Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **1**, No. 3, pp. 301-318 (2009)

June 2009

A Discontinuous Galerkin Method Based on a BGK Scheme for the Navier-Stokes Equations on Arbitrary Grids

Hong Luo[1,*], Luqing Luo[1] and Kun Xu[2]

[1] *Department of Mechanical and Aerospace Engineering North Carolina State University, Raleigh, NC, 27695, USA*

[2] *Department of Mathematics Hong Kong University of Science and Technology, Hong Kong, China*

Received 09 December 2008; Accepted (in revised version) 12 March 2009
Available online 22 April 2009

Abstract. A discontinuous Galerkin Method based on a Bhatnagar-Gross-Krook (BGK) formulation is presented for the solution of the compressible Navier-Stokes equations on arbitrary grids. The idea behind this approach is to combine the robustness of the BGK scheme with the accuracy of the DG methods in an effort to develop a more accurate, efficient, and robust method for numerical simulations of viscous flows in a wide range of flow regimes. Unlike the traditional discontinuous Galerkin methods, where a Local Discontinuous Galerkin (LDG) formulation is usually used to discretize the viscous fluxes in the Navier-Stokes equations, this DG method uses a BGK scheme to compute the fluxes which not only couples the convective and dissipative terms together, but also includes both discontinuous and continuous representation in the flux evaluation at a cell interface through a simple hybrid gas distribution function. The developed method is used to compute a variety of viscous flow problems on arbitrary grids. The numerical results obtained by this BGKDG method are extremely promising and encouraging in terms of both accuracy and robustness, indicating its ability and potential to become not just a competitive but simply a superior approach than the current available numerical methods.

AMS subject classifications: 76M10
Key words: Discontinuous Galerkin Methods, BGK scheme, compressible Navier-Stokes equations.

1 Introduction

The accuracy of many finite-volume and finite-element methods currently used in

[*] Corresponding author.
URL: `http://www.mae.ncsu.edu/directories/faculty/luo.html`
Email: hong_luo@ncsu.edu (H. Luo), lluo2@ncsu.edu (L. Luo), makxu@ust.hk (K. Xu)

computational science and engineering is at best second order. There are a number of situations where these numerical methods do not reliably yield engineering-required accuracy. The development of a practical higher-order (>2nd) solution method could help alleviate this accuracy problem by significantly decreasing time required to achieve an acceptable error level. Unfortunately, numerous reasons exist for why current finite-volume algorithms are not practical at higher order and have remained second-order. The root cause of many of these difficulties lies in the extended stencils that these algorithms employ. By contrast, discontinuous Galerkin (DG) finite element formulation introduces higher-order effects compactly within the element. While DG was originally introduced by Reed and Hill [1] for solving the neutron transport equation back in 1973, major interest did not focus on it until the nineties [2–5]. Nowadays, it is widely used in the computational fluid dynamics, computational aeroacoustics, and computational electromagnetics, to name just a few [6–17]. The discontinuous Galerkin methods (DGM) combine two advantageous features commonly associated with finite element and finite volume methods (FVM). 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 FVM. 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 FVM. What is known so far about this method offers a tantalizing glimpse of its full potential. Indeed, what sets this method apart from the crowd is many attractive features it possesses: (1) It has several useful mathematical properties with respect to conservation, stability, and convergence. (2) The method can be easily extended to higher-order (>2nd) approximation. (3) The method is well suited for complex geometries since it can be applied on unstructured grids. In addition, the method can also handle non-conforming elements, where the grids are allowed to have hanging nodes. (4) The method is highly parallelizable, as it is 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 method. (5) It 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 method allows easy implementation of *hp*-refinement, for example, the order of accuracy, or shape, can vary from element to element. (6) It has the ability to compute low Mach number flow problems without recourse to the time-preconditioning techniques normally required for the finite volume methods.

In contrast to the enormous advances in the theoretical and numerical analysis of the DGM, the development of a viable, attractive, competitive, and ultimately superior DG method over the more mature and well-established second order methods is relatively an untouched area. This is mainly due to the fact that DGM have a number of weaknesses that have to be addressed, before they can be applied to flow problems of practical interest in a complex configuration environment. In particular, how to efficiently discretize diffusion terms required for the Navier-Stokes equations remains one of unresolved issues in the DGM. DG methods is indeed a natural choice for the solution of hyperbolic problems, such as the compressible Euler equations. However, the DG formulation is far less certain and advantageous for the compressible Navier-Stokes equations, where dissipative fluxes exist. A severe difficulty raised by the application of the DG methods to the Navier-Stokes equations is the approximation of the numerical fluxes for diffusion terms, that has to properly resolve the discontinuities at the interfaces. Taking a simple arithmetic mean of the solution derivatives from the left and right is inconsistent, because the arithmetic mean of the solution derivatives does not take in account a possible jump of the solutions. A number of numerical methods have been proposed in the literature, such as those by Bassi and Rebay [18], Cockburn and Shu [19], Baumann and Oden [20] and many others. Arnold et al. have analyzed a large class of discontinuous Galerkin methods for second-order elliptic problems in a unified formulation in [21]. All these methods have introduced in some way the influence of the discontinuities in order to define correct and consistent diffusive fluxes. More recently, van Leer and Lo [22] proposed a recovery-based DG method for the diffusion equation using the recovery principle, and Gassner et al. [23] introduced a numerical scheme based on the exact solution of the diffusive generalized Riemann problem for the discontinuous Galerkin methods. Unfortunately, all these methods seem to require substantially more computational effort than the classical continuous finite element methods, which are naturally suited for the discretization of elliptic problems.

Alternatively, the numerical fluxes at the interface for the Navier-Stokes equations in the DG methods can be evaluated using a gas-kinetic formulation [24, 25], which treats the convection and dissipation effects together. The gas-kinetic formulation uses a gas-kinetic distribution function to construct the numerical fluxes, which automatically obtain the convection and dissipation effects due to the intrinsic connection between the gas-kinetic BGK model and the Navier-Stokes equations. In the BGK formulation, the fluxes at the interface for the Euler and Navier-Stokes equations are constructed based on the integral solution of the BGK model, which requires both conservative variables and their derivatives. In this regard, the BGK formulation bears a strong resemblance to the generalized Riemann solver in the evolution of fluxes at the interface, and the initial condition theoretically can be any high order polynomials on both sides of a cell interface. The BGK formulation offers a much deeper physical insight in the construction of a DG method for the convection-diffusion problems. It should be pointed out that the BGK scheme, recognized as being expensive in comparison with the traditional upwind methods for computing numerical fluxes, makes a comeback in the context of DG formulation, as it does not require a separate computation of viscous fluxes at the interfaces.

