

A Kernel-Based Explicit Unconditionally Stable Scheme for Hamilton-Jacobi Equations on Nonuniform Meshes

Andrew Christlieb^{1,*}, William Sands^{1,†} and Hyoseon Yang^{1,‡}

Abstract

In [11], the authors developed a class of high-order numerical schemes for the Hamilton-Jacobi (H-J) equations, which are unconditionally stable, yet take the form of an explicit scheme. This paper extends such schemes, so that they are more effective at capturing sharp gradients, especially on nonuniform meshes. In particular, we modify the weighted essentially non-oscillatory (WENO) methodology in the previously developed schemes by incorporating an exponential basis and adapting the previously developed nonlinear filters used to control oscillations. The main advantages of the proposed schemes are their effectiveness and simplicity, since they can be easily implemented on higher-dimensional nonuniform meshes. We perform numerical experiments on a collection of examples, including H-J equations with linear, nonlinear, convex and non-convex Hamiltonians. To demonstrate the flexibility of the proposed schemes, we also include test problems defined on non-trivial geometry.

Key Words: Hamilton-Jacobi equation; Kernel based scheme; Unconditionally stable; High order accuracy; Weighted essentially non-oscillatory methodology; Exponential basis; Nonuniform meshes.

1 Introduction

In this paper, we propose a class of high-order, weighted essentially non-oscillatory numerical schemes for approximating the viscosity solution to the Hamilton-Jacobi (H-J) equation

$$\begin{cases} \phi_t + H(\nabla \phi) = 0, & x \in \mathbb{R}^d \\ \phi(x, 0) = \phi_0(x), \end{cases} \quad (1.1)$$

where $\phi = \phi(x, t)$ is a scalar function and H is a Lipschitz continuous Hamiltonian. The H-J equations play a significant role among many fields, including optimal control, geometric optics, differential games, computer vision and image processing, as well as variational calculus. It is well known that as time evolves, the H-J equations develop continuous solutions, of which, associated derivatives might be discontinuous, even for smooth initial conditions. If the solution is redefined in a weak sense, regularity conditions on the function ϕ can be relaxed; however, such solutions may not be unique. To identify the unique, physically relevant solution, the concept of vanishing viscosity was introduced [13, 14]. In subsequent papers, [15, 32], authors addressed the convergence of general approximation schemes to the viscosity solution of (1.1).

There have been many numerical schemes developed to solve the H-J equations. Methods among the existing literature include essentially non-oscillatory (ENO) schemes [26, 27], weighted ENO (WENO) schemes [21, 36], Hermite WENO schemes [28, 29, 37, 38], as well as discontinuous Galerkin methods [7, 8, 20, 24, 35]. These schemes are typically categorized within the Method of Lines (MOL) framework, in which the spatial variable is discretized first, then the resulting initial value problems (IVPs) are solved by coupling with a suitable time integrator. This work takes an alternative approach: First, discretization is completed on the temporal variable, then, the resulting boundary value problems (BVPs) are solved at discrete time levels. To solve the BVPs, the continuous operator (in space) is inverted analytically, using an integral solution.

¹Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI, 48824, United States. *christli@msu.edu, †sandswi3@msu.edu, and ‡hyoseon@msu.edu; corresponding author.

We refer to this approach as the Method of Lines Transpose (MOL^T), which is also known as Rothes's method [3, 30, 31]. These methods are formally matrix-free, in the sense that there is no need to solve linear systems at each time step. Moreover, this integral solution extends the so-called domain-of-dependence, so that the method does not suffer from a CFL restriction. The kernel used in this formulation also exhibits pleasant numerical properties with several developments. To approximate the integral equations in the BVP, one could use a fast multipole method (FMM) [16, 23], or a Fourier-continuation alternating-direction (FC-AD) algorithm, which yields unconditionally stability at the cost of $\mathcal{O}(N \log N)$ [1, 25]. Instead, Causley *et al.* [5], with the aid of a three-term recurrence relation, managed to reduce the computational complexity of the method from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. A variety of schemes, based on the MOL^T formulation, have been developed for solving a range of time-dependent PDEs, including the wave equation [3], the heat equation (e.g., the Allen-Cahn equation [4] and Cahn-Hilliard equation [2]), Maxwell's equations [6], and the Vlasov equation [9].

Recent work on the MOL^T has involved extending the method to solve more general nonlinear PDEs, for which an integral solution is generally not applicable. This work includes the nonlinear degenerate convection-diffusion equations [10], as well as the H-J equations [11]. The key idea of these papers involved exploiting the linearity of a given *differential operator*, rather than requiring linearity in the underlying equations. This allowed derivative operators in the problems to be expressed through kernel representations developed for linear problems. Formulating applicable derivative operators in this way ultimately facilitated the stability of the schemes, since a global coupling was introduced through the integral operator. As part of this embedding process, a kernel parameter β was introduced, and through a careful selection, was shown to yield schemes which are A-stable. Remarkably, it was shown that one could couple these representations for the derivative operators with an explicit time-stepping method, such as the strong-stability-preserving Runge-Kutta (SSP-RK) methods [17] and still obtain schemes which maintain unconditional stability [10, 11]. To address shock-capturing and control non-physical oscillations, the latter two papers introduced quadrature formulas based on WENO reconstruction, along with a nonlinear filter.

This paper seeks to extend the work in [10, 11] to the H-J equations (1.1) defined on non-uniformly distributed spatial domains. In particular, several improvements are given. First, we develop the MOL^T for mapped grids using a general coordinate transformation function, which allows for a non-uniform distribution of grid points. We show that, with this mapping, our numerical scheme is able to preserve the conservation property for the derivative of the solution to the H-J equation. We also describe a novel WENO-based quadrature for the spatial discretization, which uses a basis consisting of exponential polynomials, to improve the shock capturing capabilities of the method. Another difference in this paper, compared to our previous work on H-J equations, is that we propose a different nonlinear filter, which, we believe, is more effective at minimizing oscillations in the derivative of the solution to the PDE (1.1).

The paper is organized as follows. We first review the kernel-based representations for first and second order derivative operators, and address boundary conditions for both periodic and non-periodic problems, in Section 2. In Section 3, we present our numerical scheme for H-J equations on nonuniform grids with an algorithm flowchart. A collection of numerical examples is presented to demonstrate the performance of the proposed method in Section 4. In Section 5, we conclude the paper with some remarks and directions of future work.

2 Review for the approximation of differential operators

We start with a brief review on construction of derivative operators using the methodologies proposed in [10, 11]. The second order derivative, e.g., ∂_{xx} , shall be described first, as it will be used in the representation of first derivatives. Note that representations are formed using 1D examples, but the line-by-line approach allows us to reuse these expressions, with an appropriate swapping of the direction.

