INL/CON-15-34042
PREPRINT

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

53rd AIAA Aerospace Sciences Meeting

Xiaodong Liu, Lijun Xuan,Hong Luo, Yidong Xia

January 2015

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

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

Xiaodong Liu[1], Lijun Xuan[2], and Hong Luo[3]
North Carolina State University, Raleigh, NC, 27695

Yidong Xia[4]
Idaho National Laboratory, Idaho Falls, ID, 83415

A reconstructed discontinuous Galerkin (rDG(P_1P_2)) 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(P_1P_2) method is used to compute a variety of flow problems to assess its accuracy, efficiency, and robustness. The numerical results demonstrate that the rDG(P_1P_2) 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 large eddy simulation (LES) and direct numerical simulation (DNS) of compressible turbulent flows.

I. 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. The discontinuous Galerkin methods have many attractive features:1) They have several useful mathematical properties with respect to conservation, stability, and convergence; 2) The method can be easily extended to higher-order (>2^{nd}) approximation; 3) The methods are well suited for complex geometries since they can be applied on unstructured grids. In addition, the methods can also handle non-conforming elements, where the grids are allowed to have hanging nodes; 4) The methods are highly parallelizable, as they are compact and each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal, domain decomposition can be efficiently employed. The compactness also allows for structured and simplified coding for the methods; 5) They can easily handle adaptive strategies, since refining or coarsening a grid can be achieved without considering the continuity restriction commonly associated with the conforming elements. The methods allow easy implementation of *hp*-refinement, for example, the order of accuracy, or shape, can vary from element to element; 6) They have the ability to compute low Mach number flow problems without recourse to the time-preconditioning techniques normally required for the finite volume methods. However, DGM 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

[1]Ph.D. Student, Department of Mechanical and Aerospace Engineering, Student Member AIAA.
[2]Post-Doc Research Associate, Department of Mechanical and Aerospace Engineering, Member AIAA.
[3]Professor, Department of Mechanical and Aerospace Engineering, Associate Fellow AIAA.
[4]Post-Doc Research Associate, Department of Energy Resource Recovery & Sustainability, Member AIAA.

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 and unresolved issues 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 PnPm 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 Pm 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. The beauty of rDG(P_nP_m) schemes is that they provide a unified formulation for both finite volume and DGM, and contain both classical finite volume and standard DG methods as two special cases of rDG(P_nP_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, rDG(P_0P_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 rDG(P_nP_n) scheme yields a standard DG method.

Obviously, the construction of an accurate and efficient reconstruction operator is crucial to the success of the rDG(P_nP_m) schemes. In Dumbser's work, a higher order polynomial solution is reconstructed using a L_2 projection, requiring it indistinguishable from the underlying DG solutions in the contributing cells in the weak sense. The resultant over-determined system is then solved using a least-squares method that guarantees exact conservation, not only of the cell averages but also of all higher order moments in the reconstructed cell itself, such as slopes and curvatures. However, this conservative least-squares reconstruction approach is computationally expensive, as the L_2 projection, i.e., the operation of integration, is required to obtain the resulting over-determined system. Furthermore, the reconstruction might be problematic for a boundary cell, where the number of the face-neighboring cells might be not enough to provide the necessary information to recover a polynomial solution of a desired order. Fortunately, the projection-based reconstruction is not the only way to obtain a polynomial solution of higher order from the underlying discontinuous Galerkin solutions. 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. Like the projection-based reconstruction, the strong reconstruction scheme guarantees exact conservation, not only of the cell averages but also of their slopes due to a judicious choice of the Taylor basis. 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(P_0P_1), the resultant rDG(P_1P_2) method is unstable. Although rDG(P_0P_1) 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(P_1P_2) 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(P_1P_2) 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(P_1P_2) 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. In this rDG(P_1P_2) method, a quadratic solution is first reconstructed to enhance the accuracy of the underlying DG method in two steps: (1) all second derivatives on each cell are first reconstructed using the solution variables and their first derivatives from adjacent face-neighboring cells via a strong interpolation; (2) the final second derivatives on each cell are then obtained using a WENO strategy based on the reconstructed second derivatives on the cell itself and its adjacent face-neighboring cells. This reconstruction scheme, by taking advantage of handily available and yet valuable information namely the gradients in the context of the DG methods, only involves von Neumann neighborhood and thus is compact, simple, robust, and flexible. As the underlying DG method is second-order, and the basis functions are at most linear functions, fewer quadrature points are then required for both domain and face integrals, and the number of unknowns (the number of degrees of freedom) remains the same as for the DG(P_1). Consequently, this rDG method is more efficient than its third order DG(P_2) counterpart. The gradients of the quadratic polynomial solutions are then modified using a WENO reconstruction in order to eliminate non-physical oscillations in the vicinity of strong discontinuities, thus ensuring the non-linear stability of the developed rDG method. The developed rDG(P_1P_2) method has used to compute a variety of flow problems on hybrid grids to assess its accuracy, robustness, and non-oscillatory performance. The numerical results obtained indicate that this rDG(P_1P_2) method is able to capture shock waves within once cell without any spurious oscillations, and achieve the designed third-order of accuracy: one order accuracy higher than the underlying DG(P_1) method, and thus significantly increase its accuracy without significant increase in computing costs and memory requirements.

The objective of the effort discussed in this paper is threefold: (a) extend this rDG(P_1P_2) 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(P_1P_2) method can achieve a formal third order of convergence for the compressible Navier-Stokes equations, and (c) demonstrate that the rDG(P_1P_2) method can be effectively and accurately used for the large eddy simulation of turbulent flows. In this rDG(P_1P_2) method, a quadratic solution is first reconstructed to enhance the accuracy of the underlying DG method in two steps: (1) all second derivatives on each cell are first reconstructed using the solution variables and their first derivatives from adjacent face-neighboring cells via a strong interpolation; (2) the final second derivatives on each cell are then obtained using a WENO strategy based on the reconstructed second derivatives on the cell itself and its adjacent face-neighboring cells. This reconstruction scheme, by taking advantage of handily available and yet valuable information namely the gradients in the context of the DG methods, only involves von Neumann neighborhood and thus is compact, simple, robust, and flexible. As the underlying DG method is second-order, and the basis functions are at most linear functions, fewer quadrature points are then required for both domain and face integrals, and the number of unknowns (the number of degrees of freedom) remains the same as for the DG(P_1). Consequently, the rDG(P_1P_2) method is more efficient than its third order DG(P_2) counterpart. The gradients of the quadratic polynomial solutions are then modified using a WENO reconstruction in order to eliminate non-physical oscillations in the vicinity of strong discontinuities, thus ensuring the non-linear stability of the rDG method. The discretization of the viscous and heat fluxes is carried out using the BRII method based on the reconstructed quadratic polynomial solution. The developed rDG(P_1P_2) 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(P_1P_2) method is able to achieve the designed third-order of accuracy: one order of accuracy higher than the underlying DG(P_1) method without significant increase in computing costs and memory requirements and obtain reliable and accurate solutions to the large eddy simulation (LES) and 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.

II. Governing Equations

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

$$\frac{\partial \mathbf{U}(x,t)}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U}(x,t))}{\partial x_k} = \frac{\partial \mathbf{G}_k(\mathbf{U}(x,t))}{\partial x_k} \tag{2.1}$$

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

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix} \qquad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho e + p) \end{pmatrix} \qquad \mathbf{G}_j = \begin{pmatrix} 0 \\ \sigma_{ij} \\ u_l \sigma_{lj} + q_j \end{pmatrix} \tag{2.2}$$

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

$$p = (\gamma - 1)\rho(e - \frac{1}{2} u_j u_j) \tag{2.3}$$

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

$$\sigma_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k}\delta_{ij} \qquad q_j = \frac{1}{\gamma - 1}\frac{\mu}{\mathrm{Pr}}\frac{\partial T}{\partial x_j} \tag{2.4}$$

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

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_0 + S}{T + S} \tag{2.5}$$

μ_0 denotes the viscosity at the reference temperature T_0, and S is a constant which for are assumes the value S = 110°K. The temperature of the fluid T is determined by

$$T = \gamma \frac{p}{\rho} \tag{2.6}$$

Neglecting viscous effects, the left-hand side of Eq. (2.1) represents the Euler equations governing unsteady compressible inviscid flows.

III. Reconstructed Discontinuous Galerkin Method

3.1 Discontinuous Galerkin formulation

The governing equation (2.1) 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 $V_h^{\mathrm{P_n}}$

$$V_h^{\mathrm{P_n}} = \left\{ v_h \in \left[L_2(\Omega)\right]^m : v_h\big|_{\Omega_e} \in \left[V_{\mathrm{P}_n}^m\right] \ \ \forall \Omega_e \in \Omega \right\}, \tag{3.1}$$

which consists of discontinuous vector-values polynomial functions of degree $\mathrm{P_n}$, and where m is the dimension of the unknown vector and

$$V_{\mathrm{P}_n}^m = \mathrm{span}\ \left\{ \Pi x_i^{\alpha_i} : 0 \le \alpha_i \le \mathrm{P_n}, 0 \le i \le d \right\}, \tag{3.2}$$

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 (2.1) by a test function $\mathbf{W}_h$, integrating over an element Ω_e, and then performing an integration by parts,

