Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0090

Vol. **21**, No. 5, pp. 1231-1257
May 2017

A Parallel, High-Order Direct Discontinuous Galerkin Method for the Navier-Stokes Equations on 3D Hybrid Grids

Jian Cheng[1,*], Xiaodong Liu[2], Tiegang Liu[1] and Hong Luo[2]

[1] *School of Mathematics and Systems Science, Beihang University, Beijing 100191, P.R. China.*
[2] *Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, 27695, USA.*

Received 20 June 2016; Accepted (in revised version) 18 October 2016

Abstract. A parallel, high-order direct Discontinuous Galerkin (DDG) method has been developed for solving the three dimensional compressible Navier-Stokes equations on 3D hybrid grids. The most distinguishing and attractive feature of DDG method lies in its simplicity in formulation and efficiency in computational cost. The formulation of the DDG discretization for 3D Navier-Stokes equations is detailed studied and the definition of characteristic length is also carefully examined and evaluated based on 3D hybrid grids. Accuracy studies are performed to numerically verify the order of accuracy using flow problems with analytical solutions. The capability in handling curved boundary geometry is also demonstrated. Furthermore, an SPMD (single program, multiple data) programming paradigm based on MPI is proposed to achieve parallelism. The numerical results obtained indicate that the DDG method can achieve the designed order of accuracy and is able to deliver comparable results as the widely used BR2 scheme, clearly demonstrating that the DDG method provides an attractive alternative for solving the 3D compressible Navier-Stokes equations.

AMS subject classifications: 65M60, 65M99, 35L65

Key words: Direct discontinuous Galerkin method, compressible Navier-Stokes equations, hybrid grids.

1 Introduction

Discontinuous Galerkin (DG) methods [1–4], as a typical representative in the community of high order methods, have been widely used in computational fluid dynamics,

*Corresponding author. *Email addresses:* chengjian@buaa.edu.cn (J. Cheng), xliu29@ncsu.edu (X. Liu), liutg@buaa.edu.cn (T. Liu), hong_luo@ncsu.edu (H. Luo)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

computational acoustics, and computational magneto-hydrodynamics. The reason why DG methods have been intensively studied and widely applied is because the various attractive features they possess: (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, thus, 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 by the finite volume schemes.

DG methods are indeed a natural choice for solving hyperbolic equations, such as the compressible Euler equations. However, the DG formulation is far less certain and advantageous for diffusion problems, such as the compressible Navier-Stokes equations, where viscous and heat fluxes exist and require the evaluation of the solution derivatives at the cell interface. Taking a simple arithmetic mean of the solution derivatives from the left and right is inconsistent, because it does not take into account the underlying jumps of DG solutions at the cell interface. A number of methods have been proposed in the literature to address this issue. Several of them are listed in the following table.

Table 1: Different discontinuous Galerkin methods for diffusion problems.

Method	Developed by
interior penalty(IP)	Arnold et al.
symmetric IP method(SIP)	Hartmann et al.
local DG method(LDG)	Cockburn and Shu
compact DG method(CDG)	Peraire and Persson
hybridizable DG method(HDG)	Cockburn et al.
Bassi-Rebay method(BR1/BR2)	Bassi and Rebay
reconstructed DG(rDG)	Luo et al.
recovery DG(RDG)	van Leer et al.
direct DG method(DDG)	Liu and Yan

The first attempt at using DG methods to solve elliptic and parabolic problems can be tracked back to the late 1970s and early 1980s when an interior penalty (IP) method was independently proposed and studied in [18–20]. In the IP method, a viscous flux is obtained through the average of the left and right states and then penalizing with

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

a penalty term which includes the jump of the solution at each cell interface. The IP method with different stabilization terms was further analyzed for the two dimensional compressible Navier-Stokes equations by Hartmann et al. [17], which led to a new version of IP method: symmetric IP (SIP) method. Inspired by the great success of the DG methods for a first-order system, a natural choice to solve a second-order system is to convert it into a first-order system by introducing additional variables, and then to apply DG methods directly to the first-order system. Based on different choices of numerical flux at the cell interface, there are mainly two kinds of approaches: one is the first Bassi-Rebay (BR1) scheme [15], and the other is the so-called local discontinuous Galerkin (LDG) method [12]. A variation of an LDG method, termed compact DG (CDG) method [14], was later developed by Peraire and Persson to overcome the issue that LDG method is not compact when applied for multidimensional problems. For the similar reason, Bassi and Rebay introduced a second Bassi and Rebay (BR2) scheme [11] based on BR1 scheme in order to maintain the compactness and also enhance the stability for solving pure diffusion problems. In practice, the auxiliary variables in both BR2 scheme and CDG method are usually eliminated by introducing the so-called local and global lifting operator. A kind of hybridizable DG (HDG) discretization was recently introduced for the solution of elliptic problems [27], then extended to convection-diffusion equations [28]. The HDG discretization can be regarded as a mixed method where both the state variable and its gradient can be approximated separately on each element. Inspired by the techniques commonly used for high-order finite volume schemes, van Leer et al [6,7] proposed a recovery-based DG (RDG) method for diffusion equations through recovery which recovers a smooth continuous solution that in the weak sense is indistinguishable from the discontinuous discrete solution. Similarly, Luo et al. [5] developed a reconstructed DG (rDG) method for the compressible Navier-Stokes equation on unstructured grids, where a smooth continuous solution is reconstructed at each cell interface from the discontinuous solution and the diffusive fluxes are then obtained based on the smooth reconstructed solution. Besides those methods above, Gassner et al. [21] presented a method based on the solution of the diffusive generalized Riemann problem (dGRP) to construct the numerical flux for diffusion problems. For a uniform analysis and comparison among several of the methods above, one can refer to the paper [13,16].

Recently, based on the direct weak formulation for solutions of parabolic equations, a direct discontinuous Galerkin (DDG) method was introduced by Liu and Yan [8,9] to solve diffusion problems. Compared to the IP method, the DDG method can be viewed as a multi-term penalty method. The numerical flux defined by the DDG method is simple, compact, conservative, and consistent. The most remarkable feature of the DDG method is its simplicity in implementation and its efficiency in computational cost. In their paper, the DDG method was successfully applied to one and two dimensional diffusion and convection-diffusion equations on structured meshes. Later, Kannan and Wang [10] analyzed and optimized the DDG method based on Fourier analysis and successfully used it for two dimensional Navier-Stokes equations based on spectral volume (SV) [23,24] method. Their numerical experiments demonstrated that the results obtained by the

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

DDG method are comparable to, if not better than, the ones by the LDG, BR2, and IP methods. Very recently, Cheng et al. [32] developed and extended the DDG method for solving the 2D Navier-Stokes equations on arbitrary grids in the framework of DG methods. The numerical results indicated that the DDG method can achieve the designed order of accuracy and is able to deliver comparable results as the widely used BR2 scheme with less computational cost. The DDG methods provided an attractive alternative for solving the compressible Navier-Stokes equations on arbitrary grids and demonstrated its potential for further practical application.

The major objective of the effort presented in this paper is to develop and assess the DDG method for solving the compressible Navier-Stokes equations on 3D hybrid grids. It is the first time that a comprehensive and systematic study is presented to reveal the capability and assess the performance of applying high-order DDG method for simulating three dimensional compressible flows. The formulation for the viscous discretization is detailed studied and the definition of characteristic length is also carefully examined and evaluated based on 3D hybrid grids. Accuracy studies are performed to numerically verify the order of accuracy using flow problems with analytical solutions. The capability of the method in handling curved boundary is also demonstrated and verified. A parallel strategy is proposed and implemented based on a message passing interface (MPI) programming paradigm. A number of test cases are selected to assess the performance the DDG method for solving the 3D compressible Navier-Stokes equations. Numerical experiments indicate that the DDG method can be regarded as an attractive alternative for simulating the 3D compressible flows on hybrid grids owning to its simplicity in implementation and its efficiency in computational cost.

The rest of the paper is organized as follows. The governing equations are briefly described in Section 2. The extension and implementation of the DDG method for discretizing the viscous and heat fluxes in the Navier-Stokes equations on hybrid grids are presented and discussed in Section 3. Numerical experiments are reported in Section 4. Concluding remarks are given in Section 5.