2.1 Second order derivative ∂_{xx}

In this section we will develop an approximation to ∂_{xx} based on a fast kernel method. The starting point is a Helmholtz operator of the “right sign”, meaning that the inverses is represented by a “compact” kernel. Here “compact” refers to a kernel that is represented as a function instead of an infinite sum. This representation is used to build an approximation to ∂_{xx} .

Motivated by work done for parabolic equations (see e.g., [2, 4]), we define the differential operator

$$\mathcal{L}_0 := \mathcal{I} - \frac{1}{\alpha^2} \partial_{xx}, \quad x \in [a, b], \quad (2.1)$$

where $\mathcal{I}$ is the identity operator, and α is a positive constant, which shall be specified later. We now suppose that there are two functions $w(x)$ and $v(x)$, which satisfy the equation

$$\left(\mathcal{I} - \frac{1}{\alpha^2} \partial_{xx} \right) w(x) = v(x). \quad (2.2)$$

Noting that this is a linear equation of the form

$$\mathcal{L}_0[w; \alpha](x) = v(x),$$

it follows that the solution can be obtained through an analytic inversion of the operator $\mathcal{L}_0$:

$$w(x) = \mathcal{L}_0^{-1}[v; \alpha](x).$$

Written more explicitly, the expression for $w(x)$ can be determined to be

$$w(x) = I_0[v; \alpha](x) + A_0 e^{-\alpha(x-a)} + B_0 e^{-\alpha(b-x)}, \quad (2.3)$$

where

$$I_0[v; \alpha](x) := \frac{\alpha}{2} \int_a^b e^{-\alpha|x-s|} v(s) ds, \quad (2.4)$$

is a convolution integral and the constants A_0 and B_0 are determined by boundary conditions. If the PDE is linear, e.g., the heat equation, then (2.3) is a valid expression for the update, and A_0 and B_0 can be determined using the boundary conditions specified by the problem. Otherwise, they will need to be carefully prescribed to maintain consistency. We will address this issue in Sections 2.3 and 2.4.

To develop a suitable expression for the second derivative, we introduce the related operator $\mathcal{D}_0$, which is defined as

$$\mathcal{D}_0 = \mathcal{I} - \mathcal{L}_0^{-1}. \quad (2.5)$$

Through some algebraic manipulations, one can write an alternative definition for $\mathcal{L}_0$ in terms of $\mathcal{D}_0$, i.e.,

$$\mathcal{L}_0 = (\mathcal{I} - \mathcal{D}_0)^{-1}.$$

If the operator norm for $\mathcal{D}_0$ is bounded by unity, then, using the definition (2.1), we can express the second derivative operator as a Neumann series:

$$\frac{1}{\alpha^2} \partial_{xx} = \mathcal{I} - \mathcal{L}_0 = \mathcal{L}_0 (\mathcal{L}_0 - \mathcal{I}) = -\mathcal{D}_0 (\mathcal{I} - \mathcal{D}_0)^{-1} = -\sum_{p=1}^{\infty} \mathcal{D}_0^p.$$

Here, each term in the expansion is defined successively from the previous term, i.e., $\mathcal{D}_0^p = \mathcal{D}_0[\mathcal{D}_0^{p-1}]$. Therefore, the action of ∂_{xx} on a generic function $v(x)$ is given by

$$\partial_{xx} v(x) = -\alpha^2 \sum_{p=1}^{\infty} \mathcal{D}_0^p[v; \alpha](x). \quad (2.6)$$

As previous noted, expressions in multiple spatial dimensions can be obtained by simply changing labels, e.g., x to y .

2.2 First order derivative ∂_x

As with the last section, we will use the same basic idea to construct an approximation to ∂_x that will allow us to build an integral representation that provides an up wind and down wind approximation to our operator.

In order to obtain a representation for the first derivative, we introduce two operators: $\mathcal{L}_L$ and $\mathcal{L}_R$ to account for waves traveling in different directions. The subscript on an operator is used to identify the direction associated with wave propagation, so that “ L ” and “ R ” correspond to downwinding and upwinding, respectively. The operands for this decomposition would, of course, come from a monotone splitting, depending on the problem. With this convention, we define

$$\mathcal{L}_L = \mathcal{I} - \frac{1}{\alpha}\partial_x, \quad \mathcal{L}_R = \mathcal{I} + \frac{1}{\alpha}\partial_x, \quad x \in [a, b], \quad (2.7)$$

where $\mathcal{I}$ is the identity operator and, again, α is a strictly positive constant. Using an integrating factor, we can invert these operators, similar to the case for $\mathcal{L}_0$ to find that

$$\mathcal{L}_L^{-1}[v, \alpha](x) = I_L[v, \alpha](x) + B_L e^{-\alpha(b-x)}, \quad (2.8a)$$

$$\mathcal{L}_R^{-1}[v, \alpha](x) = I_R[v, \alpha](x) + A_R e^{-\alpha(x-a)}, \quad (2.8b)$$

where

$$I_L[v, \alpha](x) = \alpha \int_x^b e^{-\alpha(s-x)} v(s) ds, \quad (2.9a)$$

$$I_R[v, \alpha](x) = \alpha \int_a^x e^{-\alpha(x-s)} v(s) ds, \quad (2.9b)$$

with constant A_R and B_L being determined by the boundary condition imposed for the operators. These expressions depend on the problem, so handling the general case requires a substantial amount of care.

As with the second derivative operator, we introduce the operators

$$\mathcal{D}_L = \mathcal{I} - \mathcal{L}_L^{-1}, \quad \mathcal{D}_R = \mathcal{I} - \mathcal{L}_R^{-1}, \quad x \in [a, b]. \quad (2.10)$$

and expand each of these into a Neumann series:

$$\frac{1}{\alpha}\partial_x^+ = \mathcal{I} - \mathcal{L}_L = \mathcal{L}_L(\mathcal{L}_L^{-1} - \mathcal{I}) = -\mathcal{D}_L/(\mathcal{I} - \mathcal{D}_L) = -\sum_{p=1}^{\infty} \mathcal{D}_L^p, \quad (2.11a)$$

$$\frac{1}{\alpha}\partial_x^- = \mathcal{L}_R - \mathcal{I} = \mathcal{L}_R(\mathcal{I} - \mathcal{L}_R^{-1}) = \mathcal{D}_R/(\mathcal{I} - \mathcal{D}_R) = \sum_{p=1}^{\infty} \mathcal{D}_R^p. \quad (2.11b)$$

As before, these operators are defined successively, but we leave the operand at each p as a generic function $v(x)$. Moreover, the $\pm$ signs on the expressions for the derivatives in (2.11a) and (2.11b) do not reflect the direction of propagation. Instead, they represent the direction of approach at an interface. For example, we use ∂_x^+ to indicate the right-sided approximation of the derivative, in x , along an interface.

