A Class of High Resolution Difference Schemes for Nonlinear Hamilton-Jacobi Equations with Varying Time and Space Grids

Huazhong Tang *   Gerald Warnecke †

May 21, 2003

Abstract

Based on a simple projection of the solution increments of the underlying partial differential equations (PDE) 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 laws (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..

*LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, P. R. China. Email: hztang@math.pku.edu.cn, or Huazhong.Tang@mathematik.uni-magdeburg.de

†Institüt für Analysis und Numerik, Otto-von-Guericke Universität Magdeburg, 39106 Magdeburg, Germany. Email: Gerald.Warnecke@mathematik.uni-magdeburg.de

Numerical errors and the experimental rate of convergence in 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.

AMS Subject Classification: 65M06, 65M99

Keywords: Hamilton-Jacobi equation, finite difference scheme, local time step discretization, Navier-Stokes equations.

1 Introduction

Consider the Hamilton-Jacobi (H-J) equations

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

with initial data $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$, where $\mathbf{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$. 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 they are 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, Evans, Lions, and many others, see e.g. [5, 6, 7]. 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 can be found in [10]. Typically, there is a close relation between the H-J equations and hyperbolic conservation laws, and as a result the concepts used for conservation laws can be transfered 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 et al. [9, 17, 18], and the central high resolution schemes proposed by Tadmor et al. [12, 13, 14]. Jin and Xin [11] investigated the numerical passage of the relaxation approximation for conservation laws 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 of Zhang and Shu [22], high order WENO schemes are developed on the unstructured meshes for two-dimensional nonlinear H-J equations. Tang et al. [19, 20] studied adaptive mesh methods for multi-dimensional hyperbolic conservation laws and H-J equations.

In practice, when solving evolutionary problems numerically, it may occur that in some spatial regions one needs a smaller 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 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 conservation laws 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 multi-time increments. In fact, they result in the same scheme. The main advantage of their schemes are the conservativity, which is very important in numerical approximations for hyperbolic conservation laws. 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 Osher and Sanders scheme [16] approximated hyperbolic conservation laws 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 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 are 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 non-conservative form. We give numerical errors and the experimental rate of convergence in 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 one–dimensional scalar H-J equations

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

subject to the initial data $\phi(x, 0) = \phi_0(x)$, where $x \in \mathbb{R}$ and the Hamiltonian $H(u) \in C^1(\mathbb{R})$. However, our schemes will be used to compute two-dimensional H-J equations with a general Hamiltonian in Section 3.

Give a partition $\{x_j\}$ 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 $u_{j+\frac{1}{2}}^n = (\phi_{j+1}^n - \phi_j^n)/h_{j+\frac{1}{2}}$, where x_j denotes a grid point, $j \in \mathbb{Z}$.

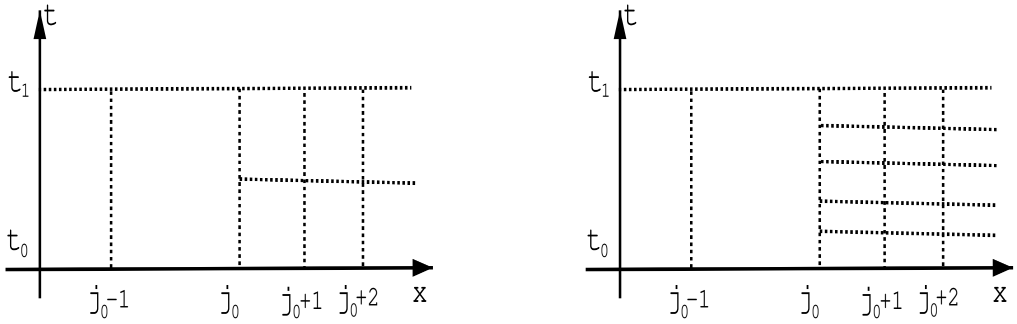
A three-point explicit scheme approximating Eq.(2.1) may be represented as

$$\phi_j^{n+1} = \phi_j^n - \tau_n \widehat{H}(u_{j-\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n), \tag{2.2}$$

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:

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

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).



(a) (b)

Figure 1: Non–uniform 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 index in the set $\mathcal{D}_1 = \{j | j \leq j_0 - 1\}$ and $\tau^{(2)}$ is for $\mathcal{D}_2 = \{j | j \geq j_0\}$, see Fig.1(a), $\tau_n = t_1 - t_0$ there.

