A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations on 3D Hybrid Grids

Xiaodong Liu^{a,*}, Lijun Xuan^a, Yidong Xia^b, Hong Luo^a

^a Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, 27695

^b Department of Energy Resource Recovery & Sustainability, Idaho National Laboratory, Idaho Falls, ID, 83415

A reconstructed discontinuous Galerkin (rDG(P₁P₂)) method, originally introduced for the compressible Euler equations, is developed for the solution of the compressible Navier-Stokes equations on 3D hybrid grids. In this method, a piecewise quadratic polynomial solution is obtained from the underlying piecewise linear DG solution using a hierarchical Weighted Essentially Non-Oscillatory (WENO) reconstruction. The reconstructed quadratic polynomial solution is then used for the computation of the inviscid fluxes and the viscous fluxes using the second formulation of Bassi and Reay (Bassi-Rebay II). The developed rDG(P₁P₂) method is used to compute a variety of flow problems to assess its accuracy, efficiency, and robustness. The numerical results demonstrate that the rDG(P₁P₂) method is able to achieve the designed third-order of accuracy at a cost slightly higher than its underlying second-order DG method, outperform the third order DG method in terms of both computing costs and storage requirements, and obtain reliable and accurate solutions to the direct numerical simulation (DNS) of compressible turbulent flows.

* Corresponding author.

E-mail address : xliu29@ncsu.edu (Xiaodong Liu).

Keywords: Discontinuous Galerkin method; WENO reconstruction; Unstructured grids;
Compressible Navier-Stokes equation

1. Introduction

The discontinuous Galerkin methods [1-31] (DGM) have recently become popular for the solution of systems of conservation laws. Nowadays, they are widely used in computational fluid dynamics, computational acoustics, and computational magneto-hydrodynamics. The discontinuous Galerkin methods combine two advantageous features commonly associated to finite element and finite volume methods. As in classical finite element methods, accuracy is obtained by means of high-order polynomial approximation within an element rather than by wide stencils as in the case of finite volume methods. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces. In this respect, the methods are therefore similar to finite volume methods. Although DGM have many attractive features [17, 31], they still have a number of weaknesses that have yet to be addressed, before they can become a viable, attractive, competitive, and ultimately superior numerical method over more mature and well-established second order finite volume methods for flow problems of practical interest in a complex configuration environment. In particular, how to effectively control spurious oscillations in the presence of strong discontinuities, and how to reduce the computing costs for the DGM remain the two most challenging in the DGM. Indeed, compared to the finite element methods and finite volume methods, the DGM require solutions of systems of equations with more unknowns for the same grids. Consequently, these methods have been recognized as expensive in terms of both computational costs and storage requirements.

In order to reduce high costs associated with the DGM, Dumbser et al [18-20] have introduced a new family of reconstructed DGM, termed P_nP_m schemes and referred to as $rDG(P_nP_m)$ in this paper, where P_n indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes. The $rDG(P_nP_m)$ schemes are designed to enhance the accuracy of the discontinuous Galerkin method by increasing the order of the underlying polynomial solution. Obviously, the construction of an accurate and efficient reconstruction operator is crucial to the success of the $rDG(P_nP_m)$ schemes. In a reconstructed DG method using a Taylor basis [26-28] developed by Luo et al. for the solution of the compressible Euler and Navier-Stokes equations on arbitrary grids, a higher order polynomial solution is reconstructed by use of a strong interpolation, requiring point values and derivatives to be interpolated on the face-neighboring cells. The resulting over-determined linear system of equations is then solved in the least-squares sense. This reconstruction scheme only involves von Neumann neighborhood, and thus is compact, simple, robust, and flexible. More recently, Zhang et al. [29-30] presented a class of hybrid DG/FV methods for the conservation laws, where the second derivatives in a cell are obtained from the first derivatives in the cell itself and its neighboring cells using a Green-Gauss reconstruction widely used in the finite volume methods. This provides a fast, simple, and robust way to obtain higher-order polynomial solutions. Lately, Luo et al. [31-32] have conducted a comparative study for these three reconstructed discontinuous Galerkin methods $rDG(P_1P_2)$ by solving 2D Euler equations on arbitrary grids. It is found that all three reconstructed discontinuous Galerkin methods can deliver the desired third order of accuracy and significantly improve the accuracy of the underlying second-order DG method, although the least-squares reconstruction method provides the best performance in terms of both accuracy and robustness.

Unfortunately, the attempt to extend our rDG method to solve 3D Euler equations on tetrahedral grids was not successful. Like the second order cell-centered finite volume methods rDG(P₀P₁), the resultant rDG(P₁P₂) method is unstable. Although rDG(P₀P₁) methods are in general stable in 2D and on Cartesian or structured grids in 3D, they suffer from the so-called *linear instability* on unstructured tetrahedral grids, when the reconstruction stencils only involve von Neumann neighborhood, i.e. adjacent face-neighboring cells [33]. The rDG(P₁P₂) method exhibits the same linear instability, which can be overcome by using extended stencils. However, this is achieved at the expense of sacrificing the compactness of the underlying DG methods. Furthermore, these linear reconstruction-based DG methods will suffer from non-physical oscillations in the vicinity of strong discontinuities for the compressible Euler equations. Alternatively, ENO, WENO, and HWENO can be used to reconstruct a higher-order polynomial solution, thereby not only enhancing the order of accuracy of the underlying DG method but also achieving both linear and non-linear stability. This type of hybrid HWENO+DG schemes has been developed on 1D and 2D structured grids by Balsara et al. [34], where the HWENO reconstruction is relatively simple and straightforward. In our latest work, a rDG(P₁P₂) method based on a Hierarchical WENO reconstruction has been developed for the solution of the compressible Euler equations on unstructured hybrid grids [35-36]. This rDG(P₁P₂) method is designed not only to reduce the high computing costs of the DGM, but also to avoid spurious oscillations in the vicinity of strong discontinuities, thus effectively addressing the two shortcomings of the DGM.

The objective of the effort discussed in this paper is threefold: (a) extend this rDG(P₁P₂) method for the solution of the three-dimensional compressible Navier-Stokes equations on hybrid grids, where the viscous and heat fluxes are discretized using Bassi-Rebay's second

method (BRII); (b) numerically assess if the developed rDG(P₁P₂) method can achieve a formal third order of convergence for the compressible Navier-Stokes equations, and (c) demonstrate that the rDG(P₁P₂) method can be effectively and accurately used for the simulation of turbulent flows. The developed rDG(P₁P₂) method is used to compute a variety of flow problems on 3D hybrid grids to assess its accuracy, robustness, and performance. The numerical results obtained indicate that this rDG(P₁P₂) method is able to achieve the designed third-order of accuracy: one order of accuracy higher than the underlying DG(P₁) method without significant increase in computing costs and memory requirements and obtain reliable and accurate solutions to the direct numerical simulation (DNS) of compressible turbulent flows. The remainder of this paper is organized as follows. The governing equations are described in Section 2. The developed reconstructed discontinuous Galerkin method is presented in Section 3. Extensive numerical experiments are reported in Section 4. Concluding remarks are given in Section 5.

2. Governing Equations

The Navier-Stokes equations governing unsteady compressible viscous flows can be expressed as

$$\frac{\partial \mathbf{U}(x,t)}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U}(x,t))}{\partial x_k} = \frac{\partial \mathbf{G}_k(\mathbf{U}(x,t))}{\partial x_k} \quad (1)$$

where the summation convention has been used. The conservative variable vector $\mathbf{U}$, advective (inviscid) flux vector $\mathbf{F}$, and viscous flux vector $\mathbf{G}$ are defined by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix}, \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ u_j(\rho e + p) \end{pmatrix}, \quad \mathbf{G}_j = \begin{pmatrix} 0 \\ \sigma_{ij} \\ u_i \sigma_{lj} + q_j \end{pmatrix} \quad (2)$$

Here ρ , p , and e denote the density, pressure, and specific total energy of the fluid, respectively, and u_i is the velocity of the flow in the coordinate direction x_i . The pressure can be computed from the equation of state

$$p = (\gamma - 1)\rho(e - \frac{1}{2}u_j u_j) \quad (3)$$

which is valid for perfect gas, where γ is the ratio of the specific heats. The components of the viscous stress tensor σ_{ij} and the heat flux vector are given by

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, \quad q_j = \frac{1}{\gamma-1} \frac{\mu}{Pr} \frac{\partial T}{\partial x_j} \quad (4)$$

In the above equations, T is the temperature of the fluid, Pr the laminar Prandtl number, which is taken as 0.7 for air. μ represents the molecular viscosity, which can be determined through Sutherland's law

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S}{T + S} \quad (5)$$

μ_0 denotes the viscosity at the reference temperature T_0 , and S is a constant which are assumed to be equal to 110 K. The temperature of the fluid T is determined by

$$T = \gamma \frac{p}{\rho} \quad (6)$$

Neglecting viscous effects, the left-hand side of Eq. (1) represents the Euler equations governing unsteady compressible inviscid flows.