2.3 Periodic boundary conditions

In this section we show how to impose periodic boundary conditions for the line by line MOL^T formulation we leverage in this work.

For problems with periodic boundary conditions, we make the requirement that

$$\mathcal{D}_L^p[v; \alpha](a) = \mathcal{D}_L^p[v; \alpha](b), \quad \mathcal{D}_R^p[v; \alpha](a) = \mathcal{D}_R^p[v; \alpha](b), \quad \mathcal{D}_0^p[v; \alpha](a) = \mathcal{D}_0^p[v; \alpha](b), \quad p \geq 1. \quad (2.12)$$

Using the definition of these operators (2.10) and (2.5), the above condition shows that at each level, we should select

$$A_R = \frac{I_R[v; \alpha](b)}{1 - \mu}, \quad B_L = \frac{I_L[v; \alpha](a)}{1 - \mu}, \quad A_0 = \frac{I_0[v; \alpha](b)}{1 - \mu}, \quad B_0 = \frac{I_0[v; \alpha](a)}{1 - \mu}, \quad (2.13)$$

where $\mu = e^{-\alpha(b-a)}$. Hence, following the idea in [10], when ϕ is a periodic function, we can approximate the first derivative $\phi_x^\pm$ with (modified) partial sums in (2.11),

$$\phi_x^+(x) \approx \mathcal{P}_k^L[\phi; \alpha](x) = \begin{cases} -\alpha \sum_{p=1}^k \mathcal{D}_L^p[\phi; \alpha](x), & k = 1, 2, \\ -\alpha \sum_{p=1}^k \mathcal{D}_L^p[\phi; \alpha](x) + \alpha \mathcal{D}_0 * \mathcal{D}_L^2[\phi; \alpha](x), & k = 3. \end{cases} \quad (2.14a)$$

$$\phi_x^-(x) \approx \mathcal{P}_k^R[\phi; \alpha](x) = \begin{cases} \alpha \sum_{p=1}^k \mathcal{D}_R^p[\phi; \alpha](x), & k = 1, 2, \\ \alpha \sum_{p=1}^k \mathcal{D}_R^p[\phi; \alpha](x) - \alpha \mathcal{D}_0 * \mathcal{D}_R^2[\phi; \alpha](x), & k = 3, \end{cases} \quad (2.14b)$$

Note that there is an extra term for $k = 3$. As remarked in [10], such a term is needed for attaining unconditional stability of the scheme. An error estimate for the approximation (2.14) regarding the truncation of the infinite sum, carried out in [10], showed that keeping k terms of the partial sums led to

$$\begin{aligned} \|\partial_x \phi(x) - \mathcal{P}_k^L[\phi; \alpha](x)\|_\infty &\leq C \left(\frac{1}{\alpha}\right)^k \|\partial_x^{k+1} \phi(x)\|_\infty, \\ \|\partial_x \phi(x) - \mathcal{P}_k^R[\phi; \alpha](x)\|_\infty &\leq C \left(\frac{1}{\alpha}\right)^k \|\partial_x^{k+1} \phi(x)\|_\infty. \end{aligned}$$

for the representation of the first derivative and

$$\|\partial_{xx} \phi(x) + \alpha^2 \sum_{p=1}^k \mathcal{D}_0^p[\phi; \alpha](x)\|_\infty \leq C \left(\frac{1}{\alpha}\right)^{2k} \|\partial_x^{2k+1} \phi(x)\|_\infty$$

for the second derivative

In numerical simulations, we will take $\alpha = \beta/(c\Delta t)$ in (2.14), with c being the maximum wave propagation speed. Here, Δt denotes the time step and β is a constant independent of Δt . Hence, the partial sums approximate ϕ_x with accuracy $\mathcal{O}(\Delta t^k)$.

2.4 Non-periodic boundary conditions

In this subsection, we will focus on the application of non-periodic boundary conditions to (1.1). Specific details for the error analysis, as well as more generic boundary conditions, can be found in our previous work on the H-J equation [11].

For non-periodic problems, additional requirements imposed on the operators $\mathcal{D}_*$ need to be consistent with the boundary condition specified on ϕ . Otherwise, this can lead to order reduction in the method. Using integration by parts, one can identify the source of the order reduction, which involves evaluations of ϕ and its derivatives, along the boundaries. To address this issue, the partial sums, as presented above, were modified to annihilate terms which resulted in the order reduction. Before introducing the modified partial sums, we specify certain requirements on the coefficients A_* and B_* used in the construction of a given operator $\mathcal{D}_*$.

2.4.1 Conditions for A_* and B_*

For reconstructions of first derivatives, suppose that C_a and C_b are given numbers. We will explain, later, how these values are obtained. If we require

$$\mathcal{D}_R[v, \alpha](a) = C_a, \quad \mathcal{D}_L[v, \alpha](b) = C_b,$$

then one can use the definition (2.10) to show that we should select

$$A_R = v(a) - C_a, \quad B_L = v(b) - C_b. \quad (2.16)$$

As an example, suppose we wish to use a first order approximation of the first partial derivative, i.e.,

$$\phi_x^+ \approx -\alpha \mathcal{D}_L[\phi; \alpha](x), \quad \phi_x^- \approx \alpha \mathcal{D}_R[\phi; \alpha](x). \quad (2.17)$$

According to an analysis of the truncation error, we should select

$$C_a \approx \frac{1}{\alpha} \phi_x(a), \quad C_b \approx -\frac{1}{\alpha} \phi_x(b),$$

to obtain a convergent approximation. The derivatives can be constructed using finite differences of a suitable order.

The case for the second derivative is a bit more cumbersome, however, it works in essentially the same way. Again, if require

$$\mathcal{D}_0[v, \alpha](a) = C_a, \quad \mathcal{D}_0[v, \alpha](b) = C_b,$$

where C_a and C_b are chosen to obtain appropriate approximation order, i.e.,

$$C_a \approx -\frac{1}{\alpha^2} \phi_{xx}(a), \quad C_b \approx -\frac{1}{\alpha^2} \phi_{xx}(b),$$

then the coefficients A_0 and B_0 are given as

$$A_0 = \frac{1}{1 - \mu^2} (\mu (I_0[v, \alpha](b) - v(b) + C_b) - (I_0[v, \alpha](a) - v(a) + C_a)), \quad (2.18a)$$

$$B_0 = \frac{1}{1 - \mu^2} (\mu (I_0[v, \alpha](a) - v(a) + C_a) - (I_0[v, \alpha](b) - v(b) + C_b)). \quad (2.18b)$$

The process of determining C_* becomes difficult to generalize if more terms in the partial sums are required. Instead, another modification was proposed in [11]: Rather than specify the conditions for A_* and B_* , the partial sums were modified so that boundary-related terms, which led to order reduction, were automatically removed.

