j}^h, \quad (2.17)$$

where  $\frac{\partial \vec{F}}{\partial Q}$  is composed of the flux Jacobian matrices, which can be computed analytically. Then the projection is approximated by the Lagrange polynomial of degree  $k$  using the flux vector divergence at the solution points, i.e.,

$$\Pi \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right) \approx \sum_j L_j^{SP}(\vec{r}) \vec{\nabla} \cdot \vec{F}(Q_{i,j}^h). \quad (2.18)$$

Numerical experiments indicate that the CR approach is much more accurate than the LP approach, at the expense of full conservation [45].

Substituting Eq. (2.14) into Eq. (2.10) we obtain the following equation

$$\frac{\partial Q_{i,j}^h}{\partial t} + \Pi_j \left( \vec{\nabla} \cdot \vec{F}(Q_i^h) \right) + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [F^n]_{f,l} S_f = 0. \quad (2.19)$$

One can clearly see that this is a collocation-like formulation with penalty-like term that comes from the element-wise correction polynomial 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 [40, 16]. For efficiency, the solution points are always chosen to coincide with the flux points. Therefore, no data interpolation is needed for flux calculation, which further reduces the computational cost. Any convergent nodal sets with enough points at the element interface are good candidates, e.g., those found in [3, 15, 49].

Finally we want to make a remark on the relationship between the CPR 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. Note that there are certainly differences how to discretize the spatial terms between the CPR using the DG coefficients (CPR-DG) and the nodal DG FEM. The same is true for the SV method. Nevertheless, it was numerically confirmed that the CPR-DG or CPR-SV is

equivalent to the DG or SV method at least for linear conservation laws in [45]. In [45], it was also shown that the resulting CPR scheme is fully conservative by using the correction coefficients for the DG, SV and SD scheme if the flux divergence term is evaluated with the LP approach. In this study, we choose the weighting function to be one of the Lagrange polynomials based on the SPs, i.e., Eq. (2.9) is identical to the DG formulation.

### 3. Discretization of the Navier-Stokes Equations

#### 3.1. Governing Equations

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

$$\frac{\partial Q}{\partial t} + \vec{\nabla} \cdot \left( \vec{F}_c(Q) - \vec{F}_v(Q, \vec{\nabla}Q) \right) = 0, \quad (3.1)$$

where  $Q$ ,  $\vec{F}_c = (F_c^x, F_c^y, F_c^z)$  and  $\vec{F}_v = (F_v^x, F_v^y, F_v^z)$  denote the conservative state vector, the inviscid and the viscous flux vectors, respectively, and are given by

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix}, \quad F_c^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{pmatrix}, \quad F_c^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{pmatrix}, \quad F_c^z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{pmatrix}. \quad (3.2)$$

$$F_v^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{pmatrix}, \quad F_v^y = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{pmatrix},$$

$$F_v^z = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{pmatrix}. \quad (3.3)$$

where  $\rho$  is the density,  $\vec{v} = (u, v, w)$  the velocity vector,  $p$  the pressure,  $e$  the total energy per unit volume. The viscous stress tensor can be represented as

$$\tau = \mu \left( \vec{\nabla} \vec{v} + (\vec{\nabla} \vec{v})^T - \frac{2}{3} (\vec{\nabla} \cdot \vec{v}) I \right). \quad (3.4)$$

where  $\mu$  is the molecular viscosity coefficient,  $I$  is the unit tensor. The heat flux is given as

$$\vec{q} = -c_p \frac{\mu}{Pr} \vec{\nabla} T. \quad (3.5)$$

Here,  $c_p$  is the specific heat capacity at constant pressure and  $T$  is the temperature. The Prandtl number  $Pr$  is assumed to be a constant of 0.72 in this study. For a perfect gas, the pressure is related to the total energy  $e$  by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2} \rho (u^2 + v^2 + w^2). \quad (3.6)$$

The specific heat ratio  $\gamma$  is set to be a constant, 1.4 for air. The computations for solving the Euler equations are performed by omitting the viscous flux.

### 3.2. CPR Formulation of the Navier-Stokes Equations

In order to discretize the Navier-Stokes equations, we follow a mixed formulation that is commonly used for the DG method [2, 6]. By introducing a new variable  $\vec{R}$ , Eq. (3.1) is rewritten in a first order system as

$$\frac{\partial Q}{\partial t} + \vec{\nabla} \cdot (\vec{F}_c(Q) - \vec{F}_v(Q, \vec{R})) = 0, \quad (3.7)$$

$$\vec{R} = \vec{\nabla} Q. \quad (3.8)$$

According to the CPR formulation by assuming  $Q_i^h \in P^k$ ,  $\vec{R}_i^h \in (P^k, P^k, P^k)$  on discretized elements  $\{V_i\}$ , we obtain

$$\frac{\partial Q_{i,j}^h}{\partial t} + \Pi_j (\vec{\nabla} \cdot \vec{F}_c(Q_i^h)) - \Pi_j^v (\vec{\nabla} \cdot \vec{F}_v(Q_i^h, \vec{R}_i^h)) + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} ([F_c^n]_{f,l} - [F_v^n]_{f,l}) S_f. \quad (3.9)$$

$$\vec{R}_{i,j}^h = (\vec{\nabla} Q_i^h)_j + \frac{1}{|V_i|} \sum_{f \in \partial V_i} \sum_l \alpha_{j,f,l} [Q]_{f,l} \vec{n}_f S_f, \quad (3.10)$$

where  $[F_c^n] \equiv F_{c,com}^n - F_c^n(Q_i^h, \vec{n})$ ,  $[F_v^n] \equiv F_{v,com}^n - F_v^n(Q_i^h, \vec{R}_i^h, \vec{n})$  and  $[Q] \equiv Q_{com}^h - Q_i^h$ .

#### 3.2.1. Inviscid Flux Calculation

We need to discretize the projection of the inviscid flux divergence  $\Pi_j (\vec{\nabla} \cdot \vec{F}_c(Q_i^h))$  and the difference of the normal inviscid flux  $[F_c^n]$  in the correction term in Eq. (3.9). To compute the inviscid flux divergence we employed the CR approach in the present study. The common inviscid flux  $F_{c,com}^n$  can be obtained with any Riemann solver. We used the Roe flux [31] for all the cases.



### 3.2.2. Viscous Flux Calculation

In the present study, we employ the BR2 scheme [2] to discretize the viscous flux. In Eq. (3.10), the common solution  $Q_{com}^h$  is simply the average of the solutions at both sides of  $f$ . The viscous flux vector at the solution points are evaluated by  $\vec{F}_v(Q_i^h, \vec{R}_i^h)$ . Then the projection of the viscous flux divergence  $\Pi_j^v \left( \vec{\nabla} \cdot \vec{F}_v(Q_i^h, \vec{R}_i^h) \right)$  is obtained through the LP approach instead of the CR approach. In the correction term, the common viscous flux  $F_{v,com}^n(Q_{com}^h, \vec{\nabla} Q_{com}^h, \vec{n})$  also needs to be determined. Besides the common solution, we also need to define a common gradient  $\vec{\nabla} Q_{com}^h$  on face  $f$ . The common gradient at a flux point  $l$  on  $f$  is evaluated as

$$\vec{\nabla} Q_{com}^h \Big|_{f,l} = \frac{1}{2} (\vec{\nabla} Q_{f,l}^- + \vec{r}_{f,l}^- + \vec{\nabla} Q_{f,l}^+ + \vec{r}_{f,l}^+), \quad (3.11)$$

where  $\vec{\nabla} Q_{f,l}^-$  and  $\vec{\nabla} Q_{f,l}^+$  are the gradients of the solution from the left and right cells.  $\vec{r}_{f,l}^-$  and  $\vec{r}_{f,l}^+$  are the local lifting corrections to the gradients only due to the common solution on face  $f$

$$\vec{r}_{f,l}^\pm = \frac{1}{|V^\pm|} \sum_{m=1} \alpha_{l,f,m} [Q]_{f,m} (\mp \vec{n}_f) S_f, \quad (3.12)$$

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. Note that there is no summation over all faces of the element in Eq. (3.12) in order to assure that the BR2 scheme maintains a compact face neighbor stencil.