Recently, the present authors have developed a DG method based on a Taylor basis for the solution of compressible Euler equations on arbitrary grids [13]. Unlike the traditional discontinuous Galerkin methods, where either standard Lagrange finite element or hierarchical node-based basis functions are used to represent numerical polynomial solutions in each element, this DG method represents the numerical polynomial solutions using a Taylor series expansion at the centroid of the cell. As a result, this new formulation [13] has a number of distinct, desirable, and attractive features and advantages in developing a DG method from a practical perspective, which can be effectively used to overcome some of the disadvantages of DGM. The objective of effort discussed in this paper is to further extend this high-order discontinuous Galerkin method for computing viscous flow problems using the BGK formulation based on the success and cornerstone of this DG method. The idea behind this approach is to combine the robustness of the BGK scheme with the accuracy of the DG methods in an effort to develop a more accurate, efficient, and robust method for the solution of the compressible Navier-Stokes equations in a wide range of flow regimes. BGKDG formulation is especially attractive for the Navier-Stokes equations, as there is no need to compute the viscous fluxes at the interfaces, thus significantly reducing the computational costs. The developed method is applied to compute a variety of viscous flow problems for a wide range of flow conditions, from subsonic to hypersonic flows. Our numerical results obtained by the BGKDG method are extremely promising and encouraging, demonstrating its ability and potential to become not just a competitive but simply a superior approach than either BGK-based and upwind-based finite volume methods or LDG methods.

2 Numerical method

Assume that we want to solve the following Navier-Stokes equations using a discontinuous Galerkin method

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial F_k(U(x,t))}{\partial x_k} = 0, \tag{2.1}$$

where U is the conservative variable vector, and F is the flux vector, which contains both convective and diffusive terms. To formulate the discontinuous Galerkin method, we first introduce the following weak formulation, which is obtained by multiplying the above conservation law by a test function W, integrating over the domain Ω and performing an integration by parts

$$\int_{\Omega} \frac{\partial U}{\partial t} W d_{\Omega} + \int_{\Gamma} F_k n_k d_{\Gamma} - \int_{\Omega} F_k \frac{\partial W}{\partial x_k} d_{\Omega} = 0, \qquad \forall W \in V, \tag{2.2}$$

where Γ $(=\partial\Omega)$ denotes the boundary of Ω, and n_j the unit outward normal vector to the boundary. Assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e, which can be triangles, quadrilaterals, polygons, or their combinations in 2D and tetrahedral, prism, pyramid, and hexahedral or their combinations in 3D. We introduce the following broken Sobolev space V_h^p

$$V_h^p = \{v_h \in [L_2(\Omega)]^m : v_h|_{\Omega_e} \in [V_p^m],\ \forall \Omega_e \in \Omega\}, \tag{2.3}$$

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

$$V_p^m = \text{span}\{\prod x_i^{\alpha_i} : 1 \le \alpha_i \le p,\ 0 \le i \le d\}, \tag{2.4}$$

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. Find $U_h \in V_h^p$ such as

$$\frac{d}{dt}\int_{\Omega_e} U_h W_h d_\Omega + \int_{\Gamma_e} F_k(U_h) n_k W_h d_\Gamma - \int_{\Omega_e} F_k(U_h)\frac{\partial W_h}{\partial x_k} d_\Omega, \quad \forall W_h \in V_h^p, \tag{2.5}$$

where U_h and W_h represent the finite element approximations to the analytical solution U and the test function W respectively, and are approximated by piecewise polynomial function of degrees p, which are discontinuous between the cell interfaces. Assume that B is the basis of polynomial function of degrees p, this is then equivalent to the following system of N equations

$$\frac{d}{dt}\int_{\Omega_e} U_h B_i d_\Omega + \int_{\Gamma_e} F_k(U_h) n_k B_i d_\Gamma - \int_{\Omega_e} F_k(U_h)\frac{\partial B_i}{\partial x_k} d_\Omega, \quad 1 \le i \le N, \tag{2.6}$$

where N is the dimension of the polynomial space.

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

$$U_h = \sum_{i=1}^{N} U_i(t) B_i(x), \tag{2.7}$$

where B_i are the finite element basis functions. As a result, the unknowns to be solved are the variables at the nodes U_i, as illustrated in Fig. 1 for linear and quadratic polynomial approximations.

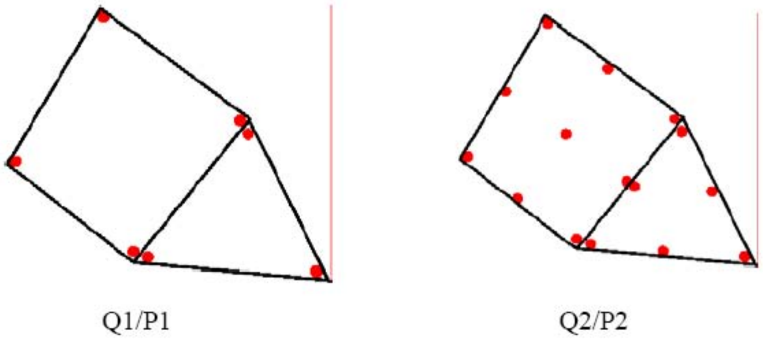

Figure 1: Representation of polynomial solutions using finite element shape functions.