3. Reconstructed Discontinuous Galerkin Method

3.1 Discontinuous Galerkin formulation

The governing equation (1) is discretized using a discontinuous Galerkin finite element formulation. We first introduce some notations. We assume that the domain Ω is subdivided into

a collection of non-overlapping elements Ω_e , which can be tetrahedra, prisms, pyramids, and hexahedra or their combinations. We introduce the following broken Sobolev space V_h^p

$$V_h^p = \{v_h \in [L_2(\Omega)]^m: v_h|_{\Omega_e} \in [V_p^m] \quad \forall \Omega_e \in \Omega\} \quad (7)$$

which consists of discontinuous vector-values polynomial functions of degree P_n , and where m is the dimension of the unknown vector and

$$V_p^m = \text{span}\{\prod x_i^{\alpha_i} : 0 \leq \alpha_i \leq p, 0 \leq i \leq d\} \quad (8)$$

where α denotes a multi-index and d is the dimension of space. To formulate the discontinuous Galerkin method, we introduce the following weak formulation, which is obtained by multiplying the above conservation law Eq. (1) by a test function $\mathbf{W}_h$, integrating over an element Ω_e , and then performing an integration by parts. Find $\mathbf{U}_h \in V_h^p$ such as

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h \mathbf{W}_h d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) n_k \mathbf{W}_h d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial \mathbf{W}_h}{\partial x_k} d\Omega = \int_{\Gamma_e} \mathbf{G}_k(\mathbf{U}_h, \nabla \mathbf{U}_h) n_k \mathbf{W}_h d\Gamma - \\ \int_{\Omega_e} \mathbf{G}_k(\mathbf{U}_h, \nabla \mathbf{U}_h) \frac{\partial \mathbf{W}_h}{\partial x_k} d\Omega, \quad \forall \mathbf{W}_h \in V_h^p \end{aligned} \quad (9)$$

where $\mathbf{U}_h$ and $\mathbf{W}_h$ are represented by piecewise-polynomial functions of degrees P_n , which are discontinuous between the cell interfaces and $\mathbf{n}_k$ denotes the unit outward normal vector to Γ_e : the boundary of Ω_e . Assume that B_i is the basis of polynomial function of degrees P_n , this is then equivalent to the following system of N equations,

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h B_i d\Omega + \int_{\Gamma_e} \mathbf{H}_k^{inv}(\mathbf{U}_h^L, \mathbf{U}_h^R, n_k) B_i d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial B_i}{\partial x_k} d\Omega = \int_{\Gamma_e} \mathbf{H}_k^{vis}(\mathbf{U}_h^L, \nabla \mathbf{U}_h^L + \\ \beta \mathbf{r} e L, \mathbf{U}_h R, \nabla \mathbf{U}_h R + \beta \mathbf{r} e R, n_k) B_i d\Gamma - \int_{\Omega_e} \mathbf{G}_k(\mathbf{U}_h, \nabla \mathbf{U}_h + R) \frac{\partial B_i}{\partial x_k} d\Omega, \quad 1 \leq i \leq N \end{aligned} \quad (10)$$

where N is the dimension of the polynomial space. The inviscid and viscous flux function appearing in Eq. (9) is replaced by numerical flux function $\mathbf{H}_k^{inv}$ and $\mathbf{H}_k^{vis}$ respectively, where $\mathbf{U}_h^L$

and $\mathbf{U}_h^R$ are the solution polynomial at the left and right states of the cell interface. In the viscous flux, r_e in the fifth term is the local lifting operator, β is called penalty parameter, and R is the global lifting operator. We need to mention that, through our experiment, there exists lower bounds of the parameter β to ensure the stability of the method, whereas lower β could give better convergences without spoiling the accuracy of the numerical solution. This scheme is called discontinuous Galerkin method of degree P_n , or in short notation DG(P_n) method. Note that discontinuous Galerkin formulations are very similar to finite volume schemes, especially in their use of numerical fluxes. Indeed, the classical first-order cell-centered finite volume scheme exactly corresponds to the DG(P_0) method, i.e., to the discontinuous Galerkin method using a piecewise constant polynomial. Consequently, the DG(P_k) methods with $k>0$ can be regarded as a natural generalization of finite volume methods to higher order methods. By simply increasing the degree P of the polynomials, the DG methods of corresponding higher order can be obtained.

The domain and boundary integrals in Eq. (9) are calculated using Gauss quadrature formulas. The number of quadrature points used is chosen to integrate exactly polynomials of order of $2P_n$ and $2P_n+1$ in the reference element for the domain and surface integrals, respectively.

In the traditional DGM, numerical polynomial solutions $\mathbf{U}_h$ in each element are expressed using either standard Lagrange finite element or hierarchical node-based basis as following

$$\mathbf{U}_h = \sum_{i=1}^N \mathbf{U}_i B_i(x_k) \quad (11)$$

where B_i are the finite element basis functions. As a result, the unknowns to be solved are the variables at the nodes $\mathbf{U}_i$. However, numerical polynomial solutions $\mathbf{U}_h$ can be expressed in other forms as well. In the present work, the numerical quadratic polynomial solutions are represented using a Taylor series expansion at the center of the cell, and further expressed as cell-averaged values and their derivatives at the center of the cell:

$$\mathbf{U}_h = \tilde{\mathbf{U}} B_1 + \mathbf{U}_x B_2 + \mathbf{U}_y B_3 + \mathbf{U}_z B_4 + \mathbf{U}_{xx} B_5 + \mathbf{U}_{yy} B_6 + \mathbf{U}_{zz} B_7 + \mathbf{U}_{xy} B_8 + \mathbf{U}_{xz} B_9 + \mathbf{U}_{yz} B_{10} \quad (12)$$

Here the basis functions are defined as follows,

$$\begin{aligned} B_1 &= 1, B_2 = \frac{(x-x_c)}{\Delta x}, B_3 = \frac{(y-y_c)}{\Delta y}, B_4 = \frac{(z-z_c)}{\Delta z}, \\ B_5 &= \frac{B_2^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_2^2}{2} d\Omega, B_6 = \frac{B_3^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_3^2}{2} d\Omega, B_7 = \frac{B_4^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_4^2}{2} d\Omega, \\ B_8 &= B_2 B_3 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2 B_3 d\Omega, B_9 = B_2 B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2 B_4 d\Omega, B_{10} = B_3 B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_3 B_4 d\Omega. \end{aligned} \quad (13)$$

About the detailed process to derive the above basis function, please refer to Appendix A. In order to improve the conditioning of the system matrix, the basis functions are normalized. Here $\Delta x = 0.5(x_{\max} - x_{\min})$, $\Delta y = 0.5(y_{\max} - y_{\min})$, and $\Delta z = 0.5(z_{\max} - z_{\min})$, and $x_{\max}$, $x_{\min}$, $y_{\max}$, $y_{\min}$, $z_{\max}$, and $z_{\min}$ are the maximum and minimum coordinates in the cell Ω_e in x-, y-, and z-directions, respectively.