## 4. Discretization on Mixed Grids with Curved Boundary

It can be observed that Eq. (2.9) 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 advantages 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  $(\xi, \eta, \zeta)$  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 CPR formulation in the standard elements. In the present study, a 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 tetrahedra attached to the wall in inviscid flows, and for only two triangular faces of prisms 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 [50]. Once the shape functions  $M_i(\xi, \eta, \zeta)$  are given, the transformation can be written as

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

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 Eq. (4.1), the Jacobian matrix  $J$  takes the following form

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

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

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

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

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial F^\xi}{\partial \xi} + \frac{\partial F^\eta}{\partial \eta} + \frac{\partial F^\zeta}{\partial \zeta} = 0, \quad (4.4)$$

where

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

Let  $\vec{S}_\xi = |J|\vec{\nabla}\xi$ ,  $\vec{S}_\eta = |J|\vec{\nabla}\eta$  and  $\vec{S}_\zeta = |J|\vec{\nabla}\zeta$ . Then we have  $F^\xi = \vec{F} \cdot \vec{S}_\xi$ ,  $F^\eta = \vec{F} \cdot \vec{S}_\eta$  and  $F^\zeta = \vec{F} \cdot \vec{S}_\zeta$ . In our implementation,  $J$ ,  $\vec{S}_\xi$ ,  $\vec{S}_\eta$  and  $\vec{S}_\zeta$  are stored at the solution points. Note that here we consider the Euler equations as the governing equations for brevity's sake. Extending the following discretization to the Navier-Stokes equations is straightforward.

#### 4.1. Discretization on a Standard Tetrahedron

On a standard tetrahedron, the CPR formulation in Eq. (2.19) can be rewritten as

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

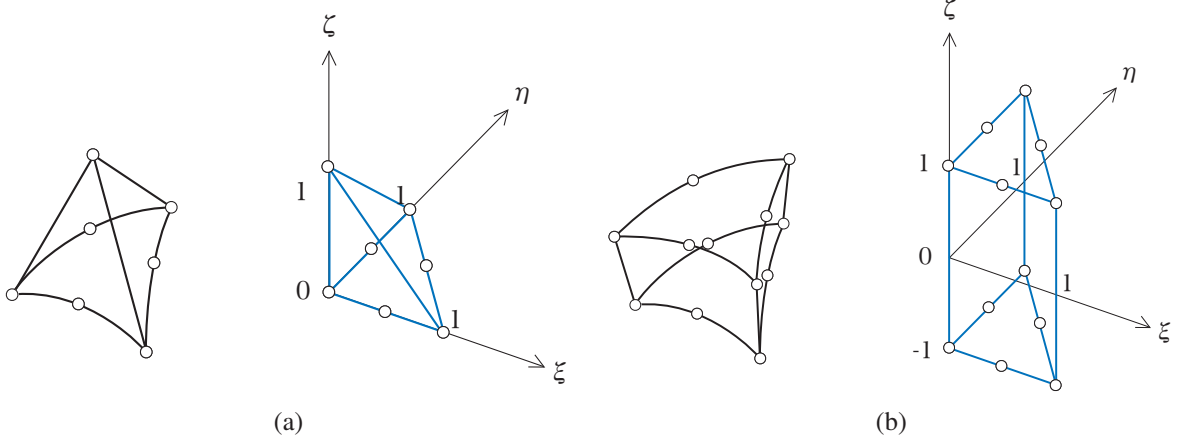


Figure 1: Transformation of curve boundary tetrahedral and prismatic cells to the standard elements.

where superscript  $(\xi)$  means the variables or operations evaluated on the computational domain. For example,  $[F^{n,(\xi)}]$  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

$$\begin{aligned}
 F^{n,(\xi)}|_{f,l} &= \vec{F}^{(\xi)}|_{f,l} \cdot \vec{n}^{(\xi)}|_f \\
 &= \vec{F}|_{f,l} \cdot \vec{S}_\xi|_{f,l} n^\xi|_f + \vec{F}|_{f,l} \cdot \vec{S}_\eta|_{f,l} n^\eta|_f + \vec{F}|_{f,l} \cdot \vec{S}_\zeta|_{f,l} n^\zeta|_f \\
 &= \vec{F}|_{f,l} \cdot \vec{S}^n|_{f,l} = F^n|_{f,l} |\vec{S}^n|_{f,l},
 \end{aligned} \tag{4.7}$$

where  $\vec{n}^{(\xi)} = (n^\xi, n^\eta, n^\zeta)$  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 in solving Eq. (4.4),  $\tilde{Q} = |J|Q$  are the solution unknowns, and are 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,

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

In 3D, to construct a complete polynomial of degree  $k$ , at least  $k(k+1)(k+2)/3!$  SPs need to be chosen. In order to achieve the most efficient implementation, SPs on edges are chosen to be the Legendre-Gauss Lobatto (LGL) points. For 4th- or higher order schemes, nodes inside the boundary triangle are chosen from [15]. For 5th- or higher order schemes, nodes inside the tetrahedron are chosen from [49]. The nodal set of the 4th-order CPR scheme is shown in Fig. 2. Note that the flux difference at a flux point corrects all solution points as shown in Eq. (4.6).

## 4.2. 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.,

$$\tilde{Q}_i^h(\xi, \eta, \zeta) = \sum_m \sum_j \tilde{Q}_{i;j,m}^h L_j^{SP}(\xi, \eta) L_m^{SP}(\zeta), \quad (4.9)$$

where  $\tilde{Q}_{i;j,m}^h$  are the state variables at the solution point  $(j, m)$ , with  $j$  the index in  $\xi - \eta$  plane and  $m$  the index in  $\zeta$  direction,  $L_j^{SP}(\xi, \eta)$  is a 2D Lagrange polynomial in the standard triangle and  $L_m^{SP}(\zeta)$  is a 1D Lagrange polynomial. Figure 3 shows the locations of the solution points for  $k = 3$ . The nodal sets on the edge and the triangle are chosen in the same manner as on the tetrahedral element.

The CPR formulation for a standard prism is

$$\begin{aligned} \frac{\partial \tilde{Q}_{i;j,m}^h}{\partial t} + \Pi_{j,m} \left( \nabla^{(\xi)} \cdot \vec{F}^{(\xi)}(\tilde{Q}_i^h) \right) \\ + \frac{1}{|V_{Tri}^{(\xi)}|} \sum_{f \in \partial V_{Tri}} \sum_l \alpha_{j,f,l} [F^{n,(\xi)}(\xi_{f,l}, \eta_{f,l}, \zeta_m)] S_f^{(\xi)} \\ - [F_{com}^\zeta(\xi_j, \eta_j, -1) - F^\zeta(\xi_j, \eta_j, -1)] g_L'(\zeta_m) \\ + [F_{com}^\zeta(\xi_j, \eta_j, 1) - F^\zeta(\xi_j, \eta_j, 1)] g_R'(\zeta_m) = 0, \end{aligned} \quad (4.10)$$

The correction process is done in a decoupled manner. The third term is the correction of the flux components in  $\xi$  and  $\eta$  directions, which is computed on a plane with fixed  $\zeta = \zeta_m$ . This is nothing but the correction used in the 2D CPR method for a triangular element. In Eq. (4.10),  $V_{Tri}$  is the area of the standard triangle,  $S_f$  the length of the edge  $f$  and  $l$  the index for flux points on  $f$ . Note that,  $[F^{n,(\xi)}(\xi_{f,l}, \eta_{f,l}, \zeta_m)]$  corrects only the solution points on the triangle with fixed  $m$  instead of all solution points in the element as shown in Fig. 3(a). The last two terms denote the correction in the  $\zeta$  direction, which is evaluated with the 1D CPR method [16].  $g_L$  and  $g_R$  are the correction functions for the left and right end points of the segment. The flux difference at an end point corrects only the solution points on the segment with fixed  $j$  as shown in Fig. 3(b). 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 is more efficient per DOF than for tetrahedrons. This decoupled procedure also facilitates the implementation employing different degrees of polynomials in  $\xi - \eta$  and  $\zeta$  directions to adapt to flow features. An attempt to employ higher order polynomials in the wall normal direction to resolve the boundary layer with coarser prism cells is shown in a 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 LGL points at each edge for all tetrahedral and prismatic elements.