2.4.2 The modified partial sums

In developing the modified sums for the first derivative, we assume that the derivatives of ϕ have been constructed, in some way, at the boundaries, i.e., $\partial_x^m \phi(a)$ and $\partial_x^m \phi(b)$, $m \geq 1$. Using this information, the schemes presented in [11], which address non-periodic boundary conditions, were, as follows:

$$\phi_x^-(x) \approx \widetilde{\mathcal{P}}_k^R[\phi; \alpha](x) = \begin{cases} \alpha \sum_{p=1}^k \mathcal{D}_R[\phi_{1,p}; \alpha](x), & k = 1, 2, \\ \alpha \sum_{p=1}^k \mathcal{D}_R[\phi_{1,p}; \alpha](x) - \alpha \mathcal{D}_0[\phi_{1,3}; \alpha](x), & k = 3, \end{cases} \quad (2.19a)$$

$$\phi_x^+(x) \approx \widetilde{\mathcal{P}}_k^L[\phi; \alpha](x) = \begin{cases} -\alpha \sum_{p=1}^k \mathcal{D}_L[\phi_{2,p}; \alpha](x), & k = 1, 2, \\ -\alpha \sum_{p=1}^k \mathcal{D}_L[\phi_{2,p}; \alpha](x) + \alpha \mathcal{D}_0[\phi_{2,3}; \alpha](x), & k = 3. \end{cases} \quad (2.19b)$$

And $\phi_{1,p}$ and $\phi_{2,p}$ are given as

$$\begin{cases} \phi_{1,1} = \phi, \\ \phi_{1,2} = \mathcal{D}_L[\phi_{1,1}; \alpha] - \sum_{m=2}^k \left(-\frac{1}{\alpha}\right)^m \partial_x^m \phi(a) e^{-\alpha(x-a)}, \\ \phi_{1,3} = \mathcal{D}_L[\phi_{1,2}; \alpha] + \sum_{m=2}^k (m-1) \left(-\frac{1}{\alpha}\right)^m \partial_x^m \phi(a) e^{-\alpha(x-a)}, \end{cases} \quad (2.20a)$$

$$\begin{cases} \phi_{2,1} = \phi, \\ \phi_{2,2} = \mathcal{D}_R[\phi_{2,1}; \alpha] - \sum_{m=2}^k \left(\frac{1}{\alpha}\right)^m \partial_x^m \phi(b) e^{-\alpha(b-x)}, \\ \phi_{2,3} = \mathcal{D}_R[\phi_{2,2}; \alpha] + \sum_{m=2}^k (m-1) \left(\frac{1}{\alpha}\right)^m \partial_x^m \phi(b) e^{-\alpha(b-x)}, \end{cases} \quad (2.20b)$$

with the boundary conditions for the operators

$$\begin{aligned} \alpha \mathcal{D}_R[\phi_{1,1}; \alpha](a) &= \phi_x(a), \quad \alpha \mathcal{D}_L[\phi_{2,1}; \alpha](b) = -\phi_x(b), \\ \alpha \mathcal{D}_L[\phi_{1,p}; \alpha](a) &= \alpha \mathcal{D}_R[\phi_{2,p}; \alpha](b) = 0, \quad \text{for } p \geq 2, \\ \alpha \mathcal{D}_0[\phi_{*,3}; \alpha](a) &= \alpha \mathcal{D}_0[\phi_{*,3}; \alpha](b) = 0, \quad * \text{ could be 1 or 2.} \end{aligned}$$

The modified partial sum (2.19) is constructed so that it agrees with the derivative values at the boundary, to preserve consistency with the boundary condition imposed on ϕ . Furthermore, the authors provided the following theorem, which verifies the accuracy of these modified sums:

Theorem 2.1. *Suppose $\phi \in \mathcal{C}^{k+1}[a, b]$, $k = 1, 2, 3$. Then, the modified partial sums (2.19) satisfy*

$$\|\partial_x \phi(x) - \widetilde{\mathcal{P}}_k^L[\phi; \alpha](x)\|_\infty \leq C \left(\frac{1}{\alpha}\right)^k \|\partial_x^{k+1} \phi(x)\|_\infty, \quad (2.21a)$$

$$\|\partial_x \phi(x) - \widetilde{\mathcal{P}}_k^R[\phi; \alpha](x)\|_\infty \leq C \left(\frac{1}{\alpha}\right)^k \|\partial_x^{k+1} \phi(x)\|_\infty. \quad (2.21b)$$

Recalling that we defined $\alpha = \beta/(c\Delta t)$ shows that the modified partial sums (2.19) approximate ϕ_x with accuracy $\mathcal{O}(\Delta t^k)$.

3 Extensions of the scheme to nonuniform grids

In this section, we describe the extension of the method to mapped grids. Section 3.1 reviews the fact that the H-J equation, under a coordinate transformation, yields yet another H-J equation, which allows us to develop a systematic approaches for both mapped and non-mapped grids. Next, in Section 3.2, we develop quadrature approximations, which employ exponential WENO to operators appearing in the MOL^T formulation on mapped grids. Finally, in Section 3.3, we outline the mapped grid algorithm used to solve the H-J equation.