2 Governing equations

The unsteady compressible Navier-Stokes equations 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 is used. The conservative variable vector $\mathbf{U}$, advective (inviscid) flux vector $\mathbf{F}$, and viscous flux vector $\mathbf{G}$ are defined by

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

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

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

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

where γ is the ratio of the specific heats. The components of the viscous stress tensor σ_{ij} and the heat flux q_j are given by

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

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 is determined through Sutherland's law

$$\frac{\mu}{\mu_0}=\Big(\frac{T}{T_0}\Big)^{\frac{3}{2}}\frac{T_0+S}{T+S}.$$

μ_0 denotes the viscosity at the reference temperature T_0, and S is a constant which is $S=110K$. The temperature of the fluid T is given by

$$T=\gamma\frac{p}{\rho}.$$

3 Formulation of direct DG method on 3D hybrid grids

3.1 DG discretization for Navier-Stokes equations

The governing equations Eq. (2.1) are discretized using a discontinuous Galerkin formulation. Firstly, we introduce the following weak form, which is obtained by multiplying the above conservation law by a test function W, integrating by parts over the domain Ω as follows

$$\int_\Omega\frac{\partial\mathbf{U}}{\partial t}Wd\Omega+\int_\Gamma\mathbf{F}_k\mathbf{n}_kWd\Gamma-\int_\Omega\mathbf{F}_k\frac{\partial W}{\partial x_k}d\Omega=\int_\Gamma\mathbf{G}_k\mathbf{n}_kd\Gamma-\int_\Omega\mathbf{G}_k\frac{\partial W}{\partial x_k}d\Omega, \tag{3.1}$$

where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω, and $\mathbf{n}_k$ the unit outward normal vector to the boundary. We assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e. We introduce the following broken Sobolev space $\mathbf{V}_h^p$

$$\mathbf{V}_h^p=\{v_h\in[L_2(\Omega)]^m:v_h|_{\Omega_e}\in[V_p^m]\ \forall\Omega_e\in\Omega\}, \tag{3.2}$$

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

$$\mathbf{V}_h^p=span\Big\{\prod x_i^{\alpha_i}:0\le\alpha\le p,0\le i\le d\Big\}, \tag{3.3}$$

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

where α denotes a multi-index and d is the dimension of space. Then, we can obtain the following semi-discrete form by applying weak formulation on each element Ω_e, i.e., to find $\mathbf{U}_h \in \mathbf{V}_h^p$, such that

$$\begin{aligned}&\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_h W_h d\Omega+\int_{\Gamma_e}\mathbf{F}_k(\mathbf{U}_h)\mathbf{n}_k W_h d\Gamma-\int_{\Omega_e}\mathbf{F}_k(\mathbf{U}_h)\frac{\partial W_h}{\partial x_k}d\Omega\\ =&\int_{\Gamma_e}\mathbf{G}_k(\mathbf{U}_h)\mathbf{n}_k W_h d\Gamma-\int_{\Omega_e}\mathbf{G}_k(\mathbf{U}_h)\frac{\partial W_h}{\partial x_k}d\Omega,\qquad \forall W_h\in\mathbf{V}_h^p,\end{aligned}\tag{3.4}$$

where $\mathbf{U}_h$ and W_h represent the finite element approximations to the analytical solution $\mathbf{U}$ and the test function W, respectively, and they are approximated by a piecewise polynomial function of degree p, which are discontinuous at the cell interfaces. Assume that B_i, $i=1,2,\cdots,N$ is a set of basis functions of upmost degree p, this is then equivalent to the following system of N equations,

$$\begin{aligned}&\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_h B_i d\Omega+\int_{\Gamma_e}\widehat{\mathbf{F}}_k(\mathbf{U}_h^L,\mathbf{U}_h^R;\mathbf{n}_k)B_i d\Gamma-\int_{\Omega_e}\mathbf{F}_k(\mathbf{U}_h)\frac{\partial B_i}{\partial x_k}d\Omega\\ =&\int_{\Gamma_e}\widehat{\mathbf{G}}_k(\mathbf{U}_h^L,\nabla\mathbf{U}_h^L,\mathbf{U}_h^R,\nabla\mathbf{U}_h^R;\mathbf{n}_k)B_i d\Gamma-\int_{\Omega_e}\mathbf{G}_k(\mathbf{U}_h)\frac{\partial B_i}{\partial x_k}d\Omega,\qquad 1\le i\le N,\end{aligned}\tag{3.5}$$

where N is the dimension of the polynomial space. The inviscid and viscous flux functions appearing in Eq. (3.4) is replaced by numerical flux functions $\widehat{\mathbf{F}}_k$ and $\widehat{\mathbf{G}}_k$, respectively, where $\mathbf{U}_h^L$ and $\mathbf{U}_h^R$ are the solution polynomials at the left and right states of the cell interface.

This scheme is called discontinuous Galerkin method of degree p, or in short notation DG(P) 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(P0) method, i.e., to the DG method using a piecewise constant polynomial. Consequently, the DG(Pk) methods with $k>0$ can be regarded as a natural generalization of finite volume schemes to higher order. By simply increasing the degree p of the polynomials, the DG methods of corresponding higher order are obtained.

Without loss of generality, the DG(P2) method is described in the following part of this section. The solution polynomial $\mathbf{U}_h$ in each element can be represented using an either nodal or modal DG formulation as follows

$$\mathbf{U}_h=\sum_{i=1}^{N}\mathbf{U}_i(t)B_i(x),\tag{3.6}$$

where N is the dimension of the polynomial solution space and B_i, $i=1,2,\cdots,N$ is a set of basis functions.

Taylor basis functions [4] are used in this work which a quadratic polynomial solution

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

$\mathbf{U}_h$ can be expressed as

$$\begin{aligned}\mathbf{U}_h=&\overline{\mathbf{U}}B_1+\frac{\partial\mathbf{U}}{\partial x}\Big|_c\Delta xB_2+\frac{\partial\mathbf{U}}{\partial y}\Big|_c\Delta yB_3+\frac{\partial\mathbf{U}}{\partial z}\Big|_c\Delta zB_4\\&+\frac{\partial^2\mathbf{U}}{\partial x^2}\Big|_c\Delta x^2B_5+\frac{\partial^2\mathbf{U}}{\partial y^2}\Big|_c\Delta y^2B_6+\frac{\partial^2\mathbf{U}}{\partial z^2}\Big|_c\Delta z^2B_7\\&+\frac{\partial^2\mathbf{U}}{\partial x\partial y}\Big|_c\Delta x\Delta yB_8,+\frac{\partial^2\mathbf{U}}{\partial x\partial z}\Big|_c\Delta x\Delta zB_9,+\frac{\partial^2\mathbf{U}}{\partial y\partial z}\Big|_c\Delta y\Delta zB_{10},\end{aligned}\tag{3.7}$$

with

$$\begin{aligned}&B_1=1,\quad B_2=\frac{x-x_c}{\Delta x},\quad B_3=\frac{y-y_c}{\Delta y},\quad B_4=\frac{z-z_c}{\Delta z},\\&B_5=\frac{(x-x_c)^2}{2\Delta x^2}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(x-x_c)^2}{2\Delta x^2}d\Omega,\\&B_6=\frac{(y-y_c)^2}{2\Delta y^2}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(y-y_c)^2}{2\Delta y^2}d\Omega,\\&B_7=\frac{(z-z_c)^2}{2\Delta z^2}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(z-z_c)^2}{2\Delta z^2}d\Omega,\\&B_8=\frac{(x-x_c)(y-y_c)}{\Delta x\Delta y}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(x-x_c)(y-y_c)}{\Delta x\Delta y}d\Omega,\\&B_9=\frac{(x-x_c)(z-z_c)}{\Delta x\Delta z}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(x-x_c)(z-z_c)}{\Delta x\Delta z}d\Omega,\\&B_{10}=\frac{(y-y_c)(z-z_c)}{\Delta y\Delta z}-\frac{1}{|\Omega_e|}\int_{\Omega_e}\frac{(y-y_c)(z-z_c)}{\Delta y\Delta z}d\Omega,\end{aligned}\tag{3.8}$$

where $\Delta x=0.5(x_{\max}-x_{\min})$, $\Delta y=0.5(y_{\max}-y_{\min})$, $\Delta z=0.5(z_{\max}-z_{\min})$ and (x_c,y_c,z_c) is the centroid of the element Ω_e. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the centroid of the element, regardless of element shapes. This formulation belongs to the so-called modal discontinuous Galerkin method and has a number of attractive features. First, cell-averaged variables and their derivatives are handily available in this formulation. It makes the implementation of reconstructed schemes and Hermite WENO limiters simple and straightforward [25, 26]. Secondly, the Taylor basis is hierarchic, thus, greatly facilitates implementation of p-multigrid method and p-refinement strategy. Thirdly, the formulation of the Taylor basis functions is uniform on arbitrary element, such as triangles and quadrilaterals in 2D, tetrahedron, pyramid, prism, and hexahedron in 3D. It significantly simplifies the implementation of DG methods on hybrid grids.

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

3.2 Direct DG formulation for viscous flux

In the present study, we employ the DDG method to discretize the viscous flux in Eq. (3.5) in which the face integral for viscous flux term can be rewritten as follows

$$\begin{aligned}&\int_{\Gamma_e}\widehat{\mathbf{G}}_k(\mathbf{U}_h^L,\nabla\mathbf{U}_h^L,\mathbf{U}_h^R,\nabla\mathbf{U}_h^R;\mathbf{n}_k)B_i d\Gamma\\&=\int_{\Gamma_e}\mathbf{G}_k(\widehat{\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R),\widehat{\nabla\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R,\nabla\mathbf{U}_h^L,\nabla\mathbf{U}_h^R,\nabla^2\mathbf{U}_h^L,\nabla^2\mathbf{U}_h^R;\mathbf{n}_k))B_i d\Gamma,\end{aligned}\tag{3.9}$$

where $\widehat{\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R)$ and $\widehat{\nabla\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R,\nabla\mathbf{U}_h^L,\nabla\mathbf{U}_h^R,\nabla^2\mathbf{U}_h^L,\nabla^2\mathbf{U}_h^R;\mathbf{n}_k)$ are the DDG viscous flux based on the conservative variables of Navier-Stokes equations.