For this special case, we may compute directly ϕ^{n+1}_j, $j \leq j_0 - 1$, by the scheme (2.2) one time, and ϕ^{n+1}_j, $j \geq j_0 + 1$, by using the scheme (2.2) two times, i.e. at the local time levels $t_n + \frac{1}{2}\tau_n$ and $t_n + \tau_n$, after $\phi^{n+\frac{1}{2}}_{j_0}$ has been computed. The remaining problem is to compute $\phi^{n+1}_{j_0}$, which will depend on the solutions $\phi^{n+\frac{1}{2}}_{j_0-1}$, $\phi^{n+\frac{1}{2}}_{j_0}$, and

$\phi_{j_0+1}^{n+\frac{1}{2}}$. Because the solution $\phi_{j_0}^{n+\frac{1}{2}}$ and $\phi_{j_0+1}^{n+\frac{1}{2}}$ have been computed with a small time step, we are considering to define $\phi_{j_0-1}^{n+\frac{1}{2}}$. There are two possible ways to define $\phi_{j_0-1}^{n+\frac{1}{2}}$. The first one is to use the direct projection of the solution as the one used in [16]. Another is to project the increment of the solution, see (2.5). The previous will suffer a loss of local consistency near a time grid interface. To show this, we assume that $H(\phi_x) = c\phi_x$ and $h_{j+\frac{1}{2}} = h$, 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 Fig.1(a), and the direct projrction of the solution is used, then we have

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

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 partial differential equation, $\phi_t + c\phi_x = 0$, 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, the solutions $\phi_{j_0-1}^{n+\frac{1}{2}}$ may be defined by

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

where

$$(\delta\phi)_{j_0-1}^{n} := -\tau_n \widehat{H}(u_{j_0-\frac{3}{2}}^{n}, u_{j_0-\frac{1}{2}}^{n}) \equiv \phi_{j_0-1}^{n+1} - \phi_{j_0-1}^{n}. \tag{2.6}$$

If we define $(\delta\phi)_{j_0-1}^{n+\frac{1}{2}} := (\delta\phi)_{j_0-1}^{n}$, then we may rewrite the scheme (2.2) at x_{j_0-1} as

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

It means that we may consider the computation of $\phi_{j_0-1}^{n+1}$ as using the scheme (2.2) two times. However, we know that it does actually not increase the computational cost, because the solution increment $(\delta\phi)_{j_0-1}^{n+\frac{1}{2}}$ is obtained by a simple projection.

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 $j \in \mathbb{Z}$:

$$(\delta\phi)_j^n := -\tau_n \widehat{H}(u_{j-\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n), \tag{2.7}$$

$$\phi_j^{n+\frac{1}{2}} = \phi_j^n + \frac{1}{2}(\delta\phi)_j^n. \tag{2.8}$$

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

$$(\delta\phi)_j^{n+\frac{1}{2}} = \begin{cases} (\delta\phi)_j^n, & j \in \mathcal{D}_1, \\ -\tau_n \widehat{H}(u_{j-\frac{1}{2}}^{n+\frac{1}{2}}, u_{j+\frac{1}{2}}^{n+\frac{1}{2}}), & j \in \mathcal{D}_2. \end{cases} \tag{2.9}$$

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

$$\phi_j^{n+1} = \phi_j^{n+\frac{1}{2}} + \frac{1}{2}(\delta\phi)_j^{n+\frac{1}{2}}. \tag{2.10}$$

The *Algorithm I* may also be rewritten in a compact form as follows:

$$\phi_j^{n+\frac{1}{2}} = \phi_j^n - \tau^{(2)} \widehat{H}(u_{j-\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n), \quad j \geq j_0 - 1, \tag{2.11}$$

$$\phi_j^{n+1} = \begin{cases} \phi_j^n - \tau^{(1)} \widehat{H}(u_{j-\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n), & j \leq j_0 - 1, \\ \phi_j^{n+\frac{1}{2}} - \tau^{(2)} \widehat{H}(u_{j-\frac{1}{2}}^{n+\frac{1}{2}}, u_{j+\frac{1}{2}}^{n+\frac{1}{2}}), & j \geq j_0. \end{cases} \tag{2.12}$$

Comparing Eqs. (2.11)-(2.12) to the global time step discretization scheme (2.2), we have:

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

$$\tau^{(i)} \left(\frac{\widehat{H}_u}{h_{j-\frac{1}{2}}} - \frac{\widehat{H}_v}{h_{j+\frac{1}{2}}} \right) \leq 1, \tag{2.13}$$

then the scheme (2.7)-(2.10) is monotone. Moreover, it is also conservative in terms of u, i.e.

$$\sum_j u_{j+\frac{1}{2}}^{n+1} h_{j+\frac{1}{2}} = \cdots = \sum_j u_{j+\frac{1}{2}}^0 h_{j+\frac{1}{2}}. \tag{2.14}$$

Proof: The monotone property can be obtained by comparing (2.11)-(2.12) to the global time step discretization scheme (2.2). Here we show conservation. According to the definition of u and *Algorithm I*, we have

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

From these, we can complete the proof of the conservation. □

The proof of the **Lemma 2.1** tells us that we derived a conservative local time step discretization scheme approximation hyperbolic conservation laws

