Reinterpretation and Simplified Implementation of a Discontinuous Galerkin Method for Hamilton-Jacobi Equations

Fengyan Li[1] and Chi-Wang Shu[1]

Abstract

In this note, we reinterpret a discontinuous Galerkin method originally developed by Hu and Shu [9] (see also [12]) for solving Hamilton-Jacobi equations. By this reinterpretation, numerical solutions will automatically satisfy the curl-free property of the exact solutions inside each element. This new reinterpretation allows a method of lines formulation, which renders a more natural framework for stability analysis. Moreover, this reinterpretation renders a significantly simplified implementation with reduced cost, as only a smaller subspace of the original solution space in [9, 12] is used and the least square procedure used in [9, 12] is completely avoided.

Keywords: Hamilton-Jacobi equations, discontinuous Galerkin method

AMS(MOS) subject classification: 65M60, 70H20

[1]Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: fengyan-l@dam.brown.edu, shu@dam.brown.edu. Research supported by ARO grant DAAD19-00-1-0405, NSF grant DMS-0207451, NASA Langley grant NCC1-01035 and AFOSR grant F49620-02-1-0113.

1 Introduction

Hamilton-Jacobi equations

$$\phi_t + H(\phi_{x_1}, \cdots, \phi_{x_n}) = 0, \text{in } \Omega \times [0,T] \subset \mathbb{R}^n \times \mathbb{R}, \quad \phi(x,0) = \phi^0(x) \tag{1.1}$$

arise in many areas of application, including calculus of variations, control theory, image processing, etc. See, e.g. [14]. Viscosity solutions of Hamilton-Jacobi equations are studied [6, 7, 14] to single out the practically relevant solutions. These viscosity solutions are Lipschitz continuous, and may have discontinuous derivatives, regardless of the smoothness of initial conditions.

Many numerical schemes have been developed for Hamilton-Jacobi equations, for example the widely used finite difference schemes [7, 15, 16, 10] which require structured meshes; and the finite volume schemes, which are based on arbitrary triangulations but rely on quite complicated reconstruction procedures [1, 18] in order to obtain higher order approximations.

Discontinuous Galerkin methods have become very popular in recent years to solve hyperbolic conservation laws because of their distinctive features, among which are the easy design of the methods with any order of accuracy and their minimal requirement on the mesh structures [5]. Adapted from these methods for conservation laws, a discontinuous Galerkin method for solving Hamilton-Jacobi equations (1.1) was developed by Hu and Shu in [9] based on the equivalence between Hamilton-Jacobi equations and hyperbolic conservation laws [11, 14]. See also [12]. In [9, 12], the Hamilton-Jacobi equations (1.1) were first rewritten as a system of conservation laws

$$(w_i)_t + (H(\mathbf{w}))_{x_i} = 0, \text{in } \Omega \times [0,T], \quad \mathbf{w}(x,0) = \nabla\phi^0(x), \tag{1.2}$$

where $\mathbf{w} = \nabla\phi$. With piecewise polynomial space as the solution space, the usual discontinuous Galerkin formulation could be obtained for (1.2) [2, 4]. Notice that $w_i, \quad i = 1, \cdots, n$ are not independent due to the restriction $\mathbf{w} = \nabla\phi$. A least square procedure was then applied in each time step (or each time stage depending on the particular time discretization used) to enforce this restriction.

In a recent work by Cockburn et al [3] and by Li and Shu [13], the locally divergence-free discontinuous Galerkin methods were developed for partial differential equations with divergence-free solutions. Compared with traditional ways to solve this type of equations, the piecewise divergence-free polynomial space, which is a subspace of the standard piecewise polynomial space, is used. With minimal change in the scheme formulation (only the solution and test space is changed to a smaller space), the computational cost is reduced, the stability and the order of accuracy of the scheme are maintained. For specific applications such as the Maxwell equations [3] and the ideal magnetohydrodynamics (MHD) equations [13], this new method even improves over the traditional discontinuous Galerkin method in terms of stability and/or accuracy while saving computational costs. The idea of this approach could be applied to more general situations, by using piecewise solution space in which functions satisfy certain properties of the exact solutions (divergence-free, or curl-free, ...). The general approximation theory can guarantee no loss of accuracy when such smaller solution space is used. This observation leads to a reinterpretation and simplified implementation of the discontinuous Galerkin method for Hamilton-Jacobi equations developed in [9, 12].

This note is organized as follows: in Section 2, we reinterpret the discontinuous Galerkin method for Hamilton-Jacobi equations developed in [9, 12] and describe the advantages of the new formulation. In Section 3, we provide a numerical example to demonstrate that the new formulation, while saving computational cost, gives identical results as those in [9, 12], as expected.

