Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 223 (2007) 398–415

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A discontinuous Galerkin finite element method for directly solving the Hamilton–Jacobi equations

Yingda Cheng, Chi-Wang Shu *

Division of Applied Mathematics, Brown University, Box F, Providence, RI 02912, United States

Received 10 May 2006; received in revised form 14 September 2006; accepted 19 September 2006
Available online 2 November 2006

Abstract

In this paper, we propose a new discontinuous Galerkin finite element method to solve the Hamilton–Jacobi equations. Unlike the discontinuous Galerkin method of [C. Hu, C.-W. Shu, A discontinuous Galerkin finite element method for Hamilton–Jacobi equations, SIAM Journal on Scientific Computing 21 (1999) 666–690.] which applies the discontinuous Galerkin framework on the conservation law system satisfied by the derivatives of the solution, the method in this paper applies directly to the solution of the Hamilton–Jacobi equations. For the linear case, this method is equivalent to the traditional discontinuous Galerkin method for conservation laws with source terms. Thus, stability and error estimates are straightforward. For the nonlinear convex Hamiltonians, numerical experiments demonstrate that the method is stable and provides the optimal $(k+1)$th order of accuracy for smooth solutions when using piecewise kth degree polynomials. Singularities in derivatives can also be resolved sharply if the entropy condition is not violated. Special treatment is needed for the entropy violating cases. Both one and two-dimensional numerical results are provided to demonstrate the good qualities of the scheme.

Keywords: Hamilton–Jacobi equations; Discontinuous Galerkin; High order accuracy; Convex Hamiltonian

1. Introduction

In this paper, we consider the numerical solution of the Hamilton–Jacobi equation

$$\varphi_t + H(\varphi_{x_1}, \ldots, \varphi_{x_d}, x_1, \ldots, x_d) = 0, \qquad \varphi(x,0) = \varphi^0(x) \tag{1.1}$$

Here d is the space dimension. In this paper, we will only consider the linear or convex Hamilton–Jacobi equations, namely the Hamiltonian H is a linear or convex function of φ_{x_i}. The solutions to the above equations are Lipschitz continuous but may admit discontinuous derivatives. For linear case with discontinuous coefficients or the nonlinear case, this is true even if the initial condition is smooth, and the solutions are also non-unique. We are only interested in the viscosity solution [6], which is the unique practically relevant solution and

* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: ycheng@dam.brown.edu (Y. Cheng), shu@dam.brown.edu (C.-W. Shu).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.09.012

satisfies the entropy condition. Fortunately, this entropy condition can be expressed in a simple form when the Hamiltonian is linear or convex.

Essentially non-oscillatory (ENO) or weighted ENO (WENO) finite difference schemes have been developed to solve the Hamilton–Jacobi equation (1.1), see, e.g. [16,10,19]. These finite difference methods work quite efficiently for Cartesian meshes, however, on unstructured meshes the scheme is quite complicated [19]. Alternatively, the Runge–Kutta discontinuous Galerkin (DG) finite element method, originally devised to solve the conservation laws [3,4,2,1,5], has the advantage of flexibility for arbitrarily unstructured meshes, with a compact stencil, and with the ability to easily achieve arbitrary order of accuracy. In [9], Hu and Shu proposed a discontinuous Galerkin method to solve the Hamilton–Jacobi equation (1.1). They use the fact that the derivatives of the solution φ satisfy a conservation law system, and apply the usual discontinuous Galerkin method on this system to advance the derivatives of φ. The solution φ itself is then recovered from these derivatives by a least square procedure for multi-dimensional cases and with an independent evolution of the cell averages of φ. Later, Li and Shu [14] reinterpreted the method of Hu and Shu by using a curl-free subspace for the discontinuous Galerkin method. The algorithm in [14] is mathematically equivalent to that in [9], but the least square procedure is avoided and the computational cost is reduced for multi-dimensional calculations. The method in [9,14] works well numerically, with provable stability results for certain special cases [9,12]. However, since this method is based on the conservation law system satisfied by the derivatives of φ, a scalar problem (1.1) is converted to a system for the multi-dimensional case, which is moreover only weakly hyperbolic at some points. This seems to have made the algorithm indirect and complicated. It is, therefore, desirable to design a discontinuous Galerkin method which solves directly the solution φ to the Hamilton–Jacobi equation (1.1). In this paper, we develop such a discontinuous Galerkin method.

This paper is organized as follows: in Section 2, we describe the formulation of our scheme for the one-dimensional case. Theoretical analysis for the linear case is provided. In Section 3, we generalize the scheme to two space dimensions. Numerical results of both one and two dimensions are presented in Section 4. Finally, in Section 5, concluding remarks are given.

2. One-dimensional case

For the simple one-dimensional case, (1.1) becomes

$$\varphi_t + H(\varphi_x, x) = 0 \qquad \varphi(x,0) = \varphi^0(x) \tag{2.1}$$

and we consider only the case where $H(\varphi_x, x)$ is a linear or convex function of φ_x. If we want to solve this equation on the interval $[a,b]$, first we divide it into N cells as follows:

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{N+\frac{1}{2}} = b \tag{2.2}$$

We denote

$$I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}), \qquad x_j = \frac{1}{2}\left(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}\right), \qquad I_{j+1/2} = \left[x_j, x_{j+1}\right] \tag{2.3}$$

and

$$\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}, \qquad h = \max_j \Delta x_j \tag{2.4}$$

Now, we define the approximation space as

$$V_h^k = \{\upsilon : \upsilon|_{I_j} \in P^k(I_j), \quad j = 1, \ldots, N\} \tag{2.5}$$

where $P^k(I_j)$ denotes all polynomials of degree at most k on I_j. We now formulate our scheme for (2.1) and give theoretical analysis for the linear case.

2.1. Formulation of the scheme

Let us denote $H_1 = \frac{\partial H}{\partial \varphi_x}$. If H_1 is always non-negative, then we can define the upwind version of our scheme as: find $\varphi_h(x,t) \in V_h^k$, such that

$$\int_{I_j} (\partial_t \varphi_h(x,t) + H(\partial_x \varphi_h(x,t),x))v_h(x)\mathrm{d}x + \max_{x\in I_{j-1/2}} H_1(\partial_x\varphi_h, x_{j-1/2})[\varphi_h]_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}} = 0, \qquad j=1,\ldots,N \tag{2.6}$$

holds for any $v_h \in V_h^k$. Here $[\varphi_h]_{j-\frac{1}{2}} = \varphi_h(x^+_{j-\frac{1}{2}},t) - \varphi_h(x^-_{j-\frac{1}{2}},t)$ denotes the jump of φ_h at the cell interface $x_{j-\frac{1}{2}}$. Our definition of the scheme is motivated by the usual discontinuous Galerkin method for the linear case, see the next subsection 2.2.

For general H, our scheme is formulated as follows: find $\varphi_h(x,t) \in V_h^k$, such that

$$\begin{aligned} &\int_{I_j} (\partial_t \varphi_h(x,t) + H(\partial_x \varphi_h(x,t),x))v_h(x)\mathrm{d}x \\ &\quad + \frac{1}{2}\left(\min_{x\in I_{j+1/2}} H_1(\partial_x\varphi_h, x_{j+1/2}) - \left|\min_{x\in I_{j+1/2}} H_1(\partial_x\varphi_h, x_{j+1/2})\right|\right)[\varphi_h]_{j+\frac{1}{2}}(v_h)^-_{j+\frac{1}{2}} \\ &\quad + \frac{1}{2}\left(\max_{x\in I_{j-1/2}} H_1(\partial_x\varphi_h, x_{j-1/2}) + \left|\max_{x\in I_{j-1/2}} H_1(\partial_x\varphi_h, x_{j-1/2})\right|\right)[\varphi_h]_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}} \\ &= 0, \qquad j=1,\ldots,N \end{aligned} \tag{2.7}$$

holds for any $v_h \in V_h^k$. This is a Roe type generalization of the upwind scheme (2.6).