And, $\tilde{\mathbf{U}}$ is the mean value of $\mathbf{U}$ in this cell. The unknowns to be solved for in this formulation are the cell-averaged variables and their modified derivatives at the center of the cells.

$$\begin{aligned} \mathbf{U}_x &= \frac{\partial \mathbf{U}}{\partial x} |_c \Delta x, & \mathbf{U}_y &= \frac{\partial \mathbf{U}}{\partial y} |_c \Delta y, & \mathbf{U}_z &= \frac{\partial \mathbf{U}}{\partial z} |_c \Delta z, \\ \mathbf{U}_{xx} &= \frac{\partial^2 \mathbf{U}}{\partial x^2} |_c \Delta x^2, & \mathbf{U}_{yy} &= \frac{\partial^2 \mathbf{U}}{\partial y^2} |_c \Delta y^2, & \mathbf{U}_{zz} &= \frac{\partial^2 \mathbf{U}}{\partial z^2} |_c \Delta z^2, \\ \mathbf{U}_{xy} &= \frac{\partial^2 \mathbf{U}}{\partial x \partial y} |_c \Delta x \Delta y, & \mathbf{U}_{xz} &= \frac{\partial^2 \mathbf{U}}{\partial x \partial z} |_c \Delta x \Delta z, & \mathbf{U}_{yz} &= \frac{\partial^2 \mathbf{U}}{\partial y \partial z} |_c \Delta y \Delta z. \end{aligned} \quad (14)$$

The discontinuous Galerkin formulation then leads to the following 10 equations

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_e} \tilde{\mathbf{U}} d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) n_k d\Gamma &= \int_{\Gamma_e} \mathbf{G}_k(\mathbf{U}_h) n_k d\Gamma \\ M_{9 \times 9} \frac{d}{dt} (\mathbf{U}_x \quad \mathbf{U}_y \quad \mathbf{U}_z \quad \mathbf{U}_{xx} \quad \mathbf{U}_{yy} \quad \mathbf{U}_{zz} \quad \mathbf{U}_{xz} \quad \mathbf{U}_{xy} \quad \mathbf{U}_{yz})^T + R_{9 \times 1} &= 0 \end{aligned} \quad (15)$$

Note that in this formulation, equations for the cell-averaged variables are decoupled from equations for their derivatives due to the judicious choice of the basis functions and the fact that

$$\int_{\Omega_e} B_1 B_i d\Omega = 0 \quad 2 \leq i \leq 10 \quad (16)$$

This formulation belongs to the so-called modal discontinuous Galerkin method and has a number of attractive, distinct, and useful features. First, cell-averaged variables and their derivatives are handily available in this formulation. This makes the implementation of both in-cell and inter-cell reconstruction schemes straightforward and simple [26-28, 30-31]. Secondly, the Taylor basis is hierarchic. This greatly facilitates implementation of p -multigrid methods [16, 17] and p -refinement. Thirdly, the same basis functions are used for any shapes of elements: tetrahedron, pyramid, prism, and hexahedron. This makes the implementation of DGM on arbitrary meshes straightforward.

3.2 Reconstructed Discontinuous Galerkin Methods

In comparison with reconstructed FV methods, the DGM have a significant drawback in that they require more degrees of freedom, additional domain integration, and more Gauss quadrature points for the boundary integration, and therefore more computational costs and storage requirements. On the one hand, the reconstruction methods that FV methods use to achieve higher-order accuracy are relatively inexpensive but less accurate and robust. On the other hand, the DGM that can be viewed as a different way to extend a FV method to higher orders are accurate and robust but costly. It is only natural and tempting to combine the efficiency of the reconstruction methods and the accuracy of the DG methods. This idea was originally introduced by Dumbser et al in the frame of $P_n P_m$ scheme [18-20], termed $rDG(P_n P_m)$ in this paper, where P_n indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the

fluxes and source terms. The beauty of $\text{rDG}(\text{P}_n\text{P}_m)$ schemes is that they provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of $\text{rDG}(\text{P}_n\text{P}_m)$ schemes, and thus allow for a direct efficiency comparison. When $n=0$, i.e. a piecewise constant polynomial is used to represent a numerical solution, $\text{rDG}(\text{P}_0\text{P}_m)$ is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m=n$, the reconstruction reduces to the identity operator, and $\text{rDG}(\text{P}_n\text{P}_m)$ scheme yields a standard DG method. Clearly, an accurate and efficient reconstruction is the key ingredient in extending the underlying DG method to higher order accuracy. Our discussion in this work is mainly focused on a third order $\text{rDG}(\text{P}_1\text{P}_2)$ method, as the benefits of higher-order ($>3\text{rd}$) methods diminish dramatically for aerodynamic applications. Nevertheless, its extension to higher order DG methods is straightforward, as demonstrated by Dumbser et al in Reference 18. The $\text{rDG}(\text{P}_1\text{P}_2)$ method is based on a Hermite WENO reconstruction and designed not only to reduce the high computing costs of the DGM, but also to ensure the linear stability of the rDG method on tetrahedral grids.

3.2.1 WENO reconstruction at P_2

The reconstruction of the second derivatives consists of two steps: a quadratic polynomial solution (P_2) is first reconstructed using a least-squares method from the underlying linear polynomial (P_1) discontinuous Galerkin solution, and the final quadratic polynomial solution is then obtained using a WENO reconstruction, which is necessary to ensure the linear stability of the rDG method [36, 37] on tetrahedral grids.

3.2.1.1 Least-squares reconstruction

In the case of $\text{DG}(\text{P}_1)$ method, a linear polynomial solution $\mathbf{U}_{hi}$ in any cell i is expressed as

$$\mathbf{U}_{hi} = \tilde{\mathbf{U}} + \mathbf{U}_{xi}B_2 + \mathbf{U}_{yi}B_3 + \mathbf{U}_{zi}B_4 \quad (17)$$

A quadratic polynomial solution $\mathbf{U}_{hi}^R$ can be reconstructed using the underlying linear polynomial DG solution in the neighboring cells as follows:

$$\begin{aligned} \mathbf{U}_{hi}^R = & \tilde{\mathbf{U}}_i^R B_1 + \mathbf{U}_{xi}^R B_2 + \mathbf{U}_{yi}^R B_3 + \mathbf{U}_{zi}^R B_4 + \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 + \mathbf{U}_{xyi}^R B_8 + \\ & \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10} \end{aligned} \quad (18)$$

In order to maintain the compactness of the DG methods, the reconstruction is required to involve only von Neumann neighborhood, i.e., the adjacent cells that share a face with the cell i under consideration. There are 10 degrees of freedom, and therefore 10 unknowns must be determined. The first four unknowns can be trivially obtained, by requiring the consistency of the rDG with the underlying DG: 1) The reconstruction scheme must be conservative, and 2) The values of the reconstructed first derivatives are equal to the ones of the first derivatives of the underlying DG solution at the centroid i . Due to the judicious choice of Taylor basis in our DG formulation, these four degrees of freedom simply coincide with the ones from the underlying DG solution, i.e.

$$\tilde{\mathbf{U}}_i^R = \tilde{\mathbf{U}}_i, \quad \mathbf{U}_{xi}^R = \mathbf{U}_{xi}, \quad \mathbf{U}_{yi}^R = \mathbf{U}_{yi}, \quad \mathbf{U}_{zi}^R = \mathbf{U}_{zi} \quad (19)$$

As a result, only six second derivatives need to be determined. This can be accomplished by requiring that the mean values and first derivatives of the reconstructed solution are consistent with these of the underlying DG solution for all the adjacent face neighboring cells. Considering an adjacent neighboring cell j , one obtains

$$\begin{aligned} \tilde{\mathbf{U}}_j = & \frac{1}{\Omega_j} \int_{\Omega_j} \tilde{\mathbf{U}}_i B_1 + \mathbf{U}_{xi} B_2 + \mathbf{U}_{yi} B_3 + \mathbf{U}_{zi} B_4 + \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 + \mathbf{U}_{xyi}^R B_8 + \\ & \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10} d\Omega_j \end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{U}}{\partial x} |_j &= \mathbf{U}_{xi} \frac{1}{\Delta x_i} + \mathbf{U}_{xxi}^R \frac{B_2}{\Delta x_i} + \mathbf{U}_{xyi}^R \frac{B_3}{\Delta x_i} + \mathbf{U}_{xzi}^R \frac{B_4}{\Delta x_i} \\
\frac{\partial \mathbf{U}}{\partial y} |_j &= \mathbf{U}_{yi} \frac{1}{\Delta y_i} + \mathbf{U}_{xyi}^R \frac{B_2}{\Delta y_i} + \mathbf{U}_{yyi}^R \frac{B_3}{\Delta y_i} + \mathbf{U}_{yzi}^R \frac{B_4}{\Delta y_i} \\
\frac{\partial \mathbf{U}}{\partial z} |_j &= \mathbf{U}_{zi} \frac{1}{\Delta z_i} + \mathbf{U}_{xzi}^R \frac{B_2}{\Delta z_i} + \mathbf{U}_{yzi}^R \frac{B_3}{\Delta z_i} + \mathbf{U}_{zzi}^R \frac{B_4}{\Delta z_i}
\end{aligned} \tag{20}$$