Find $\mathbf{U}_{\mathrm{P}_n} \in V_h^{\mathrm{P}_n}$ such as

$$\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_{\mathrm{P}_n}\mathbf{W}_{\mathrm{P}_n}d\Omega+\int_{\Gamma_e}\mathbf{F}_k(\mathbf{U}_{\mathrm{P}_n})\mathbf{n}_k\mathbf{W}_{\mathrm{P}_n}d\Gamma-\int_{\Omega_e}\mathbf{F}_k(\mathbf{U}_{\mathrm{P}_n})\frac{\partial\mathbf{W}_{\mathrm{P}_n}}{\partial x_k}d\Omega=$$
$$\int_{\Gamma_e}G_k(\mathbf{U}_{\mathrm{P}_n})\mathbf{n}_k\mathbf{W}_{\mathrm{P}_n}d\Gamma-\int_{\Omega_e}G_k(\mathbf{U}_{\mathrm{P}_n})\frac{\partial\mathbf{W}_{\mathrm{P}_n}}{\partial x_k}d\Omega,\quad\forall\mathbf{W}_{\mathrm{P}_n}\in V_h^{\mathrm{P}_n} \tag{3.3}$$

where $\mathbf{U}_{\mathrm{Pn}}$ and $\mathbf{W}_{\mathrm{Pn}}$ are represented by piecewise-polynomial functions of degrees $\mathrm{P_n}$, 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 $\mathrm{P_n}$, this is then equivalent to the following system of N equations,

$$\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_{\mathrm{P}_n}B_i d\Omega+\int_{\Gamma_e}\mathbf{F}_k(\mathbf{U}_{\mathrm{P}_n})\mathbf{n}_k B_i d\Gamma-\int_{\Omega_e}\mathbf{F}_k(\mathbf{U}_{\mathrm{P}_n})\frac{\partial B_i}{\partial x_k}d\Omega=$$
$$\int_{\Gamma_e}G_k(\mathbf{U}_{\mathrm{P}_n})\mathbf{n}_k B_i d\Gamma-\int_{\Omega_e}G_k(\mathbf{U}_{\mathrm{P}_n})\frac{\partial B_i}{\partial x_k}d\Omega,\quad 1\le \mathrm{i}\le \mathrm{N} \tag{3.4}$$

where N is the dimension of the polynomial space. Since the numerical solution $\mathbf{U}_{\mathrm{Pn}}$ is discontinuous between element interfaces, the interface fluxes are not uniquely defined. The choice of these fluxes is crucial for the DG formulation. Like in the finite volume methods, the inviscid flux function $\mathbf{F}_k(\mathbf{U}_{\mathrm{Pn}})\mathbf{n}_k$ appearing in the boundary integral can be replaced by a numerical Riemann flux function $\mathbf{H}_{\mathrm{k}}(\mathbf{U}^{L}_{\mathrm{Pn}},\mathbf{U}^{R}_{\mathrm{Pn}},\mathbf{n}_k)$ where $\mathbf{U}_{\mathrm{Pn}}{}^{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{Pn}}{}^{\mathrm{R}}$ are the conservative state vector at the left and right side of the element boundary. The computation of the viscous fluxes in the boundary integral has to properly resolve the discontinuities at the interfaces. This scheme is called discontinuous Galerkin method of degree $\mathrm{P_n}$, or in short notation $\mathrm{DG(P_n)}$ 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 $\mathrm{DG(P_0)}$ method, i.e., to the discontinuous Galerkin method using a piecewise constant polynomial. Consequently, the $\mathrm{DG(P_k)}$ methods with $\mathrm{k>0}$ 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. (3.4) are calculated using Gauss quadrature formulas. The number of quadrature points used is chosen to integrate exactly polynomials of order of $2\mathrm{P_n}$ and $2\mathrm{P_n}+1$ in the reference element for the domain and surface integrals, respectively.

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

$$\mathbf{U}_{\mathrm{P}_n}=\sum_{i=1}^{\mathrm{N}}\mathbf{U}_i(t)B_i(x_k) \tag{3.5}$$

where B_{i} are the finite element basis functions. As a result, the unknowns to be solved are the variables at the nodes $\mathbf{U}_{\mathrm{i}}$, However, numerical polynomial solutions $\mathbf{U}$ can be expressed in other forms as well. 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 follows

$$\mathbf{U}_{\mathrm{P}_2}=\mathbf{U}_c+\left.\frac{\partial\mathbf{U}}{\partial x}\right|_c(x-x_c)+\left.\frac{\partial\mathbf{U}}{\partial y}\right|_c(y-y_c)+\left.\frac{\partial\mathbf{U}}{\partial z}\right|_c(z-z_c)+\left.\frac{\partial^2\mathbf{U}}{\partial x^2}\right|_c\frac{(x-x_c)^2}{2}+\left.\frac{\partial^2\mathbf{U}}{\partial y^2}\right|_c\frac{(y-y_c)^2}{2}$$
$$+\left.\frac{\partial^2\mathbf{U}}{\partial z^2}\right|_c\frac{(z-z_c)^2}{2}+\left.\frac{\partial^2\mathbf{U}}{\partial x\partial y}\right|_c(x-x_c)(y-y_c)+\left.\frac{\partial^2\mathbf{U}}{\partial x\partial z}\right|_c(x-x_c)(z-z_c)+\left.\frac{\partial^2\mathbf{U}}{\partial y\partial z}\right|_c(y-y_c)(z-z_c) \tag{3.6}$$

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

$$
\begin{aligned}
\mathbf{U}_{\mathrm{P}_2} &= \tilde{\mathbf{U}} + \left.\frac{\partial \mathbf{U}}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial \mathbf{U}}{\partial y}\right|_c (y-y_c) + \left.\frac{\partial \mathbf{U}}{\partial z}\right|_c (z-z_c) + \left.\frac{\partial^2 \mathbf{U}}{\partial x^2}\right|_c \left(\frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d\Omega\right) \\
&+ \left.\frac{\partial^2 \mathbf{U}}{\partial y^2}\right|_c \left(\frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d\Omega\right) + \left.\frac{\partial^2 \mathbf{U}}{\partial z^2}\right|_c \left(\frac{(z-z_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(z-z_c)^2}{2}d\Omega\right) \\
&+ \left.\frac{\partial^2 \mathbf{U}}{\partial x \partial y}\right|_c \left((x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d\Omega\right) \\
&+ \left.\frac{\partial^2 \mathbf{U}}{\partial x \partial z}\right|_c \left((x-x_c)(z-z_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(z-z_c)d\Omega\right) \\
&+ \left.\frac{\partial^2 \mathbf{U}}{\partial y \partial z}\right|_c \left((y-y_c)(z-z_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(y-y_c)(z-z_c)d\Omega\right)
\end{aligned}
\tag{3.7}
$$

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

$$
\begin{aligned}
&B_1 = 1, \quad B_2 = x - x_c, \quad B_3 = y - y_c, \quad B_4 = z - z_c \\
&B_5 = \frac{B_2^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_2^2}{2}d\Omega, \quad B_6 = \frac{B_3^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_3^2}{2}d\Omega, \quad B_7 = \frac{B_4^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_4^2}{2}d\Omega, \\
&B_8 = B_2B_3 - \frac{1}{\Omega_e}\int_{\Omega_e}B_2B_3d\Omega, \quad B_9 = B_2B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}B_2B_4d\Omega, \quad B_{10} = B_3B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}B_3B_4d\Omega
\end{aligned}
\tag{3.8}
$$

The discontinuous Galerkin formulation then leads to the following 10 equations

$$
\begin{aligned}
&\frac{d}{dt}\int_{\Omega_e}\tilde{\mathbf{U}}\,\mathbf{d}\Omega + \int_{\Gamma_e}\mathbf{F}_k(\mathbf{U}_{\mathrm{P}_2})\mathbf{n}_k d\Gamma = \int_{\Gamma_e}G_k(\mathbf{U}_{\mathrm{P}_2})\mathbf{n}_k d\Gamma \\
&M_{9\mathrm{x}9}\frac{d}{dt}\left(\frac{\partial \mathbf{U}}{\partial x}\Big|_c \quad \frac{\partial \mathbf{U}}{\partial y}\Big|_c \quad \frac{\partial \mathbf{U}}{\partial z}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial x^2}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial y^2}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial z^2}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial x \partial z}\Big|_c \quad \frac{\partial^2 \mathbf{U}}{\partial y \partial z}\Big|_c\right)^T + R_{9xl} = 0
\end{aligned}
\tag{3.9}
$$

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

$$
\int_{\Omega_e}B_1B_i d\Omega = 0 \qquad 2 \le i \le 10 \tag{3.10}
$$

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix (3.3) as follows:

$$
\begin{aligned}
&B_1 = 1, \quad B_2 = \frac{x - x_c}{\Delta x}, \quad B_3 = \frac{y - y_c}{\Delta y}, \quad B_4 = \frac{z - z_c}{\Delta z} \\
&B_5 = \frac{B_2^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_2^2}{2}d\Omega, \quad B_6 = \frac{B_3^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_3^2}{2}d\Omega, \quad B_7 = \frac{B_4^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{B_4^2}{2}d\Omega, \\
&B_8 = B_2B_3 - \frac{1}{\Omega_e}\int_{\Omega_e}B_2B_3d\Omega, \quad B_9 = B_2B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}B_2B_4d\Omega, \quad B_{10} = B_3B_4 - \frac{1}{\Omega_e}\int_{\Omega_e}B_3B_4d\Omega,
\end{aligned}
\tag{3.11}
$$

where $\Delta x=0.5(x_{max}-x_{min})$, $\Delta y=0.5(y_{max}-y_{min})$, 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. A quadratic polynomial solution can then be rewritten as

$$\mathbf{U}_{\mathrm{P}_2} = \tilde{\mathbf{U}}B_1 + \mathbf{U}_x B_2 + \mathbf{U}_y B_3 + \mathbf{U}_z B_4 + \mathbf{U}_{xx} B_5 + \mathbf{U}_{yy} B_6 + \mathbf{U}_{zz} B_7 + \mathbf{U}_{xy} B_8 + \mathbf{U}_{xz} B_9 + \mathbf{U}_{yz} B_{10} \tag{3.12}$$

where

$$\begin{aligned} &\mathbf{U}_{\mathbf{x}} = \frac{\partial \mathbf{U}}{\partial x}\Big|_c \Delta x, \mathbf{U}_y = \frac{\partial \mathbf{U}}{\partial y}\Big|_c \Delta y, \mathbf{U}_z = \frac{\partial \mathbf{U}}{\partial z}\Big|_c \Delta z, \mathbf{U}_{xx} = \frac{\partial^2 \mathbf{U}}{\partial x^2}\Big|_c \Delta x^2, \mathbf{U}_{yy} = \frac{\partial^2 \mathbf{U}}{\partial y^2}\Big|_c \Delta y^2, \\ &\mathbf{U}_{zz} = \frac{\partial^2 \mathbf{U}}{\partial z^2}\Big|_c \Delta z^2, \mathbf{U}_{xy} = \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\Big|_c \Delta x \Delta y, \mathbf{U}_{xz} = \frac{\partial^2 \mathbf{U}}{\partial x \partial z}\Big|_c \Delta x \Delta z, \mathbf{U}_{yz} = \frac{\partial^2 \mathbf{U}}{\partial y \partial z}\Big|_c \Delta y \Delta z \end{aligned} \tag{3.13}$$

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_nP_m scheme[18-20], termed rDG(P_nP_m) in this paper, which can be expressed as

$$\begin{aligned} &\frac{d}{dt}\int_{\Omega_e} \mathbf{U}_{\mathrm{P}_n} B_i d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}^{\mathrm{R}}_{\mathrm{P}_m}) \mathbf{n}_k B_i d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}^{\mathrm{R}}_{\mathrm{P}_m}) \frac{\partial B_i}{\partial x_k} d\Omega = \\ &\int_{\Gamma_e} G_k(\mathbf{U}^{\mathrm{R}}_{\mathrm{P}_m}) \mathbf{n}_k B_i d\Gamma - \int_{\Omega_e} G_k(\mathbf{U}^{\mathrm{R}}_{\mathrm{P}_m}) \frac{\partial B_i}{\partial x_k} d\Omega, \quad 1 \le \mathrm{i} \le \mathrm{N} \end{aligned} \tag{3.14}$$

where $\mathbf{U}_{\mathrm{Pn}}$ indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and $\mathbf{U}_{\mathrm{Pm}}$ represents a reconstructed polynomial solution of degree of m ($m \ge n$) that is used to compute the fluxes and source terms. The beauty of rDG(P_nP_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 rDG(P_nP_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, rDG(P_0P_m) is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \ge 1$) is reconstructed from a piecewise constant solution. When m=n, the reconstruction reduces to the identity operator, and rDG(P_nP_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 rDG(P_1P_2) method, as the benefits of higher-order (>3rd) 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 rDG(P_1P_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] on tetrahedral grids.

