SIAM J. Sci. Comput.
Vol. 0, No. 0, pp. 000–000

A CLASS OF HIGH RESOLUTION DIFFERENCE SCHEMES FOR NONLINEAR HAMILTON–JACOBI EQUATIONS WITH VARYING TIME AND SPACE GRIDS*

HUAZHONG TANG† AND GERALD WARNECKE‡

Abstract. Based on a simple projection of the solution increments of the underlying partial differential equations (PDEs) at each local time level, this paper presents a difference scheme for nonlinear Hamilton–Jacobi (H–J) equations with varying time and space grids. The scheme is of good consistency and monotone under a local CFL-type condition. Moreover, one may deduce a conservative local time step scheme similar to Osher and Sanders scheme approximating hyperbolic conservation law (CL) from our scheme according to the close relation between CLs and H–J equations. Second order accurate schemes are constructed by combining the reconstruction technique with a second order accurate Runge–Kutta time discretization scheme or a Lax–Wendroff type method. They keep some good properties of the global time step schemes, including stability and convergence, and can be applied to solve numerically the initial-boundary-value problems of viscous H–J equations. They are also suitable to parallel computing.

Numerical errors and the experimental rate of convergence in the L^p-norm, $p = 1$, 2, and ∞, are obtained for several one- and two-dimensional problems. The results show that the present schemes are of higher order accuracy.

Key words. Hamilton–Jacobi equation, finite difference scheme, local time step discretization, Navier–Stokes equations

AMS subject classifications. 65M06, 65M99

DOI. 10.1137/S1064827503428126

1. Introduction.

Consider the Hamilton–Jacobi (H–J) equation

$$\phi_t + H(\mathbf{x}, t, \phi_{x_1}, \dots, \phi_{x_d}) = 0, \tag{1.1}$$

with initial data $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$, where $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$. The H–J equations have very important applications ranging from mathematical finance and differential games to front propagation and image enhancement. For this reason, there have been many theoretical and numerical studies of the H–J equations in the past two decades.

It is well known that the solutions of the above initial value problem are generally continuous, typically locally Lipschitz continuous, but with discontinuous derivatives after a finite time even if the initial data are smooth. It introduces great difficulties in theoretical analysis and obtaining numerical solutions of the H–J equations. The definition of viscosity solutions and the question of well-posedness were formulated and systematically studied by Crandall and Lions [5, 7] and Crandall, Ishi, and Lions [6]. In [7], Crandall and Lions studied the convergence of monotone finite difference schemes to the viscosity solutions of (1.1). Unfortunately, the monotone schemes are at most first order accurate, measured by local truncation errors, in the smooth regions of the solution. A rigorous analysis of convergence rates for the H–J equations

*Received by the editors May 21, 2003; accepted for publication (in revised form) June 8, 2004; published electronically DATE. This research was partially supported by the Special Funds for Major State Basic Research Projects of China, the National Natural Science Foundation of China, the Alexander von Humboldt foundation, and the Deutsche Forschungsgemeinschaft (DFG Wa 633/10-3).
http://www.siam.org/journals/sisc/x-x/42812.html

†LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China (hztang@math.pku.edu.cn).

‡Institüt für Analysis und Numerik, Otto-von-Guericke Universität Magdeburg, 39106 Magdeburg, Germany (Gerald.Warnecke@mathematik.uni-magdeburg.de).

can be found in [10]. Typically, there is a close relation between the H–J equations and hyperbolic conservation laws (CLs), and as a result the concepts used for CLs can be transferred to the H–J equations. The existing high resolution methods for solving the H–J equations include high order essentially nonoscillatory (ENO) schemes introduced by Osher and Sanders [16], Osher and Sethian [17], and Osher and Shu [18], and the central high resolution schemes proposed by Kurganov and Tadmor [12] and Lin and Tadmor [13, 14]. Jin and Xin [11] investigated the numerical passage of the relaxation approximation for CLs to the H–J equations. On the unstructured meshes, high order schemes constructed for the H–J equations are relatively rare. Abgrall [1] extended monotone-type finite volume schemes to first order H–J equations on triangular meshes and developed a high order approximation in [2]. In a recent work from Zhang and Shu [24], high order WENO schemes are developed on the unstructured meshes for two-dimensional nonlinear H–J equations. Tang and Tang [19] and Tang, Tang, and Zhang [20] studied adaptive mesh methods for multidimensional hyperbolic CLs and H–J equations.

In practice, when solving evolutionary problems numerically, it may occur that in some spatial regions one needs a smaller time step than in other regions. For example, when an adaptive grid method is introduced to resolve a singular or nearly singular solution, the allowable time step will be reduced for an explicit scheme. For an implicit scheme, the time step size is often constrained by nonlinear convergence too.

Based on a direct projection of the solutions, Osher and Sanders [16] proposed a first order accurate difference scheme for nonlinear CLs with varying space and time grids. Berger [3] did a study on conservation at space and time grid interfaces and gave a conservative scheme with multitime increments. In fact, they result in the same scheme. The main advantage of their schemes is the conservativity, which is very important in numerical approximations for hyperbolic CLs. However, they suffer a loss of consistency near a time grid interface.

The purpose of this paper is to study high resolution numerical approximations of nonlinear H–J equations with varying space and time grids. Because there is no need of conservation now, we will use the projection of the solution increments of the H–J equations to construct our local time discretization schemes, which can be conveniently implemented and are of good consistency. On the other hand, one may derive a conservative local time step scheme, similar to the Osher and Sanders scheme [16], approximating hyperbolic CLs from our schemes approximating the H–J equations. Second order accurate difference schemes will be constructed by combining the reconstruction technique with a second order accurate Runge–Kutta scheme or a Lax–Wendroff-type time discretization. The schemes can keep some good properties of the global time step schemes, including stability and convergence, and can be applied to solve numerically the initial-boundary-value (IBV) problems of general H–J equations with second order spatial derivatives. They are also suitable to parallel computing.

This paper is organized as follows. In section 2, a class of high resolution local time step discretization schemes for nonlinear H–J equations is presented based on a simple projection of the solution increments of the underlying PDEs at each local time step. Second order accurate difference schemes are constructed by combining the reconstruction technique with higher order accurate time discretization methods. In section 3, the local time step schemes are applied to several model problems, including a periodic problem of the two-dimensional Navier–Stokes equations in nonconservative form. We give numerical errors and the experimental rate of convergence in the

L^p-norm, $p = 1, 2, \infty$, to show the accuracy of the schemes. The paper is concluded with a few remarks in section 4.

2. Numerical schemes. For simplicity, in this section we will mainly restrict our attention to the one-dimensional scalar H–J equation

$$\phi_t + H(\phi_x) = 0, \tag{2.1}$$

subject to the initial data $\phi(x,0) = \phi_0(x)$, where $x \in \mathbb{R}$ and the Hamiltonian $H(u) \in C^1(\mathbb{R})$. A simple description of the scheme in two dimensions will be given at the end of section 2.1. Moreover, our schemes will be used to compute two-dimensional H–J equations with a general Hamiltonian in section 3.

Give a partition $\{x_j\}_{j\in\mathbb{Z}}$ of the physical domain $\mathbb{R}$, and denote $h_{j+\frac{1}{2}} = x_{j+1} - x_j$, $\phi_j^0 = \phi_0(x_j)$, and $u^n_{j+\frac{1}{2}} = (\phi^n_{j+1} - \phi^n_j)/h_{j+\frac{1}{2}}$, where x_j denotes a grid point, $j \in \mathbb{Z}$.

A three-point explicit scheme approximating (2.1) may be represented as

$$\phi_j^{n+1} = \phi_j^n - \tau_n \widehat{H}(u^n_{j-\frac{1}{2}}, u^n_{j+\frac{1}{2}}) =: \phi_j^n + (\delta\phi)_j^n, \tag{2.2}$$

where $\tau_n = t_{n+1} - t_n$, and $\widehat{H}_j = \widehat{H}(u_{j-\frac{1}{2}}, u_{j+\frac{1}{2}})$ is considered as a numerical Hamiltonian. In this paper, we assume that $\widehat{H}_j$ is monotone; that is to say, $\widehat{H}(u,v)$ is nondecreasing in the first variable and nonincreasing in the second variable. For the explicit scheme (2.2), there is generally a need for the time step size τ_n,

$$\tau_n \left(\frac{\widehat{H}_u}{h_{j-\frac{1}{2}}} - \frac{\widehat{H}_v}{h_{j+\frac{1}{2}}} \right) \le 1, \tag{2.3}$$

in order to keep the stability of the scheme, where $\widehat{H}_u = \frac{\partial \widehat{H}(u,v)}{\partial u}$ and $\widehat{H}_v = \frac{\partial \widehat{H}(u,v)}{\partial v}$. The condition (2.3) ensures that the right-hand side of (2.2) is a nondecreasing function with respect to ϕ^n_{j+p}, $p = -1, 0, 1$; that is to say, the scheme (2.2) is monotone under the previous assumption, and its solutions will be convergent to the viscosity solution of (2.1).