where the basis functions B are evaluated at the center of cell j , i.e. $B = B(x_j, y_j, z_j)$. This can be written in a matrix form as follows:

$$\mathbf{A} \begin{pmatrix} \mathbf{U}_{xxi}^R \\ \mathbf{U}_{yyi}^R \\ \mathbf{U}_{zzi}^R \\ \mathbf{U}_{xyi}^R \\ \mathbf{U}_{xzi}^R \\ \mathbf{U}_{yzi}^R \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{U}}_j - (\tilde{\mathbf{U}}_i B_1^j + \mathbf{U}_{xi} B_2^j + \mathbf{U}_{yi} B_3^j + \mathbf{U}_{zi} B_4^j) \\ \frac{\Delta x_i}{\Delta x_j} \mathbf{U}_{xj} - \mathbf{U}_{xi} \\ \frac{\Delta y_i}{\Delta y_j} \mathbf{U}_{yj} - \mathbf{U}_{yi} \\ \frac{\Delta z_i}{\Delta z_j} \mathbf{U}_{zj} - \mathbf{U}_{zi} \end{pmatrix} \tag{21}$$

where

$\mathbf{A}$

$$= \begin{pmatrix} \frac{1}{\Omega_j} \int_{\Omega_j} B_5 d\Omega_j & \frac{1}{\Omega_j} \int_{\Omega_j} B_6 d\Omega_j & \frac{1}{\Omega_j} \int_{\Omega_j} B_7 d\Omega_j & \frac{1}{\Omega_j} \int_{\Omega_j} B_8 d\Omega_j & \frac{1}{\Omega_j} \int_{\Omega_j} B_9 d\Omega_j & \frac{1}{\Omega_j} \int_{\Omega_j} B_{10} d\Omega_j \\ B_2^j & 0 & 0 & B_3^j & B_4^j & 0 \\ 0 & B_3^j & 0 & B_2^j & 0 & B_4^j \\ 0 & 0 & B_4^j & 0 & B_2^j & B_3^j \end{pmatrix}$$

. Similar equations can be written for all cells connected to the cell i with a common face, which leads to a non-square matrix. The number of face-neighboring cells for a tetrahedral is four. Consequently, the size of the resulting non-square matrix is 16x6. In the present work, this over-determined linear system of 16 equations for 6 unknowns is solved in the least-squares sense using both normal equation approach and the QR decomposition to obtain the second derivatives

of the reconstructed quadratic polynomial solution. One can easily verify that this least-squares reconstruction satisfies the so-called 2-exactness, i.e., it can reconstruct a quadratic polynomial function exactly. The developed rDG method could be used to deal with hybrid grids, which has the ability to computer 1D, 2D and 3D test cases in the 3D framework. Herein, we need to point out that, when involving the boundary cell, only the first formula of Eq. (21) has been considered. For 1D test case implemented in 3D code, every entry in the last row of matrix A is equal to zero. This could lead to failure with normal equation approach and the QR decomposition method. Therefore, for 1D test cases, the number of cells in y and z direction is set as 2 and 1. For 2D test cases, the number of cells in z direction is set as 1.

The BRII method is then used to discretize the viscous and heat fluxes based on this reconstructed quadratic polynomial solution. Both local and global lift operators, which are also quadratic polynomial solutions, can be easily computed using the reconstructed quadratic polynomial solution.

3.2.1.2 WENO reconstruction

When extended to solve the 3D compressible Euler equations on tetrahedral grids, this rDG method suffers from the so-called linear instability, as mentioned in the introduction. ENO/WENO can be used to reconstruct a higher-order polynomial solution, which can not only enhance the order of accuracy of the underlying DG method but also achieve both linear and non-linear stability. Specifically, the WENO scheme is adopted in this work, where an entire quadratic polynomial solution on cell i is obtained using a nonlinear WENO reconstruction as a convex combination of the least-squares reconstructed second derivatives at the cell itself and its four face-neighboring cells,

$$\frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n} |_i^{\text{weno}} = \sum_{K=1}^5 w_K \frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n} |_K \quad (22)$$

where the normalized nonlinear weights w_k are computed as

$$w_k = \widetilde{w}_k / \sum_{K=1}^5 \widetilde{w}_K \quad (23)$$

The non-normalized nonlinear weights $\widetilde{w}_i$ are functions of the linear weights λ_i and the so-called oscillation indicator o_i

$$\widetilde{w}_i = \frac{\lambda_i}{(\varepsilon + o_i)^\gamma} \quad (24)$$

where ε is a small positive number used to avoid division by zero, and γ an integer parameter to control how fast the non-linear weights decay for non-smooth stencils. The oscillation indicator for the reconstructed second order polynomials is simply defined as

$$o_k = [(\frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n} |_k)^2]^{1/2} \quad (25)$$

The linear weights λ_i can be chosen to balance the accuracy and the non-oscillatory property of the rDG method. Note that the least-squares reconstructed polynomial at the cell itself serves as the central stencil and the least-squares reconstructed polynomials on its four face-neighboring cells act as biased stencils in this WENO reconstruction. This reconstructed quadratic polynomial solution is then used to compute the domain and boundary integrals of the underlying DG(P₁) method in Eq. (10). As demonstrated in Reference 36, the resulting rDG(P₁P₂) method is able to achieve the designed third order of accuracy, maintain the linear stability, and significantly improve the accuracy of the underlying second-order DG method without significant increase in computing costs and storage requirements for the compressible Euler equations. Note that this rDG method is not compact anymore, as neighbor's neighbors are used in the solution update. However, the stencil used in the reconstruction is compact, involving only

von Neumann neighbors. Consequently, the resultant rDG method can be implemented in a compact manner.

3.3 BR II scheme

The local and global lift operator are described in this subsection. For cell i and j sharing one common face ij , the local lift operator r_i defined in cell i , could be expressed as follows,

$$\int_{\Omega_i} B_i r_h d\Omega_i = \int_{\Gamma_{ij}} \frac{1}{2} (u_{hj}^R - u_{hi}^R) n_k B_i d\Gamma_{ij}, \quad 1 \leq i \leq N \quad (26)$$

where $\mathbf{n}_k$ denotes the normal vector with respect to the common face, Γ_{ij} represents the common face shared by cell i and j , and U_{hj}^R represents a reconstructed polynomial solution of degree of 2 at the interface Γ_{ij} from cell j . Then based on the local lift operator, the global lift operator could be obtained as follows,

$$\int_{\Omega_i} B_i R_h d\Omega_i = \int_{\partial\Omega_i} \frac{1}{2} (u_{hj}^R - u_{hi}^R) n_k B_i d\Gamma_{ij}, \quad 1 \leq i \leq N \quad (27)$$

In our present work, the local and global operator are just linear polynomial solutions, meaning that the basis function of local and global operator is just the basis of polynomial function of degree P_1 .

$$r_h = \tilde{r}_i + r_{xi} B_2 + r_{yi} B_3 + r_{zi} B_4 \quad (28)$$

$$R_h = \tilde{R}_i + R_{xi} B_2 + R_{yi} B_3 + R_{zi} B_4 \quad (29)$$

Compared with basis function of polynomial function of degree P_2 , this strategy obviously could reduce the computational cost which is another great contribution to high-order method from rDG method, and will be proven to give designed order of accuracy. Whereas, we also need to mention that, when we calculate the lift operator, the value at the interface should come from the reconstructed quadratic polynomial.

4. Numerical Examples

Computational results for a number of test cases are presented in this section. The first two test cases are chosen to demonstrate that the rDG(P₁P₂) method is able to achieve the designed third-order of accuracy for Navier-Stokes equations. The last four test cases are presented to illustrate that the rDG(P₁P₂) method can be applied to obtain reliable and accurate solutions for a variety of compressible flow problems, e.g. DNS. Note that the present rDG method is developed on hybrid grids, which has the ability to compute 1D, 2D, and 3D problems using the same code, which greatly alleviates the need and pain for code maintenance and upgrade. Results for one-dimensional flows can be readily obtained by using a hexahedral mesh and by setting the number of cells in y- and z-directions to be 2 and 1 respectively. For two-dimensional problems, the number of cells in the z-direction is simply set to be 1.