On each cell, a system of $N \times N$ has to be solved, where polynomial solutions are dependent on the shape of elements. For example, for a linear polynomial approximation in 2D as shown in Fig. 1, a linear polynomial is used for triangular elements and the unknowns to be solved are the variables at the three vertices and a bi-linear polynomial is used for quadrilateral elements and the unknowns to be solved are the variables at the four vertices. However, the numerical polynomial solutions 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 centroid of the cell. For example, if we do a Taylor series expansion at the centroid of the cell, the quadratic polynomial solutions can be expressed as following

$$\begin{aligned} U_h = U_c + \frac{\partial U}{\partial x}|_c(x-x_c) + \frac{\partial U}{\partial y}|_c(y-y_c) + \frac{\partial^2 U}{\partial x^2}|_c\frac{(x-x_c)^2}{2} \\ +\frac{\partial^2 U}{\partial y^2}|_c\frac{(y-y_c)^2}{2} + \frac{\partial^2 U}{\partial x \partial y}|_c(x-x_c)(y-y_c), \end{aligned} \tag{2.8}$$

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

$$\begin{aligned} U_h = \tilde{U} + \frac{\partial U}{\partial x}|_c(x-x_c) + \frac{\partial U}{\partial y}|_c(y-y_c) + \frac{\partial^2 U}{\partial x^2}|_c\Big[\frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d_\Omega\Big] \\ +\frac{\partial^2 U}{\partial y^2}|_c\Big[\frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d_\Omega\Big] \\ +\frac{\partial^2 U}{\partial x \partial y}|_c\Big[(x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d_\Omega\Big], \end{aligned} \tag{2.9}$$

where $\tilde{U}$ is the mean value of U in this cell. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes, as shown in Fig. 2.

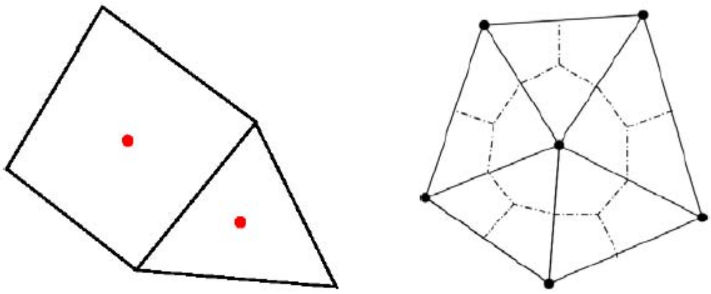

Figure 2: Representation of polynomial solutions using a Taylor series expansion for a cell-centered scheme (left) and vertex-centered scheme (right).

In this case, the dimension of the polynomial space is six and the six basis functions are $B_1 = 1$, $B_2 = x - x_c$, $B_3 = y - y_c$ and

$$\begin{aligned} &B_4 = \frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d_\Omega, \quad B_5 = \frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d_\Omega, \\ &B_6 = (x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d_\Omega, \end{aligned} \tag{2.10}$$

and the discontinuous Galerkin formulation then leads to the following six equations

$$\frac{d}{d_t}\int_{\Omega_e}\tilde{U}d_\Omega+\int_{\Gamma_e}F_k(U_h)n_k d_\Gamma=0,\qquad i=1, \tag{2.11a}$$

$$M_{5\times5}\frac{d}{d_t}\Big(\frac{\partial U}{\partial x}|_c\quad \frac{\partial U}{\partial y}|_c\quad \frac{\partial^2 U}{\partial x^2}|_c\quad \frac{\partial^2 U}{\partial y^2}|_c\quad \frac{\partial^2 U}{\partial x\partial y}|_c\Big)^T+R_{5\times1}=0. \tag{2.11b}$$

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

$$\int_{\Omega_e}B_1B_i d_\Omega=0,\qquad 2\le i\le 6. \tag{2.12}$$

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix. The details can be found in [13]. This is especially helpful and important to alleviate the stiffness of the system matrix for higher-order DG approximations.

Theoretically, this formulation allows us to clearly see the similarity and difference between DG and FV methods, and provides the evidence why the DG methods are superior to their FV counterparts. In fact, the discretized governing equations for the cell-averaged variables and the assumption of polynomial solutions on each cell are exactly the same for both methods. In other words, this DG method provides a unified formulation, where the existing finite volume methods can be recovered virtually. For example, the application of this DG method to the median dual control volume of a given mesh will lead to the classic vertex-centered finite volume scheme as shown in the right of Fig. 2, while the application of this DG method to the cell itself of any given mesh will lead to the classic cell-centered finite volume scheme as shown in the left of Fig. 2. The only difference between them is the way how they obtain high-order (>1st) polynomial solutions. In the finite volume methods, the polynomial solution of degrees p are reconstructed using information from cell-averaged values of the flow variables, which can be obtained using either TVD/MUSCL or ENO/WENO reconstruction schemes. Unfortunately, the multi-dimensional MUSCL approach, which are praised to achieve high-order accuracy for multi-dimensional problems, suffer from two shortcomings in the context of unstructured grids: (1) uncertainty and arbitrariness in choosing the stencils and methods to compute the gradients in the case of linear reconstruction; This explains why a nominally second-order finite volume scheme is hardly able to deliver a formal solution of second order accuracy in practice for unstructured grids. The situation becomes more evident, severe, and profound, when a highly stretched tetrahedral grid is used in the boundary layers. Many studies, as reported by many researchers [27–29] have demonstrated that it is difficult to obtain a second-order accurate flux reconstruction on highly stretched tetrahedral grids and that for the discretization of inviscid fluxes, the classic 1D-based upwind schemes using median-dual finite volume approximation suffer from excessive numerical diffusion due to such skewing. (2) Extended stencils required for the reconstruction of higher-order (>1st) polynomial solutions. This is exactly the reason why the current finite-volume methods using the TVD/MUSCL reconstruction are not practical at higher order and have remained second-order on unstructured grids. When the ENO/WENO reconstruction schemes are used for the construction of a polynomial of degree p on unstructured grids, the dimension of the polynomial space $N{=}N(p,d)$ depends on the degree of the polynomials of the expansion p, and the number of spatial dimensions d. One must have three, six, and ten cells in 2D and four, ten, and twenty cells in 3D for the construction of a linear, quadratic, cubic Lagrange polynomial, respectively. Undoubtedly, it is an overwhelmingly challenging, if not practically impossible, task to judiciously choose a set of admissible and proper stencils that have such a large number of cells on unstructured grids especially for higher order polynomials and higher dimensions. This explains why the application of higher-order ENO/WENO methods hardly exists on unstructured grids, in spite of their tremendous success on structured grids and their superior performance over the MUSCL/TVD methods. Unlike the FV methods, where the derivatives are reconstructed using the mean values of the neighboring cells, the present DG method computes the derivatives in a manner similar to the mean variables. This is compact, rigorous, and elegant mathematically in contrast with arbitrariness characterizing the reconstruction schemes with respect how to compute the derivatives and how to choose the stencils used in the FV methods. It is our believe that this is one of the main reasons why the second order DG methods are more accurate than the FV methods using either TVD/MUSCL or ENO/WENO reconstruction schemes, which has been demonstrated numerically. Furthermore, the higher order DG methods can be easily constructed by simply increasing the degree p of the polynomials locally, in contrast to the finite volume methods which use the extended stencils to achieve higher order of accuracy. Many other methods such as ADER scheme [26], compact finite differencing scheme [27], and CE/SE scheme [28] also solve the governing equations for the derivatives instead of using the reconstruction schemes.

