A discontinuous Galerkin method using Taylor basis for computing shock waves on arbitrary grids

H. Luo[1], J.D. Baum[2], and R. Löhner[3]

[1] *Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695 (USA)*
[2] *Center for Applied Computational Sciences, Science Applications International Corporation, McLean, VA 22102 (USA)*
[3] *School of Computational Sciences, George Mason University, Fairfax, VA 22030 (USA)*

1 Introduction

The discontinuous Galerkin methods [1] (DGM) have recently become popular for the solution of systems of conservation laws to arbitrary order of accuracy. The DGM combine two advantageous features commonly associated to finite element and finite volume methods. As in classical finite element methods, accuracy is obtained by means of high-order polynomial approximation within an element rather than by wide stencils as in the case of finite volume methods. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces.

In the traditional DG methods either standard Lagrange or hierarchical node-based finite element basis functions are used to represent numerical polynomial solutions in each element. As a result, the unknowns to be solved are the variables at the nodes and the polynomial solutions are dependent on the shape of elements. In this work, a DG formulation based a Taylor basis is presented for the solution of the compressible Euler equations on arbitrary grids, where the numerical polynomial solutions are represented using a Taylor series expansion at the centroid of the cell. The unknown variables to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes. Consequently, this formulation is able to offer the insight why the DG methods are a better approach than the finite volume methods based on either TVD/MUSCL reconstruction or essentially non-oscillatory (ENO)/weighted essentially non-oscillatory (WENO) reconstruction, and has a number of distinct, desirable, and attractive features. The developed method is used to compute a variety of shock wave problems on arbitrary grids. The numerical results obtained demonstrated the superior accuracy of this DG method in comparison with a second order finite volume method and a third order WENO method, indicating its promise and potential to become not just a competitive but simply a superior approach than its finite volume and ENO/WENO counterparts for computing shock waves of of scientific and industrial interest.

2 Numerical Method

The Euler equations governing unsteady compressible inviscid flows can be expressed in conservative form as

$$\frac{\partial \mathbf{U}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{F}_j(\mathbf{U}(\mathbf{x},t))}{\partial x_j} = 0, \qquad \text{in } \Omega \tag{1}$$

where, Ω is a bounded connected domain in $\mathbf{R}^d$, d is the number of spatial dimension, and conservative state vector $\mathbf{U}$ and inviscid flux vectors $\mathbf{F}$ are defined by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix}, \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho e + p) \end{pmatrix}, \tag{2}$$

where the summation convention has been used and ρ, p, and e denote the density, pressure, and specific total energy of the fluid, respectively, and u_i is the velocity of the flow in the coordinate direction x_i. To formulate the DG method, we first introduce the following weak formulation of (1), which is obtained by multiplying (1) by a test function $\mathbf{W}$, integrating over the domain Ω, and performing an integration by parts:

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\mathbf{W}d\Omega + \int_\Gamma \mathbf{F}_j\mathbf{n}_j\mathbf{W}d\Gamma - \int_\Omega \mathbf{F}_j\frac{\partial \mathbf{W}}{\partial x_j}d\Omega = 0, \tag{3}$$

where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω, and $\mathbf{n}_j$ 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

$$V_h^p = \{v_h \in [L_2(\Omega)]^m : v_h \mid_{\Omega_e} \in [V_p^m] \forall \Omega_e \in \Omega\}, \tag{4}$$

which consists of discontinuous vector-valued polynomial functions of degree $p \leq 0$, and where m is the dimension of conservative state vector and

$$V_p^m = \text{span}\left\{\prod_{i=1}^{d} x_i^{\alpha_i} : 0 \leq \alpha_i \leq p, 0 \leq i \leq d\right\}, \tag{5}$$

where α denotes a multi-index. Then, we can obtain the following semi-discrete form by applying weak formulation on each element Ω_e