4.1 Heat Equation

The 1D heat equation $u_t = u_{xx}$ with periodic boundary conditions is first solved from $x=0$ to $x=2\pi$ to study the convergence of the rDG methods for the discretization of diffusion operator. The analytical solution to this problem is $u(x,t) = e^{-t}\sin(x)$, which provides a valuable reference for measuring the accuracy of a numerical solution. A 3-stage TVD Runge-Kutta scheme and a very small time step of $1e-04$ are used for the time integration in order to ensure that the error from spatial discretization dominates the one from temporal discretization. In this case, numerical solutions at $t=0.01$ and $t=0.1$ to this problem are computed using rDG(P₁P₁), and rDG(P₁P₂) methods and the L_2 norm of the error between the numerical and analytical solution is used to obtain quantitative measurement of the order of accuracy and discretization errors. The detailed results for this test case are presented in Tables 1a-b. They show the mesh size, the L_2 -error of the solutions, and the order of convergence. **Fig. 1a-b** provides the spatial convergence of each

method for this numerical experiment. As expected, the DG method exhibits a full $O(h^{p+1})$ order of convergence. The rDG(P₁P₂) method does offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P₁) method. One can observe that higher order DG methods require significantly less degrees of freedom than lower order ones for the same accuracy.

Table 1a

Heat equation test case at t=0.01: Convergence order of accuracy for rDG(P₁P₁) and rDG(P₁P₂) methods.

Mesh	rDG (P1P1)		rDG (P1P2)	
	L2-error	Order	L2-error	Order
4x2x1	1.6222e-1		6.8874e-2	
8x2x1	4.0260e-2	2.017	1.0221e-2	2.761
16x2x1	1.0071e-2	2.006	1.0513e-3	3.292
32x2x1	2.5208e-3	2.005	1.1508e-4	3.202
64x2x1	6.3041e-4	2.007	1.2735e-5	3.187

Table 1b

Heat equation test case at t=0.1: Convergence order of accuracy for rDG(P₁P₁) and rDG(P₁P₂) methods.

Mesh	rDG (P1P1)		rDG (P1P2)	
	L2-error	Order	L2-error	Order
4x2x1	1.4578e-1		5.5053e-2	
8x2x1	3.7251e-2	1.975	9.9002e-3	2.482
16x2x1	9.4098e-3	1.992	1.2120e-3	3.041

32x2x1	2.3591e-3	2.002	1.2483e-4	3.291
64x2x1	5.9033e-4	2.005	1.2238e-5	3.362

(a) $t=0.01$

(b) $t=0.1$

Fig. 1. Convergence order of accuracy of the rDG(P₁P₁) and rDG(P₁P₂) methods for the heat equation test case.

4.2 Couette flow

For the second test case, Couette flow has been considered to verify the accuracy and order of convergence of the rDG(P₁P₂) method for solving the compressible Navier-Stokes equations on 3D grids. It is the laminar viscous flow between two parallel plates. The bottom plate with a fixed temperature (T_0) is stationary, while the top one with a fixed temperature (T_1) moves at a

constant velocity of U . The two plates are separated by a distance H . In this case, we have to pay attention to that the viscosity coefficient μ is a constant, otherwise the designed order of accuracy could not be achieved. There exists exact solution for this case, which is given as follows.

$$\begin{aligned} u &= \frac{y}{H}U, \quad v = 0, \quad w = 0, \\ p &= p_\infty, \quad \rho = \frac{P}{RT}, \\ T &= T_0 + \frac{y}{H}(T_1 - T_0) + \frac{y}{H}\left(1.0 - \frac{y}{H}\right)\frac{\text{Pr}U^2}{2c_p} \end{aligned}$$

where Pr denotes the Prandtl number and C_p is the specific heat capacity for constant pressure. In our numerical experiment, $H=2$, $T_0=0.8$, and $T_1=0.85$. The Mach number for the upper wall is set equal to 0.1. Reynold number $Re=100$ with a constant viscosity coefficient $\mu=0.001$. The computational domain is a rectangle ($0 \leq x \leq 2H$ and $0 \leq y \leq H$). For the boundary faces along the span-wise direction, they are set as symmetric boundary. Exact boundary condition has been set at the ghost states for all the boundary faces. A set of 4 successively refined unstructured hybrid meshes are used to assess the accuracy and order of convergence of the $rDG(P_1P_2)$ method. A 3-stage TVD Runge-Kutta scheme is used for the time advancing. The L_2 norm of the error between the numerical temperature and exact temperature is used to measure the order of convergence. The detailed results with $rDG(P_1P_1)$ and $rDG(P_1P_2)$ are presented in Table 2. The level 2 mesh used in this case is shown in **Fig. 2(a)**. And **Fig. 2(b)** provides the spatial convergence of each method for this numerical experiment.

Table 2Convergence order of accuracy for rDG(P₁P₁) and rDG(P₁P₂) methods for Couette flow

Mesh	rDG (P ₁ P ₁)		rDG (P ₁ P ₂)	
	L2-error	Order	L2-error	Order
45	2.4186e-5		4.7247e-6	
180	5.1728e-6	2.23	5.3968e-7	3.14
720	1.1974e-6	2.11	6.1218e-8	3.15
2,880	2.9285e-7	2.03	8.5549e-9	2.84
11,520	7.3626e-8	1.99	8.9279e-10	3.27

Fig. 2. (a) Level 1 hybrid mesh (b) Convergence order of accuracy of the rDG(P₁P₁) and rDG(P₁P₂) methods for Couette flow.

4.3 Subsonic viscous flow past a sphere

This test case involves a subsonic viscous flow past a sphere at a Mach number of 0.5, and an angle of attack 0° , and a Reynolds number of 118 based on the freestream velocity and the diameter of the sphere, which has been studied both experimentally [38] and numerically [39]. An adiabatic wall is assumed in this test case. The computation is then carried out using the implicit BDF1 temporal and the rDG(P₁P₂) spatial discretization. Note that only a half of the configuration is modeled, due to the symmetry of the problem. **Fig. 3** shows the computational grid used in this test case, consisting of 119,390 tetrahedral elements, 22,530 grid points, and 4,511 boundary points, and the computed Mach contours in the flow field obtained by the rDG(P₁P₂) method is shown in **Fig. 4**. The computational streamlines are compared with experimental streamlines in **Fig. 5**, where steady separation bubble is readily observed in both plots and the size of the separation region in the computation agrees very well with that of the experiment.

(a) Global view

(b) Close up near the sphere surface

Fig. 3. Unstructured mesh (nelem=119,390, npoin=22,530, nboun=4,511) for subsonic flow past a sphere at $Ma_\infty=0.5$, $Re=118$, $\alpha=0^\circ$.

Fig. 4. Computed Mach number contours by the rDG(P₁P₂) for subsonic flow past a sphere at $Ma_\infty = 0.5$, $Re=118$, $\alpha=0^\circ$.

Fig. 5. Velocity vector plot in the back of the sphere from the computation (left) and streamlines of the flow filed from experiment (right) for subsonic flow past a sphere at $M_\infty = 0.5$, $Re=118$, $\alpha=0^\circ$.