Practically, this formulation has a number of attractive, distinct, and useful features in the context of DGM, that can be exploited to develop an accurate and robust DG method for conservation laws. First, cell-averaged variables and their derivatives are handily available in this formulation. This makes implementation of WENO limiter straightforward and efficient [15], which is required to eliminate nonphysical oscillations in the vicinity of discontinuities. Secondly, the basis functions are hierarchic. This greatly facilitates implementation of p-multigrid methods [16, 17] and p-refinement. Thirdly, cell-averaged variable equations are decoupled from their derivatives equations in this formulation. This makes development of fast, low-storage implicit methods possible. Last, the same numerical polynomial solutions are used for any shapes of elements, which can be triangle, quadrilateral, and polygon in 2D, and tetrahedron, pyramid, prism, and hexahedron in 3D. Using this formulation, DGM can be easily implemented on arbitrary meshes. The numerical method based on this formulation has the ability to compute 1D, 2D, and 3D problems using the very same code, which greatly alleviates the need and pain for code maintenance and upgrade.

Furthermore, it is relatively simple and straightforward to convert an existing second order finite volume code into a higher-order DG code with minimal modification, which can be achieved by solving the equations for the derivatives instead of using the reconstruction schemes. This can be done for either cell-centered or vertex-centered scheme finite volume methods. It should be stressed that the property of this DG formulation is especially important and desirable for the wide acceptance of DG methods in industries from an application point of view, as any industries are reluctant to rewrite a DG-based code from scratch, considering the investments in their existing production codes based on a second-order finite volume method.

By taking full advantage of these features, we have developed an accurate, efficient, and robust DG method for the numerical solution of the compressible Euler equations on arbitrary grids [13]. In our implementation of this DG method, an accurate representation of the boundary normals based on the definition of the geometries is used for imposing solid wall boundary conditions for curved geometries [14]. A weighted essentially non-oscillatory reconstruction scheme based on Hermite polynomials [15] is used and applied as a limiter for the discontinuous Galerkin finite element method on unstructured grids. A physics-based shock detector [14] is introduced to effectively make a distinction between a smooth extremum and a shock wave. The limiter can be chosen to only apply to these regions identified by this shock detector in order to reduce the computational cost and maintain the high order accuracy of the DG methods. A fast, low-storage p-multigrid method [16, 17] is used to obtain steady state solutions, and the explicit three-stage third-order TVD Runge-Kutta scheme is used to advance solution in time for the unsteady flow problems. A distinct feature of this p-multigrid method [16, 17] is the application of an explicit smoother [2] on the higher level approximations ($p>0$) and an implicit smoother [29,30] on the lowest level approximation ($p=0$), resulting in a fast as well as low storage method that can be efficiently used to accelerate the convergence to a steady state solution. Furthermore, this p-multigrid method can be naturally applied to compute the flows with discontinuities, where a monotonic limiting procedure is usually required for discontinuous Galerkin methods.

Using the BGK formulation, the extension of this DG method for solving the compressible Navier-Stokes equations is simple and straightforward, as the computation of the numerical fluxes at the element interfaces is performed through a simple hybrid gas distribution function. There are two approaches that can be used to construct the numerical fluxes: one is the directional splitting method [31] and the other is the fully multidimensional method [24]. The less costly directional splitting method is used to construct the numerical fluxes at the interfaces for the Navier-Stokes equations. In this approach, the partial derivatives of the conservative variables in the direction normal to the cell interfaces, as well as the flow variables on the right and left of the interfaces will be used in the flux evaluation. It is worth to point out the fundamental difference between the BGK scheme and the upwind methods for the construction of Riemann flux function. Upwind schemes only use the flow variables on the left and right of cell interfaces to construct the Euler flux, while the BGK scheme requires both flow variables and their derivatives in order to evaluate the numerical fluxes at the interfaces. This explains why the fluxes obtained by the BGK scheme contain both the convective and dissipative terms, which is similar to the generalized Riemann solver in the evolution of fluxes at the interface. However, the BGK formulation offers a much deeper physical insight in the construction of a DG method for the convection-diffusion problems. A remarkable feature of the gas-kinetic BGK scheme in the context of the DG formulation is that a Navier-Stokes flux at the cell interfaces is obtained directly from the the flow variables and their derivatives, avoiding a separate computation of viscous fluxes at the interfaces normally required in the LDG formulation and alike approach.

3 Examples

A few examples are presented here to illustrate the high accuracy and robustness of this DG method for a wide range of flow regimes from subsonic to hypersonic.

3.1 Accuracy test for 1D Navier-Stokes equations

The first test is to solve the NavierCStokes equations with the following initial data:

$$\rho(x,0) = 1 + 0.2\sin(\pi x), \quad u(x,0) = 1, \quad p(x,0) = 1. \tag{3.1}$$

The dynamical viscosity coefficient is μ=0.0005, the Prandtl number is P_r=2/3, and the specific heat ratio is c=5/3. The computational domain is $[0,2]$ and the periodic boundary condition is used. We compute the viscous solution up to time t=2 with a

Table 1: The error and convergence order for P1 case.

N	L^∞-error	Order	L^1-error	Order	L^2-error	Order
10	3.05E-2	-	1.76E-2	-	1.99E-2	-
20	5.68E-3	2.42	3.36E-3	2.39	3.79E-3	2.39
40	1.03E-3	2.46	6.31E-4	2.41	7.07E-4	2.42
80	2.08E-4	2.31	1.28E-4	2.30	1.44E-4	2.30

Table 2: The error and convergence order for P2 case.