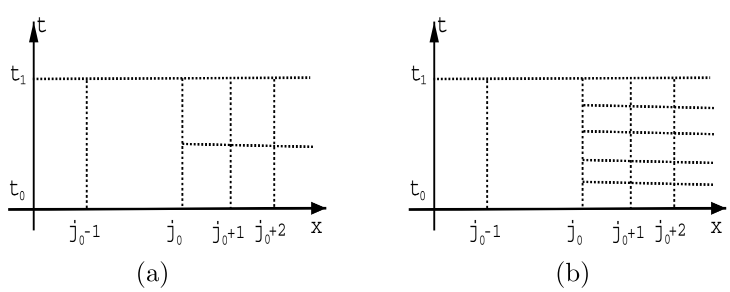

Fig. 1. *Nonuniform meshes in time and space.*

2.1. First order local time step discretization. We first consider a special case with two time increments, $\tau^{(1)} := \tau_n$ and $\tau^{(2)} := \frac{1}{2}\tau_n$, and assume that $\tau^{(1)}$ is used for the grid points with an index in the set $\mathcal{D}_1 = \{j | j \le j_0 - 1\}$ and $\tau^{(2)}$ is for $\mathcal{D}_2 = \{j | j \ge j_0\}$. See Figure 1(a); $\tau_n = t_1 - t_0$ there.

For this special case, we may compute directly ϕ_j^{n+1}, $j \le j_0 - 1$, by the scheme (2.2) once, and ϕ_j^{n+1}, $j \ge j_0 + 1$, by using the scheme (2.2) twice, i.e., at the local time levels $t_n + \frac{1}{2}\tau_n$ and $t_n + \tau_n$, after $\phi_{j_0}^{n+\frac{1}{2}}$ has been computed. The remaining problem is to compute $\phi_{j_0}^{n+1}$, which will depend on the solutions $\phi_{j_0-1}^{n+\frac{1}{2}}$, $\phi_{j_0}^{n+\frac{1}{2}}$, and $\phi_{j_0+1}^{n+\frac{1}{2}}$. Because the solutions $\phi_{j_0}^{n+\frac{1}{2}}$ and $\phi_{j_0+1}^{n+\frac{1}{2}}$ have been computed with a small time step, we consider defining $\phi_{j_0-1}^{n+\frac{1}{2}}$. There are two possible ways to define $\phi_{j_0-1}^{n+\frac{1}{2}}$. The first one is to directly project the solution as the one used in [16], i.e., evaluate $\phi_{j_0-1}^{n+\frac{1}{2}} = \phi_{j_0-1}^{n}$. Another is to project first the increment of the solution: assuming that $(\delta\phi)_{j_0-1}^{n}$ is known, then we evaluate or project $(\delta\phi)_{j_0-1}^{n+\frac{1}{2}} := (\delta\phi)_{j_0-1}^{n}$ and update $\phi_{j_0-1}^{n+\frac{1}{2}} = \phi_{j_0-1}^{n} + \frac{1}{2}(\delta\phi)_{j_0-1}^{n}$ and $\phi_{j_0-1}^{n+1} = \phi_{j_0-1}^{n+\frac{1}{2}} + \frac{1}{2}(\delta\phi)_{j_0-1}^{n+\frac{1}{2}}$ simultaneously. The former will suffer a loss of local consistency near a time grid interface, while the latter does not. To show this, we assume that $H(\phi_x) = c\phi_x$ and $h_{j+\frac{1}{2}} = h$ for any $j \in \mathbb{Z}$, and the numerical Hamiltonian in (2.2) is taken as

$$\widehat{H}_j = \frac{c}{h}(\phi_j - \phi_{j-1}),$$

where c and h are two positive constants. If there is a time grid interface at x_{j_0} as shown in Figure 1(a), then using the direct projection of the solution is used; i.e., $\phi_{j_0-1}^{n+\frac{1}{2}} = \phi_{j_0-1}^{n}$ gives

$$\begin{aligned} \phi_{j_0}^{n+1} &= \phi_{j_0}^{n+\frac{1}{2}} - \frac{c\tau_n}{2h}(\phi_{j_0}^{n+\frac{1}{2}} - \phi_{j_0-1}^{n+\frac{1}{2}}) \\ &= \phi_{j_0}^{n} - \frac{c\tau_n}{2h}(\phi_{j_0}^{n} - \phi_{j_0-1}^{n}) - \frac{c\tau_n}{2h}(\phi_{j_0}^{n+\frac{1}{2}} - \phi_{j_0-1}^{n}), \end{aligned} \tag{2.4}$$

where $\phi_{j_0}^{n+\frac{1}{2}}$ is computed by using (2.2) with a small time step size. The modified equation of the scheme (2.4) is derived by using the Taylor series expansion at x_{j_0} as

$$\phi_t + c\phi_x = -\frac{c\tau_n}{4h}\phi_t + \frac{ch}{2}\phi_{xx} - \frac{\tau_n}{2}\phi_{tt} - \frac{c\tau_n^2}{8h}\phi_{tt} + \mathcal{O}(h^2, \tau_n^2, \tau_n^3/h).$$

It is clear that the scheme (2.4) will be locally inconsistent with the underlying PDE $\phi_t + c\phi_x = 0$ if $\frac{\tau_n}{h} \to a$, a finite constant, even though $h \to 0$ and $\tau_n \to 0$.

For this reason, we will apply the projection of the solution increments to construct our schemes. For the case considered here, we first evaluate $(\delta\phi)_{j_0-1}^{n+\frac{1}{2}} := (\delta\phi)_{j_0-1}^{n}$ and then update the solutions $\phi_{j_0-1}^{n+\frac{1}{2}}$ and $\phi_{j_0-1}^{n+1}$ simultaneously as follows:

$$\phi_{j_0-1}^{n+\frac{1}{2}} = \phi_{j_0-1}^{n} + \frac{1}{2}(\delta\phi)_{j_0-1}^{n}, \tag{2.5}$$

$$\phi_{j_0-1}^{n+1} = \phi_{j_0-1}^{n+\frac{1}{2}} + \frac{1}{2}(\delta\phi)_{j_0-1}^{n+\frac{1}{2}} \equiv \phi_{j_0-1}^{n} + (\delta\phi)_{j_0-1}^{n}. \tag{2.6}$$

Obviously, (2.6) is identical to (2.2) at x_{j_0-1}. The computation of $\phi_{j_0-1}^{n+1}$ may be considered by implementing (2.2) twice. However, it does not actually increase the computational cost, because the solution increment $(\delta\phi)_{j_0-1}^{n+\frac{1}{2}}$ is evaluated by using simply the known value of $(\delta\phi)_{j_0-1}^{n}$.

The above scheme can be implemented as Algorithm I.

Step 1. Compute the solution increments $(\delta\phi)_j^n$ and the solutions $\phi_j^{n+\frac{1}{2}}$ for all $j \in \mathbb{Z}$:

$$\phi_j^{n+\frac{1}{2}} = \phi_j^n + \frac{1}{2}(\delta\phi)_j^n, \text{ or } \phi_j^{n+\frac{1}{2}} - \phi_j^n = \frac{1}{2}(\delta\phi)_j^n. \tag{2.7}$$

Step 2. Project the solution increments $(\delta\phi)_j^n$ such as

$$(\delta\phi)_j^{n+\frac{1}{2}} = \begin{cases} (\delta\phi)_j^n, & j \in \mathcal{D}_1, \\ -\tau_n \widehat{H}(u_{j-\frac{1}{2}}^{n+\frac{1}{2}}, u_{j+\frac{1}{2}}^{n+\frac{1}{2}}), & j \in \mathcal{D}_2. \end{cases} \tag{2.8}$$

Step 3. Update the solution ϕ_j^{n+1} at $t = t_{n+1}$ for all $j \in \mathbb{Z}$:

$$\phi_j^{n+1} = \phi_j^{n+\frac{1}{2}} + \frac{1}{2}(\delta\phi)_j^{n+\frac{1}{2}}. \tag{2.9}$$

Using $\tau^{(1)} := \tau_n$ and $\tau^{(2)} := \frac{1}{2}\tau_n$ as defined above, Algorithm I may be rewritten in a compact form as follows:

$$\phi_j^{n+\frac{1}{2}} = \phi_j^n + \tau^{(2)}(\delta\phi)_j^n, \quad j \geq j_0 - 1, \tag{2.10}$$

$$\phi_j^{n+1} = \begin{cases} \phi_j^n + \tau^{(1)}(\delta\phi)_j^n, & j \leq j_0 - 1, \\ \phi_j^{n+\frac{1}{2}} + \tau^{(2)}(\delta\phi)_j^{n+\frac{1}{2}}, & j \geq j_0. \end{cases} \tag{2.11}$$

Comparing (2.10)–(2.11) to the global time step discretization scheme (2.2), we have the following lemma.