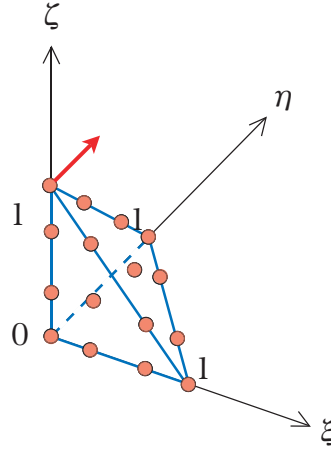


Figure 2: Solution points in the standard tetrahedral cell for degree  $k = 3$  polynomial (only points on the visible faces are shown).

## 5. Numerical Results

### 5.1. Test Cases for the Euler Equations

#### 5.1.1. Accuracy Study with Vortex Evolution Problem

To assess the order of accuracy of the developed method, the propagation of an isentropic vortex is computed with successive grid refinement. This is an idealized problem for the Euler equations in 2D used by Shu [35]. Here we consider simple extension of this problem to the 3D domain  $[0, 10] \times [0, 10] \times [0, 10]$ . 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  $S = p/\rho^\gamma$ :

$$\begin{aligned} (\delta u, \delta v, \delta w) &= \frac{\epsilon}{2\pi} e^{0.5(1-\bar{r}^2)} (-\bar{y}, \bar{x}, 0), \\ \delta T &= -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-\bar{r}^2}, \quad \delta S = 0, \end{aligned} \quad (5.1)$$

where  $\bar{r}^2 = \bar{x}^2 + \bar{y}^2$ ,  $\bar{x} = x - 5$ ,  $\bar{y} = y - 5$ , and the vortex strength  $\epsilon = 5$ . 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  $N \times N \times N$  cells is generated for the cubic domain and 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, the 3rd-order Runge-Kutta explicit scheme [34] is used. Even though the order of accuracy of the temporal scheme is less than the order for the spatial scheme,

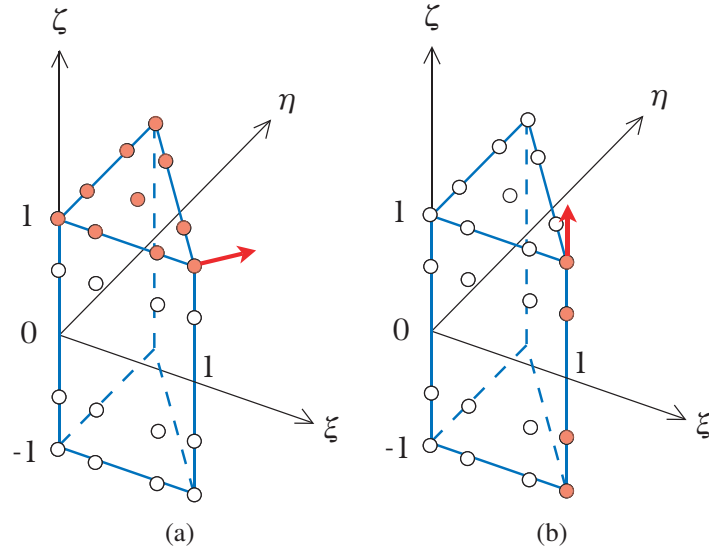


Figure 3: Solution points in the standard prism cell for degree  $k = 3$  polynomial (only points on the visible faces are shown). (a) shows the correction in the  $\xi$  and  $\eta$  directions. (b) shows the correction in the  $\zeta$  direction.

if we use a small enough timestep which does not affect the numerical solution, we can still assess the order of convergence of the spatial operator. In this study, we fixed the timestep to 0.002 for all polynomial degree  $k$  and grids. The timestep corresponds to the  $CFL$  number of 0.138 in the finest prism grid. We confirmed that using a smaller timestep than 0.002 almost did not change the computed solutions. The  $L_1$  and  $L_\infty$  norms of density error at the solution points are presented for tetrahedral grids and prismatic grids in Tables 1 and 2, respectively. The CPR-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.

### 5.1.2. 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 Figs. 4(a) and 5(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, each curved boundary face is represented with a piecewise quadratic polynomial.

The computed density contours obtained with the 2nd- to 4th-order schemes are shown at Figs. 4(b)-(d) and Figs. 5(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

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

Polynomial degree $k$	Grid size	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ 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 CPR-DG for vortex propagation problem (prismatic grids).

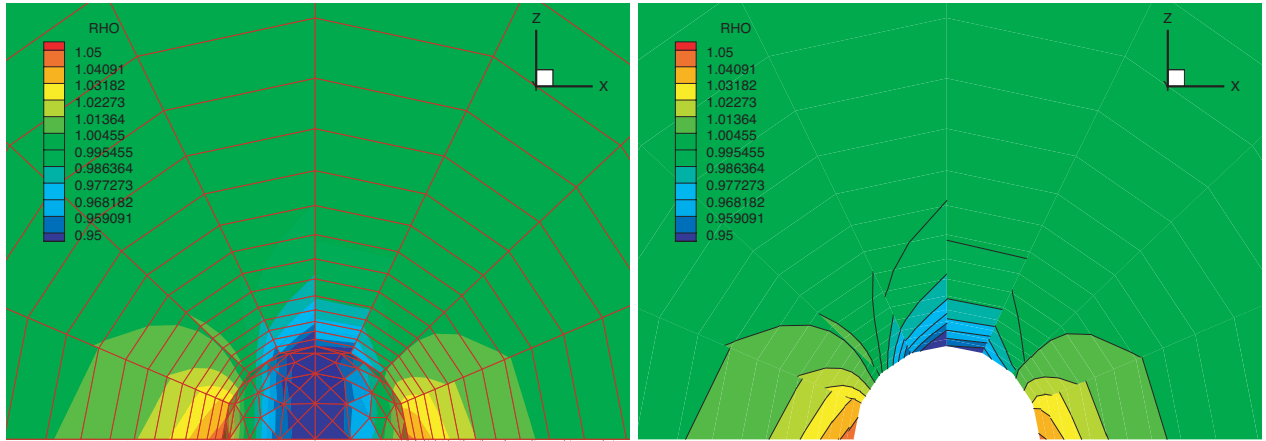
Polynomial degree $k$	Grid size	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order
1	10x10x10x2	7.37e-3	-	1.34e-1	-
	20x20x20x2	2.12e-3	1.80	4.85e-2	1.47
	40x40x40x2	5.19e-4	2.03	1.19e-2	2.03
2	10x10x10x2	2.17e-3	-	4.77e-2	-
	20x20x20x2	2.67e-4	3.02	8.65e-3	2.46
	40x40x40x2	2.88e-5	3.21	1.04e-3	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

implicit scheme [36, 13] was used to obtain steady solutions efficiently, and all the cases converged to machine zero.