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

The conservative scheme (2.15) is similar to the scheme of Osher and Sanders given in [16], but there exists a slight difference between them, because the Osher and Sanders scheme is given as

$$\begin{aligned}
u_{j+\frac{1}{2}}^{n+1} =& u_{j+\frac{1}{2}}^{n} - \frac{\tau_n}{2h_{j+\frac{1}{2}}}\left(\widehat{H}_{j+1}^{n} - \widehat{H}_{j}^{n} + \widehat{H}_{j+1}^{n} - \widehat{H}_{j}^{n}\right),\ j \le j_0 - 2,\\
u_{j_0-\frac{1}{2}}^{n+1} =& u_{j_0-\frac{1}{2}}^{n} - \frac{\tau_n}{2h_{j_0-\frac{1}{2}}}\left(\widehat{H}_{j_0}^{n} - \widehat{H}_{j_0-1}^{n} + \widehat{H}(u_{j_0-\frac{1}{2}}^{n}, u_{j_0+\frac{1}{2}}^{n+\frac{1}{2}}) - \widehat{H}(u_{j_0-\frac{3}{2}}^{n}, u_{j_0-\frac{1}{2}}^{n+\frac{1}{2}})\right),\\
u_{j_0+\frac{1}{2}}^{n+1} =& u_{j_0+\frac{1}{2}}^{n} - \frac{\tau_n}{2h_{j_0+\frac{1}{2}}}\left(\widehat{H}_{j_0+1}^{n} - \widehat{H}_{j_0}^{n} + \widehat{H}(u_{j_0+\frac{1}{2}}^{n+\frac{1}{2}}, u_{j_0+\frac{3}{2}}^{n+\frac{1}{2}}) - \widehat{H}(u_{j_0-\frac{1}{2}}^{n}, u_{j_0+\frac{1}{2}}^{n+\frac{1}{2}})\right),\\
u_{j+\frac{1}{2}}^{n+1} =& u_{j+\frac{1}{2}}^{n} - \frac{\tau_n}{2h_{j+\frac{1}{2}}}\left(\widehat{H}_{j+1}^{n} - \widehat{H}_{j}^{n} + \widehat{H}_{j+1}^{n+\frac{1}{2}} - \widehat{H}_{j}^{n+\frac{1}{2}}\right),\ j \ge j_0 + 1.
\end{aligned}$$

In the following, we extend *Algorithm I* to a more general case with multi–time increments: τ_n, $\alpha_l \tau_n$, $l = 1, \cdots, k$, where $\sum_{l=1}^{k} \alpha_l = 1$, see Fig.1(b). Define $\beta_0 = 0$, $\beta_l = \sum_{i=1}^{l} \alpha_i$, $l = 1, \cdots, k$.

The algorithm with multi–time increments can be described as follows: (We consider it as *Algorithm II*)

Step 1: Compute the solution increments $(\delta\phi)_j^n$ and the solutions $\phi_j^{n+\beta_1}$ for all

$j \in \mathbb{Z}$:

$$(\delta\phi)_j^n := -\tau_n \widehat{H}(u_{j-\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n), \tag{2.17}$$

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

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

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

$$(\delta\phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta\phi)_j^n, & j \in \mathcal{D}_1, \\ -\tau_n \widehat{H}(u_{j-\frac{1}{2}}^{n+\beta_{l-1}}, u_{j+\frac{1}{2}}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2, \end{cases} \tag{2.19}$$

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

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

For the scheme (2.17)–(2.20), the conclusions of the **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 multi-time increments as follows:

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

2.1.1 Higher–order accurate spatial discretization

In this subsection, we introduce the reconstruction technique to improve the accuracy of the previous schemes in space. Following [21], we construct a piecewise linear function:

$$w_{j+\frac{1}{2}}^{n+\beta_l}(x) = u_{j+\frac{1}{2}}^{n+\beta_l} + S_{j+\frac{1}{2}}^{n+\beta_l}(x - x_{j+\frac{1}{2}}), \ (x) \in [x_j, x_{j+1}], \tag{2.22}$$

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 $S_{j+\frac{1}{2}}^{n+\beta_l}$ is a numerical slope approximating $(u_x)_{j+\frac{1}{2}}^{n+\beta_l}$, $l = 0, 1, \cdots, k-1$. 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 many mannars to define the approximate slope $S^{n+\beta_l}_{j+\frac{1}{2}}$. The commonly used formulae are

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \text{minmod}\left(S^{n+\beta_l,L}_{j+\frac{1}{2}}, S^{n+\beta_l,R}_{j+\frac{1}{2}}\right), \tag{2.23}$$