LEMMA 2.1. *If the time step size $\tau^{(i)}$, $i = 1, 2$, satisfies*

$$\tau^{(i)} \left(\frac{\widehat{H}_u}{h_{j-\frac{1}{2}}} - \frac{\widehat{H}_v}{h_{j+\frac{1}{2}}} \right) \leq 1, \tag{2.12}$$

then the scheme (2.7)–(2.9) *is monotone and consistent. Moreover, it is also conservative with respect to $u = \phi_x$, i.e.,*

$$\sum_j u_{j+\frac{1}{2}}^{n+1} h_{j+\frac{1}{2}} = \cdots = \sum_j u_{j+\frac{1}{2}}^0 h_{j+\frac{1}{2}}. \tag{2.13}$$

Proof. The monotone property can be obtained by comparing (2.10)–(2.11) to the global time step discretization scheme (2.2).

To show consistency, we replace (2.4) by

$$\begin{aligned} \phi_{j_0}^{n+1} &= \phi_{j_0}^{n+\frac{1}{2}} - \frac{c\tau_n}{2h}(\phi_{j_0}^{n+\frac{1}{2}} - \phi_{j_0-1}^{n+\frac{1}{2}}) \\ &= \phi_{j_0}^n - \frac{c\tau_n}{2h}(\phi_{j_0}^n - \phi_{j_0-1}^n) - \frac{c\tau_n}{2h}(\phi_{j_0}^{n+\frac{1}{2}} - \phi_{j_0-1}^{n+\frac{1}{2}}). \end{aligned} \tag{2.14}$$

Again using the Taylor series expansion at x_{j_0}, we may derive the corresponding modified equation as follows:

$$\phi_t + c\phi_x = \frac{ch}{2}\phi_{xx} - \frac{\tau_n}{2}\phi_{tt} + \mathcal{O}(h^2, \tau_n^2).$$

Due to the definition of u and Algorithm I, the conservation may be found from

$$\begin{aligned} u_{j+\frac{1}{2}}^{n+1} &= u_{j+\frac{1}{2}}^n - \frac{\tau_n}{2h_{j+\frac{1}{2}}} \left(\widehat{H}_{j+1}^n - \widehat{H}_j^n + \widehat{H}_{j+1}^n - \widehat{H}_j^n \right), \ j \leq j_0 - 2, \\ u_{j_0-\frac{1}{2}}^{n+1} &= u_{j_0-\frac{1}{2}}^n - \frac{\tau_n}{2h_{j_0-\frac{1}{2}}} \left(\widehat{H}_{j_0}^n - \widehat{H}_{j_0-1}^n + \widehat{H}_{j_0}^{n+\frac{1}{2}} - \widehat{H}_{j_0-1}^n \right), \\ u_{j+\frac{1}{2}}^{n+1} &= u_{j+\frac{1}{2}}^n - \frac{\tau_n}{2h_{j+\frac{1}{2}}} \left(\widehat{H}_{j+1}^n - \widehat{H}_j^n + \widehat{H}_{j+1}^{n+\frac{1}{2}} - \widehat{H}_j^{n+\frac{1}{2}} \right), \ j \geq j_0. \end{aligned} \tag{2.15}$$

From these, the proof of Lemma 2.1 is completed. $\square$

The proof of Lemma 2.1 tells us that a conservative local time step discretization scheme (2.15) is derived for hyperbolic CLs

$$\frac{\partial u}{\partial t} + \frac{\partial H(u)}{\partial x} = 0, \tag{2.16}$$

which is similar to the Osher and Sanders scheme given in [16]. But there exists a slight difference between them, because the Osher and Sanders scheme is given as

$$\begin{aligned}
u^{n+1}_{j+\frac12} &= u^n_{j+\frac12} - \frac{\tau_n}{2h_{j+\frac12}}\left(\widehat{H}^n_{j+1} - \widehat{H}^n_j + \widehat{H}^n_{j+1} - \widehat{H}^n_j\right),\ j \le j_0 - 2,\\
u^{n+1}_{j_0-\frac12} &= u^n_{j_0-\frac12} - \frac{\tau_n}{2h_{j_0-\frac12}}\left(\widehat{H}^n_{j_0} - \widehat{H}^n_{j_0-1} + \widehat{H}\left(u^n_{j_0-\frac12}, u^{n+\frac12}_{j_0+\frac12}\right) - \widehat{H}\left(u^n_{j_0-\frac32}, u^{n+\frac12}_{j_0-\frac12}\right)\right),\\
u^{n+1}_{j_0+\frac12} &= u^n_{j_0+\frac12} - \frac{\tau_n}{2h_{j_0+\frac12}}\left(\widehat{H}^n_{j_0+1} - \widehat{H}^n_{j_0} + \widehat{H}\left(u^{n+\frac12}_{j_0+\frac12}, u^{n+\frac12}_{j_0+\frac32}\right) - \widehat{H}\left(u^n_{j_0-\frac12}, u^{n+\frac12}_{j_0+\frac12}\right)\right),\\
u^{n+1}_{j+\frac12} &= u^n_{j+\frac12} - \frac{\tau_n}{2h_{j+\frac12}}\left(\widehat{H}^n_{j+1} - \widehat{H}^n_j + \widehat{H}^{n+\frac12}_{j+1} - \widehat{H}^{n+\frac12}_j\right),\ j \ge j_0 + 1.
\end{aligned}$$

Obviously, the numerical implementation of the scheme (2.15) is more convenient than the Osher and Sanders scheme.

In the following, we extend Algorithm I to a more general case with multitime increments τ_n, $\alpha_l\tau_n$, $l = 1, \ldots, k$, where $\sum_{l=1}^k \alpha_l = 1$; see Figure 1(b). Define $\beta_0 = 0$, $\beta_l = \sum_{i=1}^l \alpha_i$, $l = 1, \ldots, k$.

The algorithm with multitime increments can be described as follows (we consider it as Algorithm II).

Step 1. Compute the solution increments $(\delta\phi)^n_j$ and the solutions $\phi^{n+\beta_1}_j$ for all $j \in \mathbb{Z}$:

$$(\delta\phi)^n_j := -\tau_n \widehat{H}(u^n_{j-\frac12}, u^n_{j+\frac12}), \tag{2.17}$$

$$\phi^{n+\beta_1}_j = \phi^n_j + \alpha_1(\delta\phi)^n_j. \tag{2.18}$$

Step 2. For $l = 2, \ldots, k$, do the following:

(a) Project the solution increments $(\delta\phi)^{n+\beta_{l-1}}_j$ at each local time level:

$$(\delta\phi)^{n+\beta_{l-1}}_j = \begin{cases} (\delta\phi)^n_j, & j \in \mathcal{D}_1,\\ -\tau_n \widehat{H}(u^{n+\beta_{l-1}}_{j-\frac12}, u^{n+\beta_{l-1}}_{j+\frac12}), & j \in \mathcal{D}_2. \end{cases} \tag{2.19}$$

(b) Update the solution $\phi^{n+\beta_l}_j$ at $t = t_n + \beta_l\tau_n$ for all j:

$$\phi^{n+\beta_l}_j = \phi^{n+\beta_{l-1}}_j + \alpha_l(\delta\phi)^{n+\beta_{l-1}}_j. \tag{2.20}$$

For the scheme (2.17)–(2.20), the conclusions of Lemma 2.1 still hold with $(\tau^{(1)}, \tau^{(2)}) = (\tau_n, \max_l\{\alpha_l\tau_n\})$, and we can also give a conservative scheme for (2.16) with multitime increments as follows:

$$\begin{aligned}
u^{n+1}_{j+\frac12} &= u^n_{j+\frac12} - \sum_{l=1}^k \frac{\alpha_l\tau_n}{2h_{j+\frac12}}\left(\widehat{H}^n_{j+1} - \widehat{H}^n_j\right),\ j \le j_0 - 2,\\
u^{n+1}_{j_0-\frac12} &= u^n_{j_0-\frac12} - \sum_{l=1}^k \frac{\alpha_l\tau_n}{2h_{j_0-\frac12}}\left(\widehat{H}^{n+\beta_{l-1}}_{j_0} - \widehat{H}^n_{j_0-1}\right),\\
u^{n+1}_{j+\frac12} &= u^n_{j+\frac12} - \sum_{l=1}^k \frac{\alpha_l\tau_n}{2h_{j+\frac12}}\left(\widehat{H}^{n+\beta_{l-1}}_{j+1} - \widehat{H}^{n+\beta_{l-1}}_j\right),\ j \ge j_0.
\end{aligned} \tag{2.21}$$