$$\begin{cases} \text{find } \mathbf{U}_h \in V_h^p \text{ such as} \\ \frac{d}{dt}\int_{\Omega_e} \mathbf{U}_h\mathbf{W}_h d\Omega + \int_{\Gamma_e} \mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j\mathbf{W}_h d\Gamma - \int_{\Omega_e} \mathbf{F}_j(\mathbf{U}_h)\frac{\partial \mathbf{W}_h}{\partial x_j}d\Omega = 0 \quad \forall \mathbf{W}_h \in V_h^p, \end{cases} \tag{6}$$

where $\Gamma_e(=\partial\Omega_e)$ denotes the boundary of Ω_e, $\mathbf{U}_h$ and $\mathbf{W}_h$ represent the finite element approximations to the analytical solution $\mathbf{U}$ and the test function $\mathbf{W}$, respectively, and both belong to the finite element space V_h^p. Assume that B_i is the basis of polynomial function of degrees p, this is then equivalent to the following system of N equations,

$$\frac{d}{dt}\int_{\Omega_e} \mathbf{U}_h B_i d\Omega + \int_{\Gamma_e} \mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j B_i d\Gamma - \int_{\Omega_e} \mathbf{F}_j(\mathbf{U}_h)\frac{\partial B_i}{\partial x_j}d\Omega = 0 \quad 1 \leq i \leq \text{N}, \tag{7}$$

where N is the dimension of the polynomial space. In the traditional DGM, numerical polynomial solutions $\mathbf{U}$ in each element are represented using either standard Lagrange finite element or hierarchical node-based basis as following

$$\mathbf{U}_h = \sum_{i=1}^{\text{N}} \mathbf{U}_i(t)B_i(\mathbf{x}). \tag{8}$$

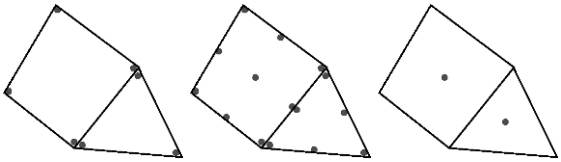

Fig. 1. Representation of polynomial solutions using finite element shape functions (two on the left) and a Taylor series expansion (one on the right)

As a result, the unknowns to be solved are the variables at the nodes $\mathbf{U}_i$, as illustrated in Fig. 1 for linear and quadratic polynomial approximations. On each cell, a system of NxN 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 $\mathbf{U}$ can be expressed in other forms as well. In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the centroid of the cell, where the quadratic polynomial solutions, for example, can be expressed as following

$$\begin{aligned}\mathbf{U}_h = \mathbf{U}_c &+ \frac{\partial \mathbf{U}}{\partial x}\,|_c\,(x-x_c) + \frac{\partial \mathbf{U}}{\partial y}\,|_c\,(y-y_c) \\ &+ \frac{\partial^2 \mathbf{U}}{\partial x^2}\,|_c\,\frac{(x-x_c)^2}{2} + \frac{\partial^2 \mathbf{U}}{\partial y^2}\,|_c\,\frac{(y-y_c)^2}{2} + \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\,|_c\,(x-x_c)(y-y_c),\end{aligned} \tag{9}$$

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

$$\begin{aligned}\mathbf{U}_h = \tilde{\mathbf{U}} &+ \frac{\partial \mathbf{U}}{\partial x}\,|_c\,(x-x_c) + \frac{\partial \mathbf{U}}{\partial y}\,|_c\,(y-y_c) \\ &+ \frac{\partial^2 \mathbf{U}}{\partial x^2}\,|_c\,(\frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d\Omega) \\ &+ \frac{\partial^2 \mathbf{U}}{\partial y^2}\,|_c\,(\frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d\Omega) \\ &+ \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\,|_c\,((x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d\Omega)\end{aligned} \tag{10}$$

where $\tilde{\mathbf{U}}$ is the mean value of $\mathbf{U}$ 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. 1. In this case, the dimension of the polynomial space is six and the six basis functions are