## 5.2. Test Cases for the Navier-Stokes Equations

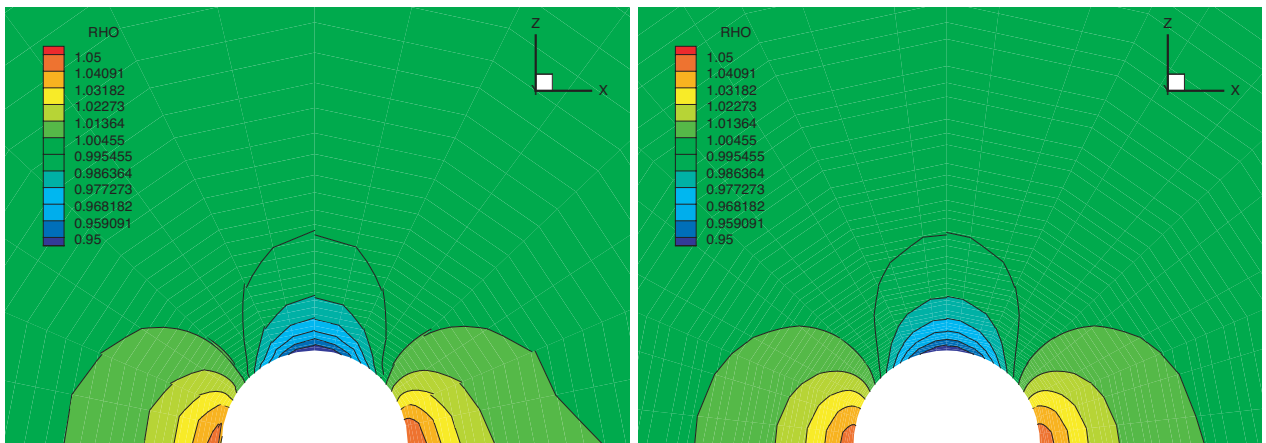
### 5.2.1. Accuracy Study with Couette Flow Problem

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  $[0, 10] \times [0, 10] \times [0, 10]$ . 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  $T_0 = 0.8$



(a) Mesh

(b)  $k = 1$

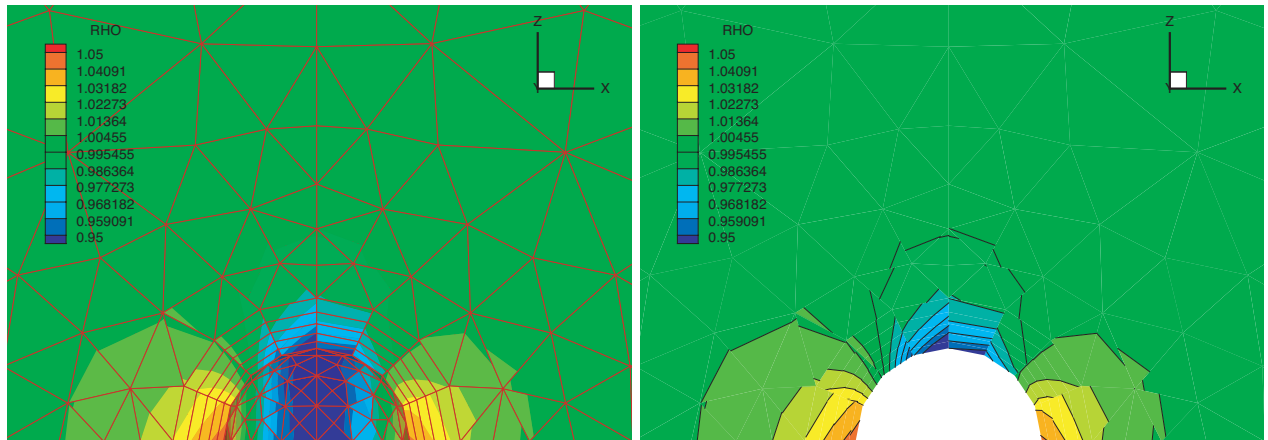


(c)  $k = 2$

(d)  $k = 3$

Figure 4: Prismatic grid and computed density contours for the inviscid flow around a sphere.





(a) Mesh

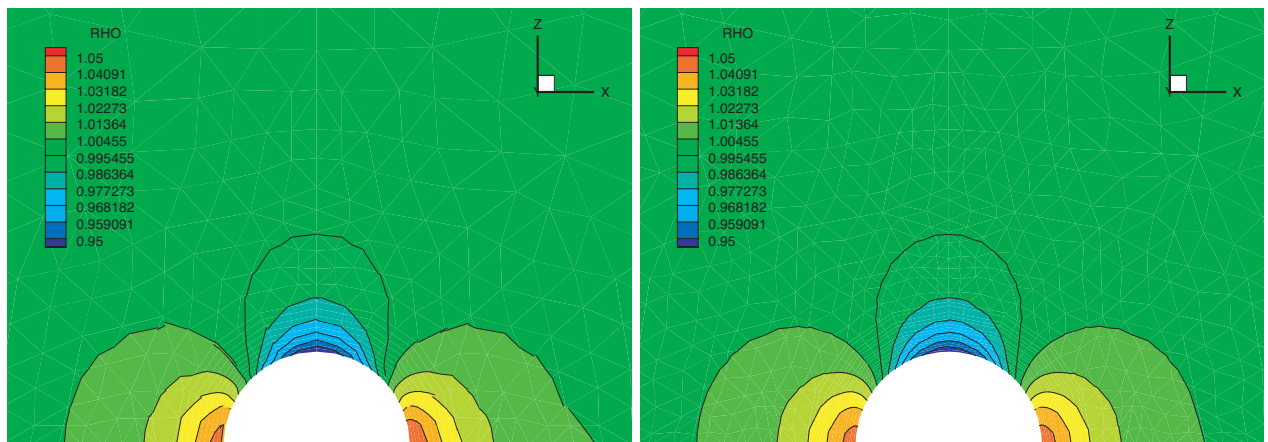
(b)  $k = 1$ (c)  $k = 2$ (d)  $k = 3$ 

Figure 5: Mixed grid (tetrahedrons and prisms) and computed density contours for the inviscid flow around a sphere.

and  $T_1 = 0.85$  respectively. The analytical solution for this case is

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

where  $\gamma$  is specific heat ratio and  $k$  is thermal conductivity. The static pressure is set to  $p_0 = 1/\gamma$  and the viscosity of the fluid is assumed to be  $\mu = 0.01$ . 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  $8$  in the same way as in the vortex propagation case. Each cube is split into 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 optimal order of accuracy is achieved for the 2nd- to 4th-order schemes.

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

Polynomial degree $k$	Grid size	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order
1	2x2x2x2	5.55e-4	-	2.40e-3	-
	4x4x4x2	1.19e-4	2.22	4.00e-4	2.59
	8x8x8x2	3.11e-5	1.94	1.16e-4	1.79
2	2x2x2x2	8.17e-6	-	2.09e-5	-
	4x4x4x2	1.29e-6	2.67	3.37e-6	2.63
	8x8x8x2	1.68e-7	2.94	5.49e-7	2.62
3	2x2x2x2	2.62e-7	-	8.20e-7	-
	4x4x4x2	2.03e-8	3.69	5.70e-8	3.85
	8x8x8x2	1.39e-9	3.87	4.21e-9	3.76

### 5.2.2. Laminar Boundary Layer over a Flat Plate

The laminar boundary layer over a flat plate is then computed using the CPR 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 formula  $\delta = 5L/\sqrt{Re_x}$ . The computational domain is selected to be  $(-2 \leq x \leq 1, 0 \leq y \leq 100\delta, 0 \leq z \leq \delta)$ . 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  $(-2 \leq x \leq 0)$  and the adiabatic wall conditions are imposed on the wall  $(0 \leq x \leq 1)$ . At the outflow boundary at  $x = 1$ , only

static pressure is prescribed. On the side boundaries at  $z = 0$  and  $\delta$ , 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. 6. 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 Fig. 7. The agreement with the Blasius's solution also becomes better with  $k$ -order refinement.