The direct discontinuous Galerkin (DDG) method can be regarded as a multi-term interior penalty method. Compared to the interior penalty method which the penalty term only involves the jump of solution values, the DDG numerical flux also takes the jump of high-order derivatives into consideration. One of the most attractive features of the DDG method lies in its simplicity with a simple flux formulation directly based on the weak form of the DG framework. In order to describe the direct DG method for the discretization of viscous terms, firstly, we introduce the following jump operator and average operator:

$$[\mathbf{U}]=\mathbf{U}^R-\mathbf{U}^L,\qquad \overline{\mathbf{U}}=\frac{1}{2}(\mathbf{U}^R+\mathbf{U}^L),\tag{3.10}$$

where $\mathbf{U}^L$ and $\mathbf{U}^R$ denote the left state and right state of a common face, respectively. With the definition above, the x-direction, y-direction and z-direction component of viscous flux for the DDG method can be given as

$$\widehat{\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R)=\overline{\mathbf{U}},\tag{3.11}$$

and

$$\begin{aligned}\widehat{\partial_x\mathbf{U}}&=\beta_0\frac{[\mathbf{U}]}{\Delta}n_x+\overline{\partial_x\mathbf{U}}+\beta_1\Delta([\partial_{xx}\mathbf{U}]n_x+[\partial_{xy}\mathbf{U}]n_y+[\partial_{xz}\mathbf{U}]n_z),\\\widehat{\partial_y\mathbf{U}}&=\beta_0\frac{[\mathbf{U}]}{\Delta}n_y+\overline{\partial_y\mathbf{U}}+\beta_1\Delta([\partial_{yx}\mathbf{U}]n_x+[\partial_{yy}\mathbf{U}]n_y+[\partial_{yz}\mathbf{U}]n_z),\\\widehat{\partial_z\mathbf{U}}&=\beta_0\frac{[\mathbf{U}]}{\Delta}n_z+\overline{\partial_z\mathbf{U}}+\beta_1\Delta([\partial_{zx}\mathbf{U}]n_x+[\partial_{zy}\mathbf{U}]n_y+[\partial_{zz}\mathbf{U}]n_z),\end{aligned}\tag{3.12}$$

with

$$\widehat{\nabla\mathbf{U}}(\mathbf{U}_h^L,\mathbf{U}_h^R,\nabla\mathbf{U}_h^L,\nabla\mathbf{U}_h^R,\nabla^2\mathbf{U}_h^L,\nabla^2\mathbf{U}_h^R;\mathbf{n})=(\widehat{\partial_x\mathbf{U}},\widehat{\partial_y\mathbf{U}},\widehat{\partial_z\mathbf{U}})\cdot\mathbf{n},\tag{3.13}$$

where Δ is the characteristic face length, $\mathbf{n}=(n_x,n_y,n_z)$ is the unit normal vector of a given face, β_0 and β_1 are regarded as constant penalty parameters. Generally, the penalty parameters can be taken as $\beta_0=1.0\sim4.0$ and $\beta_1=1/12$. In this work, we choose $\beta_0=2.0$ based on both the rigorous mathematical analysis [36,37] and our numerical experiences

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

from the implementation of DDG method for two dimensional problems. Clearly, the numerical flux defined by the DDG formulation is compact, conservative, and consistent. Further more, there are several extra merits of DDG method which should be emphasized:

- DDG numerical flux is defined directly based on the DG weak form, it does not require any additional data structure for auxiliary variables, nor intricate manipulation of the partial differential equations, as the case with LDG method;
- DDG numerical flux is not only compact but also point-wise locally, which means it does not need a preliminary integration over the face and the corresponding lifting operators, such as BR2 scheme;
- The simple definition of the DDG viscous fluxes enables us to obtain its Jacobian matrix much more easily, which is an important factor for fast convergence of any implicit method.

3.3 Calculate DDG flux on 3D hybrid Grids

In order to correctly perform the simulation using DDG method for 3D compressible flows on hybrid grids, the characteristic length Δ, also termed as mesh size, in the DDG numerical flux in Eq. (3.12) still needs to be carefully re-evaluated and examined. In our previous work [32], it showed that the characteristic length impacts both the magnitude of error and the order of convergence of the DDG method. Inappropriate definition of the characteristic length can detrimentally affect the convergence of numerical results. Several possible choices of characteristic length had been considered and tested in that work and extensive numerical experiments were used to verify the performance using different definitions of characteristic length. The one, that currently produced the best results, is given as follows.

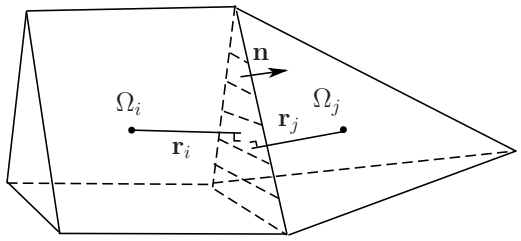


Figure 1: Definition of characteristic face length on linear mesh.

For a given internal face Γ and its left and right element Ω_i and Ω_j as shown in Fig. 1, the characteristic length for the internal face Γ is defined as

$$\Delta = |\mathbf{r}_i + \mathbf{r}_j|, \tag{3.14}$$

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the displacement vector from the center of the cell Ω_i and cell Ω_j to the common face, respectively. It can be seen clearly that this definition is consistent with the definition introduced by Liu [8,9] if only a uniform structure mesh is used and it also provides a uniform definition on 3D element of arbitrary shape.

It is well known that a second-order or even higher order curved element or curved grid is essential for DG method to handle curved geometry in real applications. The order of the curved element or curved grid depends on the order of DG spatial discretization. Currently, in this work, a second-order curved grid is used when the test problems entail curved geometry. When implement the DDG method on curved element, there are still two issues needed to be further emphasized and clarified.

Firstly, the definition of cell center used in Eq. (3.14) becomes obscure when it comes to a curved element. For a linear element, the cell centroid is commonly used as the cell center in our computation, however, for a curved element, especially a highly scratched curved element in the boundary layer region, one can imagine that the cell centroid could be far away outside the cell. Clearly, in this case, the distance from cell centroid to the common face becomes inappropriate and certainly brings pernicious effect to the computational results. Thus, when the curved element is used in the implementation, the cell center for the curved element is obtained through the nonlinear mapping from reference space to physical space as follows

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi_c,\eta_c,\zeta_c) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \tag{3.15}$$

where (ξ_c,η_c,ζ_c) is the centroid of the reference element in reference space, $M_i(\xi,\eta,\zeta)$ are the shape functions of the transformation and (x_i,y_i,z_i) are the coordinates of the element in physical space. Then, the coordinate of cell center (x_c,y_c,z_c) is obtained through mapping the centroid of the reference element in reference space into the physical space.

Another issue lies in the practical implementation of the DDG flux based on a curved element, approximate distance from the cell center to the common face, most likely a curved common face, needs to be appropriately calculated. Although, based on our experience, using the exact distance from a given point to the curved face probably is unnecessary, the characteristic length still needs to be calculated more accurately rather than a rough estimation based on the linear approximation for the curved geometry. Thus, currently in this work, in order to give an appropriate estimation of the characteristic length, the curved face is further subdivided into several linear patches, and if the linear patch is a quadrilateral patch, a further subdivision is needed to split the quadrilateral patch into two triangular patches. Then, the approximated characteristic length is obtained through calculating the minimum distance from the cell center to the all of the linear patches by a brute force search.

As for boundary face, similar to our previous work, the DDG flux is constructed with the internal first-order derivatives and consider the jump of second-order derivatives as

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

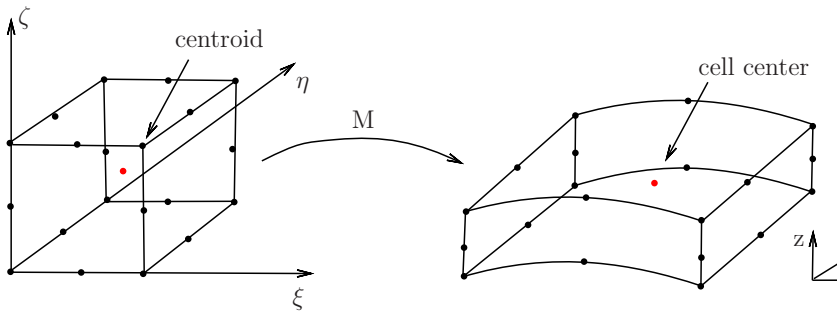


Figure 2: Definition of cell center for curved element.

zero if DG(P2) is applied. The specific form of the boundary flux can be written as

$$\widehat{\mathbf{U}}|_B = \mathbf{U}^B, \tag{3.16}$$

and

$$\begin{aligned}
\widehat{\partial_x \mathbf{U}}|_B &= \beta_0 \frac{\mathbf{U}^B - \mathbf{U}^L}{\Delta} n_x + \partial_x \mathbf{U}^L, \\
\widehat{\partial_y \mathbf{U}}|_B &= \beta_0 \frac{\mathbf{U}^B - \mathbf{U}^L}{\Delta} n_y + \partial_y \mathbf{U}^L, \\
\widehat{\partial_z \mathbf{U}}|_B &= \beta_0 \frac{\mathbf{U}^B - \mathbf{U}^L}{\Delta} n_z + \partial_z \mathbf{U}^L,
\end{aligned} \tag{3.17}$$

with

$$\widehat{\nabla \mathbf{U}}|_B = (\widehat{\partial_x \mathbf{U}}|_B, \widehat{\partial_y \mathbf{U}}|_B, \widehat{\partial_z \mathbf{U}}|_B) \cdot \mathbf{n}, \tag{3.18}$$

where $\mathbf{U}^B$ is the vector of conservative variables obtained from boundary conditions.