3.1 Problem description on the physical domain

In the one-dimensional case, (1.1) becomes

$$\phi_t + H(\phi_x) = 0, \quad a \leq x \leq b, \quad (3.1)$$

with $\phi(x, 0) = \phi^0(x)$. Assume that the spatial domain is a closed interval $[a, b]$ and partitioned with $N + 1$ points

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b,$$

with $\Delta x_i = x_{i+1} - x_i$ for $i = 0, \dots, N-1$. These grids of the physical domain could be nonuniform. Let $\phi_i(t)$ denote the solution $\phi(x_i, t)$ at mesh point x_i for $i = 0, \dots, N$. We start with the transformation from the physical domain to the computational domain. Let ξ be the uniformly distributed coordinates on our computational domain $[0, 1]$:

$$0 = \xi_0 < \xi_1 < \dots < \xi_{N-1} < \xi_N = 1,$$

so that $\xi_i = i\Delta\xi$ with $\Delta\xi = 1/N$, and define a one-to-one coordinate transformation by

$$x = x(\xi) : [0, 1] \rightarrow [a, b],$$

with $x(\xi_i) = x_i$, $x(0) = a$ and $x(1) = b$. With this transformation, we can convert the one-dimensional H-J equation (3.1) to a new H-J equation

$$\phi_t + \tilde{H}(\phi_\xi) = 0, \quad 0 \leq \xi \leq 1, \quad (3.2)$$

where

$$\tilde{H}(\phi_\xi) := H(\xi_x \phi_\xi). \quad (3.3)$$

The proposed numerical scheme on the transformed spatial domain is developed according to the semi-discrete equation

$$\frac{d}{dt}\phi_i(t) + \hat{H}(\phi_{\xi,i}^-, \phi_{\xi,i}^+) = 0, \quad i = 0, \dots, N, \quad (3.4)$$

where $\hat{H}$ is a numerical Hamiltonian which is a Lipschitz continuous monotone flux consistent with $\tilde{H}$, i.e.,

$$\hat{H}(u, u) = \tilde{H}(u).$$

Here $\phi_{\xi,i}^-$ and $\phi_{\xi,i}^+$ are the approximations to ϕ_ξ at ξ_i obtained by left-biased and right-biased methods, respectively, to take into account the direction of characteristics propagation of the H-J equation. In this work, the local Lax-Friedrichs flux

$$\hat{H}(u^-, u^+) = \tilde{H}\left(\frac{u^- + u^+}{2}\right) - \alpha_{\tilde{H}}(u^-, u^+) \frac{u^+ - u^-}{2} \quad (3.5)$$

is used with $\alpha_{\tilde{H}}(u^-, u^+) = \max_u |\tilde{H}'(u)|$ where $u \in [\min(u^-, u^+), \max(u^-, u^+)]$.

Lemma 3.1. The numerical scheme for (3.4) on the transformed domain with a newly defined numerical Hamiltonian (3.5) using a coordinate transformation is conservative.

Proof. The equation (3.4) can be discretized with n -th timestep Δt by

$$\phi_i^{n+1} = \phi_i^n - \Delta t \hat{H}(\phi_{\xi,i}^-, \phi_{\xi,i}^+) \quad (3.6)$$

where ϕ^n denotes the semi-discrete solution at t^n . Then it can be proved easily following from the fact that the scheme for (3.6) approximates hyperbolic conservation laws. First, we define a function $\Phi(x, t)$, which satisfies

$$\Phi(x_i) = \frac{1}{\Delta x_i} \int_{x_i}^{x_{i+1}} \phi_x = \frac{\phi_{i+1} - \phi_i}{\Delta x_i}, \quad i = 0, \dots, N-1,$$

and consider the time evolution of this function. Let Φ_i^n be the value of Φ at x_i at the n -th timestep t^n and so we obtain that

$$\frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t} = \frac{1}{\Delta x_i} \left[\frac{\phi_{i+1}^{n+1} - \phi_{i+1}^n}{\Delta t} - \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} \right]. \quad (3.7)$$

Denoting the Jacobian of the coordinate transformation $J = x_\xi$, we can find the relation $\phi_{\xi,i}^\pm = J_i \phi_{x,i}^\pm$ where $J_i = J|_{\xi_i}$, and using this relation with (3.3) and (3.5), the equation (3.6) is converted to

$$\begin{aligned}\phi_i^{n+1} &= \phi_i^n - \Delta t \hat{\tilde{H}}(J_i \phi_{x,i}^-, J_i \phi_{x,i}^+) \\ &= \phi_i^n - \Delta t \left[H \left(\frac{\phi_{x,i}^- + \phi_{x,i}^+}{2} \right) - \alpha_{\tilde{H}} \frac{J_i \phi_{x,i}^+ - J_i \phi_{x,i}^-}{2} \right]\end{aligned}$$

with $\alpha_{\tilde{H}} = \alpha_{\tilde{H}}(\phi_{\xi,i}^-, \phi_{\xi,i}^+) = \max |\tilde{H}'(\phi_{\xi,i})|$. Using this relation, in addition to (3.7), we obtain

$$\frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t} = \frac{1}{\Delta x_i} \left[- \left\{ H \left(\frac{\phi_{x,i+1}^- + \phi_{x,i+1}^+}{2} \right) - \alpha_{\tilde{H}} J_{i+1} \frac{\phi_{x,i+1}^+ - \phi_{x,i+1}^-}{2} \right\} + \left\{ H \left(\frac{\phi_{x,i}^- + \phi_{x,i}^+}{2} \right) - \alpha_{\tilde{H}} J_i \frac{\phi_{x,i}^+ - \phi_{x,i}^-}{2} \right\} \right],$$

which is an update equation of the form

$$\frac{\Phi_i^{n+1} - \Phi_i^n}{\Delta t} = - \frac{1}{\Delta x_i} \left[\hat{H}(\phi_{x,i+1}^-, \phi_{x,i+1}^+) - \hat{H}(\phi_{x,i}^-, \phi_{x,i}^+) \right].$$

We identify

$$\hat{H}(u^-, u^+) = H \left(\frac{u^- + u^+}{2} \right) - \alpha_H(u^-, u^+) \frac{u^+ - u^-}{2},$$

with the relation

$$\alpha_H(u^-, u^+) = \max_u |H'(u)| = \max_u |\tilde{H}'(Ju)| = J \alpha_{\tilde{H}}$$

as the monotone numerical Hamiltonian, which is consistent with H , i.e.,

$$\hat{H}(u, u) = H(u).$$

In other words, this is a conservative approximation to the hyperbolic conservation law

$$\Phi_t + H(\Phi)_x = 0.$$

□

3.2 Space discretization with exponential based WENO schemes

In this subsection, we present the detailed spatial discretization for the operators $\mathcal{D}_L$ and $\mathcal{D}_R$. We can obtain (2.9a) and (2.9b) via recurrence relations for the integral terms:

$$I_R[v; \alpha](x_i) = e^{-\alpha \Delta x_{i-1}} I_R[v; \alpha](x_{i-1}) + J_R[v; \alpha](x_i), \quad (3.8)$$

$$I_L[v; \alpha](x_i) = e^{-\alpha \Delta x_i} I_L[v; \alpha](x_{i+1}) + J_L[v; \alpha](x_i), \quad (3.9)$$

where the local integrals are defined by

$$J_R[v; \alpha](x_i) = \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} v(s) ds \quad (3.10)$$

$$J_L[v; \alpha](x_i) = \alpha \int_{x_i}^{x_{i+1}} e^{-\alpha(s-x_i)} v(s) ds. \quad (3.11)$$

By calculating the convolution integrals with this recurrence relation, we obtain a summation method which has a complexity of $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$.