3.2.1.1 Least-squares reconstruction

In the case of DG(P_1) method, a linear polynomial solution $\mathbf{U}_i$ in any cell i is expressed as

$$\mathbf{U}_i = \widetilde{\mathbf{U}}_i + \mathbf{U}_{\mathbf{x}i} B_2 + \mathbf{U}_{\mathbf{y}i} B_3 + \mathbf{U}_{zi} B_4 \tag{3.15}$$

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

$$\mathbf{U}_i^R = \widetilde{\mathbf{U}}_i^R + \mathbf{U}_{xi}^R B_2 + \mathbf{U}_{yi}^R B_3 + \mathbf{U}_{zi}^R B_4 + \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 + \mathbf{U}_{xyi}^R B_8 + \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10} \tag{3.16}$$

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.,

$$\widetilde{\mathbf{U}}_i^R = \widetilde{\mathbf{U}}_i\,, \quad \mathbf{U}_{xi}^R = \mathbf{U}_{xi}\,, \quad \mathbf{U}_{yi}^R = \mathbf{U}_{yi}\,, \quad \mathbf{U}_{zi}^R = \mathbf{U}_{zi} \tag{3.17}$$

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

$$\begin{aligned}
&\mathbf{U}_j = \widetilde{\mathbf{U}}_i + \mathbf{U}_{xi} B_2 + \mathbf{U}_{yi} B_3 + \mathbf{U}_{zi} B_4 + \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 + \mathbf{U}_{xyi}^R B_8 + \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10} \\
&\frac{\partial \mathbf{U}}{\partial x}\Big|_j = \mathbf{U}_{xi} \frac{1}{\Delta x_i} + \mathbf{U}_{xxi}^R \frac{B_2}{\Delta x_i} + \mathbf{U}_{xyi}^R \frac{B_3}{\Delta x_i} + \mathbf{U}_{xzi}^R \frac{B_4}{\Delta x_i} \\
&\frac{\partial \mathbf{U}}{\partial y}\Big|_j = \mathbf{U}_{yi} \frac{1}{\Delta y_i} + \mathbf{U}_{yyi}^R \frac{B_3}{\Delta y_i} + \mathbf{U}_{xyi}^R \frac{B_2}{\Delta y_i} + \mathbf{U}_{yzi}^R \frac{B_4}{\Delta y_i} \\
&\frac{\partial \mathbf{U}}{\partial z}\Big|_j = \mathbf{U}_{zi} \frac{1}{\Delta z_i} + \mathbf{U}_{zzi}^R \frac{B_4}{\Delta z_i} + \mathbf{U}_{xzi}^R \frac{B_2}{\Delta z_i} + \mathbf{U}_{yzi}^R \frac{B_3}{\Delta z_i}
\end{aligned} \tag{3.18}$$

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

$$\begin{pmatrix} B_5^j & B_6^j & B_7^j & B_8^j & B_9^j & B_{10}^j \\ B_2^j & 0 & 0 & B_3^j & B_4^j & 0 \\ 0 & B_3^j & 0 & B_2^j & 0 & B_4^j \\ 0 & 0 & B_4^j & 0 & B_2^j & B_3^j \end{pmatrix} \begin{pmatrix} \mathbf{U}_{xxi}^R \\ \mathbf{U}_{yyi}^R \\ \mathbf{U}_{zzi}^R \\ \mathbf{U}_{xyi}^R \\ \mathbf{U}_{xzi}^R \\ \mathbf{U}_{yzi}^R \end{pmatrix} = \begin{pmatrix} \mathbf{U}_j - (\mathbf{U}_i B_1^j + \mathbf{U}_{xi} B_2^j + \mathbf{U}_{yi} B_3^j + \mathbf{U}_{zi} B_4^j) \\ \frac{\Delta x_i}{\Delta x_j} \mathbf{U}_{xj} - \mathbf{U}_{xi} \\ \frac{\Delta y_i}{\Delta y_j} \mathbf{U}_{yj} - \mathbf{U}_{yi} \\ \frac{\Delta z_i}{\Delta z_j} \mathbf{U}_{zj} - \mathbf{U}_{zi} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^j \\ \mathbf{R}_2^j \\ \mathbf{R}_3^j \\ \mathbf{R}_4^j \end{pmatrix} \tag{3.19}$$

where $\mathbf{R}$ is used to represent the right-hand-side for simplicity. 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 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

This least-squares reconstructed discontinuous Galerkin method: rDG(P_1P_2) has been successfully used to solve the 2D compressible Euler equations for smooth flows on arbitrary grids[26-28,30-31] and is able to achieve the designed third order of accuracy and significantly improve the accuracy of the underlying second-order DG method. However, when extended to solve the 3D compressible Euler equations on tetrahedral grids, this rDG method suffers from the so-called linear instability, that is also observed in the second-order cell-centered finite volume methods, i.e., rDG(P_0P_1)[33]. This linear instability is attributed to the fact that the reconstruction stencils only involve von Neumann neighborhood, i.e., adjacent face-neighboring cells[33]. The linear stability can be achieved using extended stencils, which will unfortunately sacrifice the compactness of the underlying DG methods. Furthermore, such a linear reconstruction-based DG method cannot maintain the non-linear instability, leading to non-physical oscillations in the vicinity of strong discontinuities. Alternatively, 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 introduced by Dumber et al.[40] 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,

$$\frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n}\Big|_i^{\text{WENO}} = \sum_{k=1}^{5} w_k \frac{\partial^2 \mathbf{U}}{\partial x_m \partial x_n}\Big|_k \tag{3.20}$$

where the normalized nonlinear weights w_k are computed as

$$w_k = \frac{\widetilde{w}_k}{\sum_{i=1}^{5} \widetilde{w}_k} \tag{3.21}$$

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

$$\widetilde{w}_i = \frac{\lambda_i}{(\varepsilon + o_i)^\gamma} \tag{3.22}$$

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

$$o_k = [(\frac{\partial^2 U}{\partial x_m \partial x_n}\Big|_k)^2]^{1/2} \tag{3.23}$$

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(P_1) method in Eq. (3.3). As demonstrated in Reference 36, the resulting rDG(P_1P_2) 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.

IV. 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(P_1P_2) method is able to achieve the designed third-order of accuracy for the compressible Navier-Stokes equations. The last four test cases are presented to illustrate that the rDG(P_1P_2) method can be applied to obtain reliable and accurate solutions for a variety of compressible flow problems, ranging from RANS, to LES, and to 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 both y- and z-directions to be 1. For two-dimensional problems, the number of cells in the z-direction is simply set to be 1. All computations use an explicit four-stage Runge–Kutta time-stepping scheme to advance the solution in time.

A. Heat Equation

The 1D heat equation $u_t = u_{xx}$ with periodic boundary conditions is first solved from x=0 to x=2π to study the convergence of the rDG methods for the discretization of diffusion operator. The analytical solution to this problem is $u(x,t) = e^{-t}\sin(x)$. The initial condition is $u(x,0)=\sin(x)$. 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. Numerical solutions to this problem are computed using rDG(P_1,P_1), rDG(P_1P_2), and rDG(P_2P_2) methods and using 4, 8, 16, and 32 cells to obtain quantitative measurement of the order of accuracy and discretization errors. The detailed results for this test case are presented in Tables 1a-c. They show the mesh size, the number of degrees of freedom, the L_2-error of the solutions, and the order of convergence. Figure 1 provides details of 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(P_1P_2) method does offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P_1) method. One can observe that higher order DG methods require significantly less degrees of freedom than lower order ones for the same accuracy. Moreover, the rDG(P_1P_2) outperforms the DG(P_2) in terms of both order of convergence and magnitude of discretization error by measuring the number of the degrees of freedom required to achieve the same accuracy.

Table 1a: Heat equation test case: rDG(P_1P_1) is order of $O(h^2)$

Mesh	No. DOFs	L2-error	Order
4	16	5.56570E-02	-
8	32	1.76807E-02	1.655
16	64	4.71857E-03	1.906
32	128	1.19953E-03	1.976

Table 1b: Heat equation test case: rDG(P_1P_2) is order of $O(h^3)$

Mesh	No. DOFs	L2-error	Order
4	16	2.53169E-02	-
8	32	1.89234E-03	3.742
16	64	1.24721E-04	3.924
32	128	8.35430E-06	3.900

Table 1c: Heat equation test case: rDG(P_2P_2) is order of $O(h^3)$

Mesh	No. DOFs	L2-error	Order
4	40	2.61198E-03	-
8	80	2.44596E-04	3.417
16	160	2.73943E-05	3.159
32	320	3.31413E-06	3.047

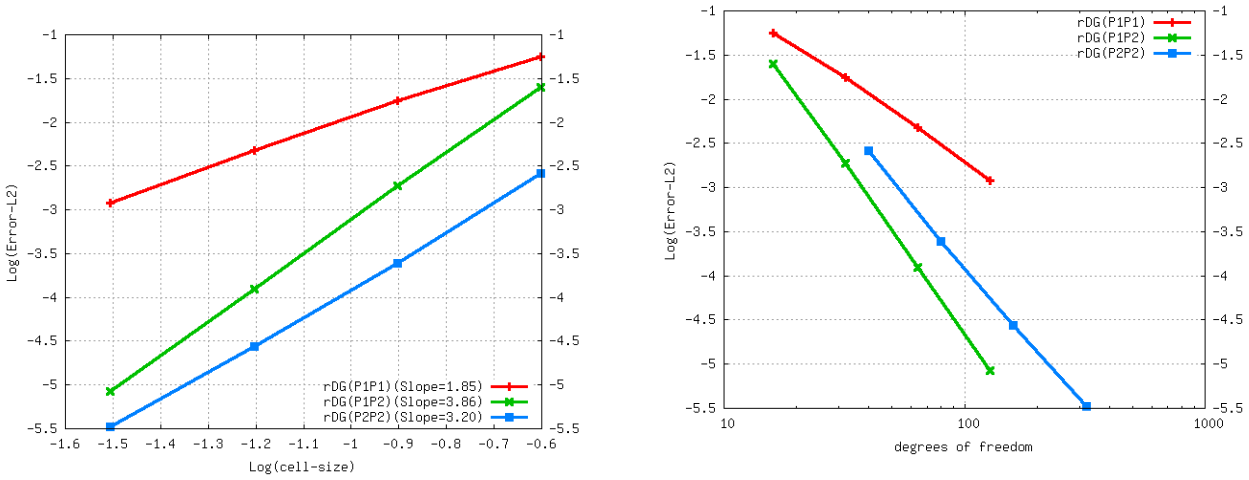


Figure 1. Convergence history for heat equation obtained by the rDG(P_1P_1), rDG(P_1P_2), and rDG(P_2P_2) methods.