3.4 Implicit temporal discretization

The spatial discretization in Eq. (3.5) leads to a system of ordinary differential equations:

$$\mathbf{M}\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = 0, \tag{3.19}$$

where $\mathbf{M}$ denotes the mass matrix, $\mathbf{U}$ is the solution vector, and $\mathbf{R}$ is residual vector. Using Euler implicit time-integration, the spatially discretized Navier-Stokes equations can be linearized in time and written as

$$\left(\frac{\mathbf{M}}{\Delta t}\mathbf{I} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)\Delta \mathbf{U} = -\mathbf{R}(\mathbf{U}^n), \tag{3.20}$$

where $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$ is the Jacobian matrix, Δt is the time increment, and $\Delta \mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^n$ is the solution difference between time level n and $n+1$. This system of linear equations at each time step is solved by a GMRES+LU-SGS method [22].

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

Note that the simple form of the viscous flux function defined by the DDG method enables us to obtain its Jacobian matrix exactly and much more easily. The use of the exact Jacobian matrix in the GMRES+LU-SGS method can significantly speed up the convergence [34], and practically deliver the quadratic convergence of Newton's method.

3.5 Parallelization

The compactness of the DDG method makes it ideally suitable for parallel computing. In the present work, an SPMD (single program, multiple data) programming paradigm based on the MPI library is adopted to achieve parallelism. The METIS library [33] is used for the partitioning of a grid into a number of sub-domains with approximately the same number of cells. A typical example is illustrated with the hybrid grid for steady flow over a sphere in Section 4.5, in which the hybrid mesh is split into 48 sub-domains as shown in Fig. 14.

The first issue we need to address in the parallelization of a 3D DDG solver on hybrid grids is the data communication among neighbor sub-domains. The data structures which need to communicate are organized in three categories: (a) geometric information, (b) elemental solution vectors, and (c) statistical indicators, such as time step size and residual et al.. The partition of whole domain into sub-domains and the distribution of geometric information among sub-domains are completed in the pre-processing part. Then, the major task becomes how to send and receive the solution vectors among the neighbor sub-domains. In our implementation, a set of ghost cells are created along the partition boundary for each sub-domain. Both the geometric information and the solution vectors of the adjacent cells from neighbor sub-domains are stored in the ghost elements. The communication in parallel mode is managed by the standard MPI commands such as, nonblocking send, nonblocking receive and wait commands.

For the implementation of explicit scheme, the solution vectors of the ghost elements only need to be updated at the end of each time step or each stage within a time step if a multi-stage scheme, such as multi-stage Runge-Kutta scheme, is used. However, for an implicit scheme, extra exchanges of the solution vectors are commonly required in the procedure of solving the linear system which stems from the linearization in Eq. (3.20). Currently, in this work, as a preconditioned GMRES solver is adopted, thus, an exchange of solution vectors among neighbor sub-domains is required whenever the matrix vector multiplication is performed in the GMRES solver. Another issue arises from the application of preconditioner, as it could be no longer a commonplace to do the matrix decomposition and preconditioning exactly as the situation in a serial program. Thus, currently, the LU-SGS preconditioning is only locally implemented within each sub-domain in a parallelized execution. This strategy can provide a satisfactory performance when the number of partitions is in a small or moderate scale, which means there are still sufficient elements in each sub-domain.

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

4 Numerical results

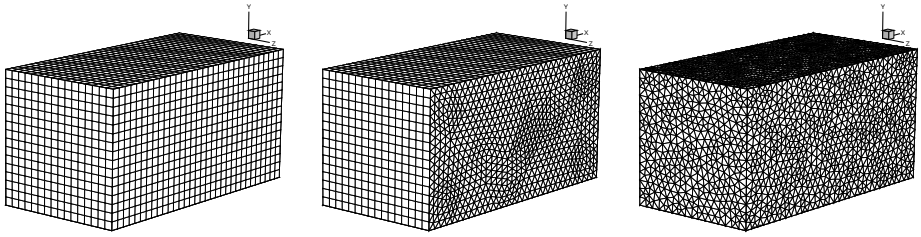
4.1 Couette flow

A compressible Couette flow, considered in this test case, is a laminar viscous flow between two parallel plates. This example is aimed at verifying the accuracy and convergence of the DDG method for solving the compressible Navier-Stokes equations on arbitrary grids. The bottom plate with a fixed temperature (T_0) is stationary, while the top one with a fixed temperature (T_1) moves at a velocity of U. The two plates are separated by a distance H. Assume that the viscosity coefficient μ is a constant. The analytic solution is given by

$$\begin{aligned} &u=\frac{y}{H}U, \qquad v=0, \qquad w=0, \\ &p=p_\infty, \qquad \rho=\frac{p}{RT}, \\ &T=T_0+\frac{y}{H}(T_1-T_0)+\frac{y}{H}(1.0-\frac{y}{H})\frac{PrU^2}{2c_p}, \end{aligned} \tag{4.1}$$

where Pr is the Prandtl number and c_p is the specific heat capacity at constant pressure.

For our numerical experiments, we take $H=2$, $T_0=0.8$, and $T_1=0.85$. The Mach number for the upper wall $M_1=0.1$, and the Reynolds number $Re=100$ with a constant viscosity coefficient. The computational domain is a cube of $4\times2\times2$. Exact boundary condition has been set at the ghost states for all the boundary faces. It should be noted that only the solution values obtained from the exact solution are used at ghost states and the boundary treatment for DDG method exactly follows the description in Section 3.3. Computations are performed on three different sets of successively globally refined grids including hexahedral grids, prismatic grids and tetrahedral girds, as shown in Fig. 3.



(a) Hexahedral grid (b) Prismatic grid (c) Tetrahedral grid

Figure 3: Three different types of grids for Couette flow.

Numerical experiments are performed to compare the performance of the DDG method with BR2 scheme. The results for the accuracy and convergence of both DDG and BR2 methods are presented in Table 2 and Table 3 for both DG(P1) and DG(P2), respectively. It

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

Table 2: Grid refinement study for Couette flow of DG(P1).

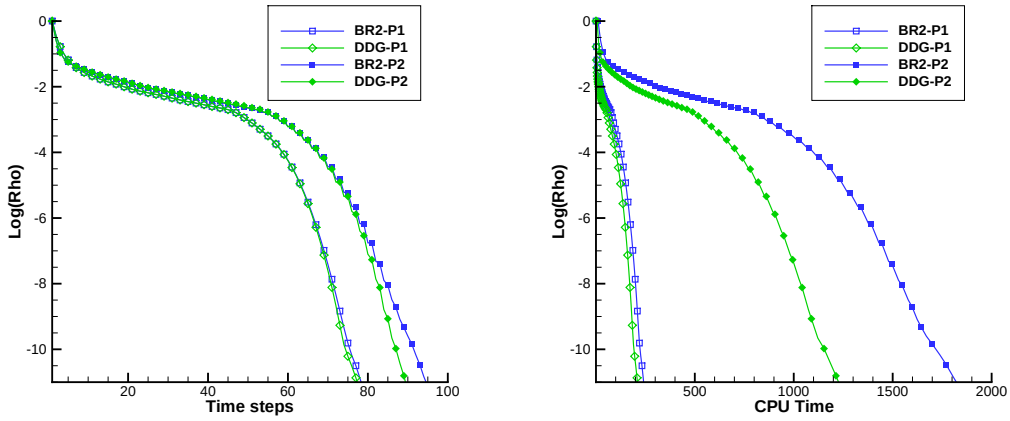
Number of cells	BR2(P1)		DDG(P1)	
Hexahedral grid	L_2-error	Order	L_2-error	Order
128	7.37e-05		7.41e-05	
1024	1.48e-05	2.31	1.52e-05	2.29
8192	3.54e-06	2.06	3.70e-06	2.03
Prismatic grid	L_2-error	Order	L_2-error	Order
344	4.76e-05		4.77e-05	
2752	1.10e-05	2.11	1.11e-05	2.10
22016	3.05e-06	1.85	2.80e-06	1.99
Tetrahedral grid	L_2-error	Order	L_2-error	Order
659	1.59e-04		1.60e-04	
5315	2.34e-05	2.76	2.41e-05	2.73
36272	6.19e-06	1.92	6.29e-06	1.94

Table 3: Grid refinement study for Couette flow of DG(P2).

Number of cells	BR2(P2)		DDG(P2)	
Hexahedral grid	L_2-error	Order	L_2-error	Order
128	6.10e-07		6.18e-07	
1024	7.61e-08	3.00	7.87e-08	2.97
8192	9.32e-09	3.03	1.00e-08	2.98
Prismatic grid	L_2-error	Order	L_2-error	Order
344	3.21e-07		3.24e-07	
2752	4.05e-08	2.99	4.19e-08	2.95
22016	4.89e-09	3.05	5.24e-09	3.00
Tetrahedral grid	L_2-error	Order	L_2-error	Order
659	1.38e-06		1.37e-06	
5315	1.19e-07	3.53	1.21e-07	3.50
36272	1.26e-08	3.24	1.21e-08	3.32

can be seen clearly that from the numerical results, both DDG and BR2 methods are able to attain the optimal rate of convergence on the three sets of grids and the magnitude of error is totally comparable. Fig. 4 shows the convergence history for both DDG and BR2 methods on pure tetrahedral grids as a representative. One can observe that the number of time steps to reach a steady state solution is comparable between DDG and BR2 methods. However, the DDG method requires much less cpu time than the BR2 method, based on our implementation, especially for DG(P2). The comparison indicates that the

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Residual history vs time steps (b) Residual history vs cpu time (s)

Figure 4: Comparison of residual history between DDG and BR2 methods for Couette flow (on tetrahedral fine grid).