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \left(\text{sign}(S^{n+\beta_l,L}_{j+\frac{1}{2}}) + \text{sign}(S^{n+\beta_l,R}_{j+\frac{1}{2}})\right) \frac{|S^{n+\beta_l,L}_{j+\frac{1}{2}}| \cdot |S^{n+\beta_l,R}_{j+\frac{1}{2}}|}{|S^{n+\beta_l,L}_{j+\frac{1}{2}}| + |S^{n+\beta_l,R}_{j+\frac{1}{2}}| + \varepsilon}, \tag{2.24}$$

$$S^{n+\beta_l}_{j+\frac{1}{2}} = \frac{(|S^{n+\beta_l,R}_{j+\frac{1}{2}}|^2 + \varepsilon^R)S^{n+\beta_l,L}_{j+\frac{1}{2}} + (|S^{n+\beta_l,L}_{j+\frac{1}{2}}|^2 + \varepsilon^L)S^{n+\beta_l,R}_{j+\frac{1}{2}}}{|S^{n+\beta_l,L}_{j+\frac{1}{2}}|^2 + |S^{n+\beta_l,R}_{j+\frac{1}{2}}|^2 + \varepsilon^L + \varepsilon^R}, \tag{2.25}$$

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

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

$\Delta_+ u^{n+\beta_l}_{j+\frac{1}{2}} = u^{n+\beta_l}_{j+\frac{3}{2}} - u^{n+\beta_l}_{j+\frac{1}{2}}$, and ε^L and ε^R are taken as $Ch_{j-\frac{1}{2}}$ and $Ch_{j+\frac{1}{2}}$, respectively. In Section 3, we have used the van Leer limiter (2.24) and the van Albada limiter (2.25) with $C = 3$ in our numerical computations. The results show that there only exist a slight difference between them. However, the Minmod limiter is more diffusive. When the limiter satisfies any constraint condition, the solutions of the high resolution H-J schemes will have some good properties, for example, they are essentially non-oscillatory.

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 Fig.1(a). When the first order scheme is used, we only need 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.

2.1.2 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, using Heun's 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 $j \in \mathbb{Z}$:

$$
\begin{aligned}
(\delta\phi)_j^n :=& -\tau_n \widehat{H}(u_{j,L}^n, u_{j,R}^n), && (2.26)\\
\phi_j^{n+\beta_1,*} =& \phi_j^n + \alpha_1(\delta\phi)_j^n, && (2.27)\\
(\delta\phi)_j^{n+\beta_1,*} =& -\tau_n \widehat{H}(u_{j,L}^{n+\beta_1,*}, u_{j,R}^{n+\beta_1,*}), && (2.28)\\
\phi_j^{n+\beta_1} =& \frac{1}{2}\left(\phi_j^n + \phi_j^{n+\beta_1,*} + \alpha_1(\delta\phi)_j^{n+\beta_1,*}\right). && (2.29)
\end{aligned}
$$

Step 3: For $l = 2, \cdots, k$, project the solution increments $(\delta\phi)_j^{n+\beta_{l-1}}$ and update the solutions $\phi_j^{n+\beta_l}$ at local time level $t = t_n + \beta_{l-1}\tau_n$:

$$
(\delta\phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta\phi)_j^n, & j \in \mathcal{D}_1,\\ -\tau_n \widehat{H}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2, \end{cases} \tag{2.30}
$$

$$
\phi_j^{n+\beta_l,*} = \begin{cases} \phi_j^n + \beta_l(\delta\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_1,\\ \phi_j^{n+\beta_{l-1}} + \alpha_l(\delta\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_2, \end{cases} \tag{2.31}
$$

$$
(\delta\phi)_j^{n+\beta_l,*} = -\tau_n \widehat{H}(u_{j,L}^{n+\beta_l,*}, u_{j,R}^{n+\beta_l,*}), \tag{2.32}
$$

$$
\phi_j^{n+\beta_l} = \begin{cases} \frac{1}{2}\phi_j^n + \frac{1}{2}\left(\phi_j^{n+\beta_l,*} + \beta_l(\delta\phi)_j^{n+\beta_l,*}\right), & j \in \mathcal{D}_1,\\ \frac{1}{2}\phi_j^{n+\beta_{l-1}} + \frac{1}{2}\left(\phi_j^{n+\beta_l,*} + \alpha_l(\delta\phi)_j^{n+\beta_l,*}\right), & j \in \mathcal{D}_2. \end{cases} \tag{2.33}
$$

To verify the accuracy of *Algorithm III* in time, we restrict to the cases shown in Fig.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 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 gives

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

Here we have used the relations

$$
\begin{aligned}
\phi_j^{n+1,*} =&\phi_j^{n+\frac{1}{2}} + \frac{\tau_n}{2}L(\phi_{j-1}^{n+\frac{1}{2}}, \phi_j^{n+\frac{1}{2}}, \phi_{j+1}^{n+\frac{1}{2}})\\
=&\phi_j^n + \tau_n L(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n) + \mathcal{O}(\tau_n^2), \quad j \ge j_0,\\
\phi_j^{n+1,*} =&\phi_j^n + \tau_n L(\phi_{j-1}^n, \phi_j^n, \phi_{j+1}^n), \quad j \in \mathcal{D}_1\\
=&\phi_j^{n+\frac{1}{2}} + \frac{\tau_n}{2}L(\phi_{j-1}^{n+\frac{1}{2}}, \phi_j^{n+\frac{1}{2}}, \phi_{j+1}^{n+\frac{1}{2}}) + \mathcal{O}(\tau_n^2), \quad j \le j_0 - 1.
\end{aligned}
$$