N	L^∞-error	Order	L^1-error	Order	L^2-error	Order
10	2.48E-3	-	1.51E-3	-	1.62E-3	-
20	2.76E-4	3.16	1.66E-4	3.18	1.86E-4	3.12
40	2.50E-5	3.47	1.54E-5	3.43	1.73E-5	3.42
80	2.54E-6	3.30	1.47E-6	3.39	1.64E-6	3.40

Table 3: The error and convergence order for P3 case.

N	L^∞-error	Order	L^1-error	Order	L^2-error	Order
10	9.05E-5	-	5.37E-5	-	5.67E-5	-
20	8.89E-6	3.35	4.23E-6	3.67	4.90E-6	3.53
40	4.90E-7	4.18	2.81E-7	3.91	3.17E-7	3.95
80	3.26E-8	3.91	1.42E-8	4.30	1.70E-8	4.22

small time step to guarantee that the spatial discretization error dominates. No limiter is used in this case. Since there is no exact solution for this problem, we evaluate the numerical error between the solutions by two successively refined meshes and use the error to estimate the numerical convergence rates. The results are shown in Tables 1, 2 and 3. From these results we can clearly notice that a $(k+1)^{th}$-order convergence rate can be obtained for P_k (k=1, 2, 3) schemes for smooth solutions.

3.2 Laminar flow past a flat plate

The laminar boundary layer over a unit flat plate is considered in this test case. This simple problem is chosen to assess the accuracy of the numerical solution obtained by the BGKDG method for solving the Navier-Stokes equations. The mesh used to compute the flat plate boundary layer is shown in Fig. 3 and contains 120 cells and 30 cells in the x and y-directions, respectively. The numerical solution is presented at a Mach number of 0.2, and Reynolds number of 100,000 based on the freestream velocity and the length of the flat plate using DG (P1) and DG (P2) methods. Fig. 4 compares the profiles of velocity component in the x-direction at five locations obtained by DG (P1) and DG (P2) solutions with Blasius solution, respectively, while the velocity profiles in the y-direction obtained by DG (P1) and DG (P2) solutions are compared with Blasius solution in Fig. 5. Both DG (P1) and DG (P2) solutions resolve boundary layers very accurately, even with as few as four cells in the boundary layer. What demonstrates the high accuracy of DG solutions is that they give the accurate prediction of velocity profiles not only in the x-direction, but also in the y-direction, which is extremely difficult to compute accurately. The computed skin friction coefficient along the flat plate is shown in Fig. 4, which is in a good agreement with the Blasius solution, indicating the high accuracy of DG methods for computing NS solutions.

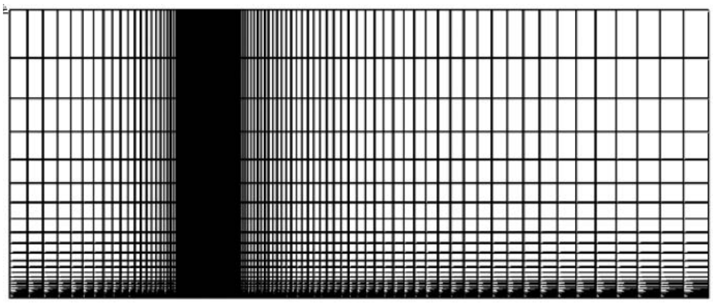

Figure 3: Mesh used for computing the laminar flow past a flat plate.

3.3 Shock boundary layer interactions

The interaction of an oblique shock wave with a laminar boundary layer is considered in this test case. The shock makes a 32.6° angle with the wall, which is located at y=0 and x>0, and hits the boundary layer on the wall at Xs=10. The Mach number of

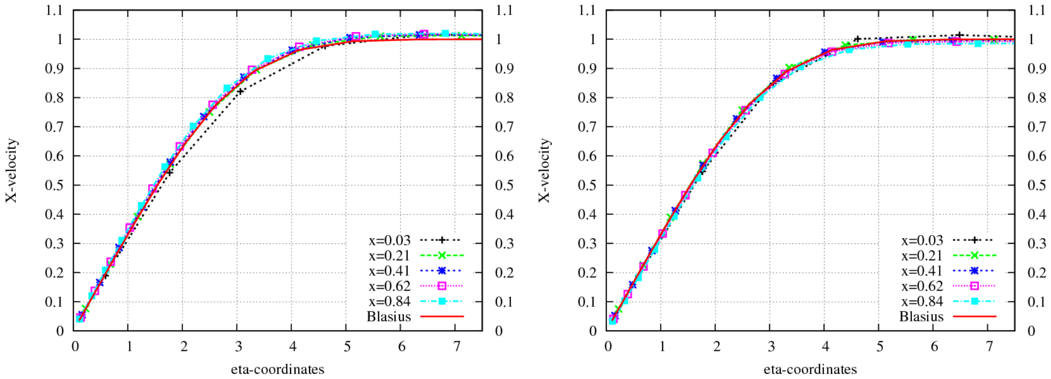

Figure 4: Comparison of the velocity profiles in the x-direction at different x-locations obtained using the DG (P1) (left) and DG (P2) (right) solutions with Blasius solution.

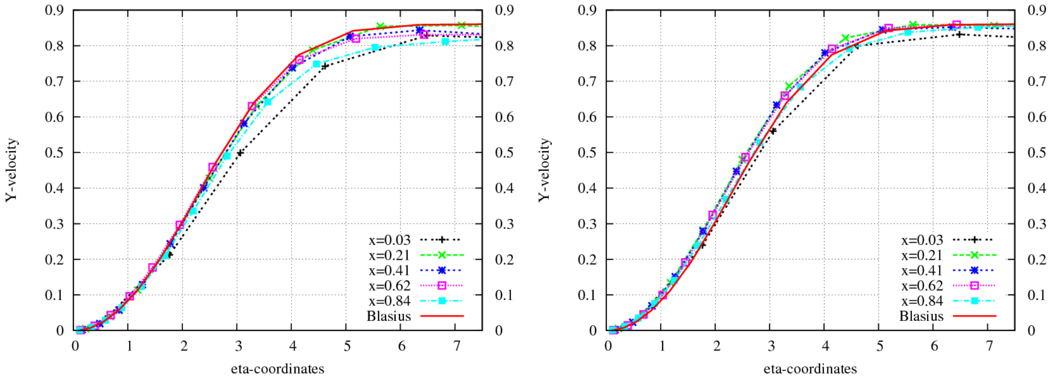

Figure 5: Comparison of the velocity profiles in the y-direction at different x-locations obtained using the DG (P1) (left) and DG (P2) (right) solutions with Blasius solution.