In order to compute $J_L[v; \alpha](x_i)$ and $J_R[v; \alpha](x_i)$, we propose a high order exponential based approximation. The process to approximate $J_L[v; \alpha](x_i)$ is simply mirror symmetric to that of $J_R[v; \alpha](x_i)$ with respect to point x_i , so we will illustrate the process only for the term $J_R[v; \alpha](x_i)$. In [18, 19], the authors introduced

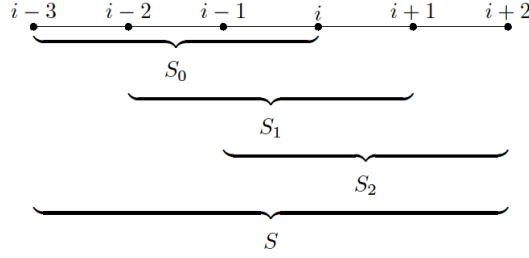


Figure 3.1: The 6-point stencil S with three 4-point stencils S_r , $r = 0, 1$ and 2 .

a sixth-order WENO scheme based on the exponential polynomial space, which we shall follow for approximating $J_R[v; \alpha](x_i)$. To begin, we consider an interpolation stencil consisting of $k+1$ points, which contains x_{i-1} and x_i :

$$S(i) = \{x_{i-r}, \dots, x_{i-r+k}\},$$

to find a unique polynomial of degree at most k , denoted as $p(x)$, which interpolates $v(x)$ at the points in $S(i)$ so that

$$J_R[v; \alpha](x_i) \approx \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} p(s) ds.$$

In this paper, a six-point stencil $S := S(i) = \{x_{i-3}, \dots, x_{i+2}\}$ is used and this stencil is subdivided into three substencils $S_0, \dots, S_2$ defined by $S_r := S_r(i) = \{x_{i-3+r}, \dots, x_{i+r}\}$ for $r = 0, 1$ and 2 . The corresponding stencil is shown in Figure 3.1.

Let $\{\phi_1, \dots, \phi_k\}$ be a set of exponential polynomials of the form $\phi(x) = x^n e^{\lambda x}$, with $n \in \mathbb{N} \cup \{0\}$ and λ being a data-dependent “tension” parameter, which can be tuned to improve the approximation. For example, we can choose $\lambda \in \mathbb{R}$ or $\lambda \in i\mathbb{R}$ ($i = \sqrt{-1}$), with the function ϕ being a trigonometric polynomial if λ is in $i\mathbb{R}$. With these exponential polynomials as basis functions, we define a rank k space Γ_k by

$$\Gamma_k := \text{span}\{\phi_1, \dots, \phi_k\},$$

which satisfies

$$\det(\phi_n(s_j) : j, n = 1, \dots, k) \neq 0 \quad (3.12)$$

for a k -point stencil $\{s_j : j = 1, \dots, k\}$. It is recommended that the polynomial $\phi(x) \equiv 1$ is contained within the basis in order for an interpolation kernel to satisfy a partition of unity, so we choose

$$\Gamma_6 := \text{span}\{1, x, x^2, x^3, e^{\lambda x}, e^{-\lambda x}\} \quad (3.13)$$

as the basis functions for global stencil S and similarly,

$$\Gamma_4 := \text{span}\{1, x, e^{\lambda x}, e^{-\lambda x}\} \quad (3.14)$$

for the four-point substencils. Here, Γ_6 and Γ_4 constitute *extended Tchebysheff systems* on $\mathbb{R}$ so that the non-singularity of the interpolation matrices in (3.12) is guaranteed, see [18, 22].

On the big stencil $S(i)$, using the basis defined on Γ_6 , we obtain the approximation as

$$J_i^R := \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} p(s) ds, \quad (3.15)$$

where p is an interpolant for v that satisfies $J_i^R = J_R[v; \alpha](x_i) + \mathcal{O}(\Delta x^6)$ if v is smooth on $S(i)$. Similarly, on each of the smaller stencils, we have

$$J_{i,r}^R := \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} p_r(s) ds, \quad (3.16)$$

where p_r is the interpolant to v using the basis from Γ_4 on nodes $S_r(i)$ which satisfies $J_{i,r}^R = J_R[v; \alpha](x_i) + \mathcal{O}(\Delta x^4)$ for smooth v . When the function v is smooth, we would like to combine approximations on the smaller stencils $S_r(i)$ so they are consistent with those on the larger stencil $S(i)$, i.e.,

$$J_i^R = \sum_{r=0}^2 d_r J_{i,r}^R. \quad (3.17)$$

The coefficients d_r are the linear weights which satisfy $\sum_{r=0}^2 d_r = 1$.

The construction of smoothness indicators is as follows: First, for $r = 0, 1, 2$, we use m th-order generalized undivided differences ($m = 2, 3$) on S_r defined by

$$D_r^m v_i := \sum_{x_n \in S_r(i)} c_{r,n}^{[m]} v(x_n), \quad (3.18)$$

which converge to $\Delta x^m v^{(m)}(x_i)$ at a higher convergence rate than classical undivided differences. Let n_r indicate the number of points inside the stencil S_r and define the coefficient vector $\mathbf{c}^{[m]} := (c_{r,n}^{[m]} : x_n \in S_r)^T$ in (3.18) by solving the linear system

$$\mathbf{V} \cdot \mathbf{c}^{[m]} = \boldsymbol{\delta}^{[m]},$$

for the non-singular matrix

$$\mathbf{V} := \left(\frac{(x_n - x_i)^\ell}{\Delta x^\ell \ell!} : x_n \in S_r, \ell = 0, \dots, n_r - 1 \right)$$

and $\boldsymbol{\delta}^{[m]} := (\delta_{m,\ell} : \ell = 0, \dots, n_r - 1)^T$. Then a simple calculation with Taylor expansion shows that

$$D_r^m v_i = \Delta x^m v^{(m)}(x_i) + \mathcal{O}(\Delta x^{n_r}), \quad (3.19)$$

on the smooth region. We now define a measurement for the smoothness of data in each substencil by

$$\beta_r := |D_r^2 v_i|^2 + |D_r^3 v_i|^2, \quad \text{for } r = 0, 1, 2, \quad (3.20)$$

and the global smoothness indicator τ is simply defined as the absolute difference between β_0 and β_2 , i.e.,