B. Convection of an isentropic vortex in a viscous flow

The convection of a 2D isentropic vortex in a viscous flow at a Reynolds number of 1,000 is considered in this test case to conduct a convergence study of the rDG(P_1,P_1), rDG(P_1P_2), and rDG(P_2P_2) methods for solving the Navier-Stokes equations. The initial condition is a linear superposition of a mean uniform flow with some perturbations δ. The free stream flow conditions are (ρ_∞, u_∞, v_∞, p_∞) = (1,1,1,1). The perturbations of the velocity components u and v, entropy S, and temperature T for the vortex are given by

$$\begin{pmatrix}\delta u\\ \delta v\end{pmatrix} = \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix}-(y-y_0)\\ (x-x_0)\end{pmatrix}, \qquad \delta S = 0, \qquad \delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2},$$

where $r^2 = (x\text{-}x_0)^2 + (y\text{-}y_0)^2$, (x_0,y_0) is the coordinate of the vortex center, and ε is the vortex strength. From $\rho=\rho_\infty+\delta\rho$, $u=u_\infty+\delta u$, $v=v_\infty+\delta v$, $T=T_\infty+\delta T$, and the isentropic relation, other physical variables can be determined as follows

$$\rho = T^{\frac{1}{\gamma-1}} = (T_\infty + \delta T)^{\frac{1}{\gamma-1}} = \left[T_\infty - \frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}\right]^{\frac{1}{\gamma-1}},$$

$$\rho u = \rho(u_\infty + \delta u) = \rho\left[u_\infty - \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (y - y_0)\right],$$

$$\rho v = \rho(v_\infty + \delta v) = \rho\left[v_\infty + \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (x - x_0)\right],$$

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

In this test case, the vortex strength ε=5, and the coordinate of the vortex center (x_0,y_0) is (5,5). The computational domain Ω is [0,10]x[0,10] and the periodic boundary conditions are imposed. For viscous flows, there is no analytical solution. A reference solution is numerically obtained by a 6-th order finite difference scheme with 3rd order TVD Runge-Kutta scheme on a uniform Cartesian grid of 500x500. The numerical solutions are obtained after one period of time t=10, and compared with the reference solution. The follow L_2-norm

$$\left\|\rho^R - \rho^r\right\|_{L^2(\Omega)} = \left(\int_\Omega \left|\rho_p^R - \rho^r\right|^2 d\Omega\right)^{\frac{1}{2}},$$

is used to measure the error between the numerical and reference solutions, where ρ^R is the rDG solutions for the density and ρ^r is the reference one. Figure 2 shows three successively refined prismatic grids having 554, 2,216, and 8,864 elements, respectively. The number of faces in each space dimension is 16, 32, and 64, respectively. Figure 3 provides the details of the spatial convergence of the rDG(P_1P_1), rDG(P_1P_2), and rDG(P_2P_2) methods for this numerical experiment. As expected, the rDG(P_1P_2) method indeed offers a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P_1) method for the Navier-Stokes equations. Higher order DG methods require significantly less degrees of freedom than lower order ones for the same accuracy. Moreover, the rDG(P_1P_2) outperforms the DG(P_2) in terms of the order of convergence, and performs as well as the DG(P_2) by measuring the number of the degrees of freedom required to achieve the same accuracy.

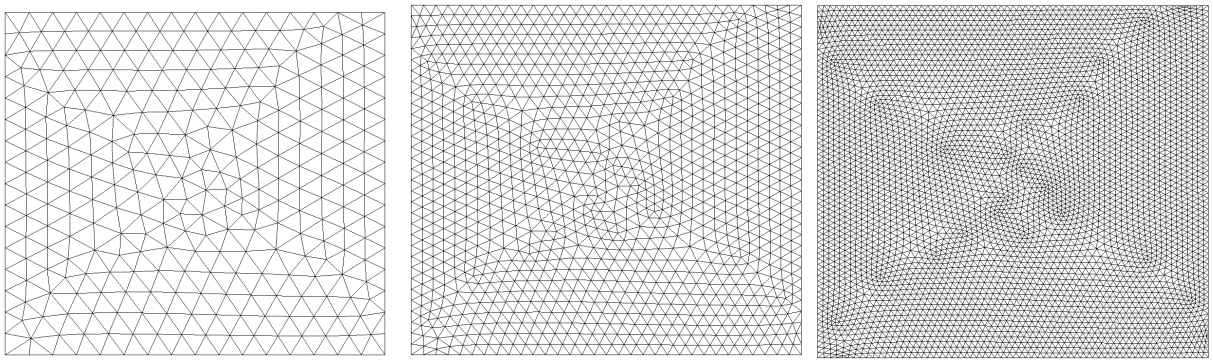


Figure 2. Sequence of triangular grids used for computing the convection of an isentropic vortex.

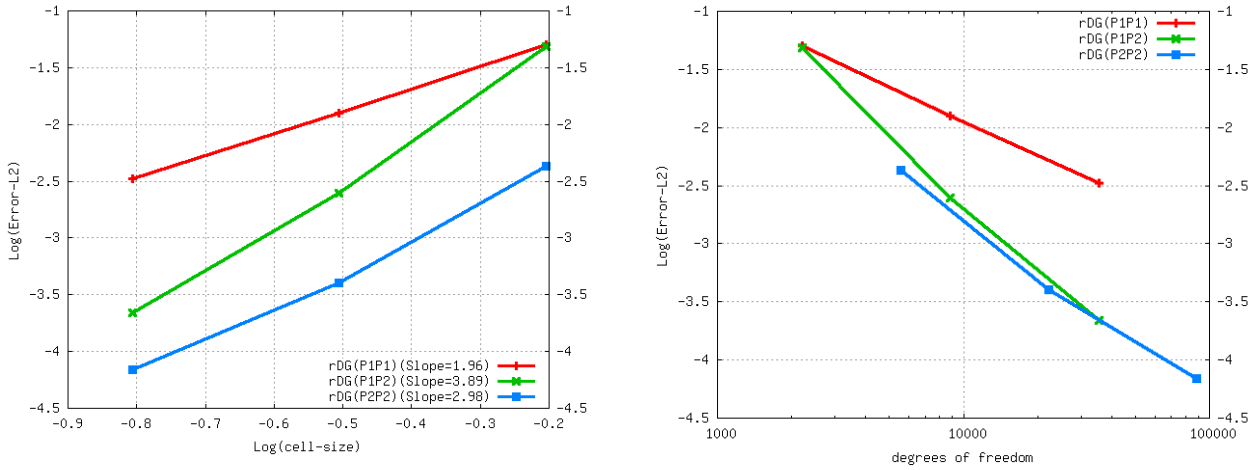


Figure 3. Convergence history of the rDG(P_1P_1), rDG(P_1P_2), and rDG(P_2P_2) methods for the convection of an isentropic vortex in a viscous flow.

C. Subsonic viscous flow past a sphere

This test case involves a subsonic viscous flow past a sphere at a Mach number of 0.3, and an angle of attack 0^o, and a Reynolds number of 118 based on the freestream velocity and the diameter of the sphere, which has been studied both experimentally[37] and numerically[38]. An adiabatic wall is assumed in this test case. Note that only a half of the configuration is modeled, due the symmetry of the problem. Figure 4 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 velocity contours in the flow field obtained by the rDG(P_1P_2) method. The computational streamlines are compared with experimental streamlines in Figure 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.

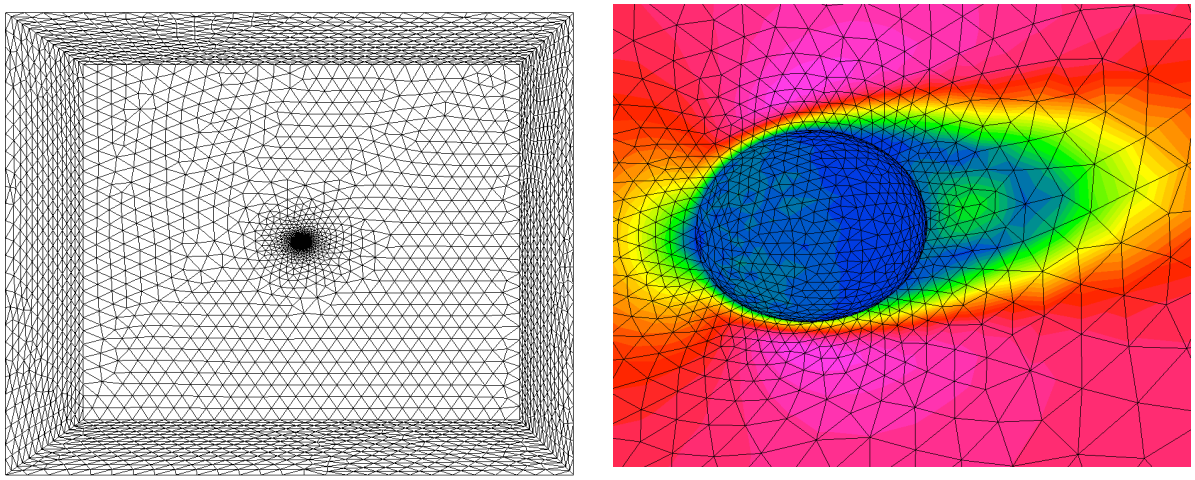


Figure 4. Unstructured mesh (left) (nelem=119,390, npoin=22,530, nboun=4,511) and computed Mach number contours by the rDG(P_1P_2) (right) for subsonic flow past a sphere at M_∞= 0.3, Re=118, $\alpha=0^o$.