In the schemes (2.6) and (2.7), we need the reconstructed information of $\partial_x\varphi_h$ on the cells $I_{j-1/2}$ and $I_{j+1/2}$. Notice that these cells include the points $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$, respectively, where the numerical solution $\varphi_h(x,t)$ is discontinuous. We use an L^2 reconstruction technique as follows. We define a polynomial $w_{j+\frac{1}{2}}(x) \in P^{2k+1}$ on $I_j \cup I_{j+1}$, such that

$$\int_{I_j} \varphi_h v \mathrm{d}x = \int_{I_j} w_{j+\frac{1}{2}} v\mathrm{d}x \tag{2.8}$$

for any $v \in P^k$ on I_j, and

$$\int_{I_{j+1}} \varphi_h v \mathrm{d}x = \int_{I_{j+1}} w_{j+\frac{1}{2}} v\mathrm{d}x \tag{2.9}$$

for any $v \in P^k$ on I_{j+1}. Then we use $\partial_x\varphi_h = \partial_x w_{j+\frac{1}{2}}$ on $I_{j+1/2}$ when taking the maximum or minimum in (2.6) and (2.7).

In the case of a uniform mesh and piecewise constant polynomials ($k = 0$), the reconstructed derivative becomes

$$\partial_x\varphi_h = \frac{\varphi_{j+1} - \varphi_j}{h} \tag{2.10}$$

on $I_{j+1/2}$. This agrees with our intuitive definition of $\partial_x\varphi_h$. For a practical implementation, once a local basis is chosen, the coefficients of $w_{j+\frac{1}{2}}(x)$ are linear combinations of the coefficients of $\varphi_h|_{I_j}$ and of $\varphi_h|_{I_{j+1}}$. These linear combination coefficients can be pre-computed to save computational cost.

We would like to remark that in (2.7), the last two terms involving the jumps of φ_h are added for stability, whereas the first integral term guarantees the accuracy of our scheme. The purpose of taking the maximum and minimum is to obtain better stability by adding more viscosity, while still maintaining accuracy since these maximum and minimum values are a $O(h)$ perturbation from $H_1(\partial_x\varphi_h(x_{j+1/2},t),x_{j+1/2})$, which guarantees accuracy according to truncation error analysis and numerical tests.

For linear Hamiltonians with discontinuous coefficients or nonlinear Hamiltonians, since our scheme is of Roe type, it may generate entropy violating solutions. We have, therefore, adopted the following entropy correction procedure:

1. For each cell I_j, determine if it is a potentially entropy violating cell. We will provide the criteria for this determination in the numerical Section 4.
   If the cell I_j is marked as a potentially entropy violating cell, then use Step 2 below to update φ_h in this cell; otherwise, update φ_h by (2.6) or (2.7).

2. Update φ_h by the DG method of Hu and Shu [9], namely, recover $\partial_x\varphi_h$ by taking the derivative of φ_h, then compute $\partial_t(\partial_x\varphi_h)$ by the usual DG method for the conservation law satisfied by $\partial_x\varphi_h$. This will determine φ_h up to a constant. The missing constant is obtained by requiring

$$\int_{I_j}(\partial_t\varphi_h(x,t)+H(\partial_x\varphi_h(x,t),x))\mathrm{d}x=0. \tag{2.11}$$

The entropy correction in Step 2 bears comparable computational cost as our scheme (2.6) or (2.7) for this one-dimensional case. It is our experience that such entropy corrections are needed only in very few isolated cells, see Section 4.

2.2. Theoretical analysis

We first consider a linear Hamilton–Jacobi equation

$$\varphi_t+a(x)\varphi_x=0 \tag{2.12}$$

and assume, for the time being, that $a(x)$ is a smooth function. If $a(x)>0$, our scheme (2.6), after replacing the maximum by the point value at $x_{j-\frac{1}{2}}$, becomes finding $\varphi_h(x,t)\in V_h^k$, such that

$$\int_{I_j}(\partial_t\varphi_h(x,t)+a(x)(\partial_x\varphi_h(x,t)))v_h(x)\mathrm{d}x+a(x_{j-\frac{1}{2}})[\varphi_h]_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}}=0,\quad j=1,\ldots,N \tag{2.13}$$

holds for any $v_h\in V_h^k$. After integration by parts, this is equivalent to

$$\begin{aligned}&\int_{I_j}\partial_t\varphi_h(x,t)v_h(x)\mathrm{d}x-\int_{I_j}a(x)\varphi_h(x,t)\partial_x v_h(x)\mathrm{d}x+(a\varphi_h)^-_{j+\frac{1}{2}}(v_h)^-_{j+\frac{1}{2}}-(a\varphi_h)^-_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}}\\&\quad=\int_{I_j}a_x(x)\varphi_h(x,t)v_h(x)\mathrm{d}x\end{aligned} \tag{2.14}$$

We observe that the scheme (2.14) is the standard DG scheme for conservation laws with source terms using upwind fluxes [4] for the equation

$$\varphi_t+(a(x)\varphi)_x=a_x(x)\varphi \tag{2.15}$$

which is equivalent to (2.12). Similarly, for general $a(x)$, our scheme (2.7), after replacing the maximum and minimum by the point value at $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$, respectively, becomes finding $\varphi_h(x,t)\in V_h^k$, such that

$$\begin{aligned}&\int_{I_j}(\partial_t\varphi_h(x,t)+a(x)(\partial_x\varphi_h(x,t)))v_h(x)\mathrm{d}x+\frac{1}{2}(a(x_{j+\frac{1}{2}})-|a(x_{j+\frac{1}{2}})|)[\varphi_h]_{j+\frac{1}{2}}(v_h)^-_{j+\frac{1}{2}}\\&\quad+\frac{1}{2}(a(x_{j-\frac{1}{2}})+|a(x_{j-\frac{1}{2}})|)[\varphi_h]_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}}=0\end{aligned} \tag{2.16}$$

holds for any $v_h\in V_h^k$. After integration by parts, this is equivalent to

$$\int_{I_j}\partial_t\varphi_h(x,t)v_h(x)\mathrm{d}x-\int_{I_j}a(x)\varphi_h(x,t)\partial_x v_h(x)\mathrm{d}x+\widehat{a\varphi_h}_{j+\frac{1}{2}}(v_h)^-_{j+\frac{1}{2}}-\widehat{a\varphi_h}_{j-\frac{1}{2}}(v_h)^+_{j-\frac{1}{2}}=\int_{I_j}a_x(x)\varphi_h(x,t)v_h(x)\mathrm{d}x \tag{2.17}$$

where

$$\widehat{a\varphi_h}_{j+\frac{1}{2}}=\frac{1}{2}(a(x_{j+\frac{1}{2}})+|a(x_{j+\frac{1}{2}})|)(\varphi_h)^-_{j+\frac{1}{2}}+\frac{1}{2}(a(x_{j+\frac{1}{2}})-|a(x_{j+\frac{1}{2}})|)(\varphi_h)^+_{j+\frac{1}{2}}$$

denotes the Roe flux. This is the standard DG scheme for conservation laws with source terms using Roe fluxes [4] for Eq. (2.15) which is equivalent to (2.12). We can, therefore, use the standard techniques in the analysis for the DG schemes to obtain the following theoretical results.

Proposition 2.1. *Suppose there is a constant β such that the derivative of $a(x)$ satisfies $a_x(x) < \beta$ for $x \in [a,b]$, then we have the following L^2 stability for our scheme (2.16):*

$$\|\varphi_h(t)\|_{L^2} \leqslant e^{\beta t/2}\|\varphi_h(0)\|_{L^2}$$

Proof. This follows from the standard proof of the cell entropy inequality for DG schemes applied to scalar conservation laws [11]. On each cell I_j, we can prove as in [11]

$$\int_{I_j}(\varphi_h)_t\varphi_h \mathrm{d}x - \int_{I_j} a_x\frac{\varphi_h^2}{2}\mathrm{d}x + \hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}} + \Theta_j = 0$$

for some entropy flux $\hat{F}_{j+\frac{1}{2}}$ and $\Theta_j \geqslant 0$. Summing over j, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b \frac{\varphi_h^2}{2}\mathrm{d}x \leqslant \int_a^b a_x\frac{\varphi_h^2}{2}\mathrm{d}x$$

Since $a_x < \beta$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b \varphi_h^2\mathrm{d}x \leqslant \beta\int_a^b \varphi_h^2\mathrm{d}x$$

Integrating over t finishes the proof. □

Proposition 2.2. *If $a(x)$ and the solution φ of (2.12) are smooth and the scheme (2.16) with the finite element space (2.5) is used, then we have the following optimal L^2 error estimate*

$$\|\varphi_h(t) - \varphi(t)\|_{L^2} \leqslant Ch^{k+1}$$

Proof. The proof is similar to that for standard DG schemes. The optimal $(k+1)$th order of convergence is obtained through a special projection in the proof, see for example [18] for the details. □