Before ending this subsection, we give a two-dimensional extension of the above schemes. The Cauchy problem for a two-dimensional H–J equation is given by

$$\begin{cases} \phi_t + H(\nabla\phi) = 0, \ (x,t) \in \mathbb{R}^2 \times (0,\infty), \\ \phi(x,0) = \phi_0(x), \ x \in \mathbb{R}^2. \end{cases} \tag{2.22}$$

Generally, (2.22) can be formulated as a system of CLs [11], simply by considering the equations satisfied by the gradient $\mathbf{u} = (u,v) = \nabla\phi(x,t)$ of the solution of the above Cauchy problem,

$$\begin{cases} \mathbf{u}_t + \nabla H(\mathbf{u}) = 0, \ (x,t) \in \mathbb{R}^2 \times (0,\infty), \\ \mathbf{u}(x,0) = \mathbf{u}_0(x) \equiv \nabla\phi_0(x), \ x \in \mathbb{R}^2. \end{cases} \tag{2.23}$$

For convenience, we restrict our attention to a regular but nonuniform mesh $\{(x_j, y_k)\}_{j,k\in\mathbb{Z}}$ for a rectangular domain Ω, for example, $\Omega = [x_L, x_R] \times [y_L, y_R]$. If a uniform time step size is used in Ω, then at each grid point (x_j, y_k), we may use the scheme

$$\phi^{n+1}_{j,k} = \phi^n_{j,k} + (\delta\phi)^n_{j,k}, \tag{2.24}$$

where $(\delta\phi)^n_{j,k} = -\tau_n \widehat{H}^n_{j,k}$, and $\widehat{H}^n_{j,k} = \widehat{H}\big(u^n_{j-\frac{1}{2},k}, u^n_{j+\frac{1}{2},k}, v^n_{j,k-\frac{1}{2}}, v^n_{j,k+\frac{1}{2}}\big)$ is any appropriate numerical Hamiltonian; see, e.g., (3.3). If $\widehat{H}_{j,k}$ is monotone with respect to its arguments, $\phi_{j\pm1,k}$ and $\phi_{j,k\pm1}$, then the scheme (2.24) is monotone under a suitable CFL condition.

We assume now that the computational domain Ω is discretized such that the half step size $h_x/2$ is within the horizontal strip $[x_a, x_b]$ and $h_y/2$ is within the vertical strip $[y_a, y_b]$, respectively, where $x_L \le x_a < x_b \le x_R$ and $y_L \le y_a < y_b \le y_R$. The half time step $\tau/2$ is used in $[x_a, x_b] \times [y_a, y_b]$, while the global step sizes h_x, h_y, and τ are taken in the rest of the domain Ω, respectively. Then the local time step scheme with two time increments may be given as follows:

$$\begin{cases} \phi^{n+\frac{1}{2}}_{j,k} = \phi^n_{j,k} + \frac{1}{2}(\delta\phi)^n_{j,k} \equiv \phi^n_{j,k} - \frac{1}{2}\tau_n \widehat{H}^n_{j,k}, & (j,k) \in \Omega^1_h, \\ \phi^{n+1}_{j,k} = \phi^{n+\frac{1}{2}}_{j,k} + \frac{1}{2}(\delta\phi)^{n+\frac{1}{2}}_{j,k} \equiv \phi^{n+\frac{1}{2}}_{j,k} - \frac{1}{2}\tau_n \widehat{H}^{n+\frac{1}{2}}_{j,k}, & (j,k) \in \Omega^1_h, \end{cases} \tag{2.25}$$

$$\begin{cases} \phi^{n+\frac{1}{2}}_{j,k} = \phi^n_{j,k} + \frac{1}{2}(\delta\phi)^n_{j,k} \equiv \phi^n_{j,k} - \frac{1}{2}\tau_n \widehat{H}^n_{j,k}, & (j,k) \in \Omega^2_h, \\ \phi^{n+1}_{j,k} = \phi^{n+\frac{1}{2}}_{j,k} + \frac{1}{2}(\delta\phi)^{n+\frac{1}{2}}_{j,k} \equiv \phi^{n+\frac{1}{2}}_{j,k} - \frac{1}{2}\tau_n \widehat{H}^n_{j,k}, & (j,k) \in \Omega^2_h, \end{cases} \tag{2.26}$$

where Ω^1_h denotes the set of the grid points in $[x_a, x_b] \times [y_a, y_b]$, and Ω^2_h denotes the set of the grid points in the rest of Ω. It is obvious that the scheme (2.25)–(2.26) is of a monotonicity property the same as that of (2.24), if $\widehat{H}_{j,k}$ is monotone with respect to its arguments as before, and the local and global CFL conditions hold within Ω^1_h and Ω^2_h, respectively. The conservation property may also be derived with respect to u and v, where $u_{j+\frac{1}{2},k} = (\phi_{j+1,k} - \phi_{j,k})/(x_{j+1} - x_j)$ and $v_{j,k+\frac{1}{2}} = (\phi_{j,k+1} - \phi_{j,k})/(y_{k+1} - y_k)$.

2.2. Higher order accurate spatial discretization. In this subsection, we introduce a reconstruction technique to improve the accuracy of the previous schemes in space. Following [23], we construct a piecewise linear function,

$$w^{n+\beta_l}_{j+\frac{1}{2}}(x) = u^{n+\beta_l}_{j+\frac{1}{2}} + S^{n+\beta_l}_{j+\frac{1}{2}}(x - x_{j+\frac{1}{2}}), \ x \in [x_j, x_{j+1}], \tag{2.27}$$

to replace the original piecewise constant function at each local time level $t = t_n + \beta_l \tau_n$, where $\beta_0 = 0$, $x_{j+\frac{1}{2}} = \frac{1}{2}(x_j + x_{j+1})$, and $S^{n+\beta_l}_{j+\frac{1}{2}}$ is a numerical slope approximating $(u_x)^{n+\beta_l}_{j+\frac{1}{2}}$, $l = 0, 1, \ldots, k-1$. High resolution local time step discretization schemes can be derived if the term $\widehat{H}(u_{j-\frac{1}{2}}, u_{j+\frac{1}{2}})$ in the previous algorithms is replaced by $\widehat{H}(u_{j,L}, u_{j,R})$, where $u_{j,L} = w_{j-\frac{1}{2}}(x_j)$ and $u_{j,R} = w_{j+\frac{1}{2}}(x_j)$.

There are many ways to define the approximate slope $S^{n+\beta_l}_{j+\frac{1}{2}}$. The commonly used formulae are

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \operatorname{minmod}\left(S^{n+\beta_l,L}_{j+\frac{1}{2}}, S^{n+\beta_l,R}_{j+\frac{1}{2}}\right), \tag{2.28}$$

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \left(\operatorname{sign}\left(S^{n+\beta_l,L}_{j+\frac{1}{2}}\right) + \operatorname{sign}\left(S^{n+\beta_l,R}_{j+\frac{1}{2}}\right)\right) \frac{\left|S^{n+\beta_l,L}_{j+\frac{1}{2}}\right| \cdot \left|S^{n+\beta_l,R}_{j+\frac{1}{2}}\right|}{\left|S^{n+\beta_l,L}_{j+\frac{1}{2}}\right| + \left|S^{n+\beta_l,R}_{j+\frac{1}{2}}\right| + \varepsilon}, \tag{2.29}$$

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \frac{\left(\left|S^{n+\beta_l,R}_{j+\frac{1}{2}}\right|^2 + \varepsilon^R\right) S^{n+\beta_l,L}_{j+\frac{1}{2}} + \left(\left|S^{n+\beta_l,L}_{j+\frac{1}{2}}\right|^2 + \varepsilon^L\right) S^{n+\beta_l,R}_{j+\frac{1}{2}}}{\left|S^{n+\beta_l,L}_{j+\frac{1}{2}}\right|^2 + \left|S^{n+\beta_l,R}_{j+\frac{1}{2}}\right|^2 + \varepsilon^L + \varepsilon^R}, \tag{2.30}$$

which are the Minmod limiter, the van Leer limiter, and the van Albada limiter, respectively. Here

$$S^{n+\beta_l,R}_{j+\frac{1}{2}} = \frac{\Delta_+ u^{n+\beta_l}_{j+\frac{1}{2}}}{\Delta_+ x_{j+\frac{1}{2}}}, \quad S^{n+\beta_l,L}_{j+\frac{1}{2}} = \frac{\Delta_+ u^{n+\beta_l}_{j-\frac{1}{2}}}{\Delta_+ x_{j-\frac{1}{2}}},$$

$\Delta_+ u^{n+\beta_l}_{j+\frac{1}{2}} = u^{n+\beta_l}_{j+\frac{3}{2}} - u^{n+\beta_l}_{j+\frac{1}{2}}$, ε is a small positive constant to avoid the denominator becoming zero, i.e., $0 < \varepsilon \ll 1$, and ε^L and ε^R are taken as $Ch_{j-\frac{1}{2}}$ and $Ch_{j+\frac{1}{2}}$, respectively. A limiter is used to ensure that the solutions of the high resolution schemes are oscillation free.