One of the concerning issues when we apply CFD solver to engineering problems is the stiffness arising from using high aspect ratio cells that are clustered near the solid wall to resolve the boundary layer especially in high Reynolds number flows. Reynolds numbers appearing in aerospace flow problems usually become  $\sim 10^6$  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 [26, 8]. 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 8 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 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. 9, the computed  $v$ -velocity profiles are shown. The computed profiles agree well with each other and also with the Blasius's solution. The convergence histories are compared in Fig. 10. For the time integration, we discretise the temporal derivative using the backward Euler algorithm and employed the block preconditioned LU-SGS scheme. We used the same timesteps for the two different meshes. To start with an impulsive condition of the uniform freestream, we set the initial timestep to 0.002 and increased it by multiplying 1.05 after every time step until it reached the prescribed maximum timestep 0.2. The initial timestep corresponds to the  $CFL$  of 3.44 and 4.58 for the grid of 728 cells and the grid of 1736 cells, respectively. 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 are about 16% less.

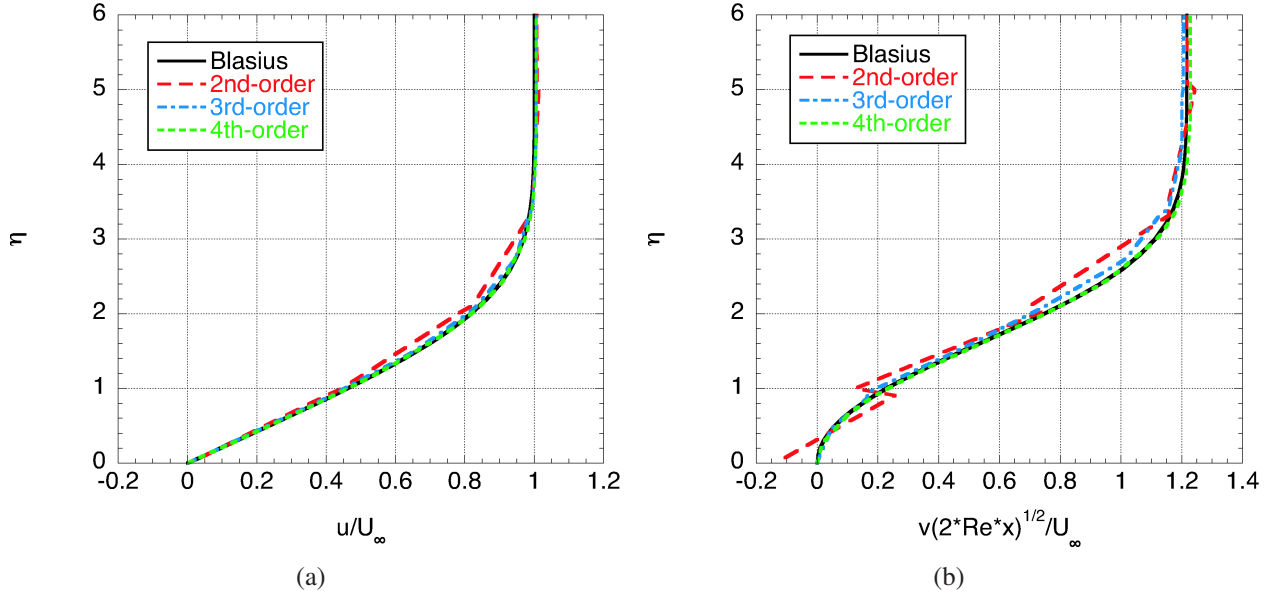


Figure 6: Comparisons of velocity profiles in the boundary layer at  $x = 0.5$ .  $u$ - and  $v$ -profiles in (a) and (b).

### 5.2.3. Steady Subsonic Flow over a Sphere at $Re=118$

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 [37] and numerical results using the SD scheme [36, 44]. 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. 11. The total number of mixed cells is 24334.

The computations were performed using the 3rd- and 4th-order schemes. The computed Mach number contours and streamlines near wake using the 4th-order CPR scheme are shown in Fig. 12 and Fig. 13, respectively. We confirmed that the computed streamlines and the size of separation region agree well with both 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. 14. The skin friction coefficients computed by the 4th-order CPR scheme and the 6th-order SD scheme are right on top of each other. The 3rd-order CPR result also agrees well with other results, though one can see only minor differences between those profiles. The predicted separation angle using the 4th-order CPR 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. 15, the computed drag coefficient by 4th-order CPR is compared to available experimental data. The agreement is also very good.

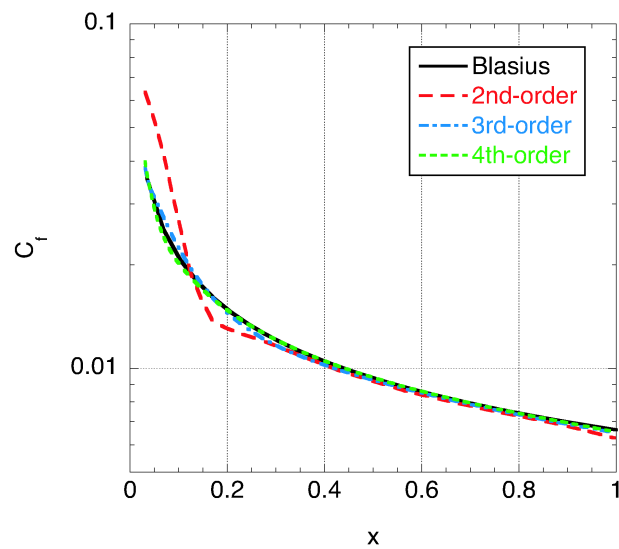


Figure 7: Comparisons of the skin friction coefficient along the plate.

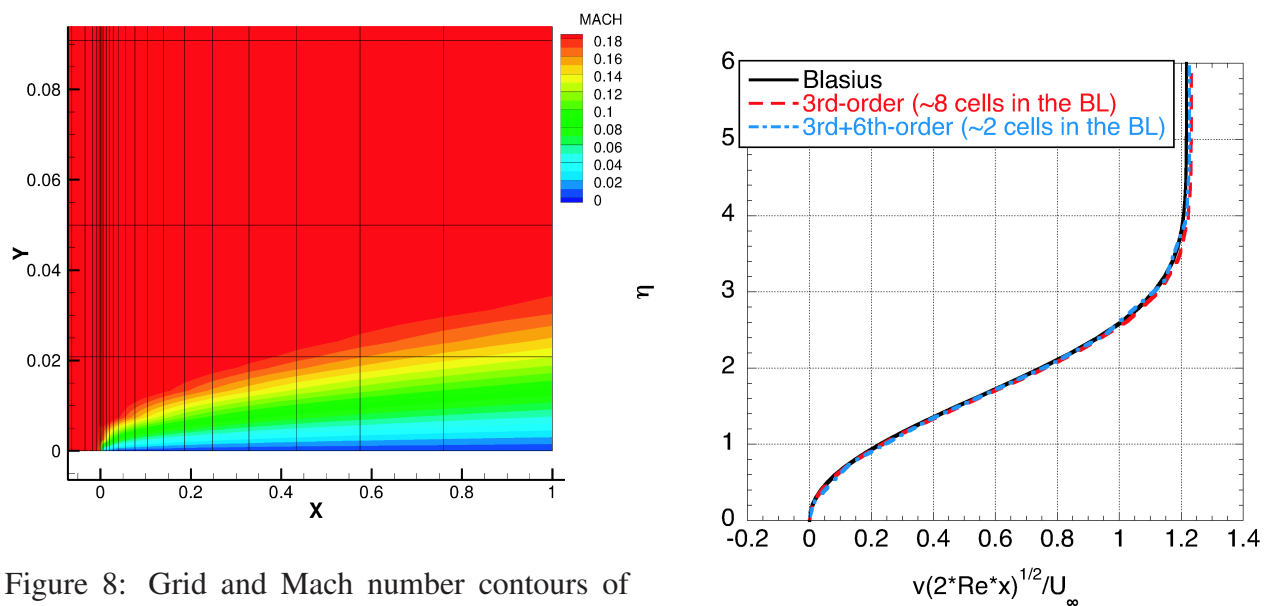


Figure 8: Grid and Mach number contours of a laminar boundary layer on a flat plate using degree  $k = 5$  polynomial in the  $y$ -direction (stretched by a factor of 10 in  $y$  direction).