For nonlinear problems, we notice that our scheme is consistent only when $k \geqslant 1$. For example, for the Burgers' equation

$$\varphi_t + \frac{\varphi_x^2}{2} = 0, \tag{2.18}$$

with $\varphi_x \geqslant 0$, our upwind scheme (2.6) with piecewise constant space (the space (2.5) with $k = 0$) gives

$$(\varphi_j)_t + \left(\frac{\varphi_j - \varphi_{j-1}}{\Delta x_j}\right)^2 = 0 \tag{2.19}$$

where $\varphi = \varphi_j$ on cell I_j. This is clearly consistent with a different equation $\varphi_t + \varphi_x^2 = 0$ and is inconsistent with the Burgers' Eq. (2.18). If $k \geqslant 1$, we can prove that our scheme is consistent, which is also verified by numerical experiments in Section 4. For example, the P^1 upwind scheme (2.6) to solve the equation $\varphi_t + H(\varphi_x) = 0$ is

$$\left(\varphi_j^0\right)_t + H\left(\frac{\varphi_j^1}{\Delta x_j}\right) + \alpha_j\frac{[\varphi_h]_{j-\frac{1}{2}}}{\Delta x_j} = 0 \tag{2.20}$$

where $\varphi = \varphi_j^0 + \varphi_j^1\frac{x-x_j}{\Delta x_j}$ on cell I_j. Since $[\varphi_h]_{j-\frac{1}{2}} = O(h^2)$ for smooth functions, the scheme is consistent.

2.3. Time discretization

Up to now, we have taken the method of lines approach and have left t continuous. We can use total variation diminishing (TVD) high-order Runge–Kutta methods [17] to solve the method of lines ODE

$$\varphi_t = L(\varphi). \tag{2.21}$$

The third-order TVD Runge–Kutta method that we use in this paper is given by

$$\begin{aligned}
\varphi^{(1)} &= \varphi^n + \Delta t L(\varphi^n) \\
\varphi^{(2)} &= \frac{3}{4}\varphi^n + \frac{1}{4}\varphi^{(1)} + \frac{1}{4}\Delta t L(\varphi^{(1)}) \\
\varphi^{n+1} &= \frac{1}{3}\varphi^n + \frac{2}{3}\varphi^{(2)} + \frac{2}{3}\Delta t L(\varphi^{(2)})
\end{aligned} \tag{2.22}$$

Detailed description of the TVD Runge–Kutta method can be found in [17], see also [7,8].

3. Two-dimensional case

In this section, we consider the case of two spatial dimensions. The equation is given by

$$\varphi_t + H(\varphi_x, \varphi_y, x, y) = 0, \qquad \varphi(x,y,0) = \varphi^0(x,y) \tag{3.1}$$

and we again only consider the case where $H(\varphi_x,\varphi_y,x,y)$ is a linear or convex function of φ_x and φ_y. For simplicity of presentation, we consider in this paper only rectangular domains and cells, although our method can be easily defined on general triangulations as other DG methods. Suppose Eq. (3.1) is solved on the domain $[a,b]\times[c,d]$. We use rectangular meshes defined as

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{N_x+\frac{1}{2}} = b, \qquad c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \ldots < y_{N_y+\frac{1}{2}} = d \tag{3.2}$$

and

$$\begin{aligned}
&I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], \quad J_i = [x_{i-1/2}, x_{i+1/2}], \quad K_j = [y_{j-1/2}, y_{j+1/2}] \\
&J_{i+1/2} = [x_i, x_{i+1}], \quad K_{j+1/2} = [y_j, y_{j+1}], \quad i = 1,\ldots,N_x, \quad j = 1,\ldots,N_y
\end{aligned} \tag{3.3}$$

We define the approximation space as

$$V_h^k = \{v : v|_{I_{i,j}} \in P^k(I_{i,j}), i = 1,\ldots,N_x, \quad j = 1,\ldots,N_y\} \tag{3.4}$$

where $P^k(I_{i,j})$ denotes all polynomials of degree at most k on $I_{i,j}$.

Let us denote $H_1 = \frac{\partial H}{\partial \varphi_x}$ and $H_2 = \frac{\partial H}{\partial \varphi_y}$. We define our scheme as: find $\varphi_h(x,t) \in V_h^k$, such that

$$\begin{aligned}
&\int_{I_{i,j}} (\partial_t \varphi_h(x,y,t) + H(\partial_x \varphi_h(x,y,t), \partial_y \varphi_h(x,y,t), x, y)) v_h(x,y) \mathrm{d}x\mathrm{d}y \\
&\quad + \frac{1}{2}\int_{K_j} \left(\min_{x\in J_{i+1/2}} H_1(\partial_x\varphi_h, \overline{\partial_y\varphi_h}, x_{i+1/2}, y) - \left| \min_{x\in J_{i+1/2}} H_1(\partial_x\varphi_h, \overline{\partial_y\varphi_h}, x_{i+1/2}, y) \right| \right) [\varphi_h](x_{i+\frac{1}{2}}, y) v_h(x^-_{i+\frac{1}{2}}, y) \mathrm{d}y \\
&\quad + \frac{1}{2}\int_{K_j} \left(\max_{x\in J_{i-1/2}} H_1(\partial_x\varphi_h, \overline{\partial_y\varphi_h}, x_{i-1/2}, y) + \left| \max_{x\in J_{i-1/2}} H_1(\partial_x\varphi_h, \overline{\partial_y\varphi_h}, x_{i-1/2}, y) \right| \right) [\varphi_h](x_{i-\frac{1}{2}}, y) v_h(x^+_{i-\frac{1}{2}}, y) \mathrm{d}y \\
&\quad + \frac{1}{2}\int_{J_i} \left(\min_{y\in K_{j+1/2}} H_2(\overline{\partial_x\varphi_h}, \partial_y\varphi_h, x, y_{j+1/2}) - \left| \min_{y\in K_{j+1/2}} H_2(\overline{\partial_x\varphi_h}, \partial_y\varphi_h, x, y_{j+1/2}) \right| \right) [\varphi_h](x, y_{j+\frac{1}{2}}) v_h(x, y^-_{j+\frac{1}{2}}) \mathrm{d}x \\
&\quad + \frac{1}{2}\int_{J_i} \left(\max_{y\in K_{j-1/2}} H_2(\overline{\partial_x\varphi_h}, \partial_y\varphi_h, x, y_{j-1/2}) + \left| \max_{y\in K_{j-1/2}} H_2(\overline{\partial_x\varphi_h}, \partial_y\varphi_h, x, y_{j-1/2}) \right| \right) [\varphi_h](x, y_{j-\frac{1}{2}}) v_h(x, y^+_{j-\frac{1}{2}}) \mathrm{d}x = 0
\end{aligned} \tag{3.5}$$

holds for any $v_h \in V_h^k$. In the above formula, we define

$$\overline{\partial_x\varphi_h} = \frac{1}{2}((\partial_x\varphi_h)^+ + (\partial_x\varphi_h)^-), \qquad \overline{\partial_y\varphi_h} = \frac{1}{2}((\partial_y\varphi_h)^+ + (\partial_y\varphi_h)^-)$$

The main idea is that, on the interfaces of cells, along the normal direction we would use the reconstructed information of the partial derivatives as in the one-dimensional case. Tangential to the interface, the average of the partial derivatives from the two neighboring cells is used. The reconstruction process is the same as that in the one-dimensional case, except that we need to fix x or y, then perform the reconstruction on the other spatial variable. If $H_1 > 0$ or $H_2 > 0$, we can apply the corresponding upwind scheme in that direction.

A similar entropy correction procedure as in the one-dimensional case is adopted here as well for the cases with linear Hamiltonians with discontinuous coefficients or nonlinear Hamiltonians:

1. For each cell $I_{i,j}$, determine if it is a potentially entropy violating cell. We will again provide the criteria for this determination in the numerical Section 4.
   If the cell $I_{i,j}$ is marked as a potentially entropy violating cell, then use Step 2 below to update φ_h in this cell; otherwise, update φ_h by (3.5).
2. Update φ_h by the DG method of Hu and Shu [9] as reinterpreted by Li and Shu [14], namely, recover $\partial_x\varphi_h$ and $\partial_y\varphi_h$ by taking the derivatives of φ_h, then compute $\partial_t(\partial_x\varphi_h)$ and $\partial_t(\partial_y\varphi_h)$ by the usual DG method for the conservation laws satisfied by $\partial_x\varphi_h$ and $\partial_y\varphi_h$ in a locally curl-free discontinuous Galerkin space. This will determine φ_h up to a constant. The missing constant is obtained by requiring

$$\int_{I_{i,j}} (\partial_t\varphi_h + H(\partial_x\varphi_h, \partial_y\varphi_h, x, y))\mathrm{d}x\mathrm{d}y = 0. \tag{3.6}$$

4. Numerical results

In this section, we provide numerical experimental results to demonstrate the behavior of our schemes.