2 A Reinterpretation

Starting with a regular triangulation $\mathcal{T}_h = \{K\}$ of Ω (edges denoted by e), the general discontinuous Galerkin formulation of (1.2) is: find $\mathbf{w} = (w_1, \cdots, w_n) \in \mathbf{V}^k$, such that

$$\frac{d}{dt}\int_K w_i v_i dx = \int_K H(\mathbf{w})(v_i)_{x_i} dx - \sum_{e\in\partial K}\int_e \hat{H}_{i,e,K} v_i ds, \quad \forall K, i = 1, \cdots, n \tag{2.1}$$

holds for all $\mathbf{v} = (v_1, \cdots, v_n) \in \mathbf{V}^k$, where $\mathbf{V}^k$ is the solution space which will be specified later, and $\hat{H}_{i,e,K}$ is the numerical flux which is an approximating Riemann solver (see [16] for more details). The strong stability preserving Runge-Kutta time discretization [17, 8] could be used in time direction. Notice (2.1) is the formulation for the derivatives of ϕ in (1.1). To recover the missing constant in ϕ (e.g. the cell average of ϕ in each element), there are two different strategies developed in [9, 12] which can be used. We refer the readers to [9, 12] for the details.

Before finalizing the scheme, we introduce the following spaces,

$$\mathbf{V}_1^k = \{(v_1, \cdots, v_n) : v_i|_K \in P^k(K), i = 1, \cdots, n, \forall K \in \mathcal{T}_h\}, \tag{2.2}$$

$$\mathbf{V}_2^k = \{(v_1, \cdots, v_n) : \mathbf{v}|_K = \nabla\phi, \phi \in P^{k+1}(K), \forall K \in \mathcal{T}_h\}, \tag{2.3}$$

where $P^k(K)$ denotes the space of polynomials in K of degree at most k. It is easy to see that $\mathbf{V}_2^k \subset \mathbf{V}_1^k$. Two formulations are obtained if $\mathbf{V}^k$ in (2.1) is specified as follows:

- *Formulation I*: $\mathbf{V}^k = \mathbf{V}_1^k$. A single polynomial $\phi \in P^{k+1}(K)$, up to a constant, is further recovered from $\mathbf{w}$ in each element by the following least square procedure

$$\left|\left|\sum_i (\phi_{x_i} - w_i)^2\right|\right|_{L^1(K)} = \min_{\psi \in P^{k+1}(K)} \left|\left|\sum_i (\psi_{x_i} - w_i)^2\right|\right|_{L^1(K)} \tag{2.4}$$

  after each time stage. This is the method proposed by Hu and Shu in [9].

- *Formulation II*: $\mathbf{V}^k = \mathbf{V}_2^k$.

Proposition: *Formulations I and II give identical results.*

Proof: There are two key components in the proof which match each other in a very good way. First, (2.1) is a Galerkin formulation. Second, the least square procedure is essentially a L^2 projection.

Notice that (2.1) could be written compactly as

$$(\frac{\partial \mathbf{w}}{\partial t}, \mathbf{v}) = < \bar{L}(\mathbf{w}), \mathbf{v} >, \tag{2.5}$$

where

$$(\mathbf{u}, \mathbf{v}) = \sum_{K \in \mathcal{T}_h, i} \int_K u_i v_i dx,$$

$$< \bar{L}(\mathbf{w}), \mathbf{v} >= \sum_{K \in \mathcal{T}_h} \left\{ \int_K H(\mathbf{w}) \nabla \cdot \mathbf{v} dx - \sum_{e \in \partial K, i} \int_e \hat{H}_{i,e,K} v_i ds \right\}.$$

First we consider the forward Euler time discretization for (2.5). Starting with $\mathbf{w}^m \in \mathbf{V}_2^k$, we want to find $\mathbf{w}^{m+1}$, so that

$$(\mathbf{w}^{m+1}, \mathbf{v}) = \Delta t < \bar{L}(\mathbf{w}^m), \mathbf{v} > +(\mathbf{w}^m, \mathbf{v}) \equiv< L(\mathbf{w}^m), \mathbf{v} >, \tag{2.6}$$

hold for some test function $\mathbf{v}$.