$$\tau = |\beta_0 - \beta_2|.$$

We form the final approximation using

$$\mathcal{A}J_i^R := \sum_{r=0}^2 \omega_r J_{i,r}^R. \quad (3.21)$$

The nonlinear weights ω_r , in the above, are defined as

$$\omega_r = \tilde{\omega}_r / \sum_{s=0}^2 \tilde{\omega}_s \quad \text{and} \quad \tilde{\omega}_r = d_r \left(1 + \frac{\tau}{\epsilon + \beta_r} + \frac{1}{2} \left(\frac{\beta_r}{\epsilon + \tau} \right)^2 \right), \quad (3.22)$$

where $\epsilon > 0$ is a small number to avoid the denominator becoming zero. In our experiments we take $\epsilon = 10^{-6}$. Note that we have employed the nonlinear weights based on the idea of the WENO-P+3 scheme proposed in [34], which enjoys less dissipation and higher resolution compared with the classical WENO schemes. This construction enables the approximation for $J_R[v; \alpha](x_i)$ to retain a high order of accuracy and it is proven in the following theorem.

Theorem 3.2. *Assume that v is smooth on the global stencil $S(i)$. Then the approximation $\mathcal{A}J_i^R$ defined in (3.21) satisfies the relation*

$$|J_R[v; \alpha](x_i) - \mathcal{A}J_i^R| = \mathcal{O}(\Delta x^6),$$

i.e., converges to $J_R[v; \alpha](x_i)$ in 6th order accuracy.

Proof. From the equations (3.17) and (3.21), it is easily obtained that

$$J_R[v; \alpha](x_i) - \mathcal{A}J_i^R = \left(J_R[v; \alpha](x_i) - \sum_{r=0}^2 d_r J_{i,r}^R \right) + \sum_{r=0}^2 (d_r - \omega_r) J_{i,r}^R$$

using the local approximations $J_{i,r}^R$ to $J_R[v; \alpha](x_i)$. By definition of the linear weights d_r in the (3.15) and (3.17), the first term on the right-hand side has the designed order of accuracy so that it is sufficient to consider the second term. Since the local integral $J_{i,r}^R$ is constructed to have convergence in (3.16) and both weights $\{d_r\}$ and $\{\omega_r\}$ fulfill a partition of unity, we have

$$\begin{aligned} \sum_{r=0}^2 (d_r - \omega_r) J_{i,r}^R &= \sum_{r=0}^2 (d_r - \omega_r) (J_R[v; \alpha](x_i) + \mathcal{O}(\Delta x^4)) \\ &= \sum_{r=0}^2 (d_r - \omega_r) \mathcal{O}(\Delta x^4) = \mathcal{O}(\Delta x^6), \end{aligned}$$

where the last equality is straightforward from (3.19) and (3.22) if we select $\epsilon = \Delta x^2$ in (3.22). $\square$

Remark 3.3. In [10,11], the authors proposed to adapt nonlinear filters $\sigma_{i,L}$ and $\sigma_{i,R}$ to control oscillations which arise when the derivative of the solution to the H-J equation develops discontinuities. For example, in the periodic case, the approximation is given by

$$\begin{aligned} \phi_{x,i}^- &= \alpha \mathcal{D}_R[\phi; \alpha](x_i) + \alpha \sum_{p=2}^k \sigma_{i,R} \mathcal{D}_R^p[\phi; \alpha](x_i), \\ \phi_{x,i}^+ &= -\alpha \mathcal{D}_L[\phi; \alpha](x_i) - \alpha \sum_{p=2}^k \sigma_{i,L} \mathcal{D}_L^p[\phi; \alpha](x_i). \end{aligned} \tag{3.23}$$

The authors applied WENO quadrature to approximate the operators $\mathcal{D}_L$ and $\mathcal{D}_R$, but only for the first step, which corresponds to $p = 1$ in (3.23). The filter was then adapted for the case $p \geq 2$, so that a cheaper linear quadrature, defined on fixed stencils, can be used.

We design a filter by defining parameters θ_i as

$$\theta_i = \frac{\min_r (|D_r^1 v_i| + |D_r^2 v_i|) + \epsilon}{\max_r (|D_r^1 v_i| + |D_r^2 v_i|) + \epsilon}, \quad \text{for } r = 0, \dots, 3, \tag{3.24}$$

where $D_r^k v_i$ are undivided differences of order k on four three-point stencils around v_i , defined in (3.19). Then, we adopt a nondecreasing map μ designed with cubic B -splines for θ_i , in Figure 3.2 and the filter σ_i is defined as

$$\sigma_i = \mu(2 \cdot \theta_i^2). \tag{3.25}$$

This definition for the filters allows us to preserve the approximation order of these schemes. If the big stencil is determined to be smooth according to the definitions (3.18) and (3.24), then we see that $\sigma_i \approx 1$. On the other hand, if a singularity develops in the vicinity of x_i , then $\sigma_i \approx 0$.

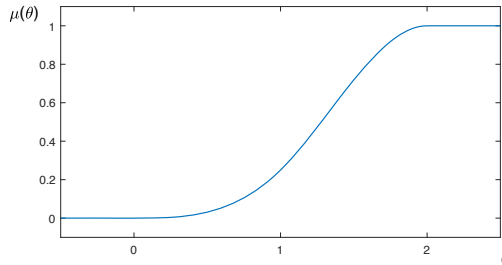


Figure 3.2: The function μ designed with cubic B -splines for filters σ .

3.3 Algorithm

In what follows, we will use H and the coordinates x , rather than $\tilde{H}$ and coordinates ξ , for purposes of convenience. For time integration, we propose to use the classic explicit SSP RK methods [17] to advance the solution from time t^n to t^{n+1} . We denote ϕ^n as the semi-discrete solution at time t^n . In this work, we use the following SSP RK methods, including the first order forward Euler scheme

$$\phi^{n+1} = \phi^n - \Delta t \hat{H}(\phi_x^{n,-}, \phi_x^{n,+}); \quad (3.26)$$

the second order SSP RK scheme

$$\begin{aligned} \phi^{(1)} &= \phi^n - \Delta t \hat{H}(\phi_x^{n,-}, \phi_x^{n,+}), \\ \phi^{n+1} &= \frac{1}{2}\phi^n + \frac{1}{2} \left(\phi^{(1)} - \Delta t \hat{H}(\phi_x^{(1),-}, \phi_x^{(1),+}) \right); \end{aligned} \quad (3.27)$$

and the third order SSP RK scheme

$$\begin{aligned} \phi^{(1)} &= \phi^n - \Delta t \hat{H}(\phi_x^{n,-}, \phi_x^{n,+}), \\ \phi^{(2)} &= \frac{3}{4}\phi^n + \frac{1}{4} \left(\phi^{(1)} - \Delta t \hat{H}(\phi_x^{(1),-}, \phi_x^{(1),+}) \right), \\ \phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3} \left(\phi^{(2)} - \Delta t \hat{H}(\phi_x^{(2),-}, \phi_x^{(2),+}) \right). \end{aligned} \quad (3.28)$$