DDG method can be more efficient than the BR2 method in solving the compressible Navier-Stokes equations.

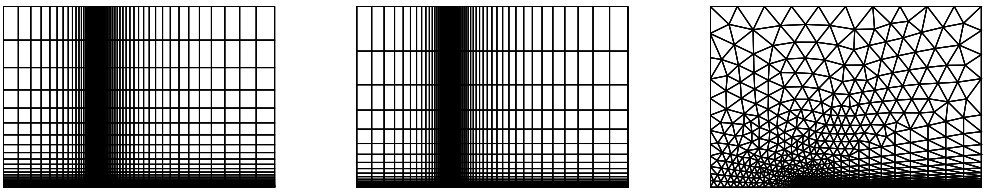
Although this is a only 3D simulation of a simple 2D test case, the results are still encouraging as it can be regarded as a bottom-line test case for a 3D solver for viscous flows. Moreover, it provides a quantitative comparison and assessment due to the existence of an analytic solution. From the numerical results, generally, both DDG and BR2 methods are of totally comparable accuracy and it clearly demonstrates the potential of applying DDG methods for solving 3D Navier-Stokes equations on arbitrary grid.

4.2 Laminar flow over a flat plate

The laminar boundary layer over an adiabatic flat plate at a free-stream Mach number of 0.5 and a Reynolds number of 100,000 based on the freestream velocity and the length of the flat plate is considered in this test case. The computational domain is bounded from -0.5 to 1.0 in the x-direction and the flat plate starts at point $(0,0)$ and extends to $(1,0)$. This problem is chosen to assess the accuracy of the DDG method for the solution of the three dimensional compressible Navier-Stokes equations, as the Blasius solution can be used to measure the accuracy of the numerical solutions. Computations are performed on four different types of grids: two hexahedral grids, one prismatic grid, and one tetrahedral gird as shown in Fig. 5.

As this is also a 3D simulation of 2D test case, thus, the first three grids are simply generated by extrusion of the 2D grids with one layer of elements in z-direction. The first two grids used in this test case have the same number of grid points 75×30, with 25×30 cells ahead of the flat plate, the same distribution of the grid points in the x-direction, but

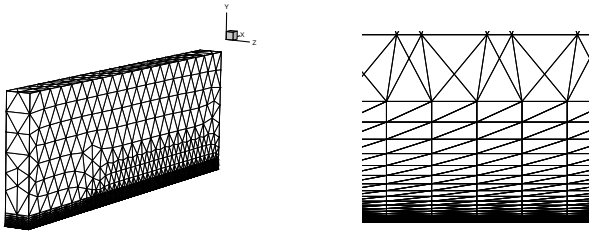
https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Hexahedral grid-1 (4,712 nodes, 2,250 elements, growth rate:1.2)

(b) Hexahedral grid-2 (4,712 nodes, 2,250 elements, growth rate:1.3)

(c) Prismatic grid (1,800 nodes, 1,693 elements)



(d) Tetrahedral grid (9,828 nodes, 47,536 elements)

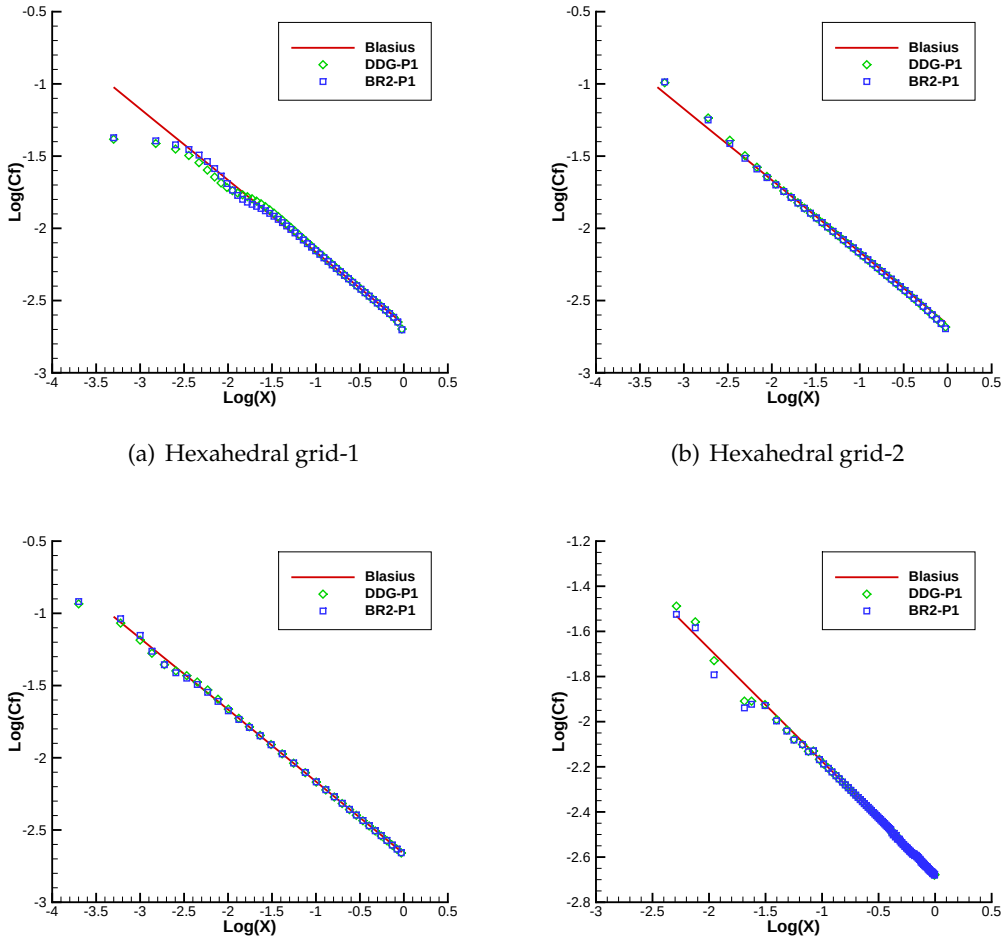
(e) Tetrahedral grid (local view at boundary layer region)

Figure 5: Four types of grids for laminar flow over a flat plate.

a different distribution of grid points in the y-direction. In order to cluster points near the wall, the point distribution in the y-direction follows a geometric stretching. A growth rate of 1.2 and 1.3 is used for the two meshes in the computation, respectively. The growth rate stands for the ratio of the heights of the two successive elements. The height of the first layer is 0.8461E-03 and 0.1145E-03, respectively. Moreover, a prismatic grid and a pure tetrahedral grid are generated for this test case in order to assess the performance of applying DDG method for highly stretched prismatic elements and tetrahedral elements within the boundary layer region.

The numerical results obtained by both DDG and BR2 methods on these four grids are presented, and compared with the theoretical one given by the well-known Blasius solution. Fig. 6 and Fig. 7 show the logarithmic plot of the computed skin friction coefficient obtained by the DG(P1) and DG(P2) solutions, respectively. In general, similar results are obtained by both DDG and BR2 methods, they both match the Blasius solution very well on the four different types of grids. However, it should be noted that the results obtained by the DDG method are slightly better than the ones obtained by the BR2 method, when grids are under-resolved as shown in Figs. 6(a) and 7(a). Moreover, the DDG method demonstrates a good capability in handling highly stretched prismatic elements and tetrahedral elements within the boundary layer region as comparable results are obtained for the prismatic and tetrahedral grids. The residual convergence history is shown in Fig. 8, similarly, the number of time steps to reach a steady state solution