Remark 2.1. When using the piecewise linear function $w^{n+\beta_l}_{j+\frac{1}{2}}(x)$ to replace the piecewise constant function $u^{n+\beta_l}_{j+\frac{1}{2}}$, the stencil of the second order scheme becomes larger than that of the first order scheme. Thus, to keep consistency, we should also enlarge the projection region of the solution increments. As an example, we consider a simple case shown in Figure 1(a). When the first order scheme is used, we need only to project solution increments at x_{j_0-1} in order to evolve the solution at x_{j_0}. However, when we use the above second order scheme, we should project the solution increments at x_{j_0-1} and x_{j_0-2} in order to evolve the solution at x_{j_0} and calculate the approximate slope.

Remark 2.2. The above scheme is a MUSCL-type or slope-limiter type method and is explicit; similar to that for hyperbolic conservation laws, the CFL number should generally be taken less than 0.5 for a one-dimensional case and 0.25 for a two-dimensional case, respectively, in order to keep stability. Certainly, it may be relaxed with some further modification of the method.

2.3. Higher order accurate time discretization. To increase the accuracy of the time discretization, we use Runge–Kutta methods or Lax–Wendroff-type methods to replace the forward Euler time discretization used in the previous version.

A second order explicit TVD Runge–Kutta method is implemented in Algorithm III.

Step 1. Compute the solution increments $(\delta\phi)_j^n$ and the solutions $\phi_j^{n+\beta_1}$ at $t = t_n + \beta_1\tau_n$ for all $j \in \mathbb{Z}$:

$$\phi_j^{n+\beta_1,*} = \phi_j^n + \alpha_1(\delta\phi)_j^n, \tag{2.31}$$

$$\phi_j^{n+\beta_1} = \frac{1}{2}\left(\phi_j^n + \phi_j^{n+\beta_1,*} + \alpha_1(\delta\phi)_j^{n+\beta_1,*}\right), \tag{2.32}$$

where $(\delta\phi)_j := -\tau_n\widehat{H}(u_{j,L}, u_{j,R})$.

Step 2. For $l = 2,\ldots,k$, project the solution increments $(\delta\phi)_j^{n+\beta_{l-1}}$ and update the solutions $\phi_j^{n+\beta_l}$ at local time level $t = t_n + \beta_{l-1}\tau_n$:

$$(\delta\phi)_j^{n+\beta_{l-1}} = \begin{cases}(\delta\phi)_j^n, & j \in \mathcal{D}_1,\\ -\tau_n\widehat{H}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2,\end{cases} \tag{2.33}$$

$$\phi_j^{n+\beta_l,*} = \begin{cases}\phi_j^n + \beta_l(\delta\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_1,\\ \phi_j^{n+\beta_{l-1}} + \alpha_l(\delta\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_2,\end{cases} \tag{2.34}$$

$$(\delta\phi)_j^{n+\beta_l,*} = -\tau_n\widehat{H}(u_{j,L}^{n+\beta_l,*}, u_{j,R}^{n+\beta_l,*}), \tag{2.35}$$

$$\phi_j^{n+\beta_l} = \begin{cases}\frac{1}{2}\phi_j^n + \frac{1}{2}\left(\phi_j^{n+\beta_l,*} + \beta_l(\delta\phi)_j^{n+\beta_l,*}\right), & j \in \mathcal{D}_1,\\ \frac{1}{2}\phi_j^{n+\beta_{l-1}} + \frac{1}{2}\left(\phi_j^{n+\beta_l,*} + \alpha_l(\delta\phi)_j^{n+\beta_l,*}\right), & j \in \mathcal{D}_2.\end{cases} \tag{2.36}$$

To verify the accuracy of Algorithm III in time, we restrict ourselves to the cases shown in Figure 1(a) and define $L(\phi_{j-1}, \phi_j, \phi_{j+1}) = (\delta\phi)_j/\tau_n$ and $L' := \sum_i L_i$, where the term L_i denotes the partial derivative of $L(u, v, w)$ with respect to the ith variable. Moreover, we will assume that $L(\phi_{j-1\pm1}, \phi_{j\pm1}, \phi_{j+1\pm1}) = L(\phi_{j-1}, \phi_j, \phi_{j+1}) + \mathcal{O}(h)$, $\mathcal{O}(h) = \mathcal{O}(\tau_n)$, for all $j \in \mathbb{Z}$.

Using Taylor series expansion and Algorithm III, we have

$$\begin{aligned}
\phi_j^{n+\frac{1}{2}} &= \phi_j^n + \frac{\tau_n}{2}\left(L + \frac{\tau_n}{4}L'L\right)(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n) + \mathcal{O}(\tau_n^3),\ j \in \mathbb{Z},\\
\phi_j^{n+1} &= \phi_j^{n+\frac{1}{2}} + \frac{\tau_n}{2}\left(L + \frac{\tau_n}{4}L'L\right)(\phi_{j-1}^{n+\frac{1}{2}}, \phi_j^{n+\frac{1}{2}}, \phi_{j+1}^{n+\frac{1}{2}}) + \mathcal{O}(\tau_n^3),\ j \le j_0 - 2,\\
\phi_j^{n+1} &= \phi_j^n + \tau_n\left(L + \frac{\tau_n}{2}L'L\right)(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n) + \mathcal{O}(\tau_n^3),\ j \ge j_0 + 1.
\end{aligned}$$

In the following, we still need to check the truncation errors of Algorithm III at x_{j_0} and x_{j_0-1}. Again using Taylor series expansion provdes the following:

$$\begin{aligned}
\phi_{j_0}^{n+1} &= \frac{1}{2}\phi_{j_0}^{n+\frac{1}{2}} + \frac{1}{2}\left(\phi_{j_0}^{n+1,*} + \tau_n L(\phi_{j_0-1}^{n+1,*}, \phi_{j_0}^{n+1,*}, \phi_{j_0+1}^{n+1,*})\right),\\
&= \phi_{j_0}^{n+\frac{1}{2}} + \frac{\tau_n}{4}\left(L(\phi_{j_0-1}^{n+\frac{1}{2}}, \phi_{j_0}^{n+\frac{1}{2}}, \phi_{j_0+1}^{n+\frac{1}{2}}) + L(\phi_{j_0-1}^{n+1,*}, \phi_{j_0}^{n+1,*}, \phi_{j_0+1}^{n+1,*})\right)\\
&= \phi_{j_0}^{n+\frac{1}{2}} + \frac{\tau_n}{2}\left(L + \frac{\tau_n}{4}L'L\right)(\phi_{j_0-1}^{n+\frac{1}{2}}, \phi_{j_0}^{n+\frac{1}{2}}, \phi_{j_0+1}^{n+\frac{1}{2}}) + \mathcal{O}(\tau_n^3).
\end{aligned}$$

Here we have used the relations

$$\phi_j^{n+1,*} = \phi_j^{n+\frac{1}{2}} + \frac{\tau_n}{2}L(\phi_{j-1}^{n+\frac{1}{2}}, \phi_j^{n+\frac{1}{2}}, \phi_{j+1}^{n+\frac{1}{2}}) = \phi_j^n + \tau_n L(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n) + \mathcal{O}(\tau_n^2), \quad j \in \mathcal{D}_2,$$

$$\phi_j^{n+1,*} = \phi_j^n + \tau_n L(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n) = \phi_j^{n+\frac{1}{2}} + \frac{\tau_n}{2}L(\phi_{j-1}^{n+\frac{1}{2}}, \phi_j^{n+\frac{1}{2}}, \phi_{j+1}^{n+\frac{1}{2}}) + \mathcal{O}(\tau_n^2), \quad j \in \mathcal{D}_1.$$

Similarly, we have

$$\phi_{j_0-1}^{n+1} = \phi_{j_0-1}^{n} + \tau_n \left(L + \frac{\tau_n}{2} L'L \right) (\phi_{j_0-2}^n, \phi_{j_0-1}^n, \phi_{j_0}^n) + \mathcal{O}(\tau_n^3).$$

The above results show that the scheme (2.31)–(2.36) is second order accurate in time in the sense of the truncation errors.