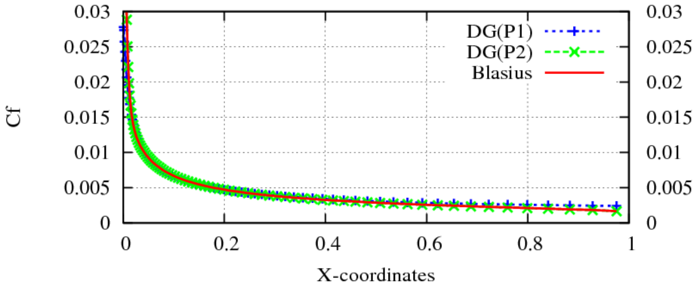

Figure 6: Comparison of computed skin friction coefficient distribution along the flat plate obtained using DG solutions with the Blasius solution.

the shock wave is equal to 2 and the Reynolds number based on the upstream flow condition and the characteristic length X_s is equal to 2.96×10^5. The Sutherland's law is used to compute the dynamic viscosity. The computation was conducted on a rectangular domain $[-1.05\leq x\leq16.09]\times[0\leq y\leq10.16]$. A nonuniform mesh with 106×73 cells, similar to the laminar boundary layer problem, is used in the computation. The computed pressure contours obtained by DG (P1) and DG (P2) solutions are shown in Fig. 7. As expected, the DG (P2) solution produces a sharper numerical shock transition than the one from DG (P1) solution due to less numerical dissipation introduced

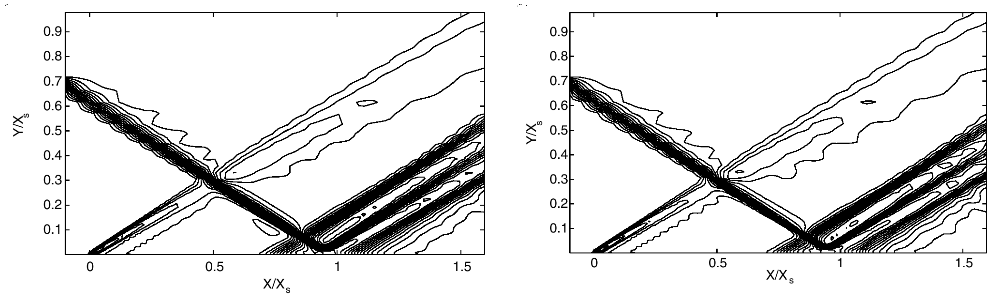

Figure 7: Computed pressure contours in the flow field for shock boundary layer interaction obtained by the DG (P1) (left) and DG (P2) (right) solutions.

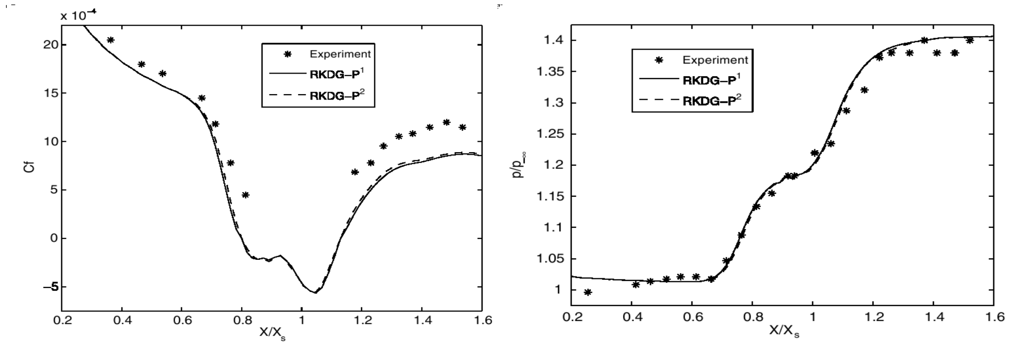

Figure 8: Computed skin friction (left) and pressure (right) distributions on the surface of the plate for the shock boundary layer interaction.

by weaker discontinuities at the cell interfaces. The skin friction and pressure distributions at the plate surface are shown in Fig. 8, where a fair agreement between the current numerical results and experimental data [32] is observed for both DG (P1) and DG (P2) solutions, and the DG (P2) solution performs slightly better than the DG (P1) solution. The discrepancy between the experimental and numerical results is mainly due to the assumption that the flow is laminar whereas the real physical one could be turbulent.

3.4 Lid-driven cavity problem

The two-dimensional lid-driven cavity flow problem was studied extensively and served as a benchmark test case for the incompressible Navier-Stokes calculations. This test case is chosen to study the accuracy and effectiveness of the DG method for computing low Mach number flows. The grid used in the computation is shown in Fig. 9. It contains 5398 elements, 2800 grid points and 200 boundary points. The computation is performed at a Mach number of 0.05 and a Reynolds number of 400 using DG (P1) method. Fig. 10 shows the computed velocity in the flow field. The u-velocity component along the vertical centerline is shown in Fig. 11. The solution of Ghia et al. [33], which is considered a standard benchmark solution, used a very fine 129×129 grid points, and is also shown for comparison. The virtually identical agreement indicates that the DG method had the ability of obtaining an accurate solution to

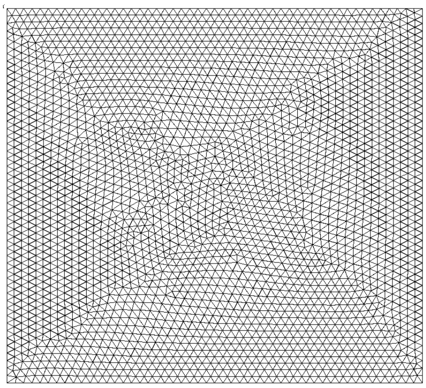

Figure 9: Mesh for lid-driven cavity problem.