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Hexahedral grid-1

(b) Hexahedral grid-2

(c) Prismatic grid

(d) Tetrahedral grid

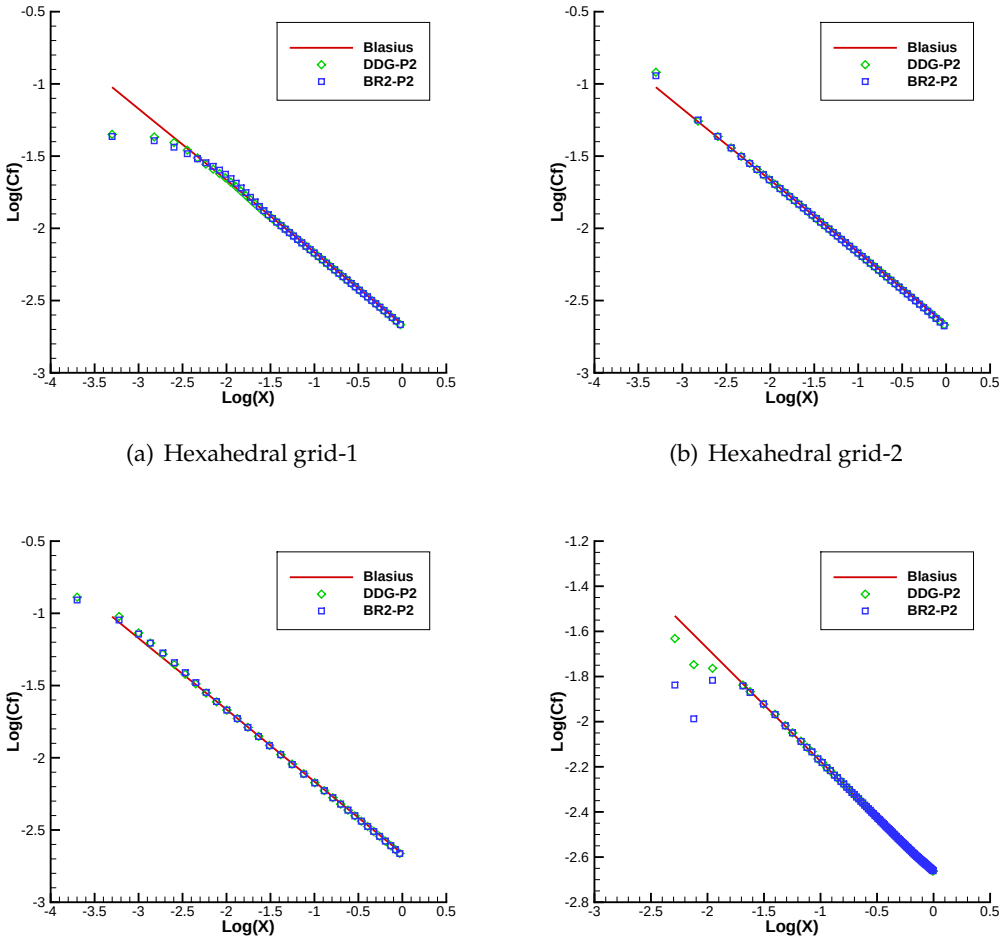
Figure 6: Comparison of skin friction coefficient (Cf) between DDG(P1) and BR2(P1).

between DDG and BR2 is more or less the same, however, based on our implementation, DDG method clearly outperforms BR2 scheme in reducing the CPU time, especially for the DG(P2) method.

4.3 Laminar flow over a Delta wing at low Reynolds number

As the last test case, we investigate a laminar flow around a Delta wing which is a well-established test case in the EU project ADIGMA [29] for high-order methods. The Delta wing has a sloped and sharp leading edge and a blunt trailing edge and geometry of the delta wing can be seen from the surface mesh shown in Fig. 9. In this test case, in order

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Hexahedral grid-1

(b) Hexahedral grid-2

(c) Prismatic grid

(d) Tetrahedral grid

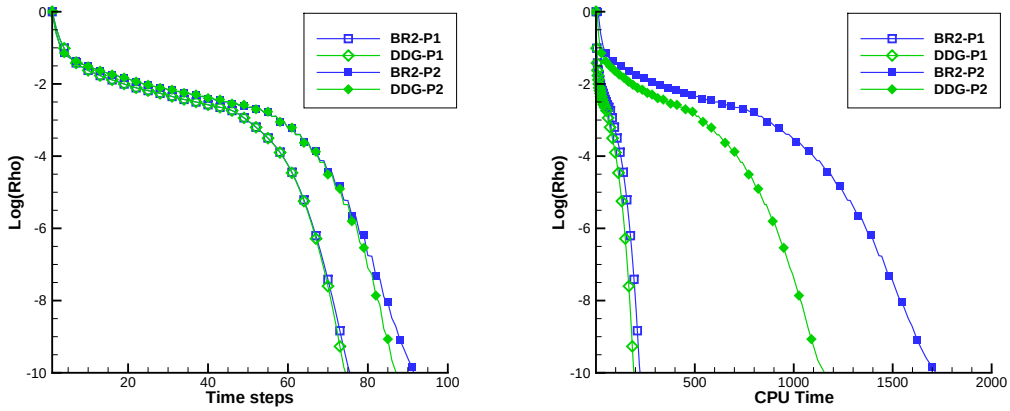
Figure 7: Comparison of skin friction coefficient (Cf) between DDG(P2) and BR2(P2).

to make a comprehensive study and comparison with the results from the high-order CFD workshop [30], the test case is considered at laminar conditions with inflow Mach number equal to 0.3, at an angle of attack equal to 12.5^o, and Reynolds number is equal to 4000 with isothermal no-slip wall boundary condition imposed on the wing geometry.

A distinguishing feature for this test case is that as the flow passing the leading edge, it rolls up and creates a vortex and a secondary vortex which remains over a distance behind the wing. Thus, this problem is widely used to verify the high-order and adaptive methods for the computation of vortex dominated external flows.

In order to present a convergence study for the lift and drag coefficients, a set of three successively refined hexahedral grids from the high-order CFD workshop is used in this

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Residual history vs time steps (b) Residual history vs cpu time (s)

Figure 8: A comparison of residual history between DDG and BR2 methods for laminar flow over a flat plate (on tetrahedral grid).

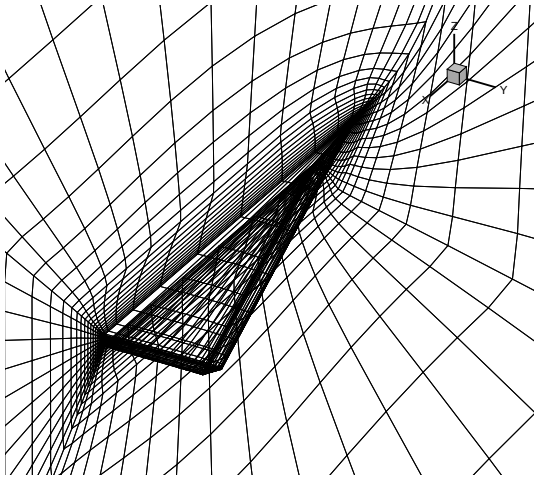
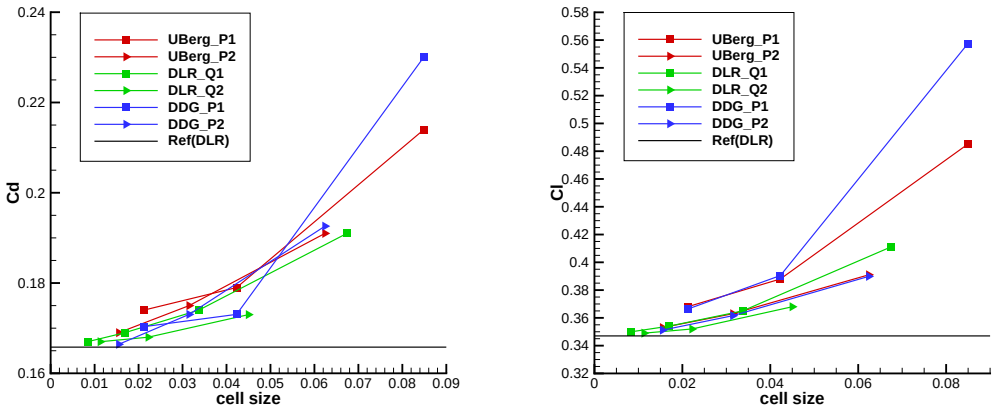


Figure 9: Surface mesh for the Delta wing (fine mesh).

test case, which contains 408, 3,264 and 26,112 elements, respectively. The convergence study of both lift and drag coefficient compared to the reference results obtained from the high-order CFD workshop is shown in Fig. 10(a) and Fig. 10(b). The reference value of drag coefficient is C_d=0.1658 and lift coefficient is C_l=0.347 based on the literature [31]. In order to make a comprehensive comparison, two sets of reference data for both DG(P1) and DG(P2) methods from the high-order CFD workshop are selected and denoted as 'UBerg_Pk' and 'DLR_Qk', respectively. The reference values of 'UBerg_Pk' are provided by University of Bergamo with modal basis DG and BR2 scheme for viscous discretiza-

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Comparison of Cd vs cell size

(b) Comparison of Cl vs cell size

Figure 10: Convergence study of lift and drag coefficients for laminar flow over a Delta wing.

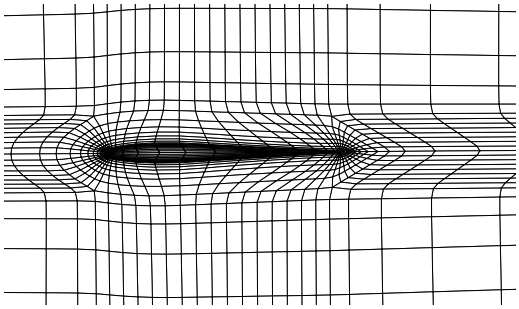
tion. The reference values of 'DLR_Qk' are provided by DLR with nodal basis DG and BR2 scheme for viscous discretization.

From the grid convergence study, it can be seen clearly that both DDG(P1) and DDG(P2) methods can deliver satisfactory and reliable results. Clearly, DDG(P2) method outperforms DDG(P1) method in predicting both lift and drag coefficient. Compared to the reference results, DDG methods yields a relatively better prediction on drag coefficient when the grid becomes reasonably fine. One should also note that the difference of results between DG(Pk) and DG(Qk) is due to the use of different basis functions. The grid convergence study, again, indicates that the DDG method is an attractive and reliable alternative for the discretization of viscous flux in the DG framework and shows its promising for practical applications.