Figure 9: Comparison of  $v$ -velocity profiles using different degrees of polynomial and grids.

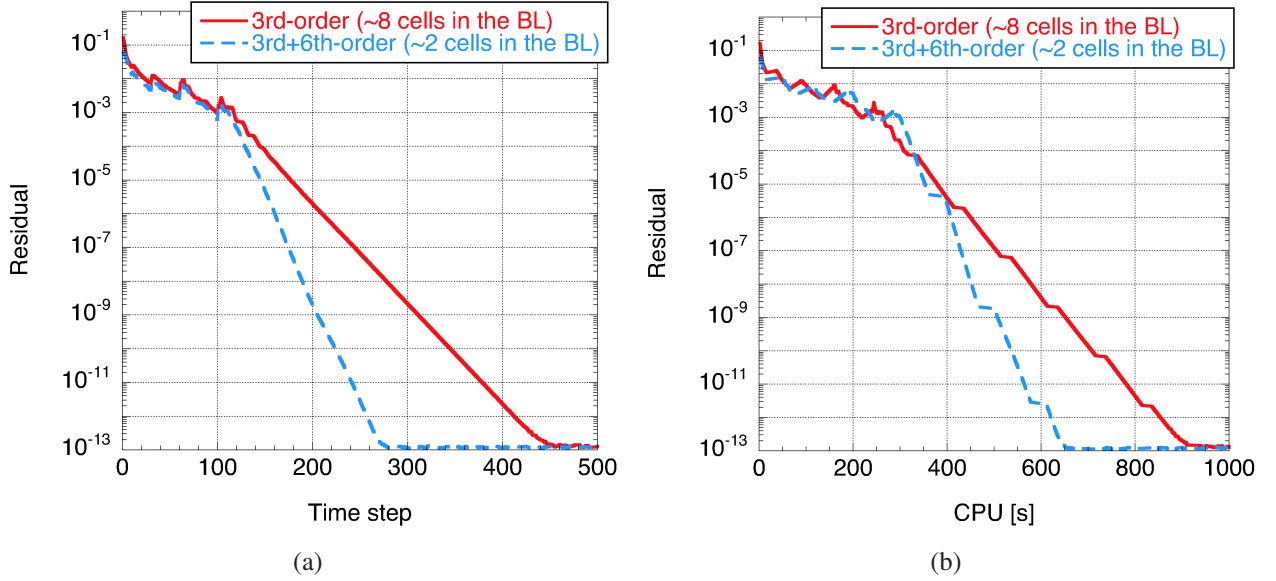


Figure 10: Comparisons of convergence histories using different degrees of polynomial and grids in terms of time step and cpu time in (a) and (b).

#### 5.2.4. Unsteady Subsonic Flow over a Sphere at $Re=300$

We consider an unsteady flow case over the sphere with radius  $r = 1$  at the Reynolds number of 300 based on the diameter of the sphere. The inflow Mach number is assumed to be 0.3 in this case. The computational mesh is shown in Fig. 16. To resolve shedding vortices, the mesh is generated to have finer cells in the wake region. The total number of mixed cells is 54312. Local grid size around the sphere is  $\sim 0.2$  and the size in the wake region is  $\sim 0.8$ . In this case, we employed the 3rd-order TVD Runge-Kutta method for the time integration and computed by the MPI parallelized code using 8 cores of a cluster machine to reduce the wall clock time.

The computed Q isosurface colored by local Mach number using the 4th-order CPR scheme is shown in Fig. 17. The obtained plain symmetric wake vortex structure is comparable to the available experimental and computational results in [10, 19] at least qualitatively. In Fig. 18 we plot the history of the drag coefficient  $C_d$  in terms of non-dimensional time  $t$ . The computed drag coefficient and the oscillating amplitude of drag and the Strouhal number  $St$  are shown in Table 4. For comparison, results from Gassner [10] using the 4th-order DG scheme on tetrahedral grid and from Tomboulides [38] and Johnson and Patel [19] obtained by incompressible simulation, are shown as well. The results computed by the CPR method reasonably agree with those reference values.

## 6. Conclusions

The CPR method is successfully extended to 3D hybrid unstructured meshes using tetrahedral and prismatic elements. The CPR formulation for tetrahedral elements is directly derived in the same

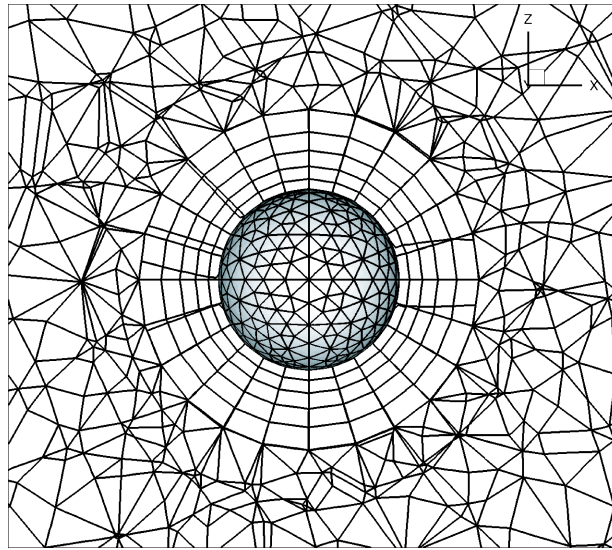


Figure 11: Computational grid around a sphere for the steady viscous flow at  $Re = 118$ .

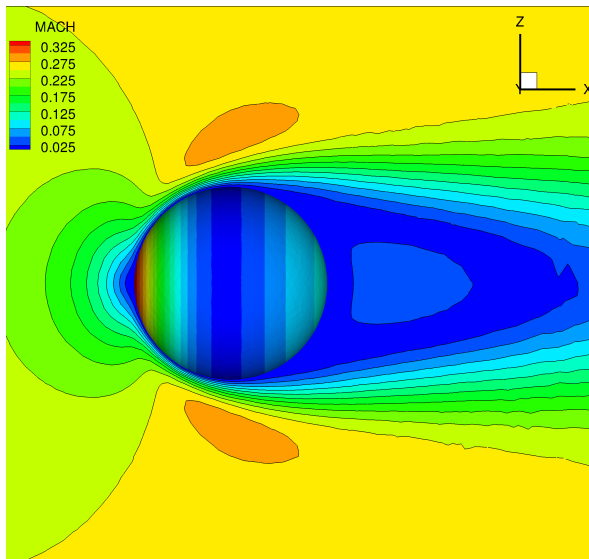


Figure 12: Computed pressure (on the sphere) and Mach number (on  $y = 0$  plane) distributions using the 4th-order CPR scheme.

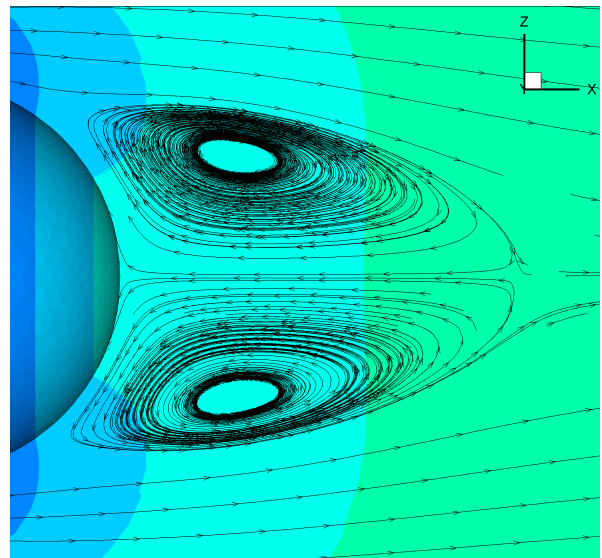


Figure 13: Computed streamlines near the wake region behind the sphere using the 4th-order CPR scheme.