4.4 Viscous flow past an SD7003 airfoil

A viscous flow past an SD7003 airfoil at a free-stream Mach number of $Ma_\infty = 0.1$, an angle of attach of $\alpha = 4^\circ$, and a Reynolds number of $Re = 10,000$ is considered in this test case. The computation is initialized with constant values in the entire domain with no-slip, adiabatic boundary conditions on the solid wall. **Figs. 6(a) – 6(c)** shows the grid used in the computation, which consists of 50,781 prismatic elements, 52,176 grid point, 101,562 triangular boundary

faces, and 279 quadrilateral boundary faces. The computation is performed using IRK3 temporal and rDG(P₁P₂) spatial discretization schemes, and a fixed time-step size of $\Delta t = 0.01$, corresponding to a maximal CFL number of about 5000. Typical computed pressure contours in the flow field are compared with those obtained using a compact scheme [40-41] in **Figs. 7(a)** and **7(b)**, while the comparison for the vorticity magnitude contours between the rDG(P₁P₂) method and the compact method is shown in **Figs. 8(a)** and **8(b)**. Quantitatively, both solutions look very similar, which capture the same flow features: separation of the flow on the upper surface of the airfoil and shedding of the trailing vortices, in spite of the fact that a round trailing edge is used in the compact solution and a sharp trailing edge is used in the rDG solution. The computed pressure contours in the flow field are also presented along with the grid in **Fig. 9(a)** to illustrate that accurate and smooth solutions are obtained using the rDG(P₁P₂) method in spite of the highly stretched grid used in the boundary layer. **Fig. 9(b)** shows the computed velocity vectors in the flow field, where the development of the boundary layers and flow separation on the upper surface of the airfoil are clearly visible.

(a)

(b)

(c)

Fig. 6. The prismatic grid used for computing a viscous flow past an SD7003 airfoil: (a) global view, (b) leading edge, and (c) trailing edge.

(a)

(b)

Fig. 7. Computed pressure contours obtained by (a) the compact method, and (b) the IRK3 and the rDG(P₁P₂) method in the flow field for the viscous flow past an SD7003 airfoil at a free-stream Mach number of $Ma_\infty = 0.1$, an angle of attach of $\alpha = 4^\circ$, and a Reynolds number of $Re_\infty = 10,000$.

(a)

(b)

Fig. 8: Computed vorticity contours obtained by (a) the compact method, and (b) the IRK3 and rDG(P₁P₂) method in the flow field for the viscous flow past an SD7003 airfoil at a free-stream Mach number of $Ma_\infty = 0.1$, an angle of attach of $\alpha = 4^\circ$, and a Reynolds number of $Re_\infty = 10,000$.

(a)

(b)

Fig. 9. Computed pressure contours on the upper surface of the airfoil (a) and velocity vectors near the airfoil (b) for the viscous flow past a SD7003 airfoil.

4.5. Lid driven cavity flow in 3D

A transitional flow in a three-dimensional lid driven cavity at $Re_\infty = 10,000$ is considered in this test case to assess the ability of the rDG method for complex flows, as the accuracy of the numerical results obtained can be assessed through comparison with both the experimental data [42] and the LES data [43]. The cavity dimensions are 1 unit in the stream-wise x direction and vertical y direction, and 0.5 unit in the span-wise z direction. A hexahedral grid consisting of $64 \times 64 \times 32$ grid points is used in computation, as shown in **Fig. 10**. While being equally distributed in the z direction, the grid points are clustered near the walls in the x - y plane, with the grid spacing geometrically stretched away from the wall with the first element thickness being 0.005 ($y^+ = 3.535$). On the bottom and sidewalls, the no-slip, adiabatic boundary conditions are prescribed. Along the top “lid”, the no-slip, adiabatic boundary conditions with a lid velocity $V_b = (0.2, 0, 0)$ are imposed to ensure an essentially incompressible flow field. The computation is conducted in two stages. At stage I, the computation is started with a zero-velocity field, and sufficient steps are taken to evolve the field into a cyclically oscillating state by using the BDF1 + rDG(P_1P_2) method with $CFL = 500$. The solution obtained at the end of stage I is used as the initial solution for stage II. The computation is then carried out using the IRK3 temporal and the rDG(P_1P_2) spatial discretization methods for 30,000 time steps with a constant step size of $\Delta t = 0.1$, during which the instantaneous solutions are written every 300 time steps for time averaging calculations. The computed mean velocity and components of Reynolds stress along the center-lines on the span-wise mid-plane are presented in **Fig. 11**, **Fig. 12** and **Fig. 13**, respectively. Those profiles are obtained by using a linear polynomial interpolation of the elemental solutions at the intersected nodes cut through by the center- lines. For example, in **Fig. 11**, the mean x -velocity u is plotted along the vertical center-line (vs. y -coordinate), and the mean y -velocity v is plotted along the horizontal center-line (vs. x -coordinate). One can observe that the results

obtained by the rDG(P₁P₂) method are in good agreement with the experimental data and LES data, demonstrating the ability of the rDG(P₁P₂) method for the computing complex flows.

Fig.10. Hexahedral grid (64x64x32 points) used for the LES of a lid driven cavity ($x:y:z=1:1:0.5$) at $Re=10,000$.

Fig.11. Comparison of the normalized mean velocity components $\langle u \rangle / u_B$ and $\langle v \rangle / u_B$ in the span-wise mid-plane with the classical experimental data by Prasad & Koseff and numerical data by Zang et al, for the LES of a lid driven cavity ($x:y:z=1:1:0.5$) at $Re=10,000$.

Fig.12. Comparison of the scaled RMS velocity components $10\langle u'u' \rangle^{0.5}/u_B$ and $10\langle v'v' \rangle^{0.5}/u_B$ in the span-wise midplane with the classical experimental data by Prasad & Koseff (1989) and numerical data by Zang et al, for the LES of a lid driven cavity ($x:y:z=1:1:0.5$) at $Re=10,000$.

Fig.13. Comparison of the scaled mean Reynolds stress tensor components $500\langle u'v' \rangle/u_B^2$ in the span-wise mid-plane with the classical experimental data by Prasad & Koseff and numerical data by Zang et al, for the LES of a lid driven cavity ($x:y:z=1:1:0.5$) at $Re=10,000$.

4.6 Direct Numerical Simulation of the Taylor-Green Vortex

The Taylor-Green vortex flow problem, one of the benchmark cases in the 3rd International Workshop on high order CFD methods, is chosen in this test case to assess the accuracy and performance of the rDG(P₁P₂) method for the Direct Numerical Simulation (DNS) of turbulent flows. The Taylor-Green vortex flow was originally developed to study the dynamics of turbulence numerically. The initial conditions are smooth, but the flow quickly transitions to turbulence with the creation of small scales and begins to decay, mimicking homogeneous non-isotropic turbulence. The initial conditions of the Taylor-Green vortex flow are given by:

$$\begin{aligned} u_1 &= V_0 \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ u_2 &= -V_0 \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ u_3 &= 0 \\ p &= p_0 + \rho_0 V_0^2 \left[\frac{1}{16} \left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right) \right] \end{aligned}$$

where $\rho_0=1$, $p_0=1/\gamma$, and u_1 , u_2 , and u_3 are the components of the velocity in the x-, y- and z- directions respectively and p is the pressure. The flow is initialized to be isothermal ($p/\rho=p_0/\rho_0=RT_0$). To minimize the effects of compressibility, the free-stream Mach number is set to 0.1. The Reynolds number in this case is 1,600, which corresponds to a peak Taylor microscale Reynolds number of about 22. The flow is computed in a periodic and square box, which spans $[-\pi L, \pi L]$ in each coordinate direction. The physical duration of the computation is 20 based on the characteristic convective time defined as $t_c=L/V_0$, i.e., $t_{\text{final}} = 20t_c$. The computation is then carried out using explicit 4-stage Runge-Kutta temporal and the rDG(P₁P₂) spatial discretization. A grid convergence study to this problem is performed using the rDG(P₁P₂) method on 4 hexahedral meshes, which have 40^3 , 81^3 , 161^3 and 256^3 elements respectively. **Fig. 14** shows the computed vortex detection criterion Q at $t=8 t_c$, on the finest mesh. One can observe that the

vortex structure obtained by the rDG(P₁P₂) solution looks very similar to the one from Reference [44].

Fig. 15 and **Fig. 16** compare the time history of the kinetic energy $E_k = \frac{1}{\Omega} \int \frac{1}{2} \rho v v d\Omega$ and the kinetic energy dissipation rate $e = -\frac{dE_k}{dt}$ computed from the data at the space-time quadrature points, respectively, for all 4 grids with the result from an incompressible simulation using a spectral code on a mesh of 512^3 grid points [45]. There remains significant numerical dissipation on the coarse mesh and the point of peak dissipation is poorly captured. The higher the grid resolution is, the more accurate the numerical solutions are. The results from the rDG(P₁P₂) solution on the finest mesh agree very well with the ones from the spectral code solution. In addition, the temporal evolution of the enstrophy integrated on the whole domain is shown in **Fig. 17**. Compared with the kinetic energy dissipation rate, it seems that it is a little more difficult to match well with reference solution for the enstrophy.