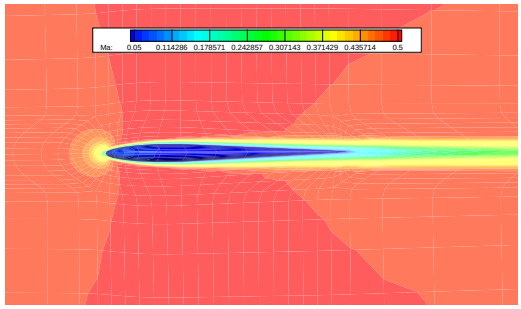
4.4 Laminar flow over a streamlined 3D body

In this test case, we consider laminar flow over a streamlined 3D body in order to test and assess the capability of the DDG solver in handling curved geometry. The definition of the geometry for the streamlined 3D body comes from the European ADIGMA project where it is called BTC0 test case. The geometry is based on a 10% thick airfoil with boundaries constructed by a surface of revolution. The airfoil is defined analytically by an elliptical leading edge and straight lines. The inflow Mach number equals to 0.5, angle of attack $\alpha = 1^o$ and Reynolds number $Re = 5000$ with adiabatic no-slip wall boundary condition imposed. A set of three successively refined second-order curved hexahedral grids which include 768, 6,144 and 49,152 cells, respectively, is used in this test case. As the solution is symmetric, only half of the configuration is modeled in the current numerical experiments.

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

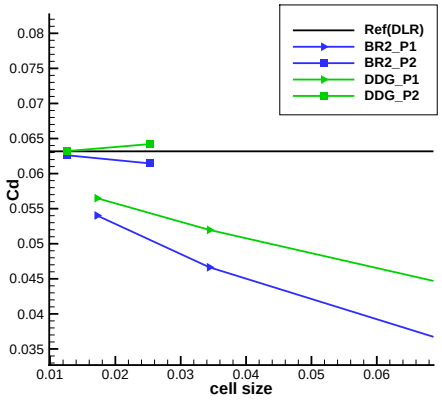


(a) Surface mesh of the streamlined 3D body, medium grid

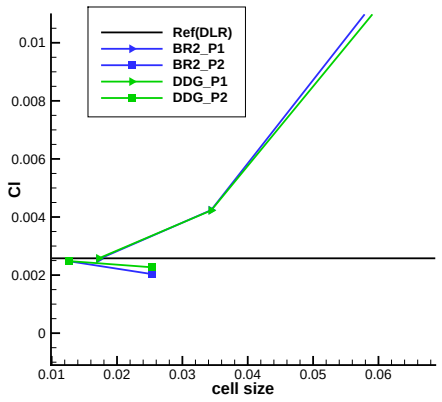


(b) 36 Mach number contours from 0.05 to 0.5, DDG(P2), fine grid

Figure 11: Surface mesh and Mach number contours for laminar flow over a streamlined 3D body.



(a) Comparison of Cd vs cell size



(b) Comparison of Cl vs cell size

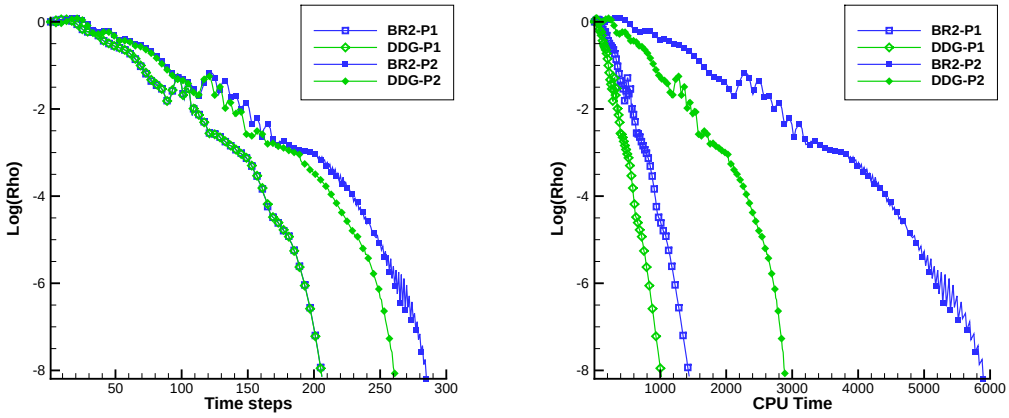
Figure 12: Convergence study of lift and drag coefficients for laminar flow over a streamlined 3D body.

The surface mesh and the computed Mach number contours of the flow field are shown in Fig. 11. A comparison of lift and drag coefficients between the DDG method and the BR2 scheme is present in Fig. 12. Based on the same second-order curved grids, DDG method yields sightly better results than BR2 scheme for DG(P1) and generally provide comparable results for DG(P2). The results demonstrate that the DDG method is well capable in handling curved geometry with high-order elements for laminar flows. The residual convergence history shown in Fig. 13 indicates that the DDG method is more efficient than BR2 based on our implementation, especially for the DDG(P2) method.

4.5 Steady flow over a 3D sphere

A steady viscous flow around a sphere is computed to validate the developed DDG solver on an unstructured hybrid grid with curved boundary geometry. The Reynolds

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Residual history vs time steps (b) Residual history vs cpu time (s)

Figure 13: Comparison of residual history between DDG and BR2 methods for laminar flow over a streamlined 3D body.

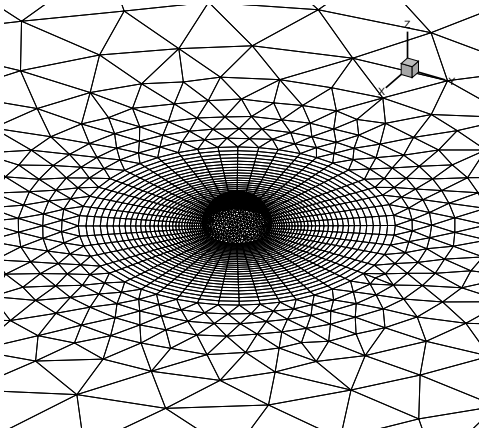
number is chosen to be 118 based on the diameter of the sphere and the Mach number is equal to 0.2535. The objective for this test case is again to assess and verify the performance of DDG methods for handling curved boundaries.

As the solution is symmetric, only half of the configuration is modeled in the current numerical experiments. In order to verify the capability of DDG method in handling hybrid grids with curved geometry, a second-order hybrid grid is generated for this test case which consists 194,007 grid nodes, 80,434 grid cells including 35,760 prismatic cells and 44,674 tetrahedral cells. The surface mesh is shown in Fig. 14, as one can see that there are 30 layers of prismatic cells around the sphere and isotropic tetrahedral cells for the remaining regions.

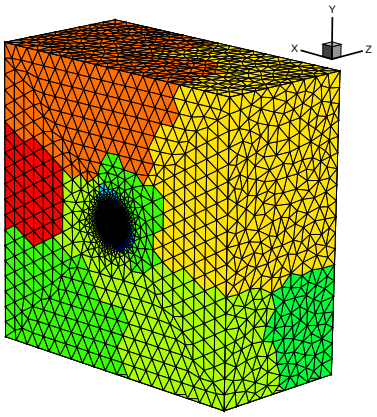
The computed Mach number contours of the flow field and the streamlines over the cylinder are shown in Fig. 15 and one can observe the symmetric steady separation bubble clearly. A comparison of separation angle θ_s, wake length L_w and drag coefficient between the DDG method and the BR2 scheme is present in Table 4. It can be seen that the results obtained by both DDG method and BR2 scheme are similar based on the same grid. The results demonstrate that the DDG method is well capable in handling curved geometry with high-order elements for laminar flows. The residual convergence history based on our implementation, shown in Fig. 16, again indicates that the DDG method is more efficient than BR2 scheme, especially for the DDG(P2) method.

In order to assess the scalability of our parallelization strategy implemented in the current work, we perform a scaling test based on the current test case. The computation is conducted on a small cluster which consist of 32 compute nodes, 2 cores per node. The scaling test is carried out for both DDG(P1) and DDG(P2) methods with implicit temporal discretization as described in Section 3.4. The results of the speedup ratio obtained

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

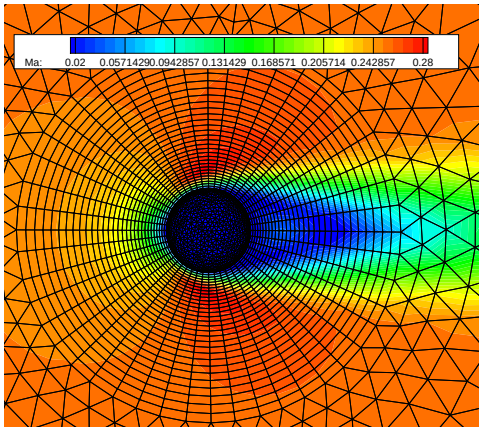


(a) Surface mesh of the hemisphere with the symmetric plane Z=0

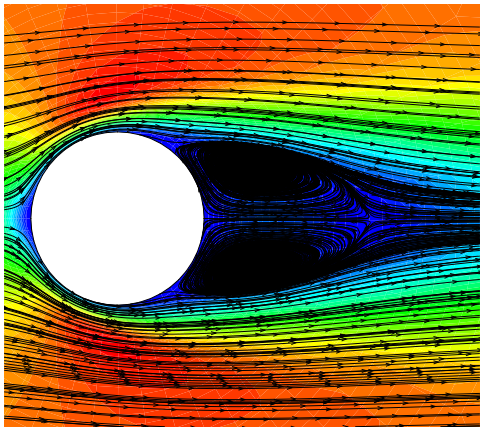


(b) Domain decomposition using METIS

Figure 14: Grid and domain decomposition for steady flow over a sphere.