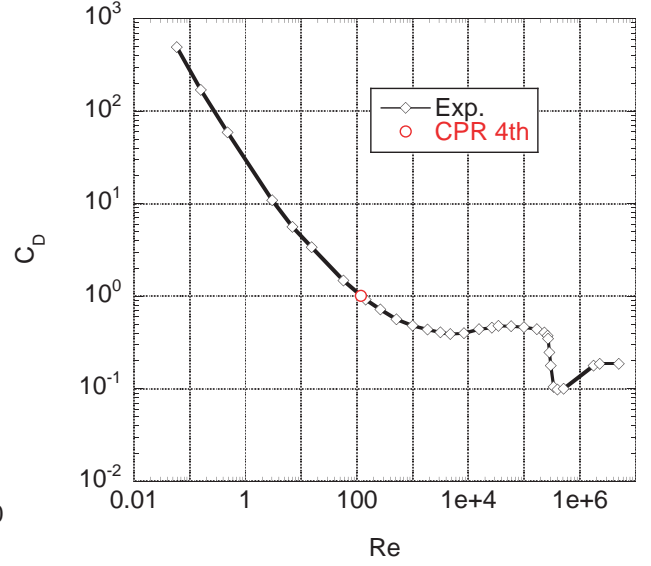
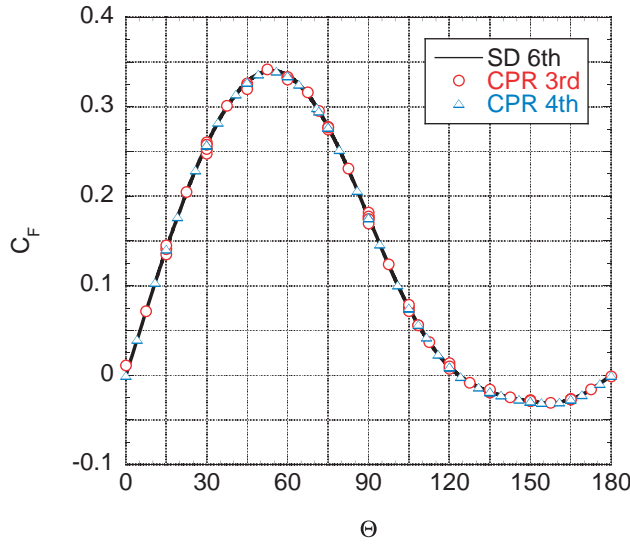


Figure 14: Comparison of computed skin friction coefficients using the 3rd- and 4th order CPR drag coefficient schemes and 6th-order SD scheme in [44].

Figure 15: Comparison between the computed drag coefficient using the 4th-order CPR scheme and experimental data for a sphere.

Table 4: Comparisons of the averaged drag coefficient, the amplitude of drag and the Strouhal number.

Method	$C_d$	$\Delta C_d$	$St$
Present	0.670	0.0032	0.131
Gassner [10]	0.673	0.0031	0.135
Tomboulides [38]	0.671	0.0028	0.136
Johnson & Patel [19]	0.656	0.0035	0.137

manner as for 2D triangular elements and the one for prism is obtained by just a combination of the 1D and 2D schemes. The resulting scheme needs no explicit integrations and no data reconstructions. This numerical efficiency is 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 CPR scheme is verified with grid convergence studies for an inviscid flow and a viscous flow, indicating that the developed scheme is capable of achieving nearly the optimal order of accuracy. Then, several validation cases are computed for solving the 3D Euler equations and the 3D NS equations. The CPR method performs very well to obtain high-order accurate solutions for all cases. Future studies include extension to adopt hexahedral and pyramidal cells for more flexible geometry discretizations and hp-adaptation techniques for realizing practical high accurate CFD simulations.



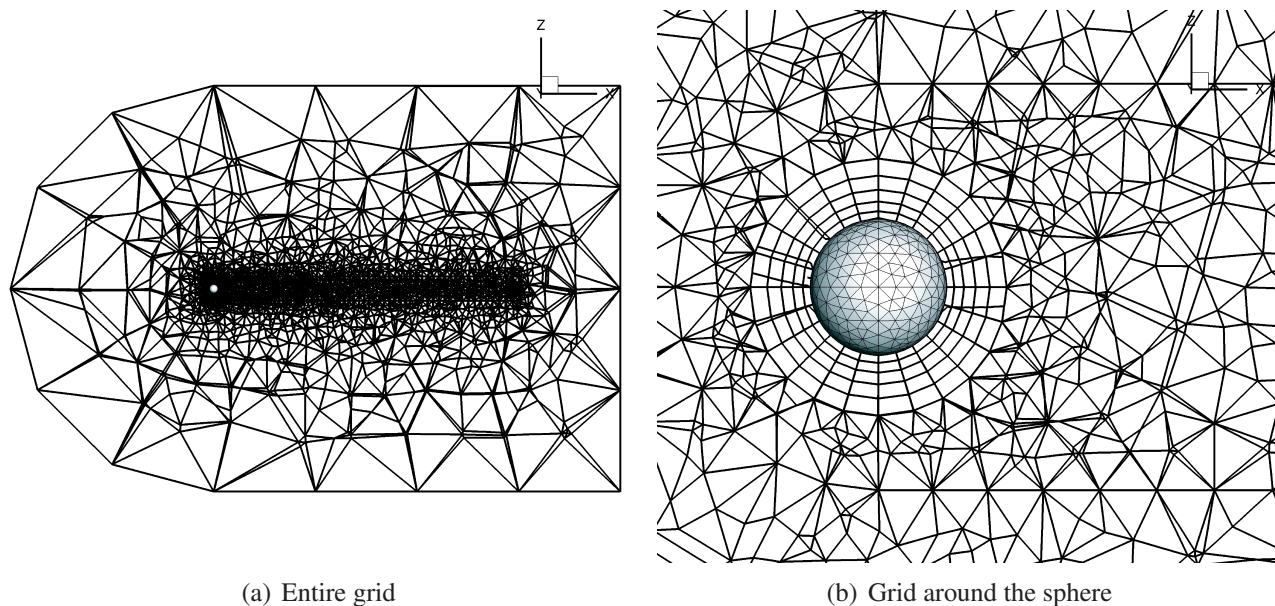


Figure 16: Computational grid around a sphere for the unsteady viscous flow at  $Re = 300$ .

## Acknowledgements

This study has been supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-09-1-0128. 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.

## 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 (1997), 267–279.
- [2] F. Bassi, S. Rebay. *GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations*. In B. Cockburn, G.E. Karniadakis, and C. W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computations and Applications*, volume 11 of *Lecture Note in Computational Science and Engineering*. Springer, 2000.
- [3] Q. Chen and I. Babuska. *Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle*. Comput. Methods Appl. Mech. Eng., 128 (1995), 405–417.