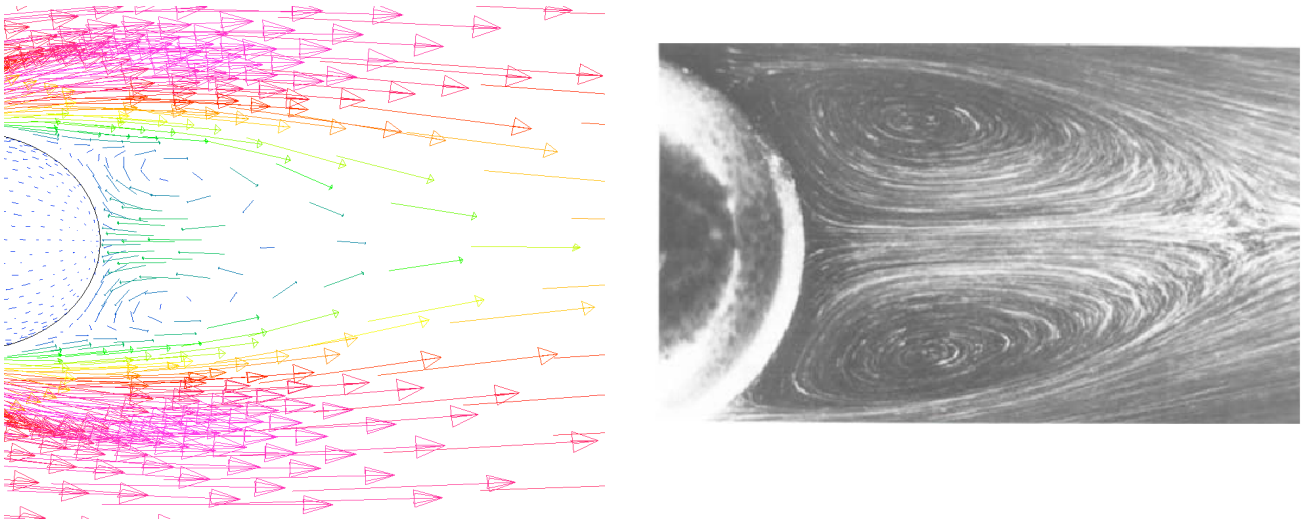
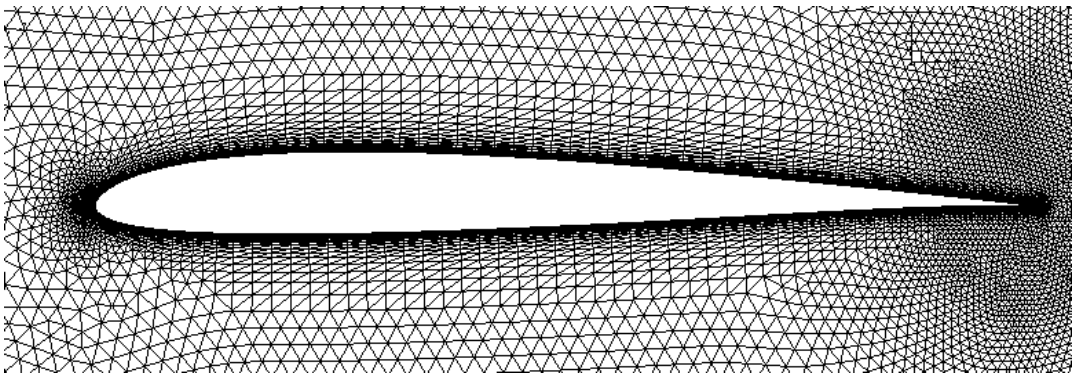


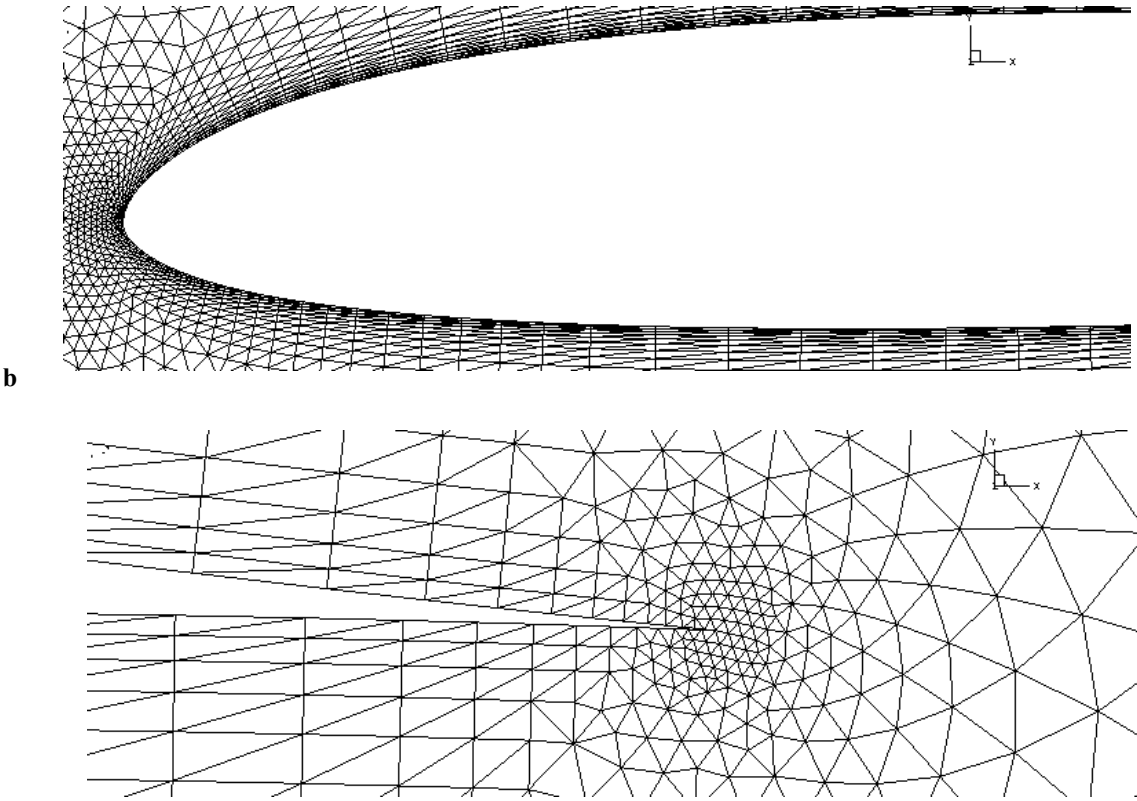
Figure 4. Velocity vector plot in the back of the sphere from the computation (left) and streamlines of the flow filed from experiment (right) for subsonic flow past a sphere at M_∞= 0.3, Re=118, $\alpha=0^o$.

D. Viscous flow past an SD7003 airfoil

A viscous flow past an SD7003 airfoil at a free-stream Mach number of $M_\infty = 0.1$, 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. 5(a)–5(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(P_1P_2) 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[39, 40] in Figs. 6(a) and 6(b), while the comparison for the vorticity magnitude contours between the rDG(P_1P_2) method and the compact method is shown in Figs. 7(a) and 7(b). In addition, the computed entropy contours obtained by the IRK3 and rDG(P_1P_2) method is displayed in 7(c). 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 tailing 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. 8(a) to illustrate that accurate and smooth solutions are obtained using the rDG(P_1P_2) method in spite of the highly stretched grid used in the boundary layer. Fig. 8(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.



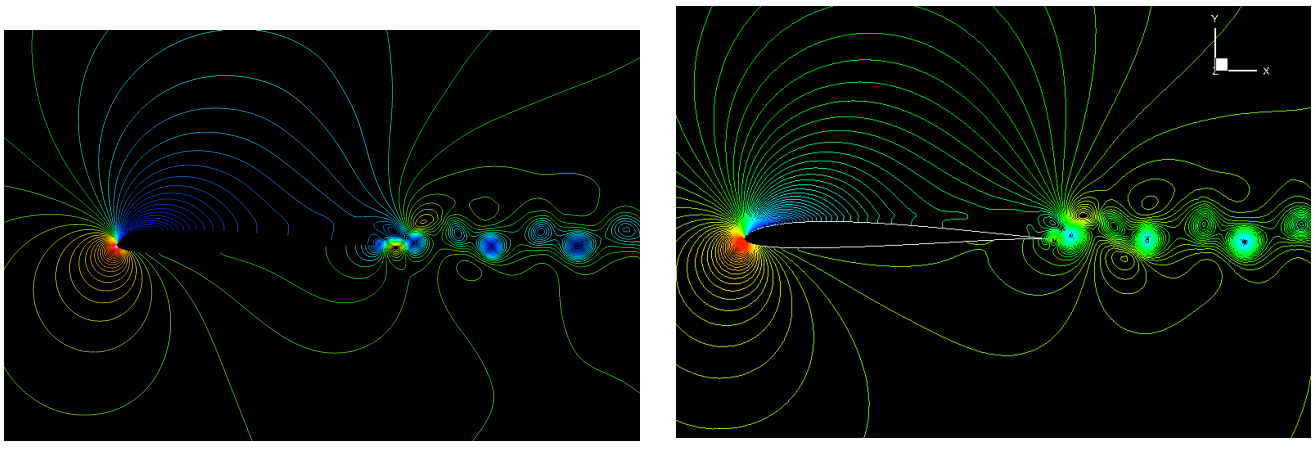
a



b

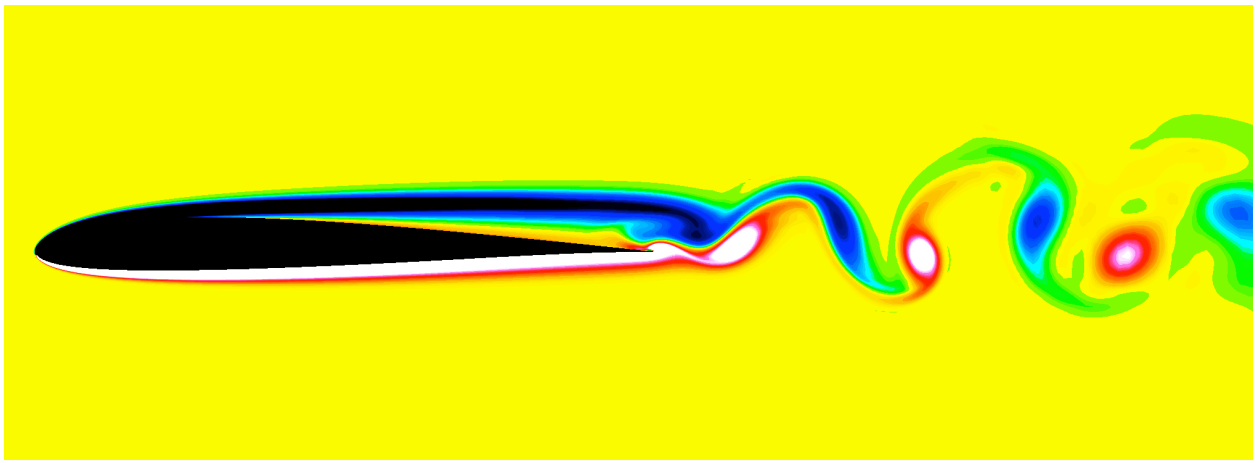
c

Figure 5. The prismatic grid used for computing a viscous flow past an SD7003 airfoil: (a) global view, (b) leading edge, and (c) trailing edge