4.1. Linear smooth problems

In this subsection, linear smooth problems are computed using our scheme. In this case, our scheme is equivalent to the standard DG scheme for conservation laws with source terms.

Example 4.1.1. We solve the one-dimensional problem

$$\begin{cases} \varphi_t + \sin(x)\varphi_x = 0 \\ \varphi(x,0) = \sin(x) \\ \varphi(0,t) = \varphi(2\pi,t) \end{cases} \tag{4.1}$$

The exact solution is

$$\varphi(x,t) = \sin\left(2\tan^{-1}\left(\mathrm{e}^{-t}\tan\left(\frac{x}{2}\right)\right)\right) \tag{4.2}$$

We use the general scheme (2.16) and list the results in Tables 4.1–4.4 for P^0, P^1, P^2 and P^3, respectively. We clearly observe $(k+1)$th order of accuracy for P^k polynomials.

Example 4.1.2. We solve the two-dimensional linear Hamilton–Jacobi equation with variable coefficients

$$\varphi_t - y\varphi_x + x\varphi_y = 0. \tag{4.3}$$

Table 4.1
Errors and numerical orders of accuracy for Example 4.1.1 when using P^0 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.49E−01		0.62E−01		0.29E+00	
80	0.25E−01	0.95	0.32E−01	0.93	0.16E+00	0.86
160	0.13E−01	0.97	0.17E−01	0.96	0.83E−01	0.96
320	0.65E−02	0.98	0.84E−02	0.98	0.42E−01	0.99
640	0.33E−02	0.99	0.42E−02	0.99	0.21E−01	1.00

Final time $t = 1$. CFL = 0.9.

Table 4.2
Errors and numerical orders of accuracy for Example 4.1.1 when using P^1 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.12E−02		0.25E−02		0.15E−01	
80	0.31E−03	1.96	0.68E−03	1.90	0.43E−02	1.81
160	0.78E−04	1.97	0.18E−03	1.94	0.11E−02	1.92
320	0.20E−04	1.98	0.46E−04	1.97	0.29E−03	1.96
640	0.50E−05	1.99	0.12E−04	1.98	0.74E−04	1.98

Final time $t = 1$. CFL = 0.3.

Table 4.3
Errors and numerical orders of accuracy for Example 4.1.1 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.48E−04		0.10E−03		0.52E−03	
80	0.60E−05	2.99	0.14E−04	2.88	0.88E−04	2.58
160	0.75E−06	3.00	0.18E−05	2.90	0.14E−04	2.70
320	0.94E−07	2.99	0.24E−06	2.93	0.20E−05	2.78
640	0.12E−07	2.99	0.31E−07	2.95	0.27E−06	2.85

Final time $t = 1$. CFL = 0.1.

Table 4.4
Errors and numerical orders of accuracy for Example 4.1.1 when using P^3 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.21E−05		0.51E−05		0.29E−04	
80	0.14E−06	3.96	0.35E−06	3.88	0.22E−05	3.75
160	0.87E−08	3.97	0.23E−07	3.93	0.16E−06	3.78
320	0.55E−09	3.97	0.15E−08	3.96	0.10E−07	3.91
640	0.35E−10	3.98	0.94E−10	3.98	0.68E−09	3.95

Final time $t = 1$. CFL = 0.05.

The computational domain is $[-1,1]^2$. The initial condition is given by

$$\varphi_0(x,y) = \begin{cases} 0 & 0.3 \leqslant r \\ 0.3 - r & 0.1 < r < 0.3 \\ 0.2 & r \leqslant 0.1 \end{cases} \tag{4.4}$$

where $r = \sqrt{(x-0.4)^2 + (y-0.4)^2}$. We also impose periodic boundary condition on the domain. This is a solid body rotation around the origin. The exact solution can be expressed as

$$\varphi(x,y,t) = \varphi_0(x\cos(t) + y\sin(t), -x\sin(t) + y\cos(t)) \tag{4.5}$$

For this problem, the derivatives of φ are not continuous. Therefore, we do not expect to obtain $(k+1)$th order of accuracy for P^k polynomials, see Table 4.5.

At $t = 2\pi$, i.e. the period of rotation, we take a snapshot at the line $x = y$ in Fig. 4.1. We can see that a higher order scheme can yield better results for this nonsmooth initial condition.

Example 4.1.3. We solve the same Eq. (4.3) as that in Example 4.1.2, but with a different initial condition as

$$\varphi_0(x,y) = \exp\left(-\frac{(x-0.4)^2 + (y-0.4)^2}{2\sigma^2}\right) \tag{4.6}$$

Table 4.5
Errors and numerical orders of accuracy for Example 4.1.2 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of $N \times N$ cells

$N \times N$	L^1 error	Order	L^2 error	Order	L^∞ error	Order
20×20	0.41E−03		0.13E−02		0.11E−01	
40×40	0.14E−03	1.58	0.55E−03	1.26	0.65E−02	0.82
80×80	0.47E−04	1.54	0.24E−03	1.22	0.36E−02	0.84
160×160	0.15E−04	1.62	0.10E−03	1.23	0.21E−02	0.81

Final time $t = 1$. CFL = 0.1.

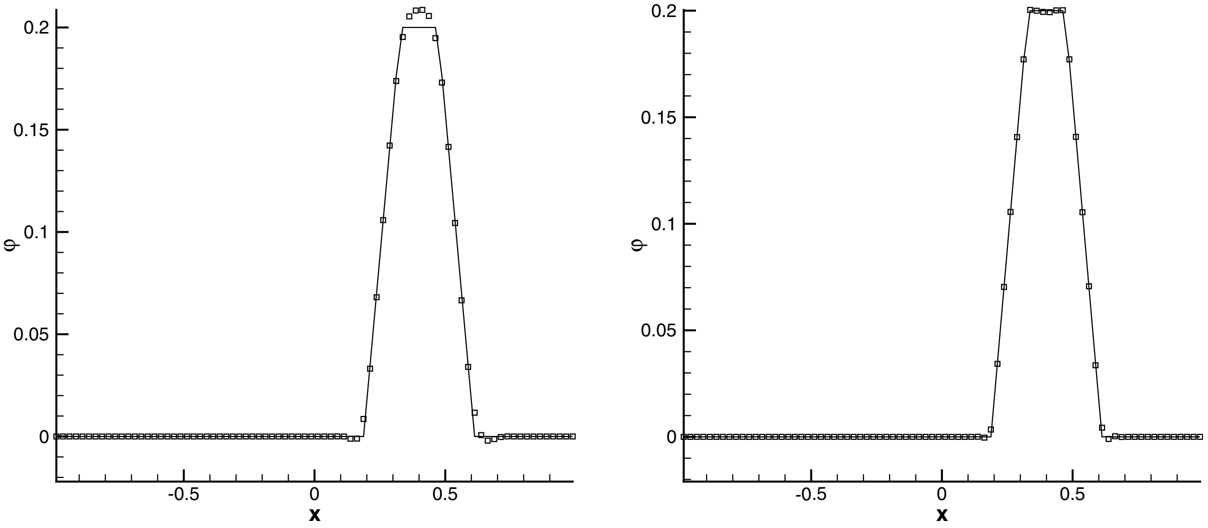

Fig. 4.1. Example 4.1.2. 80×80 uniform mesh. $t = 2\pi$. Solid line: the exact solution; rectangles: the numerical solution. One-dimensional cut of 45° with the x axis. Left: P^1 polynomial; right: P^2 polynomial.

We take $\sigma = 0.05$ such that at the domain boundary, φ is very small, hence imposing a periodic boundary condition will lead to small non-smoothness errors. We then observe the desired order of accuracy in Table 4.6.