(a) 36 Ma number contours from 0.02 to 0.28, DDG(P2)



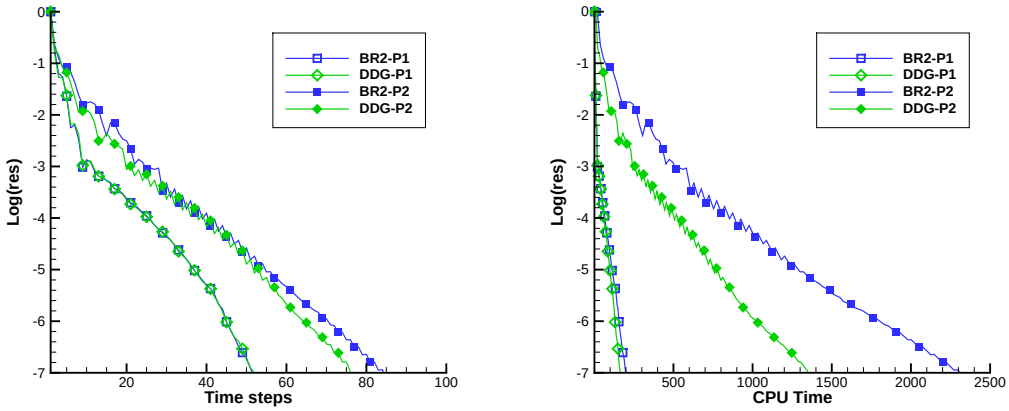
(b) Streamlines near wake region, DDG(P2)

Figure 15: Computed Ma number contours and streamlines near the wake region for steady flow over a sphere.

Table 4: Comparison of separation angle θ_s, wake length L_w and drag coefficient C_d for steady flow past a sphere

Method	θ_s	L_w	C_d
DG(P1)-DDG	127.5^o	0.92D	0.9782
DG(P2)-DDG	123.7^o	0.96D	1.0162
DG(P1)-BR2	127.6^o	0.92D	0.9513
DG(P2)-BR2	123.6^o	0.96D	1.0137
Ref [35]	$123.6^o \pm 0.1$	—	1.0162

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press



(a) Residual history vs time steps (b) Residual history vs cpu time (s)

Figure 16: Comparison of residual history between DDG and BR2 methods for steady flow over a sphere.

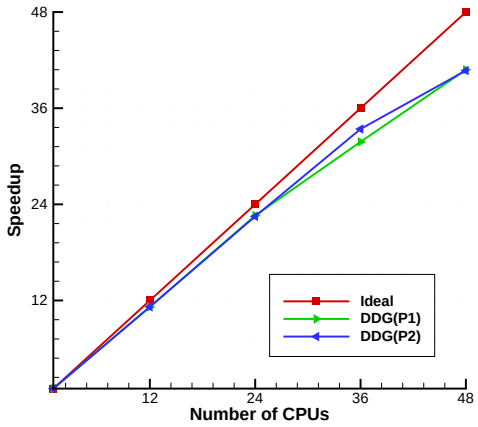


Figure 17: Parallel speedup ratio for steady flow over a sphere.

from the scaling test are shown in Fig. 17. From the results, it can be seen clearly that the current parallelization strategy applied in our implementation delivers a satisfactory result with respect to the scale of the problem in this test case.

5 Conclusions

The DDG method has been developed and extended for the discretization of the viscous and heat fluxes for 3D compressible Navier-Stokes equations on hybrid grids. The

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

formulation of the DDG discretization has been detailed studied and the definition of characteristic length is also carefully examined and evaluated based on 3D hybrid, both linear and second-order curved grids. A number of test cases have been presented to assess the performance of the DDG method for solving the 3D compressible Navier-Stokes equations. Numerical experiments demonstrate that the DDG method can achieve the designed order of accuracy and is able to deliver totally comparable results as the widely used BR2 method at a significantly reduced cost based on our practical experience. Owning to its simplicity in implementation and efficiency with respect to computational cost, the DDG method provides an attractive alternative for solving the Navier-Stokes equations. Future study including the application of DDG method for simulating 3D high Reynolds number turbulent flows is ongoing.

Acknowledgments

The authors would like to acknowledge the National Natural Science Foundation of China No. 91530325 and No. 11601024, Fundamental Research of Civil Aircraft No. MJ-F-2012-04.

References

[1] W.H. Reed, 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, C.-W. Shu, TVD Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation, 55(1990), pp. 545-581.
[3] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional system, J. Comput. Phys., 141(1998), pp. 199-224.
[4] H. Luo, J.D. Baum, R. Löhner, A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids, J. Comput. Phys., 227(2008), pp. 8875-8893.
[5] H. Luo, L.Q. Luo, R. Nourgaliev, V.A. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids, J. Comput. Phys., 229(2010), pp. 6961-6978.
[6] B. van Leer, S. Nomura, Discontinuous Galerkin method for Diffusion, AIAA-2005-5108.
[7] B. van Leer, M. Lo, A discontinuous Galerkin method for diffusion based on recovery, AIAA-2007-4083.
[8] H.L. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems SIAM J. Numer. Anal., 47(2009), pp. 675-698.
[9] H.L. Liu, J. Yan, The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections, Commun. Comput. Phys., 8(2010), pp. 541-564.
[10] R. Kannan, Z.J. Wang, The direct discontinuous Galerkin (DDG) viscous flux scheme for the high order spectral volume method, Computers & Fluids, 39(2010), pp. 2007-2021.
[11] F. Bassi, S. Rebay, Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-w$ turbulence model equations, Computers & Fluids, 34(2005), pp. 507-540.

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

[12] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time dependent convection-diffusion systems, SIAM J. Numer. Anal., 35(1998), pp. 2440-2463.
[13] H.Y. Gao, Z.J. Wang, H.T. Huynh, Differential formulation of discontinuous Galerkin and related methods for the Navier-Stokes equations, Commun. Comput. Phys., 13(2013), pp. 1013-1044.
[14] J. Peraire, P.O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput., 30(2008), pp. 1806-1824.
[15] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of compressible Navier-Stokes equations, J. Comput. Phys., 131(1997), pp. 267-279.
[16] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods fro elliptic problems, SIAM J. Numer. Anal., 19(2002), pp. 742-760.
[17] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations, J. Comput. Phys., 227(2008), pp. 9670-9685.
[18] J. Douglas, T. Dupont, Interior penalty procedures for eplliptic and parabolic Galerkin methods, in Computing Methods in Applied Sciences, Lecture Notes in Phys., Springer, Berlin, 1976, pp. 207-216.
[19] D.N. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal., 19(1982), pp. 742-760.
[20] M.F. Wheeler, An elliptic collocation-finite element method with interior penalties, SIAM J. Numer. Anal., 15(1978), pp. 152-161.
[21] G. Gassner, F. Lörcher, C.-D. Munz, A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes, J. Comput. Phys., 224(2207), pp. 1049-1063.
[22] H. Luo, J.D. Baum, R. Löhner, A fast, matrix-free implicit method for compressible flows on unstructured grids, J. Comput. Phys., 146(1998), pp. 664-690.
[23] Y. Sun, Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids: extension to viscous flow, J. Comput. Phys., 215(2006), pp. 41-58.
[24] Y. Liu, M. Vinokur, Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, J. Comput. Phys., 212(2006), pp. 454-472.
[25] H. Luo, Y.D. Xia, S.J. Li, R. Nourgaliev, C.P. Cai, A Hermite WENO reconstruction-based discontinuous Galerkin method for the Euler equations on tetrahedral grids, J. Comput. Phys., 231(2012), pp. 5489-5503.
[26] H. Luo, Y.D. Xia, S. Spiegel, R. Nourgaliev, Z.G. Jiang, A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids, J. Comput. Phys., 236(2013), pp. 477-492.
[27] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal., 47(2009), pp. 1319-1365.
[28] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations, J. Comput. Phys., 228(2009), pp. 3232-3254.
[29] N. Kroll, ADGIMA, A European project on the development of adaptive higher-order variational methods for aerospace applications, AIAA-2009-176.
[30] The third international workshop on high-order CFD methods,

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press

https://www.grc.nasa.gov/hiocfd/.
[31] T. Leicht, R. Hartmann, Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations, J. Comput. Phys., 229(2010), pp. 7344-7360.
[32] J. Cheng, X.Q. Yang, T.G. Liu, H. Luo, A direct discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids, AIAA-2016-1344.
[33] G. Karypis, V. Kumar, Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0, 1995.
[34] X.Q. Yang, J. Cheng, C.J. Wang, H. Luo, J.T. Si, A fast, implicit discontinuous Galerkin method based on analytical Jacobians for the compressible Navier-Stokes equations, AIAA-2016-1326.
[35] P. Castonguay, High-order energy stable flux reconstruction schemes for fluid flow simulations on unstructured grids, Ph.D. thesis, Stanford University, 2012.
[36] H.L. Liu, Optimal error estimates of the direct discontinuous Galerkin method for convection-diffusion equations, Math. Comp., 84(2015), pp. 2263-2295.
[37] W.-X. Cao, H.L. Liu, Z.-M. Zhang, Superconvergence of the direct discontinuous Galerkin method for convection-diffusion equations, Numer. Methods Partial Differential Eq., doi:10.1002/num.22087, 2016.

https://doi.org/10.4208/cicp.OA-2016-0090 Published online by Cambridge University Press