Similarly, we have

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

The above results show that the scheme (2.26)-(2.33) 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 Taylor series expansion, we have

$$
\phi_j^{n+\beta_{l+1}} = \phi_j^{n+\beta_l} + \alpha_l\tau_n(\phi_t)_j^{n+\beta_l} + \frac{1}{2}(\alpha_l\tau_n)^2(\phi_{tt})_j^{n+\beta_l} + \mathcal{O}(\tau_n^3), \tag{2.34}
$$

where

$$(\phi_t)_j^{n+\beta_l} = -\left(H(\phi_x)\right)_j^{n+\beta_l},$$

and

$$(\phi_{tt})_j^{n+\beta_l} = \left(H' \frac{\partial}{\partial x} H(\phi_x)\right)_j^{n+\beta_l} = \left((H')^2 \phi_{xx}\right)_j^{n+\beta_l}.$$

They may be approximated by

$$(\phi_t)_j^{n+\beta_l} \approx -\widehat{H}(u_{j,L}^{n+\beta_l}, u_{j,R}^{n+\beta_l}), \tag{2.35}$$

$$(\phi_{tt})_j^{n+\beta_l} \approx \left(H'\Big(\frac{1}{2}(u_{j,L}^{n+\beta_l} + u_{j,R}^{n+\beta_l})\Big)\right)^2 \frac{u_{j+\frac{1}{2}}^{n+\beta_l} - u_{j-\frac{1}{2}}^{n+\beta_l}}{\frac{1}{2}(h_{j+\frac{1}{2}} + h_{j-\frac{1}{2}})}, \tag{2.36}$$

respectively.

The corresponding *Algorithm IV* is:

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

$$\begin{cases} (\delta_t \phi)_j^n := -\tau_n \widehat{H}(u_{j,L}^n, u_{j,R}^n), \\ (\delta_{tt}\phi)_j^n := \frac{(\tau_n)^2}{2} \widehat{G}(u_{j,L}^n, u_{j,R}^n), \end{cases} \tag{2.37}$$

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

$$\phi_j^{n+\beta_1} = \phi_j^n + \alpha_1 (\delta_t \phi)_j^n + (\alpha_1)^2 (\delta_{tt}\phi)_j^n, \qquad j \in \mathbb{Z}. \tag{2.38}$$

Step 3: For $l = 2, \cdots, k$, 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 $t = t_n + \beta_{l-1}\tau_n$:

$$(\delta_t \phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta_t \phi)_j^n, & j \in \mathcal{D}_1, \\ -\tau_n \widehat{H}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2, \end{cases} \tag{2.39}$$

and

$$(\delta_{tt}\phi)_j^{n+\beta_{l-1}} = \begin{cases} (\delta_{tt}\phi)_j^n, & j \in \mathcal{D}_1, \\ \frac{(\tau_n)^2}{2} \widehat{G}(u_{j,L}^{n+\beta_{l-1}}, u_{j,R}^{n+\beta_{l-1}}), & j \in \mathcal{D}_2, \end{cases} \tag{2.40}$$

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

$$\phi_j^{n+\beta_l} = \begin{cases} \phi_j^n + \beta_l (\delta_t \phi)_j^{n+\beta_{l-1}} + (\beta_l)^2 (\delta_{tt}\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_1, \\ \phi_j^{n+\beta_{l-1}} + \alpha_l (\delta_t \phi)_j^{n+\beta_{l-1}} + (\alpha_l)^2 (\delta_{tt}\phi)_j^{n+\beta_{l-1}}, & j \in \mathcal{D}_2. \end{cases} \tag{2.41}$$

In the above Lax-Wendroff type algorithm, $\widehat{G}(u_{j,L}^{n+\beta_l}, u_{j,R}^{n+\beta_l})$ denotes the right hand side of (2.36), and two "solution increments" have been projected at each local time level.

Table 1: Example 3.1: The errors and the rate of convergence for the case $(a, c) = (0, 1)$ obtained by *Algorithm III* with van Leer limiter.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
25	3.07e-2	–	6.39e-2	–	3.40e-2	–
50	7.93e-3	1.95	1.41e-2	2.18	7.48e-3	2.18
100	1.97e-3	2.01	3.31e-3	2.09	1.68e-3	2.15
200	4.71e-4	2.06	8.15e-4	2.02	3.93e-4	2.10

Table 2: Example 3.1: Same as Table 1, except for van Albada limiter.