Example 4.1.4. We solve the two-dimensional linear Hamilton–Jacobi equation with variable coefficients

$$\varphi_t + f(x,y,t)\varphi_x + g(x,y,t)\varphi_y = 0 \tag{4.7}$$

The computational domain is still $[-1,1]^2$, and the advection coefficients are

$$f(x,y,t) = \sin^2(\pi x)\sin(2\pi y)\cos\left(\frac{t}{T}\pi\right), \quad g(x,y,t) = -\sin^2(\pi y)\sin(2\pi x)\cos\left(\frac{t}{T}\pi\right)$$

where T is the period of deformation. The initial condition is given by

Table 4.6
Errors and numerical orders of accuracy for Example 4.1.3 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of $N \times N$ cells

$N \times N$	L^1 error	Order	L^2 error	Order	L^∞ error	Order
20×20	0.14E−02		0.10E−01		0.28E+00	
40×40	0.15E−03	3.21	0.15E−02	2.81	0.53E−01	2.41
80×80	0.11E−04	3.82	0.11E−03	3.73	0.58E−02	3.19
160×160	0.11E−05	3.30	0.12E−04	3.26	0.90E−03	2.69

Final time $t = 1$. CFL = 0.1.

$$\varphi_0(x,y)=\begin{cases}0 & 0.3\leqslant r\\ 0.3-r & 0.1<r<0.3\\ 0.2 & r\leqslant 0.1\end{cases}\tag{4.8}$$

where $r=\sqrt{(x-0.4)^2+(y-0.4)^2}$. This is a numerical test for incompressible flow first introduced by LeVeque in [13]. During the evolution, the initial data are severely deformed, then it returns to the original shape after one period. At $t=1.5$, i.e. the period of rotation, we take a snapshot at the line $x=y$ in Fig. 4.2. We can clearly observe that a higher order scheme yields better results for this nonsmooth initial condition.

4.2. Linear nonsmooth problems

In this subsection, the Hamiltonian H is a linear function of $\nabla\varphi$ with nonsmooth coefficients.

Example 4.2.1. We solve the model problem

$$\begin{cases}\varphi_t+\operatorname{sign}(\cos(x))\,\varphi_x=0\\ \varphi(x,0)=\sin(x)\\ \varphi(0,t)=\varphi(2\pi,t)\end{cases}\tag{4.9}$$

The exact solution is given by

- if $0\leqslant t\leqslant \pi/2$

$$\varphi(x,t)=\begin{cases}\sin(x-t) & \text{if } 0\leqslant x\leqslant\frac{\pi}{2}\\ \sin(x+t) & \text{if } \frac{\pi}{2}<x\leqslant\frac{3\pi}{2}-t\\ -1 & \text{if } \frac{3\pi}{2}-t<x\leqslant\frac{3\pi}{2}+t\\ \sin(x-t) & \text{if } \frac{3\pi}{2}+t<x\leqslant 2\pi\end{cases}\tag{4.10}$$

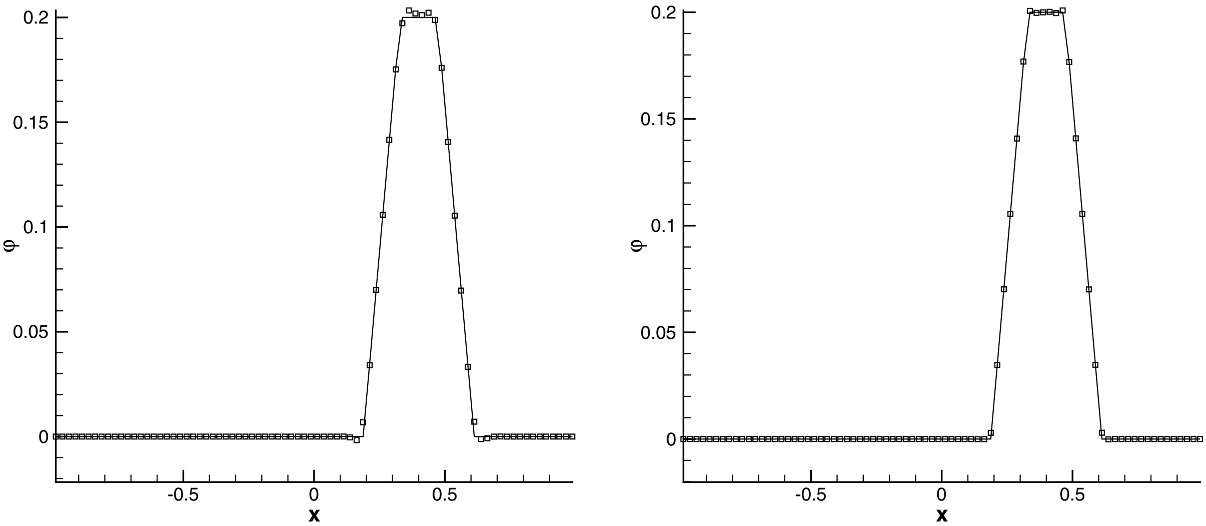

Fig. 4.2. Example 4.1.4. 80 × 80 uniform mesh. $t=1.5$. Solid line: the exact solution; rectangles: the numerical solution. One-dimensional cut of 45° with the x axis. Left: P^1 polynomial; right: P^2 polynomial.

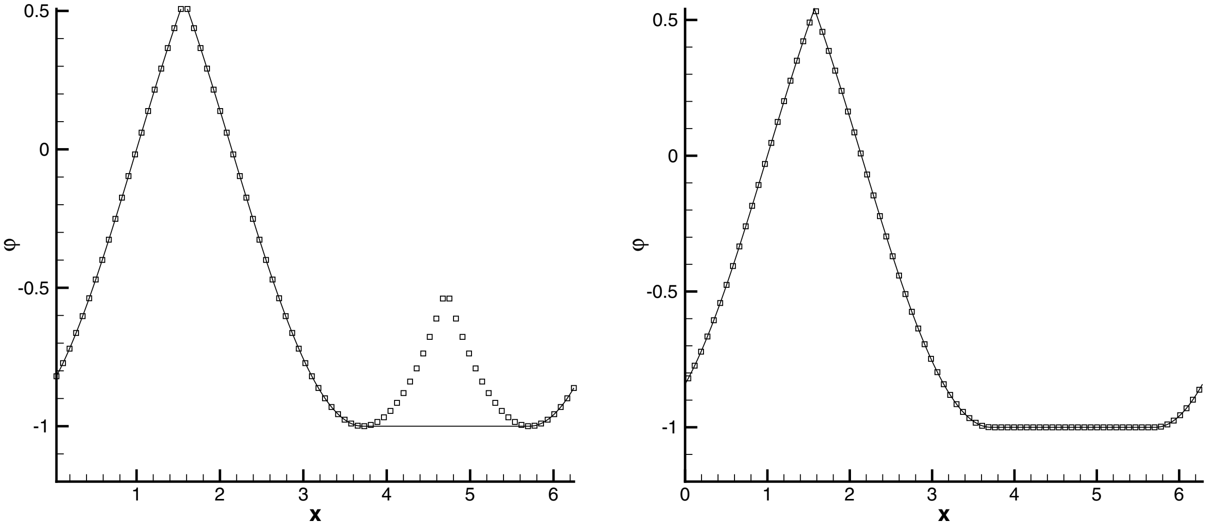

Fig. 4.3. Example 4.2.1. $t = 1$, CFL = 0.1, using P^2 polynomials. Solid line: the exact solution; rectangles: the numerical solution. Left: $N = 80$; right: $N = 81$.

- if $\pi/2 \leqslant t \leqslant \pi$

$$\varphi(x,t) = \begin{cases} -1 & \text{if } 0 \leqslant x \leqslant t - \frac{\pi}{2} \\ \sin(x-t) & \text{if } t - \frac{\pi}{2} < x \leqslant \frac{\pi}{2} \\ \sin(x+t) & \text{if } \frac{\pi}{2} < x \leqslant \frac{3\pi}{2} - t \\ -1 & \text{if } \frac{3\pi}{2} - t < x \leqslant 2\pi \end{cases} \tag{4.11}$$

- if $t \geqslant \pi$

$$\varphi(x,t) = -1. \tag{4.12}$$

For the viscosity solution, at $x = \frac{\pi}{2}$, there will be a shock forming in φ_x, and at $x = \frac{3\pi}{2}$, there is a rarefaction wave.

We first test the scheme without any entropy correction. If we take N to be a multiple of 4, then the discontinuity of $a(x)$ is exactly located at a cell interface. In this case, the entropy condition is violated by our scheme at the two cells neighboring $\frac{3\pi}{2}$, and the numerical solution obtained is not close to the viscosity solution, see Fig. 4.3, left. If instead, we take other values of N such that the discontinuity of $a(x)$ is not at the cell interface, then the entropy condition is not violated and the numerical solution obtained approximates the viscosity solution very well, see Fig. 4.3, right.