$$B_1 = 1, \quad B_2 = x - x_c, \quad B_3 = y - y_c, \quad B_4 = \frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d\Omega$$

$$B_5 = \frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d\Omega,$$

$$B_6 = (x - x_c)(y - y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x - x_c)(y - y_c)d\Omega \tag{11}$$

and the discontinuous Galerkin formulation (7) then leads to the following six equations

$$\frac{d}{dt}\int_{\Omega_e}\tilde{\mathbf{U}}d\Omega + \int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j d\Gamma = 0 \qquad i = 1, \tag{12}$$

$$\sum_{j=2}^{6}\int_{\Omega_e}B_i B_j d\Omega \frac{d}{dt}\begin{pmatrix} \frac{\partial \mathbf{U}}{\partial x}\mid_c \\ \frac{\partial \mathbf{U}}{\partial y}\mid_c \\ \frac{\partial^2 \mathbf{U}}{\partial x^2}\mid_c \\ \frac{\partial^2 \mathbf{U}}{\partial y^2}\mid_c \\ \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\mid_c \end{pmatrix} + \int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j B_i d\Gamma - \int_{\Omega_e}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial B_i}{\partial x_j}d\Omega = 0 \quad 2 \le i \le 6, \tag{13}$$

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

$$\int_{\Omega_e} B_1 B_i d\Omega = 0, \qquad 2 \le i \le 6. \tag{14}$$

Using this formulation, the similarity and difference between DG and FV methods become clear, and the advantage of the DGM is especially evident in comparison with the FV methods. In fact, the discretized governing equations for cell-averaged variables (12) and the assumption of polynomial solutions on each cell (10) are exactly the same for both methods. The DG(p_0) method, i.e., the DG method using piecewise constant polynomials, exactly corresponds to the first order cell-centered finite volume scheme. The only difference between them is the way how they obtain the high-order polynomial solutions (> 1). In the finite volume methods, the polynomial solutions of degree p are reconstructed using cell-averaged variables from neighboring cells, which can be obtained using either TVD/MUSCL or ENO/WENO reconstruction schemes. Unfortunately, the multi-dimensional TVD/MUSCL reconstruction schemes of arbitrary order based on the extension of one-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; 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, 2) extended stencils required for the reconstruction of higher-order ($> 1^{\text{st}}$) 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, and 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 exist 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. which is natural, unique, compact, rigorous, and elegant mathematically in contrast with arbitrariness characterizing the reconstruction schemes used in the FV methods with respect how to compute the derivatives and how to choose the stencils. This formulation has a number of distinct, desirable, and attractive features and advantages in the context of DG methods. First, 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, DG method 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. Secondly, cell-averaged variables and their derivatives are handily available in this formulation. This makes implementation of WENO limiter straightforward and efficient [2], which is required to eliminate nonphysical oscillations in the vicinity of discontinuities. Thirdly, the basis functions are hierarchic. This greatly facilitates implementation of p-multigrid methods [3, 4] and p-refinement. Last, cell-averaged variable equations are decoupled from their derivatives equations in this formulation. This makes development of fast, low-storage implicit methods possible.

3 Numerical examples

Due to page limitation, only a few illustrative examples are presented in this section to demonstrate the accuracy, robustness, and versatility of this DG method. Hancock scheme is used to advance the solution in time in order to achieve the efficiency for time accurate problems and a p-multigrid method [4] is used to accelerate the convergence of the Euler equations to a steady state solution. A WENO-based limiter [2] is used to eliminate nonphysical oscillations in the vicinity of discontinuities.

Example 1: Subsonic flow past a cylinder

This test case is chosen to numerically compare accuracy between DG and FV methods, where a grid convergence study has been conducted for subsonic flow past a circular cylinder at a Mach number of 0.38, and the numerical results are presented in Fig. 2. One can see that the second order DG(P1) solutions on an given meshes are more accurate than the second order finite volume solutions FV(P1) on the globally refine meshes, clearly demonstrating the high accuracy of the DG method.