The Lax–Wendroff-type method is another way to get a higher order accurate time discretization scheme. Using (2.1) and Taylor series expansion, we have

$$\phi_j^{n+\beta_{l+1}} = \phi_j^{n+\beta_l} - \alpha_l \tau_n \left(H(\phi_x)\right)_j^{n+\beta_l} + \frac{1}{2}(\alpha_l \tau_n)^2 \left((H')^2 \phi_{xx}\right)_j^{n+\beta_l} + \mathcal{O}(\tau_n^3). \tag{2.37}$$

If the term $(H(\phi_x))_j^{n+\beta_l}$ is approximated as before, and $\widehat{G}(u_{j,L}^{n+\beta_l}, u_{j,R}^{n+\beta_l})$ is used to denote numerical approximation of the term $((H')^2\phi_{xx})_j^{n+\beta_l}$, for example,

$$\widehat{G}(u_{j,L}^{n+\beta_l}, u_{j,R}^{n+\beta_l}) = \left(H'\left(\frac{1}{2}(u_{j,L}^{n+\beta_l} + u_{j,R}^{n+\beta_l}) \right) \right)^2 \frac{u_{j+\frac{1}{2}}^{n+\beta_l} - u_{j-\frac{1}{2}}^{n+\beta_l}}{\frac{1}{2}(h_{j+\frac{1}{2}} + h_{j-\frac{1}{2}})}, \tag{2.38}$$

then the corresponding Algorithm IV may be given as follows.

Step 1. Compute the increments $(\delta_t\phi)_j^n$ and $(\delta_{tt}\phi)_j^n$ for all $j \in \mathbb{Z}$:

$$(\delta_t\phi)_j^n := -\tau_n \widehat{H}(u_{j,L}^n, u_{j,R}^n), \quad (\delta_{tt}\phi)_j^n := \frac{(\tau_n)^2}{2} \widehat{G}(u_{j,L}^n, u_{j,R}^n), \tag{2.39}$$

Step 2. Update the solutions at $t = t_n + \beta_1 \tau_n$:

$$\phi_j^{n+\beta_1} = \phi_j^n + \alpha_1 (\delta_t\phi)_j^n + (\alpha_1)^2 (\delta_{tt}\phi)_j^n, \qquad j \in \mathbb{Z}. \tag{2.40}$$

Step 3. For $l = 2, \dots, k$, do the following:

(a) Project the increments $(\delta_t\phi)_j^{n+\beta_{l-1}}$ and $(\delta_{tt}\phi)_{j2}^{n+\beta_{l-1}}$ at $t = t_n + \beta_{l-1}\tau_n$:

$$(\delta_t\phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta_t\phi)_j^n, & j \in \mathcal{D}_1, \\ -\tau_n \widehat{H}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2, \end{cases} \tag{2.41}$$

and

$$(\delta_{tt}\phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta_{tt}\phi)_j^n, & j \in \mathcal{D}_1, \\ \frac{(\tau_n)^2}{2} \widehat{G}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2. \end{cases} \tag{2.42}$$

(b) Update the solutions at $t = t_n + \beta_l \tau_n$ for all j:

$$\phi_j^{n+\beta_l} = \begin{cases} \phi_j^n + \beta_l (\delta_t\phi)_j^{n+\beta_{l-1}} + (\beta_l)^2 (\delta_{tt}\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_1, \\ \phi_j^{n+\beta_{l-1}} + \alpha_l (\delta_t\phi)_j^{n+\beta_{l-1}} + (\alpha_l)^2 (\delta_{tt}\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_2. \end{cases} \tag{2.43}$$

It is worth noting that two “solution increments” have been projected at each local time level in the above algorithm.

3. Numerical experiments. Several examples will be considered in this section. All of them have been used by several authors to test various numerical schemes. Three limiters listed in section 2.2 have been checked, but to save space we will only give the results computed with the van Albada limiter (2.30). The results computed with the van Leer limiter (2.29) are similar to those shown in the following, but those obtained with the Minmod limiter are more diffusive and less accurate. In our computations, the parameter C is taken as 3 in (2.30), and the CFL number is taken as 0.48 and 0.24 for the one- and two-dimensional cases, respectively, unless stated otherwise.

3.1. One-dimensional problems. For the computations of one-dimensional problems, the numerical Hamiltonian

$$\widehat{H}(u_{j,L}, u_{j,R}) = H\left(\frac{u_{j,L}+u_{j,R}}{2}\right) - \frac{\max_{u\in I_j}\{|H'(u)|\}}{2}(u_{j,R}-u_{j,L}) \tag{3.1}$$

is used, where $I_j = [\min\{u_{j,L}, u_{j,R}\}, \max\{u_{j,L}, u_{j,R}\}]$. The second order derivative ϕ_{xx} will be discretized as one given in (2.38). The L^p-errors, $p = 1, 2, \infty$, are estimated as follows:

$$\begin{aligned}
e_N^1 &= \sum_j |\phi_N(x_j,T) - \phi(x_j,T)|\frac{1}{2}(h_{j+\frac{1}{2}} + h_{j-\frac{1}{2}}),\\
e_N^2 &= \sqrt{\sum_j |\phi_N(x_j,T) - \phi(x_j,T)|^2\frac{1}{2}(h_{j+\frac{1}{2}} + h_{j-\frac{1}{2}})},\\
e_N^\infty &= \max_j\{|\phi_N(x_j,T) - \phi(x_j,T)|\},
\end{aligned}$$

where N denotes the number of grid points, and $\phi_N(x_j,T)$ and $\phi(x_j,T)$ denote the approximate and exact solutions at $t = T$, respectively. The experimental rate of convergence is computed as $p^i = \log(e_N^i/e_{2N}^i)/\log(2)$, $i = \infty$, 1, or 2.

Example 3.1. We solve a linear convection-diffusion equation

$$\begin{cases}\phi_t + c\phi_x = a\phi_{xx}, & (x,t) \in [0,2\pi[\, \times\,]0,T[\,,\\ \phi(x,0) = \sin(x), & x \in [0,2\pi[\,,\\ \phi(x+2\pi,t) = \phi(x,t), & t \in [0,T],\end{cases} \tag{3.2}$$

where c and $a \geq 0$ are both constants. The exact solution is $\phi(x,t) = e^{-at}\sin(x-ct)$ [4]. We compute the solutions up to $T = 2$ for two cases: $(a,c) = (0,1)$ and $(a,c) = (1,1)$, and set that the half step sizes $h/2$ and $\tau/2$ are used within $[\pi, 1.5\pi]$, while the global step sizes h and τ are in other regions. Tables 1–3 show the errors and convergence order obtained by Algorithms III and IV, respectively.

The numerical results show that a second order rate of convergence is obtained for the problem in (3.2). When three local time steps 0.5τ, 0.1τ, and 0.4τ are used within $[\pi, 1.5\pi]$ instead of the previous two time increments, we have obtained fully the same data.

Example 3.2. This example is to solve the H–J equation (2.1) with a convex H (Burgers' equation)

$$H(u) = \frac{1}{2}(u+1)^2.$$

Further 2π-periodic initial data

$$\phi(x,0) = -\cos\bigl(\pi(x-x_0)\bigr), \quad x \in [-1,1[\,,$$

are taken as in [20], where $x_0 = 0.85$.

To verify the convergence rate for the local time step discretization schemes, we take the half step sizes $h/2$ and $\tau/2$ within $[-0.2, 0.2]$, while the global step sizes h and τ are used in the other regions, and solve the problem up to $t = 0.5/\pi^2$, when the solution is still smooth.

TABLE 1
Example 3.1: *The errors and rate of convergence for the case* $(a, c) = (0, 1)$ *in* (3.2) *obtained by Algorithm* III.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
25	2.02e-2	–	5.37e-2	–	2.58e-2	–
50	5.03e-3	2.01	1.33e-2	2.01	6.27e-3	2.04
100	1.22e-3	2.04	3.30e-3	2.01	1.54e-3	2.03
200	2.96e-4	2.04	8.24e-4	2.00	3.79e-4	2.02

TABLE 2
Example 3.1: *Same as Table* 1, *except for* $(a, c) = (1, 1)$ *in* (3.2).