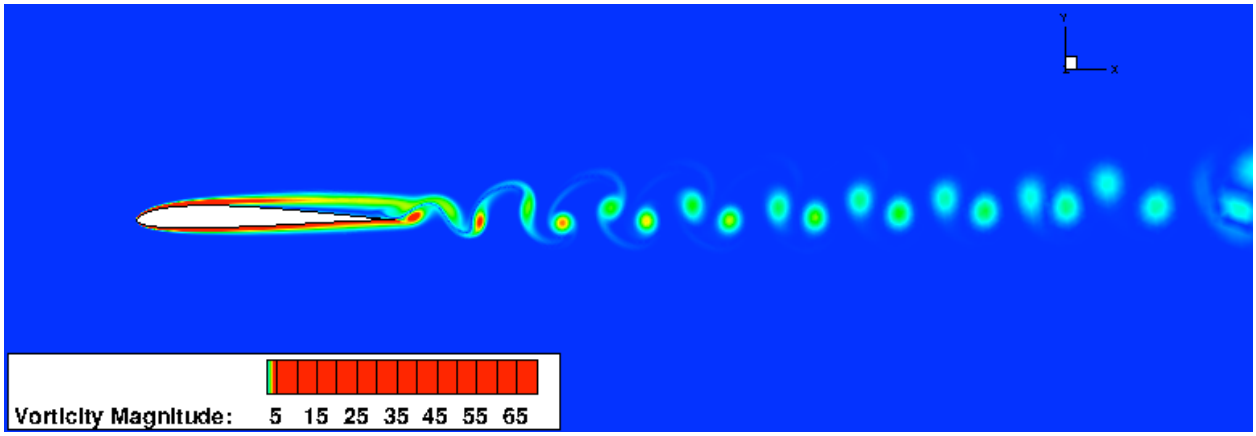


(a) **(b)**

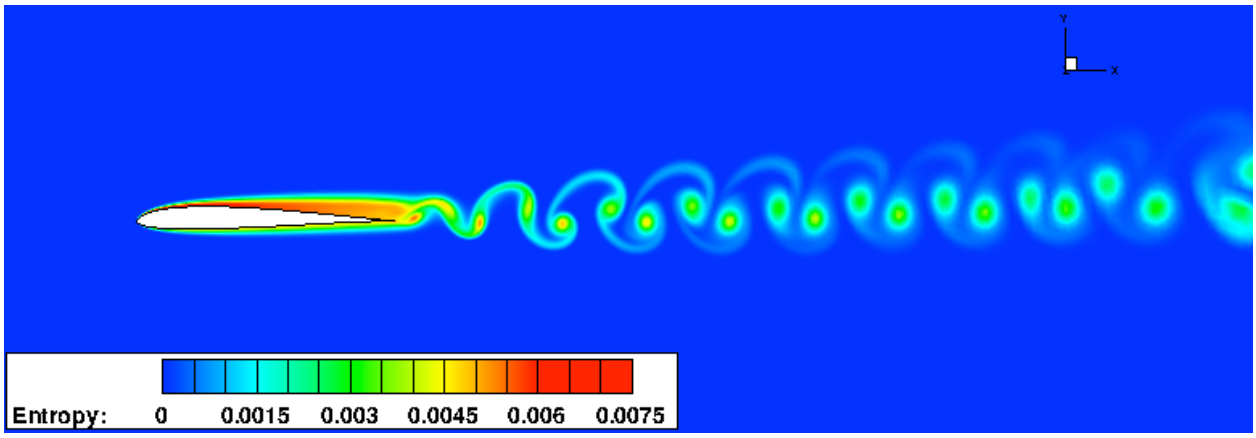
Figure 6. Computed pressure contours obtained by (a) the compact method, and (b) the IRK3 and the rDG(P_1P_2) method in the flow field for the viscous flow past an SD7003 airfoil at a free-stream Mach number of $M_\infty = 0.1$, an angle of attach of $\alpha = 4^\circ$, and a Reynolds number of Re = 10, 000.



(a)

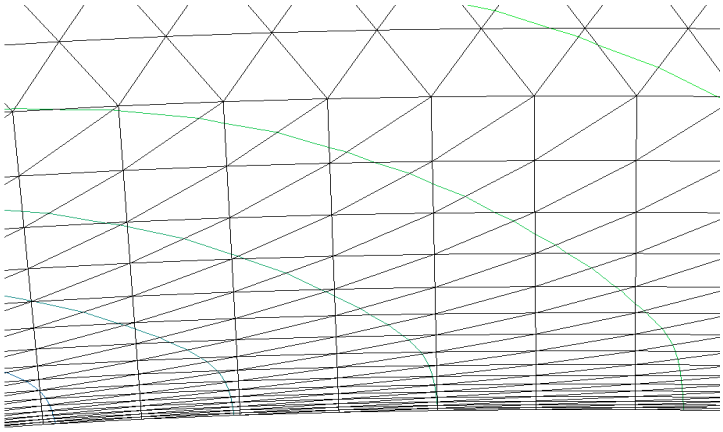


(b)

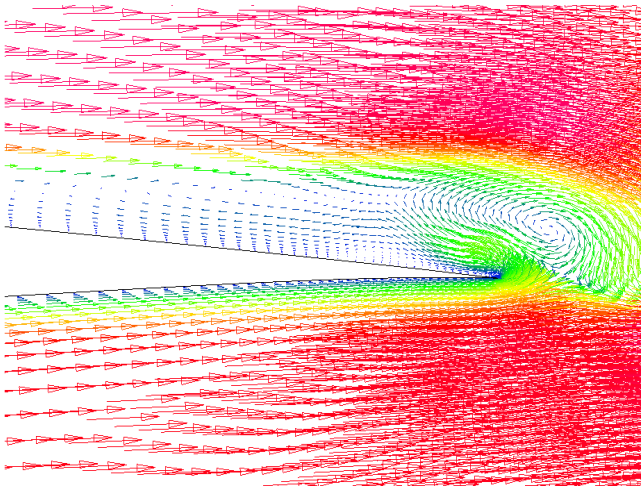


(c)

Figure 7: Computed vorticity contours obtained by (a) the compact method, and (b) the IRK3 and rDG(P_1P_2) method, and (c) computed entropy contours by the IRK3 and rDG(P_1P_2) method in the flow field for the viscous flow past an SD7003 airfoil at a free-stream Mach number of $M_\infty = 0.1$, an angle of attach of $\alpha = 4^\circ$, and a Reynolds number of Re = 10, 000.



(a)



(b)

Figure 8. 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.

E. Lid driven cavity flow in 3D

A transitional flow in a three-dimensional lid driven cavity at Re_∞ = 10, 000 is considered in this test case to assess the ability of the rDG method for large Eddy Simulation (LES) of turbulent flows, as the accuracy of the numerical results obtained can be assessed through comparison with both the experimental data[41] and the LES data[42]. 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 × 64 × 32 grid points is used in computation, as shown in Figure 9. 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 Figures 10-12, 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 Figure 10, 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(P_1P_2) method are in good agreement with the experimental data and LES data, demonstrating the ability of the rDG(P_1P_2) method for the accurate large eddy simulation of turbulent flows.

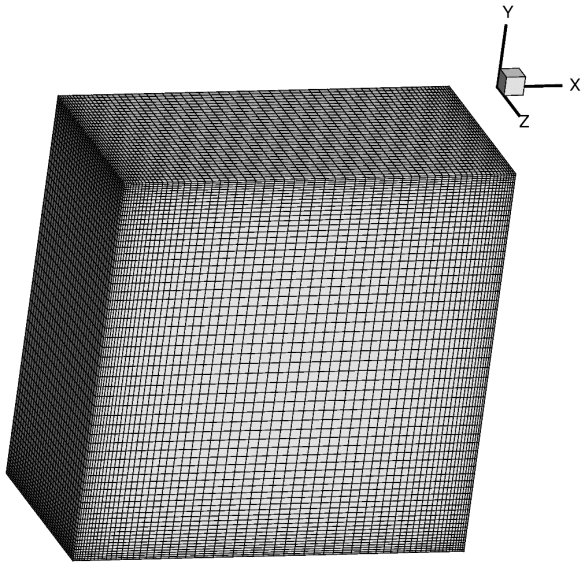


Figure 9. Hexahedral grid (64x64x32 points) used for the LES of a lid driven cavity (x:y:z=1:1:0.5) at Re=10,000.

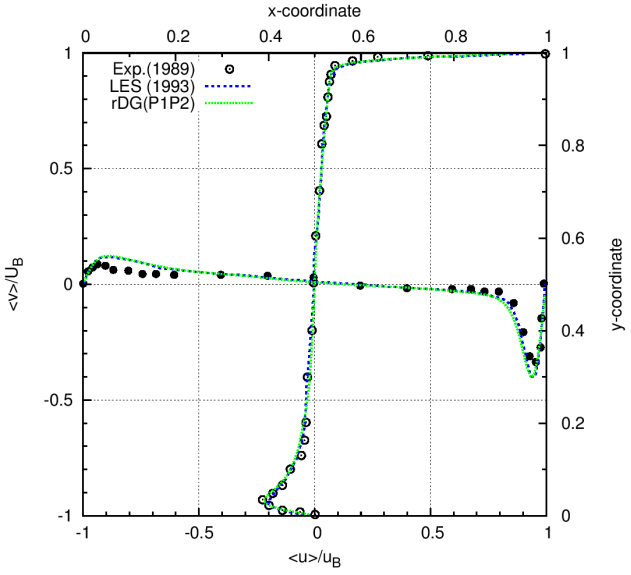


Figure 10. Comparison of the normalized mean velocity components <u>/u_B and <v>/u_B in the span-wise mid-plane with the classical experimental data by Prasad & Koseff and numerical data by Zang et al, for the LES of a lid driven cavity (x:y:z=1:1:0.5) at Re=10,000.