Formulation II then becomes: find $\mathbf{w}_2^{m+1} \in \mathbf{V}_2^k$, such that

$$(\mathbf{w}_2^{m+1}, \mathbf{v}) =< L(\mathbf{w}^m), \mathbf{v} >, \quad \forall \mathbf{v} \in \mathbf{V}_2^k. \tag{2.7}$$

For Formulation I, we first obtain $\bar{\mathbf{w}}_\mathbf{1}^{m+1} \in \mathbf{V}_1^k$, which satisfies

$$(\bar{\mathbf{w}}_\mathbf{1}^{m+1}, \mathbf{v}) =< L(\mathbf{w}^m), \mathbf{v} >, \quad \forall \mathbf{v} \in \mathbf{V}_1^k, \tag{2.8}$$

then the least square procedure provides us $\mathbf{w}_1^{m+1} \in \mathbf{V}_2^k$ by

$$(\mathbf{w}_1^{m+1}, \mathbf{v}) = (\bar{\mathbf{w}}_1^{m+1}, \mathbf{v}) =< L(\mathbf{w}^m), \mathbf{v} >= (\mathbf{w}_2^{m+1}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V}_2^k \tag{2.9}$$

where the second equality holds because of (2.8) and $\mathbf{V}_2^k \subset \mathbf{V}_1^k$, and the last one is from (2.7). Therefore $\mathbf{w_1}^{m+1} = \mathbf{w_2}^{m+1}$ and hence Formulations I and II give the same numerical results. The conclusion can go directly to general explicit Runge-Kutta time discretizations, as these Runge-Kutta methods can be written as the linear combination of forward Euler methods (with proper time step) and the L^2 projection is a linear operator. ■

Remarks: Although the two formulations provide identical results by the previous proposition, Formulation II does have several advantages over Formulation I:

1. Formulation II allows the method of lines version of the scheme, while Formulation I does not have a method of lines version due to the least square procedure which is

applied after each time step or stage. The method of lines version allows more natural and direct analysis for stability and accuracy of discontinuous Galerkin methods, e.g. the results in [12].

2. The implementation of the algorithm is significantly simplified by using Formulation II since a smaller solution space is used and the least square procedure is completely avoided. If we characterize the computational cost of (2.1) per time step per element simply by the dimension of $\mathbf{V}^k|_K$, we can get

$$n_1 = dim(\mathbf{V}_1^k|_K) = n\sum_{r=0}^{k} C_{r+n-1}^{n-1}, \quad n_2 = dim(\mathbf{V}_2^k|_K) = \sum_{r=1}^{k+1} C_{r+n-1}^{n-1}.$$

For example, for the two dimensional case $n = 2$, $n_1 = (k+2)(k+1)$, $n_2 = \frac{(k+4)(k+1)}{2}$, hence $\frac{n_2}{n_1} \to \frac{1}{2}$ as $k \to \infty$, i.e. the cost is reduced to about half for higher order schemes. For the three dimensional case $n = 3$, $n_1 = \frac{k^3+6k^2+11k+6}{2}$, $n_2 = \frac{(k+1)(k^2+8k+18)}{6}$, hence $\frac{n_2}{n_1} \to \frac{1}{3}$ as $k \to \infty$, i.e. the cost is reduced to about one third for higher order schemes.

3 A Numerical Example

We only present one numerical example here, the two dimensional Burgers equation, to show the identical numerical results of Formulations I and II. Consider

$$\phi_t + \frac{(\phi_x + \phi_y + 1)^2}{2} = 0$$

on $[-2, 2] \times [-2, 2]$ with $\phi(x, y, 0) = -\cos\left(\frac{\pi(x+y)}{2}\right)$ and periodic boundary conditions. In Table 3.1 we show errors and orders of accuracy at $t = 0.5/\pi^2$, when the exact solution is still smooth. One can see that the differences of the errors between solutions from the two formulations are just on machine error level. We also plot the numerical solutions for ϕ at $t = 1.5/\pi^2$ in Figure 3.1 when the solution has discontinuous derivatives, obtained by Formulation II. The computation is performed on non-uniform rectangular meshes, which are obtained by a random perturbation of 10% from uniform meshes. The local Lax-Friedrichs numerical flux [9], and Runge-Kutta time discretization of compatible accuracy order are

Table 3.1: L^∞ errors and orders of accuracy for the 2D Burgers equations, non-uniform rectangular meshes, $t = \frac{0.5}{\pi^2}$.