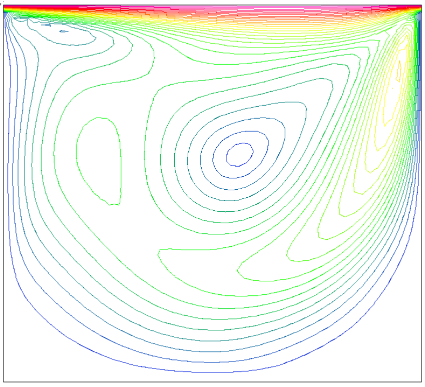

Figure 10: Computed velocity contours for lid-driven cavity problem at M_∞=0.05, R_e=400.

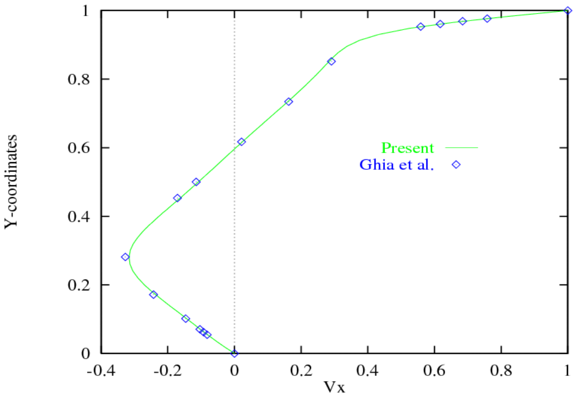

Figure 11: Computed x-direction velocity distribution along vertical geometric centerline at M_∞=0.05, R_e=400.

the low Mach number flows without recourse to the time-preconditioning techniques normally required for the finite volume methods.

3.5 Hypersonic laminar flow past a circular cylinder

This test case, taken from the experiment done by Wieting [34], where the flow condition is given as M_∞=8.03, T_∞=124.94K, T_W=294.44K, $R_e=1.835\times10^5$, is chosen to demonstrate the robustness of the BGKDG method for accurate and reliable prediction of heat flux in the hypersonic regime. The application of the LDG method (Bassi-Rebay II scheme) for this hypersonic flow problem is not encouraging, as shown in Fig. 12, where the mesh used in the computation, the computed pressure and temperature contours in the flow field, are presented. Note that different approximate Riemann solvers have been tried to compute the inviscid fluxes in an attempt to improve the numerical solutions. However, all solutions obtained are not as accurate as the one obtained using the BGKDG method, as demonstrated in Fig. 13 where the computed pressure, Mach number and temperature contours in the flow field are shown. Fig. 14 compares the computed normalized pressure and heat flux at the cylindrical surface

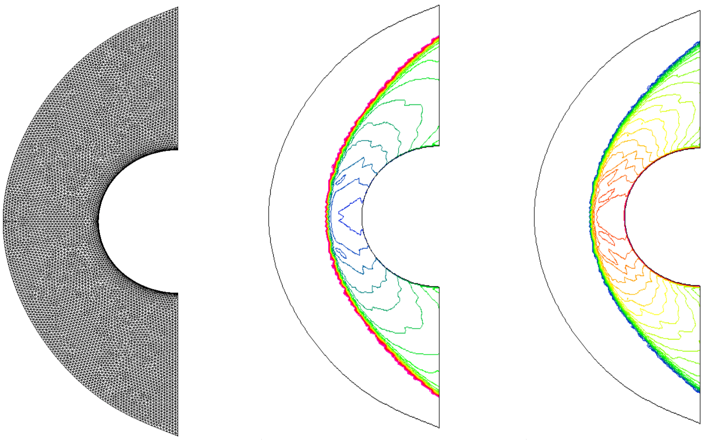

Figure 12: Computed mesh (left), computed Mach number (middle) and temperature (right) contours in the flow field obtained using LDG (P1) solution.

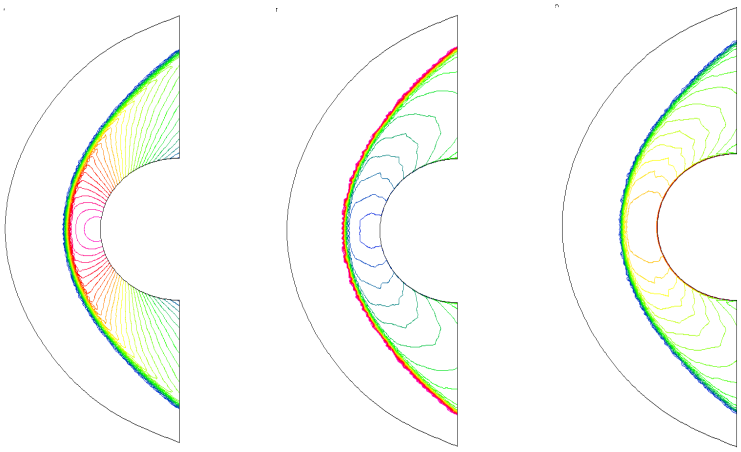

Figure 13: Computed pressure (left), Mach number (middle) and temperature (right) contours in the flow field obtained using BGKDG (P1) solution.

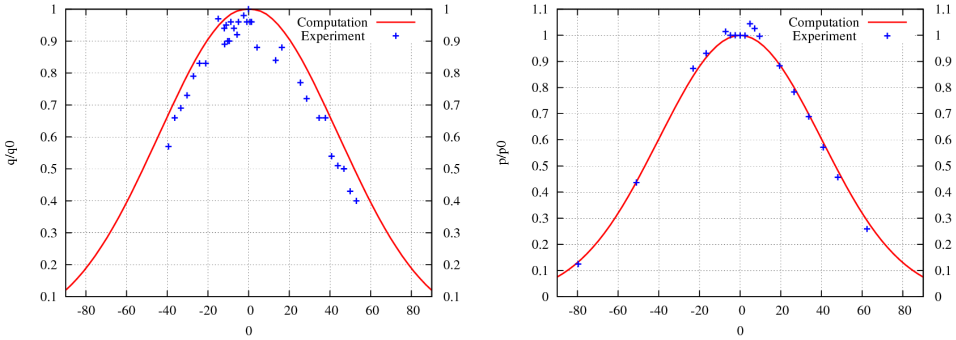

Figure 14: Comparison of the computed head flux (left) and pressure distributions obtained by the BGKDG solution along the cylindrical surface with the experimental data.

with the experimental data, where a fairly good agreement can be observed. This example clearly indicates the potential and promise of the DG method for accurate and reliable prediction of heat flux in the hypersonic regime.