The individual terms in the kinetic energy evolution equation can be used to assess the accuracy of the numerical solutions. The kinetic energy dissipation rate in compressible flows is given by the sum of three contributions $e = e_1 + e_2 + e_3 = -\frac{dE_k}{dt}$, where $e_1 = \frac{1}{\Omega} \int 2\mu s_{ij} s_{ij} d\Omega$, $e_2 = \frac{1}{\Omega} \int \mu_v u_{kk} u_{kk} d\Omega$, and $e_3 = -\frac{1}{\Omega} \int p u_{kk} d\Omega$. Here, $s_{ij} = \frac{1}{2}(u_{ij} + u_{ji})$ is the strain-rate tensor. In this case, the gas is assumed to have zero bulk viscosity, $\mu_v = 0$. Therefore, the dissipation due to the bulk viscosity is always equal to zero, meaning that $e_2 = 0$. Since the flow is nearly incompressible, the dissipation due to the pressure-dilatation term (e_3) can be expected to be small. The kinetic energy dissipation rate, e , is then approximately equal e_1 . However, for the compressible simulation this does not hold exactly. Time histories of the computed e , e_1 , and e_3 on the finest mesh are presented in **Fig. 18**. The pressure-dilatation, e_3 , has a significant bias,

contributing large net positive kinetic energy dissipation. With increasing mesh refinement, the biased pressure-dilatation term decreases toward zero as shown in **Fig. 19**. Compressibility effects are evident in oscillations of the pressure dilatation term (e_3).

Runs [46] are performed on the ARC at the NC State University. All machines are 2-way SMPs with AMD Opteron 6128 (Magny Core) processors with 8 cores per socket (16 cores per node). Finally, the simulation details and computational costs for different meshes are listed in **Table 3**.

Fig. 14. Iso-surfaces of the computed vortex detection criterion Q colored by the velocity magnitude at time $t=8t_c$ by the rDG(P_1P_2) solution on the finest mesh.

Fig. 15. Evolution of the dimensionless kinetic energy as a function of the dimensionless time.

Fig. 16. Evolution of the dimensionless kinetic energy dissipation rate as a function of the dimensionless time.

Fig. 17. Evolution of the enstrophy as a function of the dimensionless time.

Fig. 18. Kinetic energy dissipation balance for the Taylor-Green vortex problem on the finest mesh.

Fig. 19. Biased pressure-dilatation term for the Taylor-Green vortex problem, with mesh refinement.

Table 3

The simulation details and computational costs of the DNS of Taylor-Green vortex.

Mesh	Memory (MB)	Time-step size	Time steps	No. of cores	CPU time (sec)
40x40x40	4.88	4.00e-03	2000	32	2963.7
81x81x81	40.52	2.00e-03	4000	32	36049.4
161x161x161	318.21	8.00e-04	10000	32	581799.6

5. Conclusions

A reconstructed discontinuous Galerkin method has been developed for the solution of the compressible Navier-Stokes equations on hybrid grids. A piecewise quadratic polynomial solution is obtained from the underlying piecewise linear DG solution using a hierarchical WENO reconstruction. The reconstructed quadratic polynomial solution is then used for the computation of the inviscid fluxes and the viscous fluxes using the BR11 scheme. The developed rDG(P₁P₂) method is used to compute a variety of flow problems to demonstrate its accuracy,

efficiency, robustness, and versatility. The numerical results indicate that the rDG(P₁P₂) method is able to achieve the designed third order of convergence, one order higher accuracy than the underlying DG method for the compressible Navier-Stokes equations, outperforms its third order DG method in terms of both computing costs and storage requirements, and is able to obtain accurate and reliable solutions for the direct numerical simulation of compressible turbulent flows.

Acknowledgments

The authors would like to acknowledge the partial support for this work provided by DOE under Nuclear Engineering University Program.

References

- [1] Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. Los Alamos Scientific Laboratory Report, LA-UR-73-479; 1973.
- [2] Cockburn B, Hou S, Shu CW. TVD runge-kutta local projection discontinuous galerkin finite element method for conservation laws IV: the multidimensional case. *Math Comput* 1990; 55:545–581.
- [3] Cockburn B, Shu CW. The runge-kutta discontinuous galerkin method for conservation laws v: multidimensional system. *J Comput Phys* 1998; 141:199–224.
- [4] Cockburn B, Karniadakis G, Shu CW. The development of discontinuous galerkin method, theory, computation, and applications. In: *Lecture notes in computational science and engineering*, 11. Springer-Verlag; 2000. p. 5–50.
- [5] Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2d euler equations. *J Comput Phys* 1997; 138:251–85.
- [6] Atkins HL, Shu CW. Quadrature free implementation of discontinuous galerkin method for hyperbolic equations. *AIAA J* 1998; 36:775–82.
- [7] Bassi F, Rebay S. GMRES discontinuous galerkin solution of the compressible navier-stokes equations. In: *Lecture notes in computational science and engineering*, 11. Springer-Verlag; 2000. p. 197–208.
- [8] Warburton TC, Karniadakis GE. A discontinuous galerkin method for the viscous MHD equations. *J Comput Phys* 1999; 152:608–41.

- [9] Hesthaven JS, Warburton T. Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Springer Science & Business Media; 2007.
- [10] Rasetarinera P, Hussaini MY. An efficient implicit discontinuous spectral galerkin method. *J Comput Phys* 2001; 172:718–38.
- [11] Helenbrook BT, Mavriplis D, Atkins HL. Analysis of p-multigrid for continuous and discontinuous finite element discretizations. *AIAA* 2003:2003–3989.
- [12] Fidkowski KJ, Oliver TA, Lu J, Darmofal DL. P-multigrid solution of high-order discontinuous galerkin discretizations of the compressible navier-stokes equations. *J Comput Phys* 2005; 207:92–113.
- [13] Luo H, Baum JD, Löhner R. A discontinuous galerkin method using taylor basis for compressible flows on arbitrary grids. *J Comput Phys* 2008a; 227:8875–93.
DOI: 210.1016/j.jcp.2008.06.035
- [14] Luo H, Baum JD, Löhner R. On the computation of steady state compressible flows using discontinuous galerkin method. *Int J Numer Meth Eng* 2008b; 73:597–623.
- [15] Luo H, Baum JD, Löhner R. A hermite WENO-based limiter for discontinuous galerkin method on unstructured grids. *J Comput Phys* 2007; 225(1):686–713.
- [16] Luo H, Baum JD, Löhner R. A p-multigrid discontinuous galerkin method for the euler equations on unstructured grids. *J Comput Phys* 2006; 211:767–83.
- [17] Luo H, Baum JD, Löhner R. A fast, p-multigrid discontinuous galerkin method for compressible flows at all speeds. *AIAA J* 2008; 46:635–52.
- [18] Dumbser M, Balsara DS, Toro EF, Munz CD. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *J Comput Phys* 2008; 227:8209–53.
- [19] Dumbser M, Zanotti O. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *J Comput Phys* 2009; 228: 6991-7006.
- [20] Dumbser M. Arbitrary high-order pnpm schemes on unstructured meshes for the compressible navier-stokes equations. *Comput Fluids* 2010; 39:60–76.
- [21] Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. *J Comput Phys* 1997b; 131:267–79.