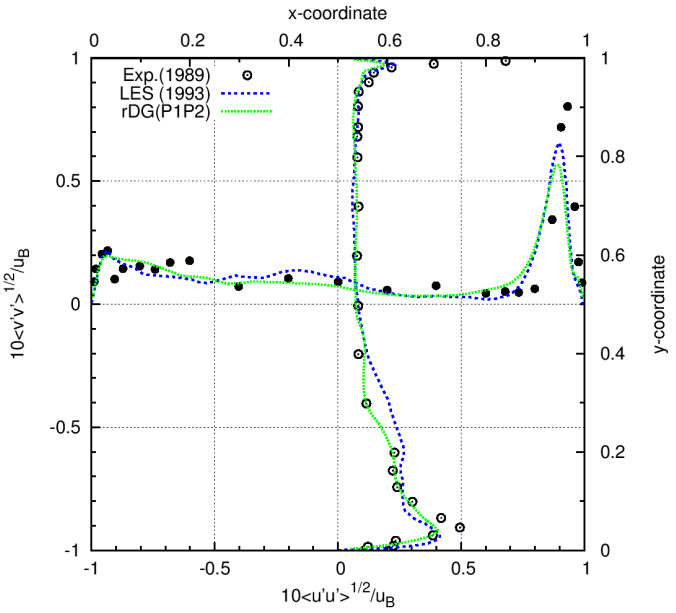


Figure 11. Comparison of the scaled RMS velocity components $10\langle u'u'\rangle^{0.5}/u_B$ and $10\langle v'v'\rangle^{0.5}/u_B$ in the span-wise midplane with the classical experimental data by Prasad & Koseff (1989) and numerical data by Zang et al, for the LES of a lid driven cavity (x:y:z=1:1:0.5) at Re=10,000.

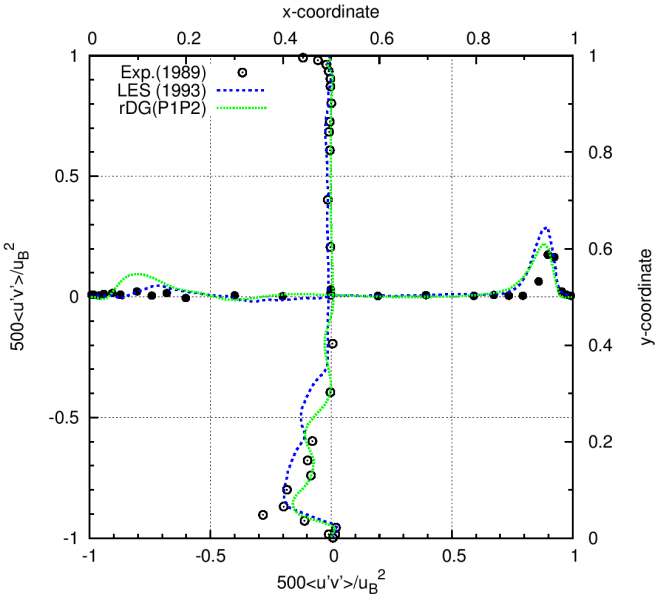


Figure 12. Comparison of the scaled mean Reynolds stress tensor components $500\langle u'v'\rangle/u^2_B$ in the span-wise mid-plane with the classical experimental data by Prasad & Koseff and numerical data by Zang et al, for the LES of a lid driven cavity (x:y:z=1:1:0.5) at Re=10,000.

F. 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(P_1P_2) 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:

$$u_1 = V_0 \sin(\frac{x}{L})\cos(\frac{y}{L})\cos(\frac{z}{L})$$

$$u_2 = -V_0 \cos(\frac{x}{L})\sin(\frac{y}{L})\cos(\frac{z}{L})$$

$$u_3 = 0$$

$$p = p_0 + \rho_0 V_0^2[\frac{1}{16}(\cos(\frac{2x}{L}) + \cos(\frac{2y}{L}))(\cos(\frac{2z}{L}) + 2)]$$

where ρ_0=1, p_0=1/γ, 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 [-πL, πL] 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., $t_{final} = 20t_c$.

A grid convergence study to this problem is performed using the rDG(P_1P_2) method on three hexahedral meshes, which have 40^3, 81^3, and 161^3 elements with 256,000, 2,125,764, and 16,693,124 degrees of freedom, respectively. Figure 13 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(P_1P_2) solution looks very similar to the one from Reference 43.

Figures 14 and 15 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 $\epsilon = -\frac{dE_k}{dt}$ computed from the data at the space-time quadrature points, respectively, for all three grids with the result from an incompressible simulation using a spectral code on a mesh of 512^3 grid points[44]. 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(P_1P_2) solution on the finest mesh agree very well with the ones from the spectral code solution.

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 $\epsilon=\epsilon_1+\epsilon_2+\epsilon_3=-\frac{dE_k}{dt}$:

$$\epsilon_1 = \frac{1}{\Omega}\int 2\mu s_{ij}\, s_{ij} d\Omega$$

$$\epsilon_2 = \frac{1}{\Omega}\int \mu_v u_{kk}\, u_{kk} d\Omega$$

$$\epsilon_3 = -\frac{1}{\Omega}\int p u_{kk}\, d\Omega$$

where $s_{ij} = \frac{1}{2}(u_{ij} + u_{ji})$ is the strain-rate tensor. In this case, the gas is assumed to have zero bulk viscosity, μ_v=0. Therefore, the dissipation due to the bulk viscosity is always equal to zero, meaning that ϵ_2=0. Since the flow is nearly incompressible, the dissipation due to the pressure-dilatation term (ϵ_3) can be expected to be small. The kinetic energy dissipation rate is then approximately equal to $\epsilon=\epsilon_1$. However, for the compressible simulation this does not hold exactly. Time histories of the computed ϵ, ϵ_1, and ϵ_3 on the finest mesh are presented in Figure 16. The pressure-dilatation, ϵ_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 Figure 17. Compressibility effects are evident in oscillations of the pressure dilatation term (ϵ_3).

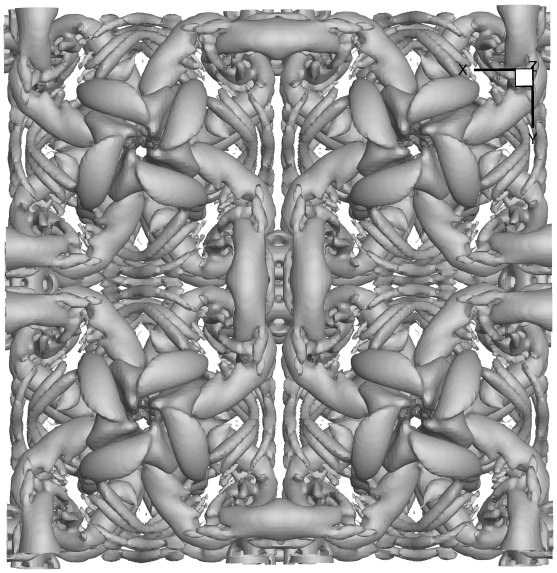


Figure 13. Iso-surfaces of the computed vortex detection criterion Q at time $t=8t_c$ by the rDG(P_1P_2) solution on the finest mesh.

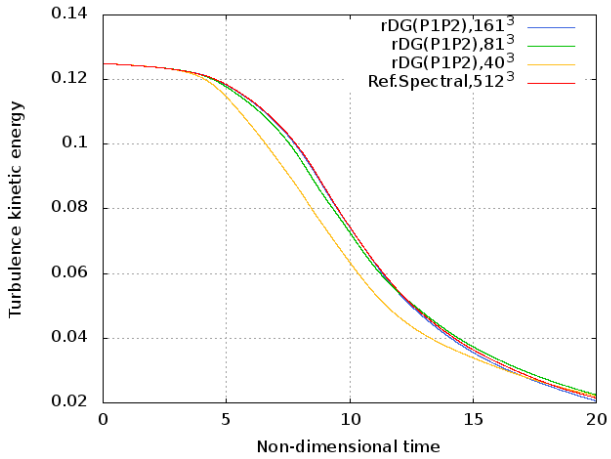


Figure 14. Evolution of the dimensionless kinetic energy as a function of the dimensionless time.

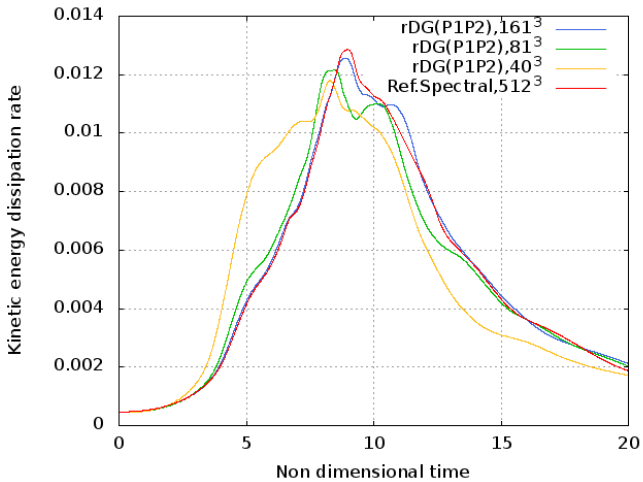


Figure 15. Evolution of the dimensionless kinetic energy dissipation rate as a function of the dimensionless time.

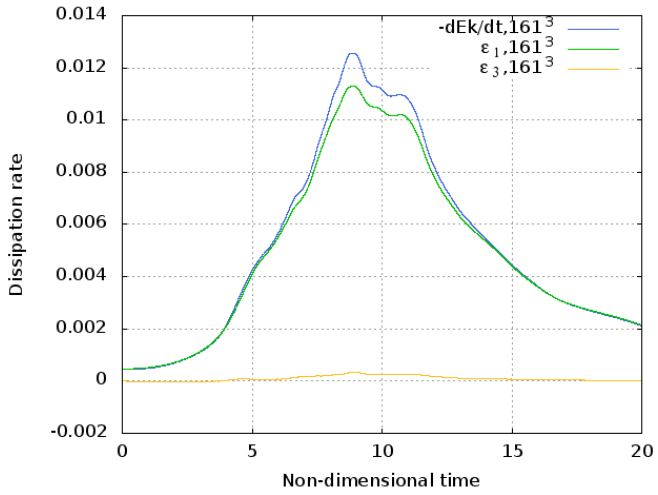


Figure 16. Kinetic energy dissipation balance for the Taylor-Green vortex problem on the finest mesh.

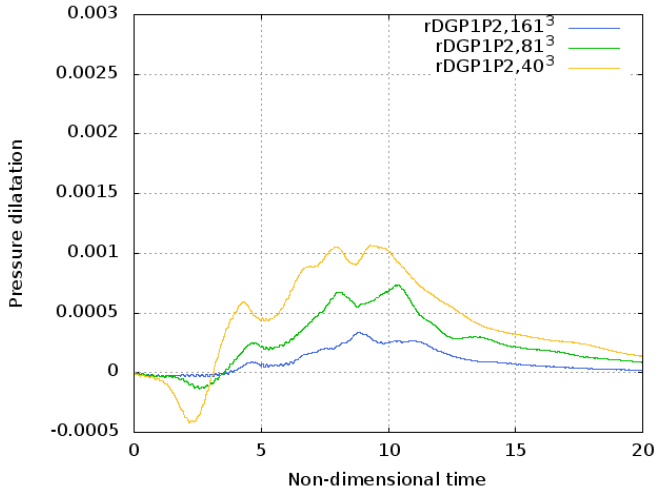


Figure 17. Biased pressure-dilatation term for the Taylor-Green vortex problem, with mesh refinement.