$N \times N$	error for Formulation I	error for Formulation II	order
$k = 2$			
10×10	4.677844483874727E-2	4.677844483874738E-2	
20×20	1.227397975233135E-2	1.227397975233091E-2	1.93
40×40	3.539268278996754E-3	3.539268278997088E-3	1.80
80×80	1.153353110058708E-3	1.153353110058597E-3	1.62
160×160	2.722372199789325E-4	2.722372199783774E-4	2.08
$k = 3$			
10×10	1.001317293031612E-2	1.001317293031612E-2	
20×20	1.394123796094382E-3	1.394123796094604E-3	2.85
40×40	2.294961878530621E-4	2.294961878529511E-4	2.60
80×80	5.113292444847151E-5	5.113292444935968E-5	2.16
160×160	7.157782964561932E-6	7.157782965006021E-6	2.84

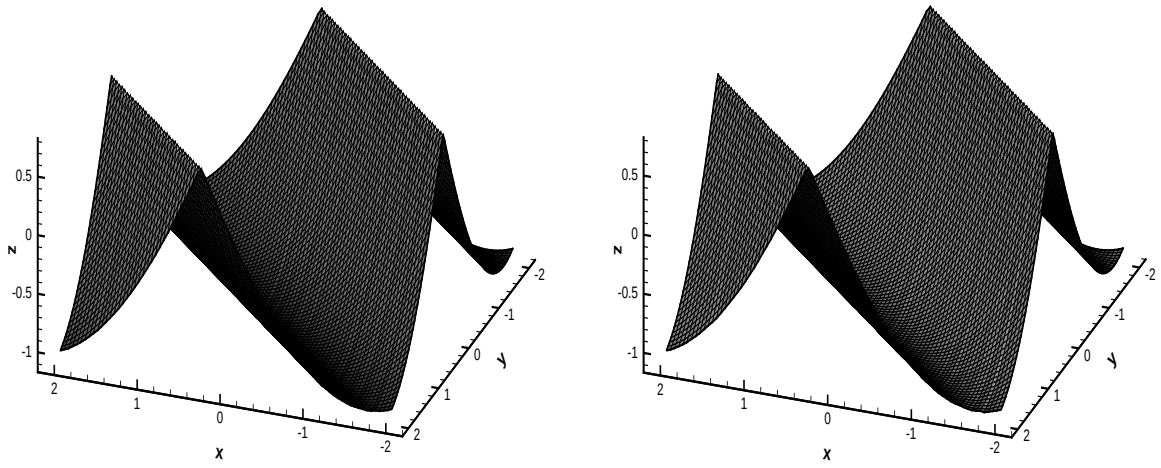

Figure 3.1: 2D Burgers equations, non-uniform rectangular meshes, $t = \frac{1.5}{\pi^2}$. Left: $k = 2$; right: $k = 3$.

used. The constant in ϕ is recovered first along $y = -2$ from $(-2, -2)$, then along $x =$ *constant* from bottom to top. See [9] for details.

References

[1] R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation*, Journal of Computational Physics, v114 (1994), pp.45-58.

[2] B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*, Mathematics of Computation, v54 (1990), pp.545-581.

[3] B. Cockburn, F. Li and C.-W. Shu, *Locally divergence-free discontinuous Galerkin methods for the Maxwell equations*, Journal of Computational Physics, to appear.

[4] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, Journal of Computational Physics, v141 (1998), pp.199-224.

[5] B. Cockburn and C.-W. Shu, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, Journal of Scientific Computing, v16 (2001), pp.173-261.

[6] M. Crandall and P.L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society, v277 (1983), pp.1-42.

[7] M. Crandall and P.L. Lions, *Two Approximations of solutions of Hamilton-Jacobi equations*, Mathematics of Computation, v43 (1984), pp.1-19.

[8] S. Gottlieb, C.-W. Shu and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, v43 (2001), pp.89-112.

[9] C. Hu and C.-W. Shu, *A discontinuous Galerkin finite element method for Hamilton-Jacobi equations*, SIAM Journal on Scientific Computing, v21 (1999), pp.666-690.

[10] G. Jiang and D. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM Journal on Scientific Computing, v21 (1999), pp.2126-2143.

[11] S. Jin and Z. Xin, *Numerical passage from systems of conservation laws to Hamilton-Jacobi equations, and relaxation schemes*, SIAM Journal on Numerical Analysis, v35 (1998), pp.2385-2404.

[12] O. Lepsky, C. Hu and C.-W. Shu, *Analysis of the discontinuous Galerkin method for Hamilton-Jacobi equations*, Applied Numerical Mathematics, v33 (2000), pp.423-434.

[13] F. Li and C.-W. Shu, *Locally divergence-free discontinuous Galerkin methods for MHD equations*, Journal of Scientific Computing, to appear.

[14] P.L. Lions, *Generalized solutions of Hamilton-Jacobi equations*, Pitman, Boston, 1982.

[15] S. Osher and J. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, v79 (1988), pp.12-49.

[16] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis, v28 (1991), pp.907-922.

[17] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, v77 (1988), pp.439-471.

[18] Y.-T. Zhang and C.-W. Shu, *High order WENO schemes for Hamilton-Jacobi equations on triangular meshes*, SIAM Journal on Scientific Computing, v24 (2003), pp.1005-1030.