- [22] Bassi F, Rebay S. Discontinuous galerkin solution of the reynolds-averaged navier-stokes and $k - \omega$ turbulence model equations. *J Comput Phys* 2005; 34:507–40.
- [23] Cockburn B, Shu CW. The local discontinuous galerkin method for time-dependent convection-diffusion system. *SIAM, J Numer Anal* 2001;16:2440–63.
- [24] Baumann CE, Oden JT. A discontinuous hp finite element method for the euler and navier-stokes equations. *Int J Numer Meth Fluids* 1999; 31:79–95.
- [25] Luo H, Luo L, and Xu K. A discontinuous galerkin method based on a BGK scheme for the Navier-Stokes equations on arbitrary grids. *Adv App Math Mech* 2009; 1(3):301-18.
- [26] Luo H, Luo LQ, Nourgaliev R. A reconstructed discontinuous galerkin method for the compressible euler equations on arbitrary grids. *Commun Computat Phys* 2012; 12:1495–519.
- [27] Luo H, Luo LQ, Nourgaliev R, Mousseau VA, Dinh N. A reconstructed discontinuous galerkin method for the compressible navier-stokes equations on arbitrary grids. *J Comput Phys* 2010; 229:6961–78.
- [28] Luo H, Luo LQ, Ali A, Nourgaliev R, Cai CP. A parallel, reconstructed discontinuous galerkin method for the compressible flows on arbitrary grids. *Commun Computat Phys* 2011a; 9:363–89.
- [29] Zhang LP, Liu W, He LX, Deng XG, Zhang HX. A class of hybrid DG/FV methods for conservation laws I: basic formulation and one-dimensional system. *J Comput Phys* 2012; 231:1081-103.
- [30] Zhang LP, Liu W, He LX, Deng XG, Zhang HX. A class of hybrid DG/FV methods for conservation laws II: two-dimensional cases. *J Comput Phys* 2012; 231:1104–20.
- [31] Luo H, Xia Y, Nourgaliev R, and Cai C. A class of reconstructed discontinuous galerkin methods for the compressible flows on arbitrary grids. *AIAA* 2011: 2011-0199.
- [32] Luo H, Xiao H, Nourgaliev R, Cai C. A comparative study of different reconstruction schemes for reconstructed discontinuous galerkin methods for the compressible flows on arbitrary grids. *AIAA* 2011b:2011–3839.
- [33] Haider DF, Croisille JP, and Courbet, B. Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids. **Numirische Mathematik**, 2009; 113: 555-600.
- [34] Balsara D, Altmann C, Munz CD, Dumbser M. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG + HWENO schemes. *J Comput Phys*, 2007; 226: 586–620.

- [35] Luo H, Xia Y, Li S, Nourgaliev R, Cai C. A hermite WENO reconstruction-based discontinuous galerkin method for the Euler equations on tetrahedral grids. *J Comput Phys*, 2012; 231: 5489-5503.
- [36] Luo H, Xia YD, Spiegel S, Nourgaliev R, Jiang ZG. A reconstructed discontinuous galerkin method based on a hierarchical WENO reconstruction for compressible flows on tetrahedral grids. *J Comput Phys* 2013; 236:477–92.
- [37] Xia YD, Liu XD, Luo H. A finite volume method based on weno reconstruction for compressible flows on hybrid grids. *AIAA* 2014: 2014-0939.
- [38] Taneda S. Experimental Investigations of the wake behind a sphere at low Reynolds numbers, *J Phys Soc Jpn* 1956; 11: 1104-1108.
- [39] Sun Y, Wang ZJ, Liu Y. High-Order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, *Commun Computat Phys*, 2007; 2: 310-333.
- [40] Gordnier RE, Visbal MR. Compact difference scheme applied to simulation of Low-Sweep Delta wing flow. *AIAA J*, 2005; 43:1744-1752.
- [41] Visbal MR and Gaitonde DV. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, *J Comput Phys*, 2002; 181: 155-185.
- [42] Prasad AK, Koseff J. Reynolds number and end-wall effects on a lid-driven cavity flow. *Phys Fluids A*, 1989; 1: 208-218.
- [43] Zang Y, Street RL, Koseff JR. A dynamic mixed subgrid-scale model and its application to turbulent recirculating Flows. **Phys Fluids A**, 1993; 5: 3186-3196.
- [44] Bull JR, Jameson A. Simulation of the compressible Taylor Green vortex using High-Order flux reconstruction schemes. *AIAA* 2014: 2014-3210.
- [45] Van Rees WM, Leonard A, Pullin DI, Koumoutsakos P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds number. *J Comput Phys*, 2011; 230: 2794-2805.
- [46] Liu XD, Xia YD, Luo H. Assessment of the RDGFLO3D for a Delta Wing at Low Reynolds Number (C2.2) and DNS of the Taylor-Green Vortex (C3.3), January 3-4, 2015. https://www.grc.nasa.gov/hio CFD/wp-content/uploads/sites/22/C3.3_NCState.pdf

Appendix A

In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the center of the cell. For example, if we do a Taylor series expansion at the center of the cell, a quadratic polynomial solution can be expressed as follow

$$\begin{aligned} \mathbf{U}_h = & \mathbf{U}_c + \frac{\partial \mathbf{U}}{\partial x}|_c (x - x_c) + \frac{\partial \mathbf{U}}{\partial y}|_c (y - y_c) + \frac{\partial \mathbf{U}}{\partial z}|_c (z - z_c) + \frac{\partial^2 \mathbf{U}}{\partial x^2}|_c \frac{(x - x_c)^2}{2} + \frac{\partial^2 \mathbf{U}}{\partial y^2}|_c \frac{(y - y_c)^2}{2} + \\ & \frac{\partial^2 \mathbf{U}}{\partial z^2}|_c \frac{(z - z_c)^2}{2} + \frac{\partial^2 \mathbf{U}}{\partial x \partial y}|_c (x - x_c)(y - y_c) + \frac{\partial^2 \mathbf{U}}{\partial x \partial z}|_c (x - x_c)(z - z_c) + \frac{\partial^2 \mathbf{U}}{\partial y \partial z}|_c (y - y_c)(z - z_c) \end{aligned} \quad (\text{A.1})$$

which can be further expressed as cell-averaged values and their derivatives at the center of the cell:

$$\begin{aligned} \mathbf{U}_h = & \tilde{\mathbf{U}} + \frac{\partial \mathbf{U}}{\partial x}|_c (x - x_c) + \frac{\partial \mathbf{U}}{\partial y}|_c (y - y_c) + \frac{\partial \mathbf{U}}{\partial z}|_c (z - z_c) + \frac{\partial^2 \mathbf{U}}{\partial x^2}|_c \left(\frac{(x - x_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\Omega \right) + \\ & \frac{\partial^2 \mathbf{U}}{\partial y^2}|_c \left(\frac{(y - y_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\Omega \right) + \frac{\partial^2 \mathbf{U}}{\partial z^2}|_c \left(\frac{(z - z_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(z - z_c)^2}{2} d\Omega \right) + \frac{\partial^2 \mathbf{U}}{\partial x \partial y}|_c \left((x - x_c)(y - y_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\Omega \right) + \\ & \frac{\partial^2 \mathbf{U}}{\partial x \partial z}|_c \left((x - x_c)(z - z_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(z - z_c) d\Omega \right) + \frac{\partial^2 \mathbf{U}}{\partial y \partial z}|_c \left((y - y_c)(z - z_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (y - y_c)(z - z_c) d\Omega \right) \end{aligned} \quad (\text{A.2})$$

where $\tilde{\mathbf{U}}$ is the mean value of $\mathbf{U}$ in this cell. The unknowns to be solved for in this formulation are the cell-averaged variables and their derivatives at the center of the cells. The dimension of the polynomial space is 10 and the basis functions are

$$\begin{aligned} B_1 = 1, B_2 = (x - x_c), B_3 = (y - y_c), B_4 = (z - z_c), \\ B_5 = \frac{B_2^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_2^2}{2} d\Omega, B_6 = \frac{B_3^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_3^2}{2} d\Omega, B_7 = \frac{B_4^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_4^2}{2} d\Omega, \end{aligned}$$

$$B_8 = B_2B_3 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2B_3 d\Omega, B_9 = B_2B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2B_4 d\Omega, B_{10} = B_3B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_3B_4 d\Omega. \quad (\text{A.3})$$

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix as follows:

$$\begin{aligned} B_1 &= 1, B_2 = \frac{(x-x_c)}{\Delta x}, B_3 = \frac{(y-y_c)}{\Delta y}, B_4 = \frac{(z-z_c)}{\Delta z}, \\ B_5 &= \frac{B_2^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_2^2}{2} d\Omega, B_6 = \frac{B_3^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_3^2}{2} d\Omega, B_7 = \frac{B_4^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{B_4^2}{2} d\Omega, \\ B_8 &= B_2B_3 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2B_3 d\Omega, B_9 = B_2B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_2B_4 d\Omega, B_{10} = B_3B_4 - \frac{1}{\Omega_e} \int_{\Omega_e} B_3B_4 d\Omega. \end{aligned} \quad (\text{A.4})$$

Finally, the basis functions Eq.(13) and unknowns Eq.(14) used in our work could be derived.