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
25	3.01e-3	–	8.99e-3	–	4.42e-3	–
50	8.20e-4	1.88	2.42e-3	1.89	1.19e-4	1.89
100	2.13e-4	1.94	6.26e-4	1.95	3.07e-4	1.95
200	5.43e-5	1.97	1.59e-4	1.98	7.79e-5	1.98

TABLE 3
Example 3.1: *Same as Table* 1, *except for Algorithm* IV.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
25	1.06e-2	–	1.45e-2	–	1.14e-2	–
50	2.28e-3	2.22	2.99e-2	2.28	2.37e-3	2.27
100	5.20e-3	2.13	6.66e-3	2.17	5.33e-3	2.15
200	1.25e-4	2.06	1.54e-4	2.11	1.25e-4	2.09

TABLE 4
Example 3.2: *The errors and convergence order obtained by Algorithm* III *at* $t = 0.5\pi^2$.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
30	1.34e-2	–	1.71e-2	–	1.33e-2	–
60	2.74e-3	2.29	3.24e-3	2.40	2.55e-3	2.38
120	5.83e-4	2.23	7.00e-4	2.21	5.55e-4	2.20
240	1.26e-4	2.21	1.59e-4	2.14	1.27e-4	2.13

TABLE 5
Example 3.2: *Same as Table* 4, *except for Algorithm* IV.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
30	1.06e-2	–	1.34e-2	–	1.05e-2	–
60	2.36e-3	2.17	2.92e-3	2.20	2.32e-3	2.18
120	5.29e-4	2.16	6.42e-4	2.19	5.13e-4	2.19
240	1.15e-4	2.20	1.47e-4	2.13	1.19-4	2.11

Tables 4–5 show the errors and convergence order obtained by Algorithm III as well as Algorithm IV, respectively. The data show that a second order rate of convergence has been obtained by using the local time step discretization methods to solve nonlinear H–J equations.

Figures 2–3 show the solution $\phi_N(x_j, t)$ and $u^N_{j+\frac{1}{2}} = (\phi_{j+1} - \phi_j)/h_{j+\frac{1}{2}}$ approximating $\phi_x(x_{j+\frac{1}{2}}, t)$ at $t = 1.5/\pi^2$ obtained by Algorithm III as well as Algorithm IV, respectively, when the discontinuity in the ϕ_x is well developed. Here the number of grid cells is 60, and the solid line denotes the solution calculated by the global time step scheme on a uniform mesh with 2000 grid cells. The ability of the local time step discretization methods to capture and follow the moving discontinuity is clearly

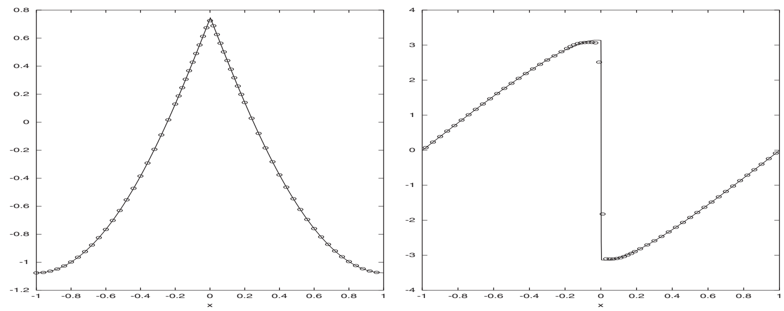

FIG. 2. *Comparison of the computed solutions ("○") with the "exact" solutions (solid line) of Example* 3.2 *given at* $t = 1.5\pi^2$. *Left:* $\phi(x,t)$. *Right:* $\phi_x(x,t)$.

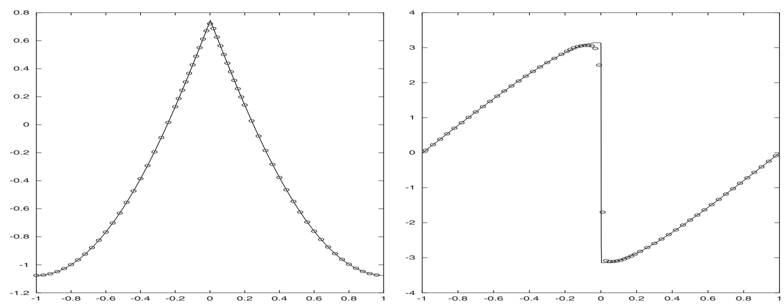

FIG. 3. *Same as Figure* 2*, except for Algorithm* IV.

demonstrated in these figures. The solutions obtained by using two different limiters are consistent.

When three local time steps 0.5τ, 0.1τ, and 0.4τ are used within $[-0.2, 0.2]$ instead of the previous two time increments, we have obtained the same data too.

Here, we just used fixed nonuniform meshes to demonstrate the performance of our present schemes. The adaption is now being considered combining the present local time step schemes with our adaptive grid methods in [20]. In principle, there is no big difficulty that the method creates, because we have resolved to move singularity for hyperbolic conservation laws with the local time step schemes in Example 5 of our paper [21]. The adaptive idea can also be found in [22].

3.2. Two-dimensional problems. In the following computations, we restrict ourselves to a regular but nonuniform mesh $\{(x_j, y_k)\}_{j,k\in\mathbb{Z}}$. The numerical Hamiltonian

$$\widehat{H}(u_{j-\frac{1}{2},k}, u_{j+\frac{1}{2},k}, v_{j,k-\frac{1}{2}}, v_{j,k+\frac{1}{2}}) = H\left(\frac{u_{j+\frac{1}{2},k} + u_{j-\frac{1}{2},k}}{2}, \frac{v_{j,k+\frac{1}{2}} + v_{j,k-\frac{1}{2}}}{2}\right)$$
$$- \frac{\max_{u\in I^u_{j,k}}\{|\alpha(u)|\}}{2}(u_{j+\frac{1}{2},k} - u_{j-\frac{1}{2},k}) - \frac{\max_{v\in I^v_{j,k}}\{|\beta(v)|\}}{2}(v_{j,k+\frac{1}{2}} - v_{j,k-\frac{1}{2}}) \tag{3.3}$$

is used to approximate the Hamiltonian $H(\phi_x, \phi_y)$, where $\alpha(u) = H_u(u, v_{j,k})$, $\beta(v) = H_v(u_{j,k}, v)$, and

$$u_{j\pm\frac{1}{2},k} = \frac{\phi_{j\pm1,k} - \phi_{j,k}}{x_{j\pm1} - x_j}, \quad v_{j,k\pm\frac{1}{2}} = \frac{\phi_{j,k\pm1} - \phi_{j,k}}{y_{k\pm1} - y_j},$$

$$I^u_{j,k} = [\min\{u_{j-\frac{1}{2},k}, u_{j+\frac{1}{2},k}\}, \max\{u_{j-\frac{1}{2},k}, u_{j+\frac{1}{2},k}\}],$$
$$I^v_{j,k} = [\min\{v_{j,k-\frac{1}{2}}, v_{j,k+\frac{1}{2}}\}, \max\{v_{j,k-\frac{1}{2}}, v_{j,k+\frac{1}{2}}\}],$$

as well as $H_w = \frac{\partial H(u,v)}{\partial w}$, $w = u$ or v.

Example 3.3. The first two-dimensional example is to solve scalar IBV problem [18]:

$$\phi_t + H(\phi_x, \phi_y) = 0, \quad \phi(x, y, 0) = -\cos\big(\pi(x+y)/2\big), \tag{3.4}$$

with a convex H: $H(u, v) = \frac{1}{2}(u+v+1)^2$, $-2 \le x, y \le 2$. It is a real two-dimensional H–J problem. We can use the one-dimensional exact solution to analyze our numerical results because under the transformation $\xi = (x+y)/2$ and $\eta = (x-y)/2$, the above IBV problem becomes the one-dimensional IBV problem in the ξ-direction in Example 3.2. However, since we use (x, y) coordinates, this is a true two-dimensional test problem. We compute to $t = t_1 = 0.5/\pi^2$ as well as $t = t_2 = 1.5/\pi^2$. The computational domain is discretized such that the half step sizes $h_x/2$ and $h_y/2$ are taken within $[-0.5, 0.5]$, respectively, and $\tau/2$ is used in $[-0.5, 0.5] \times [-0.5, 0.5]$, while the global step sizes h_x, h_y, and τ are taken in other domains. The results are presented in Table 6 and Figure 4.

TABLE 6
Example 3.3: *The errors and convergence order for solutions at* $t = 0.5/\pi^2$.