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

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. In our computations, 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

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

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 (2.36). 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 the 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

$$p^i = \log(e_N^i/e_{2N}^i)/\log(2), \quad i = \infty, \text{or } 1, \text{ or } 2.$$

Table 3: Example 3.1: Same as Table 1, except for $(a, c) = (1, 1)$.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
25	2.86e-3	–	8.93e-3	–	4.35e-3	–
50	7.76e-4	1.88	2.39e-3	1.90	1.17e-4	1.89
100	2.08e-4	1.90	6.21e-4	1.94	3.03e-4	1.95
200	5.38e-5	1.95	1.59e-4	1.97	7.75e-5	1.97

Table 4: Example 3.1: Same as Table 2, except for $(a, c) = (1, 1)$.

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 5: 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	3.72e-2	–	6.47e-2	–	3.74e-2	–
50	9.45e-3	1.98	1.54e-2	2.07	8.35e-3	2.16
100	2.28e-3	2.05	3.75e-3	2.04	1.90e-3	2.14
200	5.46e-4	2.06	9.28e-4	2.01	4.49e-4	2.08

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 1-6 show the errors and convergence order obtained by *Algorithm III* and *Algorithm IV* with two limiters, van Leer limiter (2.24) and van Albada limiter (2.25), respectively.

The numerical results show that a second-order rate of convergence is obtained for the initial boundary problem (3.2). When three local time steps $0.5\tau, 0.1\tau$, and

Table 6: Example 3.1: Same as Table 2, 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

0.4τ are used within $[\pi, 1.5\pi]$ insteady of the previous two time increments, we have obtained fully the same data.

Example 3.2. This example is to solve the H-J equations (2.1) with a convex 2π-periodic initial data

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

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

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

Tables 7-10 show the errors and convergence order obtained by *Algorithm III* and *Algorithm IV* with two limiters given in (2.24) and (2.25), respectively. The data show that a second-order rate of convergence has been obtained by using our local time step discretization methods to solve nunlinear H-J equations.

Figs. 2-5 show the solution $\phi_N(x_j,t)$ and $\phi_x(x_{j+\frac{1}{2}},t) \approx u_N(x_{j+\frac{1}{2}},t)$ at $t = 1.5/\pi^2$ obtained by *Algorithm III* and *Algorithm IV* with van Leer limiter (2.24) and van Albada limiter (2.25), respectively, when the discontinuity in the ϕ_x is well developed. Here the number of grid cells is 60, and the exact solution is 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 demonstrated in these figures. The solutions obtained by using two different limiters are consistent.

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

Table 7: Example 3.2: The errors and convergence order obtained by *Algorithm III* at $t = 0.5\pi^2$ with van Leer limiter.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
30	1.06e-2	–	1.45e-2	–	1.14e-2	–
60	2.28e-3	2.22	2.99e-3	2.28	2.37e-3	2.27
120	5.20e-4	2.13	6.66e-4	2.17	5.33e-4	2.15
240	1.25e-4	2.06	1.54e-4	2.11	1.25e-4	2.09

Table 8: Example 3.2: Same as Table 7, except for van Albada limiter.

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 9: Example 3.2: Same as Table 7, except for *Algorithm IV*.

N	e_N^∞	p^∞	e_N^1	p^1	e_N^2	p^2
30	8.48e-3	–	1.19e-2	–	9.35e-3	–
60	2.01e-3	2.08	2.68e-3	2.15	2.14e-3	2.13
120	4.71e-4	2.09	6.10e-4	2.14	4.91e-4	2.12
240	1.14e-4	2.05	1.43e-4	2.09	1.16e-4	2.08

Table 10: Example 3.2: Same as Table 8, 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

3.2 Two-dimensional problems

In the following computations, we restrict to a regular but non-uniform mesh $\{(x_j, y_k)\}$, and the numerical Hamiltonian

$$\begin{aligned}\widehat{H}(u_{j-\frac{1}{2},k},u_{j+\frac{1}{2},k},v_{j,k-\frac{1}{2}},v_{j,k+\frac{1}{2}}) = H\left(\frac{u_{j+\frac{1}{2},k}+u_{j-\frac{1}{2},k}}{2},\frac{v_{j,k+\frac{1}{2}}+v_{j,k-\frac{1}{2}}}{2}\right)\\ -\frac{\max\limits_{u\in I^u_{j,k}}\{|H_u(u,v_{j,k})|\}}{2}(u_{j+\frac{1}{2},k}-u_{j-\frac{1}{2},k})\\ -\frac{\max\limits_{v\in I^v_{j,k}}\{|H_v(u_{j,k},v)|\}}{2}(v_{j,k+\frac{1}{2}}-v_{j,k-\frac{1}{2}}),\end{aligned} \tag{3.3}$$

is used to approximate the Hamiltonian $H(\phi_x, \phi_y)$, where

$$\begin{aligned}u_{j\pm\frac{1}{2},k} &= (\phi_{j\pm1,k}-\phi_{j,k})/(x_{j\pm1}-x_j),\\ v_{j,k\pm\frac{1}{2}} &= (\phi_{j,k\pm1}-\phi_{j,k})/(y_{k\pm1}-y_j),\end{aligned}$$