The test above indicates the necessity of an entropy correction in this case. The criteria for the entropy correction is as follows. For the cell $I_j = (x_{j\text{-}1/2}, x_{j+1/2})$, if

$$a^-(x_{j-1/2}) < 0 < a^+(x_{j-1/2}) \tag{4.13}$$

or

$$a^-(x_{j+1/2}) < 0 < a^+(x_{j+1/2}) \tag{4.14}$$

is satisfied, we will compute $(\varphi_x)_t$ on the cell I_j by solving the conservation law for $\varphi_x = u$ as

$$u_t + (\operatorname{sign}(\cos(x))u)_x = 0 \tag{4.15}$$

using the standard DG method with polynomials in $P^{k\text{-}1}$, and then recover φ by requiring

$$\int_{I_j} (\varphi_t + \operatorname{sign}(\cos(x))\varphi_x)\mathrm{d}x = 0. \tag{4.16}$$

For this one-dimensional example it does not increase the computational cost. We can see in Fig. 4.4 that, after this entropy correction, the numerical solution approximates the viscosity solution very well.

The numerical errors and order of accuracy are shown in Table 4.7. Since the exact solution is not smooth, we do not expect the full $(k+1)$th order accuracy.

4.3. Nonlinear smooth problems

In this subsection, the Hamiltonian H is a nonlinear smooth function of $\nabla\varphi$.

Example 4.3.1. One-dimensional Burgers' equation

$$\begin{cases} \varphi_t + \frac{\varphi_x^2}{2} = 0 \\ \varphi(x,0) = \sin(x) \\ \varphi(0,t) = \varphi(2\pi,t) \end{cases} \tag{4.17}$$

The exact solution when φ is still smooth is obtained by the characteristics methods. First solve x_0 from

$$x = x_0 + \cos(x_0)t \tag{4.18}$$

then get φ as

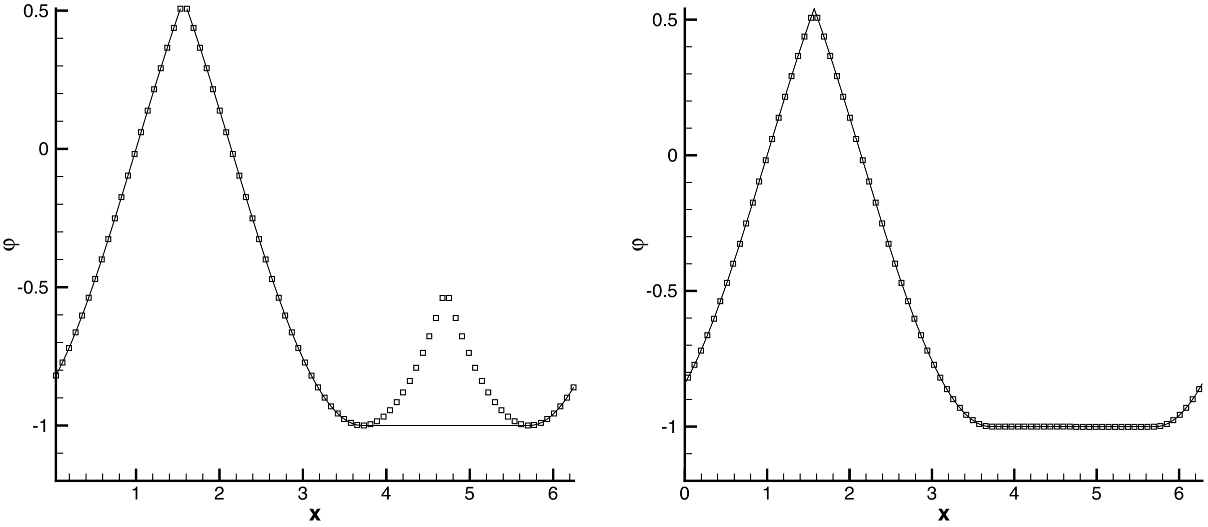

Fig. 4.4. Example 4.2.1. $t=1$, CFL $=0.1$, $N=80$, using P^2 polynomials. Solid line: the exact solution; rectangles: the numerical solution. Left: without entropy correction; right: with entropy correction.

Table 4.7
Errors and numerical orders of accuracy for Example 4.2.1 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.64E−03		0.15E−02		0.41E−02	
80	0.16E−03	1.97	0.40E−03	1.94	0.10E−02	2.00
160	0.41E−04	1.99	0.10E−03	1.97	0.26E−03	2.00
320	0.10E−04	2.00	0.25E−04	1.99	0.64E−04	2.00
640	0.26E−05	2.00	0.64E−05	2.00	0.16E−04	2.00

Final time $t=1$. CFL $=0.1$.

$$\varphi(x,t)=\sin(x_0)+\frac{\cos(x_0)^2}{2}t \tag{4.19}$$

When $t = 0.5$, the solution is still smooth, and the expected third order accuracy is obtained for P^2 polynomials, see Table 4.8. After $t = 1$, a shock will form in φ_x, our scheme can resolve the derivative singularity sharply, see Fig. 4.5.

Example 4.3.2. One-dimensional Burgers' equation with a nonsmooth initial condition

$$\begin{cases}\varphi_t+\frac{\varphi_x^2}{2}=0\\ \varphi(x,0)=\begin{cases}\pi-x & \text{if } 0\leqslant x\leqslant\pi\\ x-\pi & \text{else where in } [0,2\pi],\end{cases}\\ \varphi(0,t)=\varphi(2\pi,t)\end{cases} \tag{4.20}$$

For the viscosity solution, the sharp corner at π will be smoothed out, and a rarefaction wave will form in the derivative. Since the entropy condition is violated by our Roe type scheme, we need to apply the entropy correction procedure. Fig. 4.6 shows the comparison of the numerical solution with and without the entropy correction. Clearly the entropy correction is needed to obtain a good approximation to the entropy solution. The criteria for the entropy correction is as follows. For the cell $I_j = (x_{j\text{-}1/2}, x_{j+1/2})$, if either

$$\varphi_x^-(x_{j-1/2})<0<\varphi_x^+(x_{j-1/2}) \tag{4.21}$$

Table 4.8
Errors and numerical orders of accuracy for Example 4.3.1 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.13E−04		0.22E−04		0.84E−04	
80	0.17E−05	2.97	0.29E−05	2.93	0.12E−04	2.86
160	0.22E−06	2.98	0.37E−06	2.96	0.15E−05	2.92
320	0.27E−07	2.98	0.47E−07	2.97	0.20E−06	2.95
640	0.34E−08	2.99	0.59E−08	2.99	0.25E−07	2.97

Final time $t = 0.5$. CFL = 0.1.

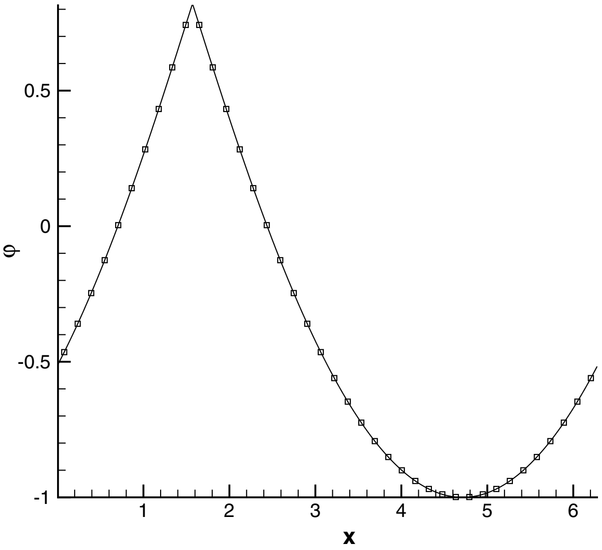

Fig. 4.5. Example 4.3.1. Numerical solution. Solid line: $N = 500$; rectangles: $N = 40$. Final time $t = 1.5$, CFL = 0.05, P^2 polynomials.