N×N	e^∞_N	p^∞	e^1_N	p^1	e^2_N	p^2
20×20	2.15e-2	–	4.97e-2	–	2.07e-2	–
40×40	5.95e-3	1.85	1.36e-2	1.87	5.91e-3	1.81
80×80	1.41e-3	2.08	3.40e-3	2.00	1.44e-3	2.04
160×160	3.11e-4	2.18	8.36e-4	2.03	3.39e-4	2.09

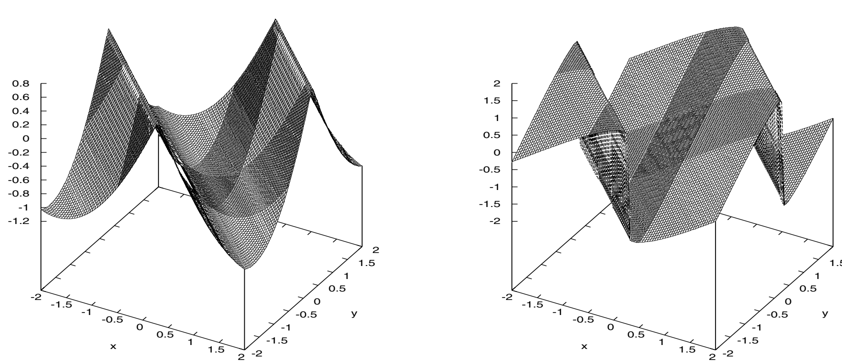

FIG. 4. *The computed solutions of Example* 3.3 *at* $t = 1.5/\pi^2$, 100×100 *cells. Left:* $\phi(x, y, t)$. *Right:* $\phi_\xi(x, y, t)$.

Example 3.4. This example is to compute two-dimensional Navier–Stokes equations [15]:

$$\begin{cases} \omega_t + \mathbf{u}\cdot\nabla\omega = \frac{1}{Re}\Delta\omega, \\ \Delta\psi = \omega, \quad \mathbf{u} = \nabla^{\perp}\psi, \end{cases} \tag{3.5}$$

and check the accuracy of Algorithm III with the van Albada limiter, where $(x, y) \in [0, 2\pi[\times [0, 2\pi[$ and $\nabla^{\perp} = (-\partial_y, \partial_x)$. The two-dimensional incompressible Navier–Stokes equation (3.5) may be considered as an H–J equation with a viscosity. Our purpose of solving (3.5) is to check effectiveness as well as accuracy of our schemes for an H–J-type equation with higher order spatial dervatives.

For this problem, the periodic boundary conditions are specified on four boundaries of the computational domain, and the Reynolds number Re is taken as 100. The discrete Poisson equation for the stream function ψ is solved iteratively by a Jacobi-type iteration.

The initial condition is taken such that the exact solution of the problem is known as

$$\omega(x,y,t) = -2\sin(x)\sin(y)e^{-\frac{2t}{Re}}, \qquad \psi(x,y,t) = \sin(x)\sin(y)e^{-\frac{2t}{Re}},$$
$$u(x,y,t) = -\sin(x)\cos(y)e^{-\frac{2t}{Re}}, \qquad v(x,y,t) = \cos(x)\sin(y)e^{-\frac{2t}{Re}}.$$

The computational domain is discretized such that the half step sizes $h_x/2$ and $h_y/2$ are taken within $[0.8, 1.2]$, respectively, and $\tau/2$ is used in $[0.8, 1.2] \times [0.8, 1.2]$, while the global step sizes h_x, h_y, and τ are taken in other domains. Table 7 shows the errors and convergence orders for the vorticity function at $t = 2$.

TABLE 7
Example 3.4: *The errors and convergence order for solutions at* $t = 2$.

N×N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
24×24	1.71e-2	–	1.93e-1	–	4.07e-2	–
48×48	3.06e-3	2.48	3.22e-2	2.58	6.63e-3	2.63
96×96	5.06e-4	2.60	5.54e-3	2.54	1.16e-3	2.51
192×192	1.13e-4	2.16	1.04e-3	2.41	2.14e-4	2.44

4. Concluding remarks. A class of high resolution local time step schemes have been presented for nonlinear H–J equations (1.1) in this paper, based on a simple projection of the solution increments at each local time step.

Second order accurate difference schemes were constructed using the reconstruction technique, and the Runge–Kutta- or Lax–Wendroff-type time discretization method. The local time step schemes are of good consistency, keep some good properties of the global time step schemes, including stability and convergence, and can be applied to solve numerically the IBV problems of general H–J equations with higher order spatial derivatives. They are suitable to parallel computing too. Moreover, from our schemes, one may derive a conservative local time scheme approximating hyperbolic conservation laws similar to Osher and Sanders scheme. The main idea can be used in construction of finite element methods, etc., with varying time and space grids.

The present schemes have been used to solve numerically several model problems, including a periodic problem of the two-dimensional incompressible Navier–Stokes equations. The numerical results show that a second-order rate of convergence could be obtained by the presented schemes in computations of one- and two-dimensional problems.

In the future, we will apply the local time step schemes to improve the efficiency of the adaptive grid algorithms and analyze the computational cost of the local time step schemes. Another interesting topic is to construct third and higher order accurate schemes with locally varying time and space grids.

REFERENCES

[1] R. Abgrall, *Numerical discretization of the first order Hamilton-Jacobi equation on triangular meshes*, Comm. Pure Appl. Math., 49 (1996), pp. 1339–1373.

[2] S. Augoula and R. Abgrall, *High order numerical discretization for Hamilton-Jacobi equations on triangular meshes*, J. Sci. Comput., 15 (2000), pp. 197–229.

[3] M. J. Berger, *On conservation at grid interfaces*, SIAM J. Numer. Anal., 24 (1987), pp. 967–984.

[4] B. Cockburn and C. W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463.

[5] M. G. Crandall and P. L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 277 (1983), pp. 1–42.

[6] M. G. Crandall, H. Ishi, and P. L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. (N.S.), 27 (1992), pp. 1–67.

[7] M. G. Crandall and P. L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19.

[8] C. Q. Hu and C. W. Shu, *A discontinuous Galerkin finite element method for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 666–690.

[9] G. S. Jiang and D. P. Peng, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.

[10] E. R. Jakobsen, K. H. Karlsen, and N. H. Risebro, *On the convergence rate of operator splitting for Hamilton–Jacobi equations with source terms*, SIAM J. Numer. Anal., 39 (2001), pp. 499–518.

[11] S. Jin and Z. Xin, *Numerical passage from systems of conservation laws to Hamilton–Jacobi equations, and relaxing schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 2385–2404.

[12] A. Kurganov and E. Tadmor, *New high-resolution semi-discrete central schemes for Hamilton-Jacobi equations*, J. Comput. Phys., 160 (2000), pp. 720–742.

[13] C.-T. Lin and E. Tadmor, *High-resolution nonoscillatory central schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2163–2186.

[14] C.-T. Lin and E. Tadmor, *L^1-stability and error estimates for approximate Hamilton-Jacobi solutions*, Numer. Math., 87 (2002), pp. 701–735.

[15] J.-G. Liu and C.-W. Shu, *A high order discontinuous Galerkin method for 2D incompressible flows*, J. Comput. Phys., 160 (2000), pp. 577–596.

[16] S. Osher and R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp., 41 (1983), pp. 321–336.

[17] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.

[18] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 28 (1991), pp. 907–922.

[19] H.-Z. Tang and T. Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., 41 (2003), pp. 487–515.

[20] H.-Z. Tang, T. Tang, and P. W. Zhang, *An adaptive mesh redistribution method for nonlinear Hamilton-Jacobi equations in two- and three-dimensions*, J. Comput. Phys., 188 (2003), pp. 543–572.

[21] H.-Z. Tang and G. Warnecke, *A Class of High Resolution Schemes for Hyperbolic Conservation Laws and Convection-Diffusion Equations with Varying Time and Space Grids*, http://www.math.ntnu.no/conservation/2003/036.html, 2003.

[22] Z. J. Tan, Z. R. Zhang, Y. Q. Huang, and T. Tang, *Moving Mesh Methods with Locally Varying Time Steps*, http://www.math.hkbu.edu.hk/~ttang/Papers/submit_03.pdf, 2003.

[23] B. van Leer, *Towards the ultimate conservative difference scheme,* V. *A second order sequel to Godunov's method*, J. Comput. Phys., 32 (1979), pp. 101–136.

[24] Y. T. Zhang and C. W. Shu, *High order WENO schemes for Hamilton–Jacobi equations on triangular meshes*, SIAM J. Sci. Comput., 24 (2003), pp. 1005–1030.