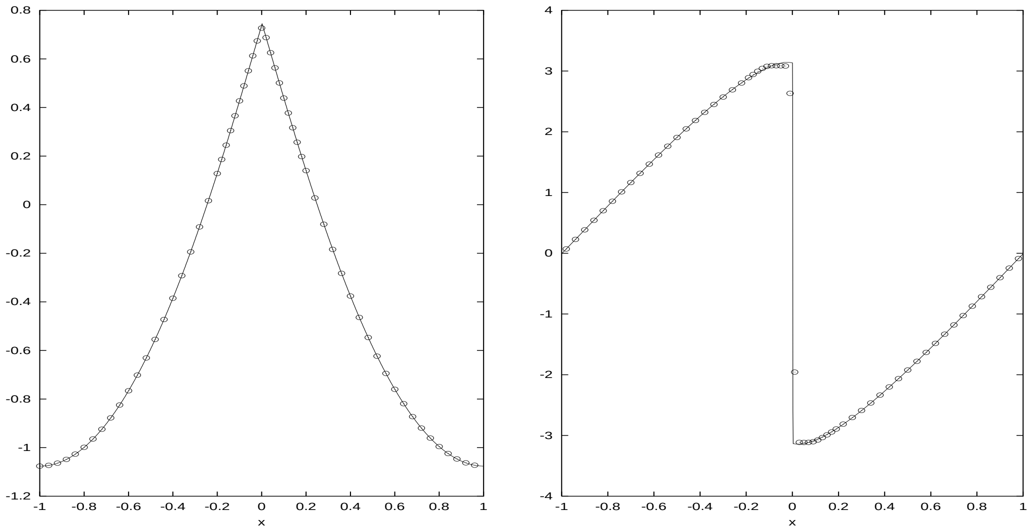


Figure 2: Comparison of the computed solutions ("$\circ$") with the exact solutions (solid line) of Example 3.1 given at $t = 1.5\pi^2$. Left: $\phi(x,t)$; Right: $\phi_x(x,t)$.

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

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

Example 3.3. This example is to compute two-dimensional Navier–Stokes equations [15]:

$$
\begin{cases}
\omega_t + \mathbf{u}\cdot\nabla\omega = \frac{1}{Re}\Delta\omega,\\
\Delta\psi = \omega, \quad \mathbf{u} = \nabla^{\perp}\psi,
\end{cases}
\tag{3.4}
$$

and check the accuracy of *Algorithm III* with van Albada limiter, where $(x,y) \in [0,2\pi[\times[0,2\pi[$ and $\nabla^{\perp} = (-\partial_y, \partial_x)$. 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 Possion 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

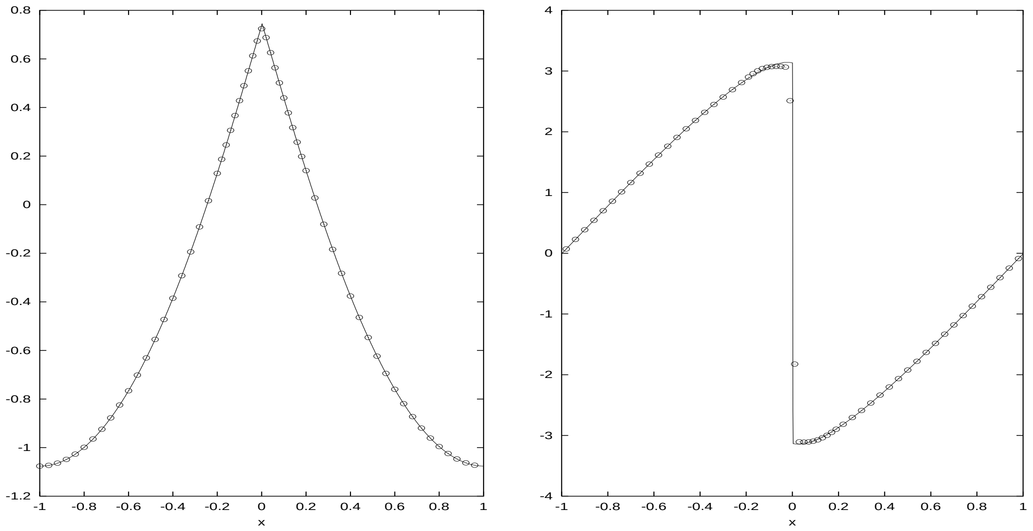


Figure 3: Same as Fig. 2, except for van Albada limiter.

known as

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

The computational domain is discretized such that the half step sizes $h_x/2$ and $h_y/2$ are within $[0.8, 1.2]$, respectively, and $\tau/2$ is in $[0.8, 1.2] \times [0.8, 1.2]$, while the global step sizes h_x, h_y and τ are in other domain. Table 11 shows the errors and convergence orders for the vorticity function at $t = 2$.