4 Conclusions and Outlook

A discontinuous Galerkin method based on a Taylor basis has been extended for solving the compressible Navier-Stokes equations on arbitrary grids. Unlike the traditional discontinuous Galerkin methods which normally use a local discontinuous Galerkin formulation to discretize the viscous fluxes, the present DG method uses BGK formulation for the discretization of the Navier-Stokes equations, which has the ability of treating both convective and dissipative effects together using a gas-kinetic distribution function. As a result, there is no need to compute the viscous fluxes at the interfaces separately, thus significantly reducing the computational costs. The developed method has been used to compute a variety of viscous flow problems on arbitrary grids. The numerical results obtained by the BGKDG method are extremely promising and encouraging, indicating its ability and potential to become not just a competitive but simply a superior approach than the current available numerical methods. Further effort will be focused on conducting a systematic study on accuracy, convergence, and cost between BGKDG and LDG methods for solving the Navier-Stokes equations and extending this BGKDG method for three dimensional problems.

Acknowledgments

The first author would like to acknowledge the financial support provided by NCSU new faculty start-up fund and NCSU Faculty Research and Development Fund.

References

[1] 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 V: the Multidimensional Case*, Mathematics of Computation, 55 (1990), pp. 545–581.

[3] B. COCKBURN AND C. W. SHU, *The Runge-Kutta discontinuous Galerkin method for conservation laws IV: multidimensional system*, J. Comput. Phys., 141 (1998), pp. 199–224.

[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, 11 (2000), pp. 5–50.

[5] F. BASSI AND S. REBAY, *High-order accurate discontinuous finite element solution of the 2D Euler equations*, J. Comput. Phys., 138 (1997), pp. 251–285.

[6] H. L. ATKINS AND C. W. SHU, *Quadrature free implementation of discontinuous Galerkin method for hyperbolic equations*, AIAA Journal, 36 (1998), No. 5.

[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, 11 (2000), pp. 197–208.

[8] T. C. Warburton and G. E. Karniadakis, *A discontinuous Galerkin method for the viscous MHD equations*, J. Comput. Phys., 152 (1999), pp. 608–641.

[9] J. S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Texts in Applied Mathematics, 56 (2008).

[10] P. Rasetarinera and M. Y. Hussaini, *An efficient implicit discontinuous spectral Galerkin Method*, J. Comput. Phys., 172 (2001), pp. 718–738.

[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*, J. Comput. Phys., No. 1, 207 (2005), pp. 92–113.

[13] H. Luo, J. D. Baum and R. Löhner, *A discontinuous Galerkin method using Taylor Basis for compressible flows on arbitrary grids*, J. Comput. Phys., 227 (2008), pp.8875-8893.

[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, No. 5, 73 (2008), pp. 597–623.

[15] H. Luo, J. D. Baum and R. Löhner, *A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids*, J. Comput. Phys., No. 1, 225 (2007), pp. 686–713.

[16] H. Luo, J. D. Baum and R. Löhner, *A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids*, J. Comput. Phys., No. 2, 211 (2006), pp. 767–783.

[17] H. Luo, J. D. Baum and R. Löhner, *A fast, p-multigrid discontinuous Galerkin method for compressible flows at all speeds*, AIAA Journal, 46 (2008), No.3, pp.635-652.

[18] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267–279.

[19] B. Cockburn and C. W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion system*, SIAM, J. Numer. Ana., 16 (2001).

[20] 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, 31 (1999).

[21] D. N. Arnold, F. Brezzi, B. Cockburn and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM Journal on Numerical Analysis, No. 5., 39 (2002), pp. 1749–1779.

[22] B. van Leer and M. Lo, *A discontinuous Galerkin method for diffusion based on recovery*, AIAA Paper 2007-4083, 2007.

[23] G. Gassner, F. Lorcher and C. D. Munz, *A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes*, J. Comput. Phys., No. 2, 224 (2007), pp. 1049-1063.

[24] K. Xu, M. Mao and L. Tang, *A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow*, J. Comput. Phys., No. 2, 203 (2005), pp. 405–421.

[25] H. Liu and K. Xu, *A Runge-Kutta discontinuous Galerkin method for viscous flow equations*, J. Comput. Phys., In press, 2007.

[26] V.A. Titarev and E. F. Toro, *ADER schemes for three-dimensional nonlinear hyperbolic systems*, J. Comput. Phys., 204 (2005), pp. 715-736.

[27] M. R. Visbal and D. V. Gaitonde, *On the use of higher-order finite-difference schemes on curvilinear and deforming meshes*, J. Comput. Phys., No. 1, 181 (2002), pp. 155–185.

[28] S. C. Chang, *The method of space-time conservation element and solution element- a new approach for solving the Navier-Stokes and Euler equations*, J. Comput. Phys., No. 2, 110 (1995),

pp. 259–324.

[29] H. Luo, J. D. Baum and R. Löhner, *A fast, matrix-free implicit method for compressible flows on unstructured grids*, J. Comput. Phys., No. 2, 146 (1998), pp. 664–690.

[30] H. Luo, D. Sharov, J. D. Baum and R. Löhner, *A class of matrix-free implicit methods for compressible flows on unstructured grids*, Proceedings of the First International Conference on Computational Fluid Dynamics, Kyoto, Japan, 10-14, July 2000.

[31] K. Xu, *A gas kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method*, J. Comput. Phys., No. 1, 171 (2001), pp. 289–335.

[32] R. J. Hakkinen, L. Greber, L. Trilling and S. S. Abarbanel, The Intersection of an Oblique Shock Wave with a Laminar Boundary Layer, NASA Memo. 2-18-59W, NASA, 1959.

[33] U. Ghia, K. N. Ghia and C. T. Shin, *High-resolution for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys., 48 (1982), pp. 87–411.

[34] A. R. Wieting, Experimental Study of Shock Wave Interface Heating on a Cylindrical Leading Edge, NASA TM-100484, 1987.