V. 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 BRII scheme. The developed rDG(P_1P_2) 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(P_1P_2) 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 large eddy simulation and 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, W.H. Reed and T.R. Hill, ``Triangular Mesh Methods for the Neutron Transport Equation,'' **Los Alamos Scientific Laboratory Report**, LA-UR-73-479, 1973.

[2]B. Cockburn, S. Hou, and C. W. Shu, ``TVD Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for conservation laws IV: the Multidimensional Case,'' **Mathematics of Computation**, Vol. 55, pp. 545-581, 1990.eorge, P. L., Automatic Mesh Generation, J. Wiley & Sons, 1991.

[3]B. Cockburn, and C. W. Shu, ``The Runge-Kutta Discontinuous Galerkin Method for conservation laws V: Multidimensional System,'' **Journal of Computational Physics**, Vol. 141, pp. 199-224, 1998.

[4]B. Cockburn, G. Karniadakis, and C. W. Shu, ``The Development of Discontinuous Galerkin Method'', in Discontinuous Galerkin Methods, Theory, Computation, and Applications, edited by B. Cockburn, G.E. Karniadakis, and C. W. Shu, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2000, Vol. 11 pp. 5-50, 2000.

[5]F. Bassi and S. Rebay, ``High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations,'' **Journal of Computational Physics**, Vol. 138, pp. 251-285, 1997.

[6]H. L. Atkins and C. W. Shu, ``Quadrature Free Implementation of Discontinuous Galerkin Method for Hyperbolic Equations,'' **AIAA Journal**, Vol. 36, No. 5, 1998.

[7]F. Bassi and S. Rebay, ``GMRES discontinuous Galerkin solution of the Compressible Navier-Stokes Equations,'' **Discontinuous Galerkin Methods, Theory, Computation, and Applications**, edited by B. Cockburn, G.E. Karniadakis, and C. W. Shu, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2000, Vol. 11 pp. 197-208, 2000.

[8]T. C. Warburton, and G. E. Karniadakis, ``A Discontinuous Galerkin Method for the Viscous MHD Equations,'' **Journal of Computational Physics**, Vol. 152, pp. 608-641, 1999.

[9]J. S. Hesthaven and T. Warburton, ``Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications,'' Texts in **Applied Mathematics**, Vol. 56, 2008.

[10]P. Rasetarinera and M. Y. Hussaini, ``An Efficient Implicit Discontinuous Spectral Galerkin Method,'' **Journal of Computational Physics**, Vol. 172, pp. 718-738, 2001.

[11]B. T. Helenbrook, D. Mavriplis, and H. L. Atkins, ``Analysis of *p*-Multigrid for Continuous and Discontinuous Finite Element Discretizations,'' AIAA Paper 2003-3989, 2003.

[12]K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal, ``*p*-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations,'' **Journal of Computational Physics**, Vol. 207, No. 1, pp. 92-113, 2005.

[13]H. Luo, J. D. Baum, and R. Löhner, ``A Discontinuous Galerkin Method Using Taylor Basis for Compressible Flows on Arbitrary Grids,'' **Journal of Computational Physics**, DOI: 210.1016/j.jcp.2008.06.035, Vol. 227, No 20, pp. 8875-8893, October 2008.

[14]H. Luo, J.D. Baum, and R. Löhner, ``On the Computation of Steady-State Compressible Flows Using a Discontinuous Galerkin Method'', **International Journal for Numerical Methods in Engineering,** Vol. 73, No. 5, pp. 597-623, 2008.

[15]H. Luo, J. D. Baum, and R. Löhner, ``A Hermite WENO-based Limiter for Discontinuous Galerkin Method on Unstructured Grids,'' **Journal of Computational Physics**, Vol. 225, No. 1, pp. 686-713, 2007.

[16]H. Luo, J.D. Baum, and R. Löhner, ``A *p*-Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids'', **Journal of Computational Physics**, Vol. 211, No. 2, pp. 767-783, 2006.

[17]H. Luo, J.D. Baum, and R. Löhner, ``A Fast, *p*-Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds'', **AIAA Journal** , Vol. 46, No. 3, pp.635-652, 2008.

[18]M. Dumbser, D.S. Balsara, E.F. Toro, C.D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. **Journal of Computational Physics**, 227:8209-8253, 2008.

[19]M. Dumbser, O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. **Journal of Computational Physics**, 228:6991-7006, 2009.

[20]M. Dumbser. Arbitrary High Order PNPM Schemes on Unstructured Meshes for the Compressible Navier-Stokes Equations. **Computers & Fluids**, 39: 60-76. 2010.

[21]F. Bassi and S. Rebay, ``A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations,'' **Journal of Computational Physics**, Vol. 131, pp. 267-279, 1997.

[22]F. Bassi and S. Rebay, ``Discontinuous Galerkin Solution of the Reynolds-Averaged Navier-Stokes and k-ω Turbulence Model Equations, **Journal of Computational Physics**, Vol. 34, pp. 507-540, 2005.

[23]B. Cockburn and C.W. Shu, ``The Local Discontinuous Galerkin Method for Time-dependent Convection-Diffusion System, **SIAM, Journal of Numerical Analysis**, Vo. 16, 2001.

[24]C. E. Baumann and J. T. Oden, ``A Discontinuous *hp* Finite Element Method for the Euler and Navier-Stokes Equations,'' **International Journal for Numerical Methods in Fluids**, Vol. 31, 1999.

[25]H. Luo, L. Luo, and K. Xu - A Discontinuous Galerkin Method Based on a BGK Scheme for the Navier-Stokes Equations on Arbitrary Grids, **Advances in Applied Mathematics and Mechanics**, Vol. 1, No. 3, pp. 301-318, 2009.

[26]H. Luo, L. Luo, and R. Nourgaliev, A Reconstructed Discontinuous Galerkin Method for the Euler Equations on Arbitrary Grids, **Communication in Computational Physics,** Vol. 12, No. 5, pp. 1495-1519, 2012.

[27]H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, and N. Dinh, ``A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations on Arbitrary Grids", **Journal of Computational Physics**, Vol. 229, pp. 6961-6978, 2010.

[28]H. Luo, L. Luo, A. Ali, R. Nourgaliev, and C. Cai, ``A Parallel, Reconstructed Discontinuous Galerkin Method for the Compressible Flows on Arbitrary Grids", **Communication in Computational Physics**, Vol. 9, No. 2, pp. 363-389, 2011.

[29]L.P. Zhang, W. Liu, L.X. He, X.G. Deng, and H.X. Zhang, A Class of Hybrid DG/FV Methods for Conservation Laws I: Basic Formulation and One-Dimensional System, **Journal of Computational Physics,** Vol. 23, No. 4, pp. 1081-1103, 2012.

[30] L.P. Zhang, W. Liu, L.X. He, X.G. Deng, and H.X. Zhang, A Class of Hybrid DG/FV Methods for Conservation Laws II: Two dimensional Cases, **Journal of Computational Physics,** Vol. 23, No. 4, pp. 1104-1120, 2012.

[31]H. Luo, Y. Xia, R. Nourgaliev, and C. Cai, A Class of Reconstructed discontinuous Galerkin Methods for the Compressible Flows on Arbitrary Grids, AIAA-2011-0199, 2011.

[32]H. Luo, H. Xiao, R. Nourgaliev, and C. Cai, A Comparative Study of Different Reconstruction Schemes for Reconstructed Discontinuous Galerkin Methods for the Compressible Flows on Arbitrary Grids, AIAA-2011-3839, 2011.

[33]D. F. Haider, J.P. Croisille, and B. Courbet, Stability Analysis of the Cell Centered Finite-Volume MUSCL Method on Unstructured Grids, **Numirische Mathematik**, Vol. 113, No. 4 pp. 555-600, 2009.

[34]D. Balsara, C. Altmann, C.D. Munz and M. Dumbser, A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG + HWENO schemes, **Journal of Computational Physics**, Vol. **226**, pp. 586–620, 2007.

[35]H. Luo, H., Y. Xia, S. Li, R. Nourgaliev, and C. Cai, A Hermite WENO Reconstruction-Based Discontinuous Galerkin Method for the Euler Equations on Tetrahedral Grids, **Journal of Computational Physics**, Vol. 231, pp. 5489-5503, 2012.

[36]H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, Z. Jiang, A Reconstructed Discontinuous Galerkin Method Based on a Hierarchical WENO Reconstruction for Compressible Flows on Tetrahedral Grids, **Journal of Computational Physics**, Vol. 236, pp. 477-492, 2013

[37]S. Taneda, ``Experimental Investigations of the wake behind a sphere at low Reynolds numbers, J. Phys. Soc. Jpn., 11 (1956), pp 1104-1108.

[38]Y. Sun, Z.J. Wang, and Y. Liu, High-Order Multidomain Spectral Difference Method for the Navier-Stokes Equations on Unstructured Hexahedral Grids, **Communications in Computational Physics**, Vol. 2, No., 2, pp. 310,333, 2007.

[39]R. E. Gordnier and M. R. Visbal, Compact Difference Scheme Applied to Simulation of Low-Sweep Delta Wing Flow, **AIAA Journal**, Vol. 43, No. 8, 00.1744-1752, 2005.

[40]M. R. Visbal and D. V. Gaitonde, On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes, **Journal of Computational Physics**, Vol. 181, No. 1, pp. 155-185, 2002.

[41]A.K. Prasad and J. Kose, Reynolds Number and End-wall Effects on a Lid-driven Cavity Flow. **Phys Fluids A**, Vol. 1, pp. 208-218, 1989.

[42]Y. Zang, R.L.S., Kose_, J.R.. A Dynamic Mixed Subgrid-scale Model and its Application to Turbulent Recirculating Flows, **Phys Fluids A**, Vol. 5, pp. 3186-3196, 1993.

[43]J. R. Bull and A. Jameson, Simulation of the Compressible Taylor Green Vortex using High-Order Flux Reconstruction Schemes. AIAA Paper, 2014-3210, 7th AIAA Theoretical Fluid Mechanics Conference, 16-20 June 2014, Atlanta, GA.

[44]W. van Ress, A. Leonard, D. Pullin, and P. Koumoutsakos., A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds number, **Journal of Computational Physics**, Vol. 230, pp. 2794-2805, 2011.