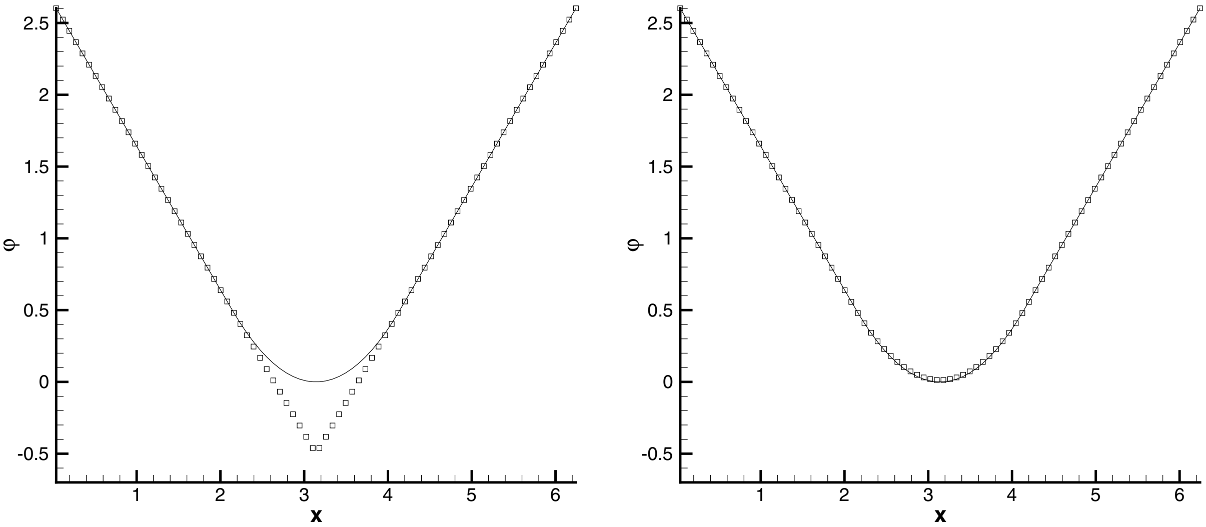

Fig. 4.6. Example 4.3.2. $t = 1$, CFL $= 0.05$, $N = 80$, using P^2 polynomials. Solid line: the exact solution; rectangles: the numerical solution. Left: without entropy correction; right: with entropy correction.

or

$$\varphi_x^-(x_{j+1/2}) < 0 < \varphi_x^+(x_{j+1/2}) \tag{4.22}$$

is satisfied, then the entropy correction is needed.

Example 4.3.3. Two-dimensional Burgers' equation.

$$\begin{cases} \varphi_t + \frac{(\varphi_x+\varphi_y+1)^2}{2} = 0 \\ \varphi(x,y,0) = -\cos(x+y) \end{cases} \tag{4.23}$$

with periodic boundary condition on the domain $[0,2\pi]^2$.

We use a uniform rectangular mesh. At $t = 0.1$, the solution is still smooth. Numerical errors and order of accuracy are listed in Table 4.9, demonstrating the expected order of accuracy. At $t = 1$, the solution is no longer smooth. We plot the numerical solution in Fig. 4.7. We observe good resolution of the kinks in the solution.

4.4. Nonlinear nonsmooth problems

In this subsection, the Hamiltonian H is a nonlinear nonsmooth function of $\nabla\varphi$.

Table 4.9
Errors and numerical orders of accuracy for Example 4.3.3 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of $N \times N$ cells

$N \times N$	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10 × 10	0.30E−02		0.43E−02		0.35E−01	
20 × 20	0.38E−03	2.98	0.58E−03	2.90	0.56E−02	2.64
40 × 40	0.48E−04	2.97	0.77E−04	2.91	0.80E−03	2.81
80 × 80	0.66E−05	2.87	0.11E−04	2.83	0.14E−03	2.55

Final time $t = 0.1$. CFL $= 0.1$.

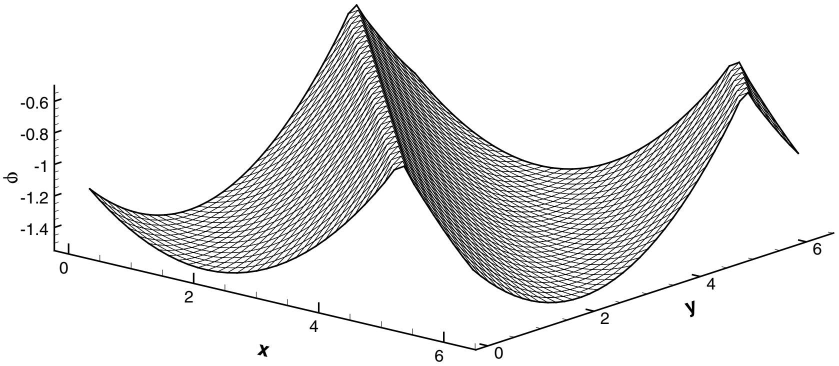

Fig. 4.7. Example 4.3.3. Numerical solution when $t = 1$, CFL = 0.1, 40 × 40 uniform mesh, using P^2 polynomials.

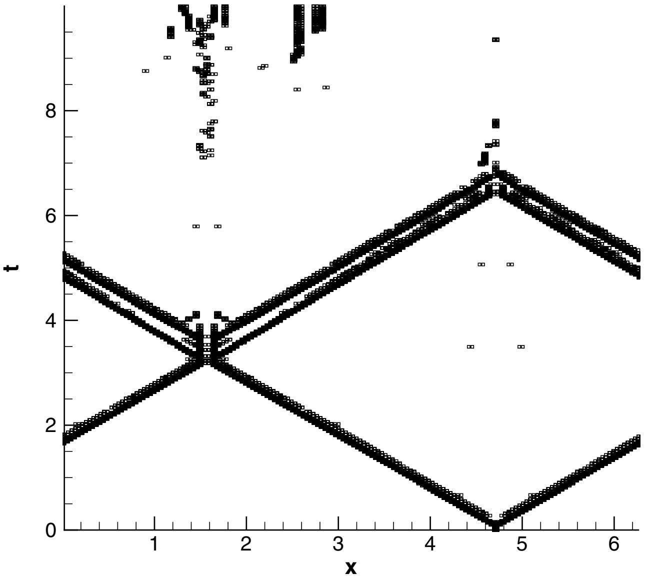

Fig. 4.8. Example 4.4.1. $t = 10$, CFL = 0.1, $N = 160$ uniform mesh, using P^2 polynomials. $\varepsilon = 10^{-10}$. Rectangular symbols mark the cells in which the entropy correction is performed. Those cells are plotted every five time steps.

Example 4.4.1. We solve the Eikonal equation given by

$$\begin{cases} \varphi_t + |\varphi_x| = 0 \\ \varphi(x,0) = \sin(x) \\ \varphi(0,t) = \varphi(2\pi,t) \end{cases} \tag{4.24}$$

The exact solution is the same as the exact solution of Example 4.2.1, given by (4.10)–(4.12). Because the entropy condition is violated by our scheme in some cells, we need to apply the entropy correction technique. The criteria are as follows. If we denote $u = \varphi_x$, then for the cell $I_j = (x_{j\text{-}1/2}, x_{j+1/2})$, if

$$u^-(x_{j-1/2}) < -\varepsilon \quad \text{and} \quad \varepsilon < u^+(x_{j-1/2}) \tag{4.25}$$

or

$$u^-(x_{j+1/2}) < -\varepsilon \quad \text{and} \quad \varepsilon < u^+(x_{j+1/2}) \tag{4.26}$$

are satisfied, we use the entropy correction on I_j. We take the parameter $\varepsilon = 10^{-10}$ in the calculation, which is introduced to avoid unnecessary entropy corrections due to small numerical errors in the computation.

Table 4.10
Errors and numerical orders of accuracy for Example 4.4.1 when using P^2 polynomials and Runge–Kutta third order time discretization on a uniform mesh of N cells

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
40	0.87E−03		0.15E−02		0.28E−02	
80	0.23E−03	1.89	0.41E−03	1.88	0.76E−03	1.89
160	0.64E−04	1.86	0.11E−03	1.86	0.21E−03	1.85
320	0.18E−04	1.85	0.31E−04	1.84	0.59E−04	1.85

Final time $t = 1$. CFL = 0.1.