Example 2: A Mach 3 wind tunnel with a Step

The test case is a classical example for testing the accuracy of numerical schemes for computing unsteady shock waves. The problem under consideration is a Mach 3 flow in a wind tunnel with a step. Fig. 3 shows the computed density contours obtained by the DG method and a third order WENO method, respectively. Note that the same mesh resolution is used for both computations. One can see that the shock resolution of the 3rd order WENO scheme is slightly more diffusive than the present second DG scheme, and the slip line coming from the lambda shock is also more visible in the 2nd DG solution than 3rd order WENO solution, qualitatively demonstrating that the present second order DG solution is as accurate as, if not more accurate than, the third order WENO solution.

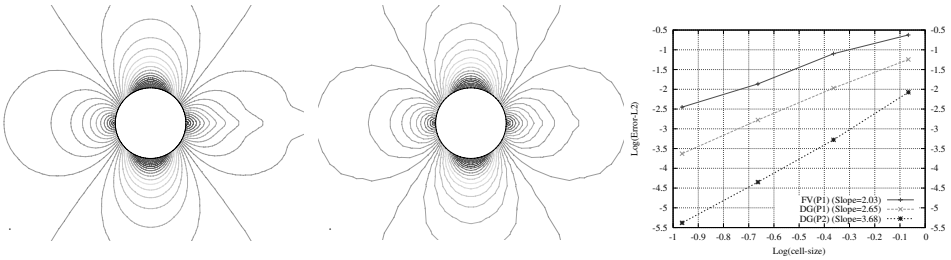

Fig. 2. Computed Mach number contours in the flow field obtained using FV(P1) solution on the 126x33 grid (left), and DG(P1) solution on the 64x17 grid (middle), and the behavior of the error norms with grid refinement for FV(P1), DG(P1), and DG(P2) for subsonic flow past a circular cylinder.

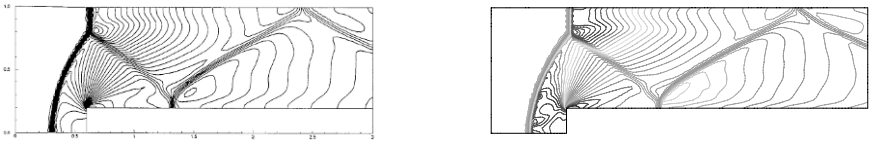

Fig. 3. Computed density contours in the flow field obtained using the 3rd WENO solution (left) and 2nd DG(P1) solution (right)

4 Conclusion

A DG method based a Taylor basis has been presented for the solution of the compressible Euler equations on arbitrary grids. The developed method has been used to compute a variety of time-accurate flow problems on arbitrary grids. The numerical results demonstrated the superior accuracy of this DG method in comparison with a second order finite volume method and a third order WENO method, indicating its promise and potential to become not just a competitive but simply a superior approach than its finite volume and ENO/WENO counterparts for solving flow problems of scientific and industrial interest.

References

1. Cockburn B. and C. W. Shu C. W. : The Runge-Kutta Discontinuous Galerkin Method for conservation laws V: Multidimensional System. journal of computational physics, Vol. 141, pp. 199-224, 1998.
2. Luo H., Baum J.D., and Löhner R. : A Hermit WENO-based limiter for DG method on unstructured grids. journal of computational physics, doi:10.1016/j.jcp.2006,12.017, 2007.
3. Luo H., Baum J.D., and Löhner R. : On the computation of steady-state compressible flows using a discontinuous Galerkin method. International Journal for Numerical Methods in Engineering, doi:10.1002/nme.208, 2007.
4. Luo H., Baum J.D., and Löhner R. : A p-multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids. journal of computational physics, Vol. 211, No. 2, pp. 767-783, 2006.