4 Concluding Remarks

A class of high resolution local time step schemes have been presented for nonlinear Hamilton-Jacobi 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 by 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 prop-

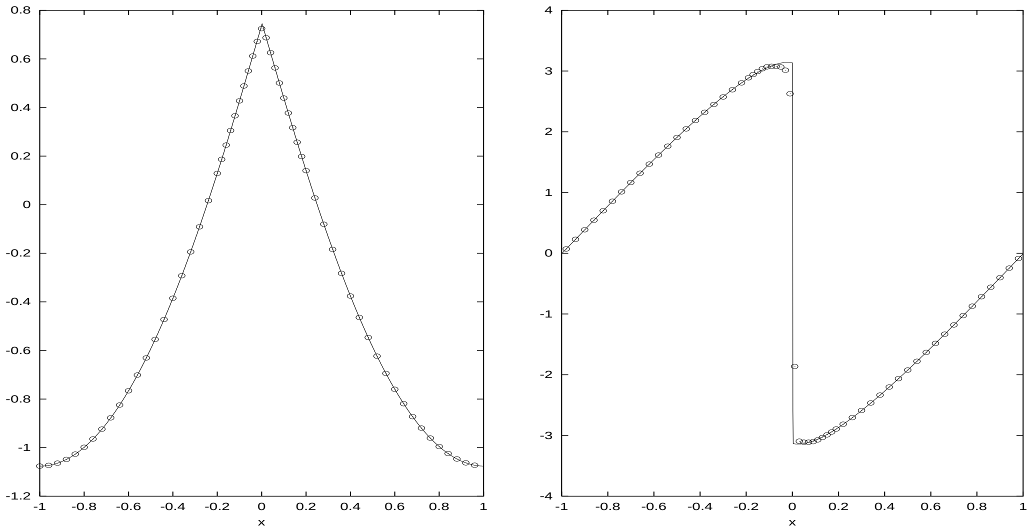


Figure 4: Same as Fig. 2, except for *Algorithm IV*.

Table 11: Example 3.3: 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

erties of the global time step schemes, including stability and convergence, and can be applied to solve numerically the initial boundary value 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 constructions 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

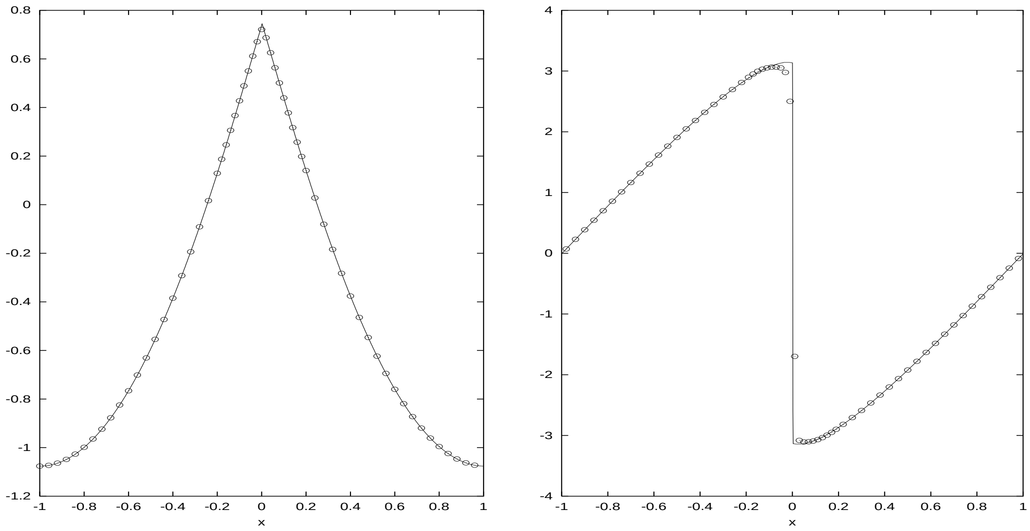


Figure 5: Same as Fig. 3, except for *Algorithm IV.*

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. It is another interesting topic to construct third– and higher–order accurate schemes with locally varying time and space grids.

Acknowledgment

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).

References

[1] R. Abgrall, Numerical discretization of first order Hamilton-Jacobi equations on triangular meshes, *Commm. 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. Scientific Computing* 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.* 27(1992), pp. 1–67.

[7] M.G. Crandall and P.L: Lions, Two approximations of solutions of Hamilton-Jacobi equations, *Math. Comput.* 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.P. 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 non-oscillatory central scheme 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(2000), 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. Comput.* 40(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, to appear in *J. Comput. Phys.*, 2003. Also available at `http://www.math.hkbu.edu.hk/~ttang`

[21] 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.

[22] 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.