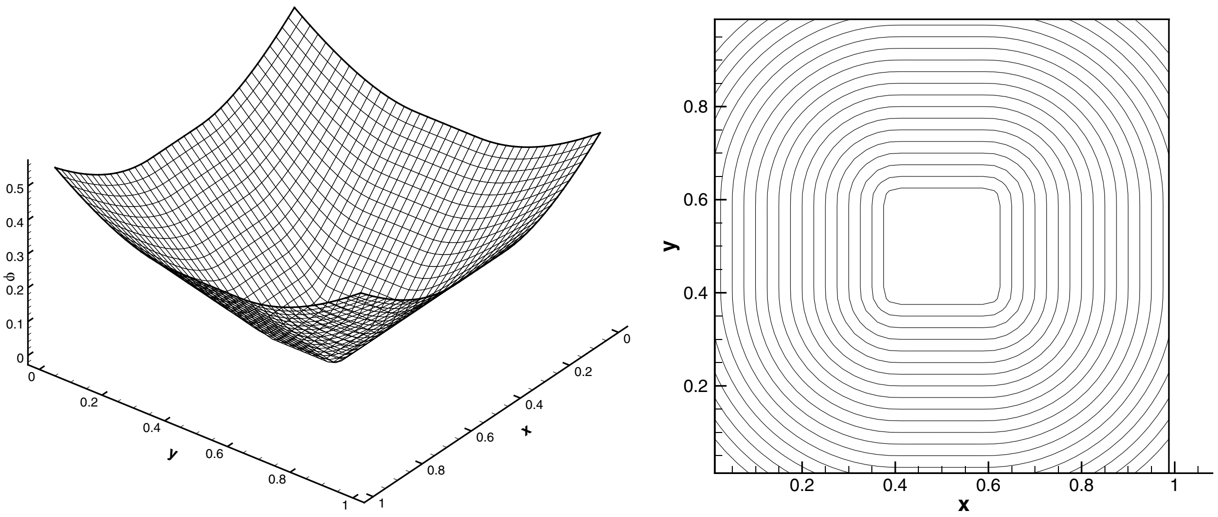

Fig. 4.9. Example 4.4.2. Steady state solution with 40 × 40 uniform mesh, using P^2 polynomials. Left: three-dimensional plot; right: contour plot.

Fig. 4.8 shows the space–time location where the entropy correction is applied. We observe that the correction is mostly applied at a few cells neighboring the boundary of the rarefaction wave. The number of cells in which the correction is performed is relatively small compared to the total number of cells.

The numerical errors and the order of accuracy are listed in Table 4.10. Since the solution is not smooth, we do not expect the full $(k + 1)$th order accuracy.

Example 4.4.2. We solve the two-dimensional Eikonal equation

$$\varphi_t + \sqrt{\varphi_x^2 + \varphi_y^2} = 1 \qquad (4.27)$$

First of all, we consider the case of the computational domain being $[0,1]^2 \backslash [0.4,0.6]^2$. For the inner boundary along $[0.4,0.6]^2$, we impose the boundary condition $\varphi = 0$. On the other hand, we impose free outflow boundary conditions on the outer boundary. The initial condition is taken as $\varphi_0(x,y) = \max\{|x - 0.5|, |y - 0.5|\} - 0.1$. The steady state solution should give us a function that is equal to the distance of the point to the inner boundary. For the outer boundary cells, we use the upwind version of our scheme according to the direction of the local wind. For all other cells, the general scheme (3.5) is used. We plot the numerical steady state solution in Fig. 4.9.

Next, we consider this example with a point source condition; namely, we take the inner boundary to be the center point (0.5, 0.5). In this case, we would need φ in the center cell to be the L^2 projection of the exact distance function. For all other cells, the computation is the same as for the previous case. The initial

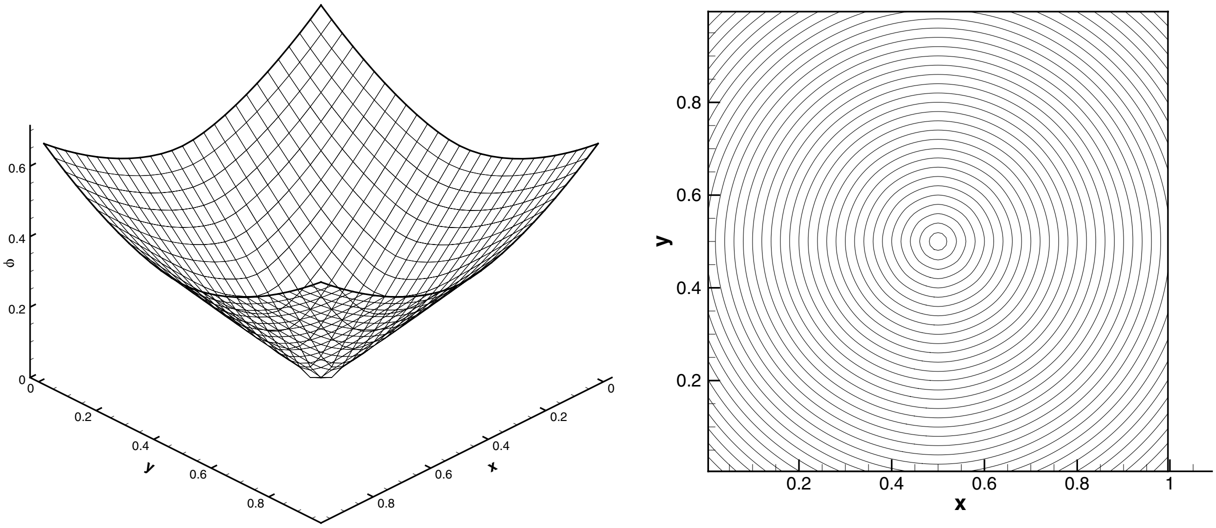

Fig. 4.10. Example 4.4.2. Steady state solution with 39 × 39 uniform mesh, using P^2 polynomials. Left: three-dimensional plot; right: contour plot.

condition is taken as $\varphi_0(x,y) = \max\{|x-0.5|,\ |y-0.5|\}$. We plot the numerical steady state solution in Fig. 4.10. We can see that in both cases we obtain very good resolution to the viscosity solution.

5. Concluding remarks

We have developed a discontinuous Galerkin finite element method for solving Hamilton–Jacobi equations approximating directly the solution variable rather than its derivatives as in the earlier work in [9,14]. Both linear and convex nonlinear Hamiltonians are considered in this paper, while the case for non-convex Hamiltonians is left for future study. One and two-dimensional numerical results demonstrate that the method approximates the viscosity solutions very well. In the future, we will also explore more direct entropy correction techniques without resorting to the techniques in [9,14]. We remark that Osher and Yan [15] has recently developed another class of discontinuous Galerkin type scheme for solving Hamilton–Jacobi equations, which also approximates directly the solution variable rather than its derivatives. A comparison of these two different approaches would be interesting.

Acknowledgments

Research supported by ARO Grant W911NF-04-1-0291, NSF Grant DMS-0510345 and AFOSR Grant FA9550-05-1-0123.

References

[1] B. Cockburn, S. Hou, C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation 54 (1990) 545–581.

[2] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, Journal of Computational Physics 84 (1989) 90–113.

[3] B. Cockburn, C.-W. Shu, The Runge–Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, Mathematical Modelling and Numerical Analysis 25 (1991) 337–361.

[4] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation 52 (1989) 411–435.

[5] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, Journal of Computational Physics 141 (1998) 199–224.

[6] M. Crandall, P.L. Lions, Viscosity solutions of Hamilton–Jacobi equations, Transactions of the American Mathematical Society 277 (1983) 1–42.

[7] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Mathematics of Computation 67 (1998) 73–85.

[8] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high order time discretization methods, SIAM Review 43 (2001) 89–112.

[9] C. Hu, C.-W. Shu, A discontinuous Galerkin finite element method for Hamilton–Jacobi equations, SIAM Journal on Scientific Computing 21 (1999) 666–690.

[10] G. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM Journal on Scientific Computing 21 (1999) 2126–2143.

[11] G. Jiang, C.-W. Shu, On cell entropy inequality for discontinuous Galerkin methods, Mathematics of Computation 62 (1994) 531–538.

[12] O. Lepsky, C. Hu, C.-W. Shu, The Analysis of the discontinuous Galerkin method for Hamilton–Jacobi equations, Applied Numerical Mathematics 33 (2000) 423–434.

[13] R. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (1996) 627–665.

[14] F. Li, C.-W. Shu, Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton–Jacobi equations, Applied Mathematics Letters 18 (2005) 1204–1209.

[15] S. Osher, J. Yan, A new discontinuous Galerkin method for the Hamilton–Jacobi equation, in preparation.

[16] S. Osher, C.-W. Shu, High order essentially non-oscillatory schemes for Hamilton–Jacobi equations, SIAM Journal on Numerical Analysis 28 (1991) 907–922.

[17] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.

[18] J. Yan, C.-W. Shu, A local discontinuous Galerkin method for KdV type equations, SIAM Journal on Numerical Analysis 40 (2002) 769–791.

[19] Y.-T. Zhang, C.-W. Shu, High order WENO schemes for Hamilton–Jacobi equations on triangular meshes, SIAM Journal on Scientific Computing 24 (2003) 1005–1030.