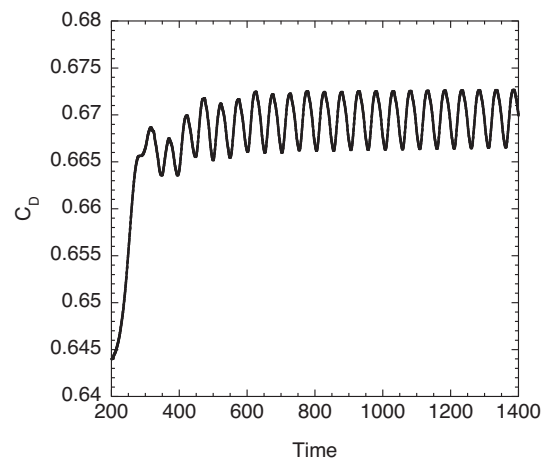
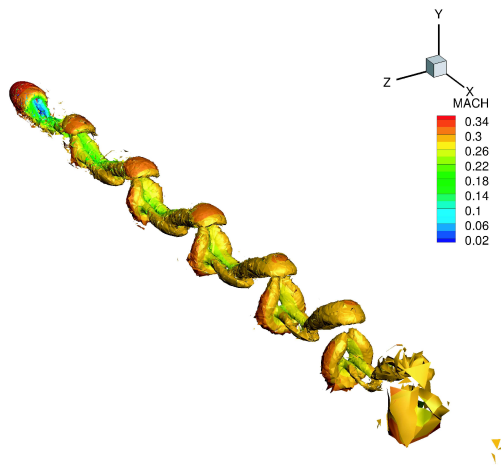


Figure 17: Computed  $Q$  isosurfaces in the wake region of the viscous laminar flow over a sphere at  $Re = 300$ .

Figure 18: Time history of the drag coefficient for unsteady flow over a sphere at  $Re = 300$ .

- [4] B. Cockburn, S. Y. Lin, C. W. Shu. *TVD Runge-Kutta local projection discontinuous Galerkin Finite element method for conservation laws III: one-dimensional systems*. J. Comput. Phys., 84 (1989), 90-113.
- [5] B. Cockburn, C. W. Shu. *TVD Runge-Kutta local projection discontinuous Galerkin Finite element method for conservation laws II: general framework*. Math. Comput., 52 (1989) 411-435.
- [6] B. Cockburn, C. W. Shu. *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*. SIAM J. Numer. Anal., 35 (1998), No. 6, 2440-2463.
- [7] B. Cockburn, C. W. Shu. *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*. J. Comput. Phys., 141 (1998), 199-224.
- [8] K. Fidkowski, T. A. Oliver, J. Lu, D. Darmofal. *p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations*. J. Comput. Phys., 207 (2005), 92-113.
- [9] 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, 2009.
- [10] G. J. Gassner, F. Lorcher, C-D. Munz, and J. S. Hesthaven. *Polymorphic nodal elements and their application in discontinuous Galerkin methods*. J. Comput. Phys., 228 (2009), 1573-1590.
- [11] S. K. Godunov. *A difference scheme for numerical computation of discontinuous solutions of equations of fluid dynamics*. Math. Sbornik, 47 (1959), 271-306, In Russian.

- [12] T. Haga, M. Furudate, K. Sawada. *RANS simulation using high-order spectral volume method on unstructured tetrahedral grids*. AIAA Paper 2009–404, 2009.
- [13] 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*, 6 (2009), No.5, 978-996.
- [14] R. Harris, Z. J. Wang, Y. Liu. *Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation*. *J. Comput. Phys.*, 227 (2008), 1620-1642.
- [15] J. S. Hesthaven. *From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex*. *SIAM J. Numer. Anal.*, 35 (1998), No.2, 655-676.
- [16] H. T. Huynh. *A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods*. AIAA Paper 2007–4079, 2007.
- [17] H. T. Huynh. *A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion*. AIAA Paper 2009–403, 2009.
- [18] 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.
- [19] T. A. Johnson and V. C. Patel. *Flow past a sphere up to a Reynolds number of 300*. *J. Fluid Mech.*, 378 (1999), 19-70.
- [20] D. A. Kopriva and J. H. Koliass. *A conservative staggered-grid Chebyshev multidomain method for compressible flows*. *J. Comput. Phys.*, 125 (1996), 244–261.
- [21] M. S. Liou. *A sequel to AUSM, Part II: AUSM+-up for all speeds*. *J. Comput. Phys.*, 214 (2006), 137-170.
- [22] Y. Liu, M. Vinokur, and 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 2004.
- [23] Y. Liu, M. Vinokur, and Z. J. Wang. *Spectral difference method for unstructured grids I: Basic formulation*. *J. Comput. Phys.*, 216 (2006), 780-801.
- [24] 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.
- [25] H. Luo, J. D. Baum, and R. Lohner. *A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids*. *J. Comput. Phys.*, 227 (2008), 8875-8893.

- [26] D. J. Mavriplis. *Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes*. J. Comput. Phys., 145 (1998), 141-165.
- [27] G. May, A. Jameson. *A spectral difference method for the Euler and Navier-Stokes equations*. AIAA Paper 2006-304, 2006.
- [28] C. R. Nastase, D. J. Mavriplis. *High-order discontinuous Galerkin methods using an hp-multigrid approach*. J. Comput. Phys., 213 (2006), 330-357.
- [29] S. Osher. *Riemann solvers, the entropy condition, and difference approximations*. SIAM J. Numer. Anal., 21 (1984), 217-235.
- [30] W. H. Reed, T. R. Hill. *Triangular mesh methods for the neutron transport equation*. Los Alamos Scientific Laboratory Report LA-UR-73-479, 1973.
- [31] P. L. Roe. *Approximate Riemann solvers, parameter vectors, and difference schemes*. J. Comput. Phys., 43 (1981) 357-372.
- [32] V. V. Rusanov. *Calculation of interaction of non-steady shock waves with obstacles*. J. Comput. Math. Phys., 1 (1961), 267-279.
- [33] S. J. Sherwin, G. E. Karniadaks. *A new triangular and tetrahedral basis for high-order (hp) finite element methods*. Int. J. Num. Meth. Eng., 38 (1995), 3775-3802.
- [34] C. W. Shu. *Total-variation-diminishing time discretizations*. SIAM Journal on Scientific and Statistical Computing, 9 (1988), 1073-1084.
- [35] C. W. Shu. *Essentially non-oscillatory and weighted and non-oscillatory schemes for hyperbolic conservation laws*. In B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor, editors, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Note in Mathematics*. Springer, 1998.
- [36] 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, 2 (2007), 310-333.
- [37] S. Taneda. *Experimental investigations of the wake behind a sphere at low reynolds numbers*. J. Phys. Soc. Japan, 11 (1956), 11041108.
- [38] A. G. Tomboulides, S. A. Orzag. *Numerical investigation of transitional and weak turbulent flow past a sphere*. J. Fluid Mech., 416 (2000), 45-73.
- [39] K. Van den Abeele and C. Lacor. *An accuracy and stability study of the 2D spectral volume method*. J. Comput. Phys., 226 (2007), 1007-1026.
- [40] K. Van den Abeele, C. Lacor, Z. J. Wang. *On the stability and accuracy of the spectral difference method*. J. Sci. Comput., 37 (2008), 162-188.

- [41] B. Van Leer. *Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method*. J. Comput. Phys., 32 (1979), 110-136.
- [42] B. Van Leer, S. Nomura. *Discontinuous Galerkin for diffusion*. AIAA Paper 2005-5108, 2005.
- [43] Z. J. Wang. *Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation*. J. Comput. Phys., 178 (2002), 210-251.
- [44] Z. J. Wang. *High-order methods for the Euler and Navier-Stokes equations on unstructured grids*. Progress in Aerospace Sciences, 43 (2007), 1-41.
- [45] Z. J. Wang, 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.
- [46] 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.
- [47] Z. J. Wang, Y. Liu. *Spectral (finite) volume method for conservation laws on unstructured grids III: One-dimensional systems and partition optimization*. Journal of Scientific Computing, 20 (2004), No.1, 137-157.
- [48] Z. J. Wang, L. Zhang, Y. Liu. *Spectral (finite) volume method for conservation laws on unstructured grids IV: Extension to two-dimensional systems*. J. Comput. Phys., 194 (2004), 716-741.
- [49] T. Warburton. *An explicit construction of interpolation nodes on the simplex*. J. Eng. Math., 56 (2006), 247-262.
- [50] O. C. Zienkiewicz, R. L. Taylor. *The Finite Element Method The Basics*, vol. 1. Butterworth-Heinemann, Oxford, England, 2000.