In addition, linear stability of the proposed kernel-based schemes has been established in [11]:

Theorem 3.4. *For the linear equation $\phi_t + c\phi_x = 0$, (i.e. the Hamiltonian is linear) with periodic boundary conditions, we consider the k^{th} order SSP RK method as well as the k^{th} partial sum in (2.14), with $\alpha = \beta/(|c|\Delta t)$. Then there exists a constant $\beta_{k,\max} > 0$ for $k = 1, 2, 3$, such that the scheme is unconditionally stable provided $0 < \beta \leq \beta_{k,\max}$. The constants $\beta_{k,\max}$ for $k = 1, 2, 3$ are summarized in following:*

k	1	2	3
$\beta_{k,\max}$	2	1	1.243

We summarize the proposed scheme for solving the one dimensional periodic boundary case in the following algorithm flowchart.

Algorithm: MOL^T-type scheme for solving one-dimensional H-J equation on nonuniform grids

We solve (3.1) until the final time $t = T$. Let the given nonuniform grid on the physical domain $[a, b]$ be $\{x_i\}$, $i = 0, \dots, N$. We denote the numerical solution at n -th time step $t = t^n$ by $\{\phi_i^n := \phi^n(x_i)\}$. Start with $n = 0$ noting that $\{\phi_i^0\}$ is given.

1. Define a computational grid $\{\xi_i\}$, $i = 0, \dots, N$, on $[0, 1]$ with uniform distribution $\Delta\xi = 1/N$.
2. Approximate the associated Jacobian for the transformation x_ξ by a fourth-order finite difference scheme:

$$\{x_\xi|_i = \frac{x_{i-2} - 8x_{i-1} + 8x_{i+1} - x_{i+2}}{12\Delta\xi}\}, \quad (3.29)$$

and then we now solve the transformed H-J equation (3.2).

3. Set β depending on desired order k of the scheme according to Theorem 3.4 and the time step size Δt^0 by CFL condition (4.1).

While $t < T$, given ϕ^n and Δt^n at time $t = t^n$, $n \geq 0$,

4. Approximate the integrals J_L and J_R using the exponential polynomial based WENO quadrature. For $i = 0, \dots, N$,
 - (a) Construct the local approximations $J_{i,r}^L$ and $J_{i,r}^R$ based on exponential polynomials (3.16) on each substencil S_r for $r = 0, 1, 2$, determine the linear weights d_r , and form the approximation given by equation (3.17).
 - (b) Compute the smoothness indicators of local data via (3.20) and construct nonlinear weights (3.22) for the final approximations J_i^L and J_i^R in (3.21).
 5. Using the previously computed approximations $\mathcal{A}J_i^L$ and $\mathcal{A}J_i^R$, apply the recurrence relations (3.8) and (3.9) to obtain the convolution integrals. Form the inverse operators (2.8a) and (2.8b) by applying boundary conditions (see Section 2.3). Once we have $\mathcal{L}_L^{-1}$ and $\mathcal{L}_R^{-1}$, construct the operators $\mathcal{D}_L$ and $\mathcal{D}_R$ defined in (2.10).
 6. Repeat step 4. and 5. k times to approximate the first derivatives ϕ_ξ^- and ϕ_ξ^+ with k partial sums of $\mathcal{D}_L$ and $\mathcal{D}_R$, respectively. If multiple terms in the partial sums are desired, i.e., $k \geq 2$, then apply WENO quadrature only to the first terms and use the cheaper linear quadrature on the remaining ones. For this case, additional filters (3.25), obtained in WENO quadrature, are needed. Filters are applied according to (2.14).
 7. Form the local Lax-Friedrichs Hamiltonian (3.5) for the transformed H-J equation (3.4) using the inverse of associated Jacobian (3.29), i.e., $\{\xi_x\}$. Then, update the time step from t^n to $t^{n+1} = t^n + \Delta t^n$ by applying an appropriate RK scheme (3.26) - (3.28). One should couple an order k RK method to an approximation for the partial derivatives of an equivalent order for consistency.
 8. If $t^{n+1} < T$, set the time step size Δt^{n+1} by (4.1) with updated wave speeds. Otherwise, $t^{n+1} + \Delta t^{n+1} > T$, so we set $\Delta t^{n+1} = T - t^{n+1}$. Execute another time step of the process, beginning with 4., until time T .
-

3.4 Two-dimensional implementation

Consider the two-dimensional H-J equation

$$\phi_t + H(\phi_x, \phi_y) = 0, \quad (3.30)$$

with $\phi(x, y, 0) = \phi^0(x, y)$. We assume (x_i, y_j) refers to the (i, j) -th node of a two-dimensional orthogonal grid. The spacing between points is denoted by $\Delta x_i = x_i - x_{i-1}$ and $\Delta y_j = y_j - y_{j-1}$. In addition, we shall take $\phi(x_i, y_j, t) = \phi_{i,j}(t)$ as the discrete solution to (3.30) on the grid.

As with the one-dimensional case, we assume the existence of one-to-one coordinate transformations $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$ from the computational domain $[0, 1] \times [0, 1]$ to the physical domain. Here, the computational domain is distributed by a fixed uniform mesh given by $\xi_i = i\Delta\xi$ and $\eta_j = j\Delta\eta$ with $\Delta\xi = 1/N$ and $\Delta\eta = 1/N$. Then, the H-J equation (3.30) defined on irregular domain becomes

$$\phi_t + \tilde{H}(\phi_\xi, \phi_\eta) = 0 \quad (3.31)$$

defined on uniform spatial domain, where $\tilde{H}(\phi_\xi, \phi_\eta) := H(\xi_x \phi_\xi + \eta_x \phi_\eta, \xi_y \phi_\xi + \eta_y \phi_\eta)$. Below, we will use H and coordinates x and y instead of $\tilde{H}$ and coordinates ξ and η . In the two-dimensional examples, we shall use the semi-discrete scheme

$$\frac{d}{dt} \phi_{i,j}(t) + \hat{H}(\phi_{x,i,j}^-, \phi_{x,i,j}^+, \phi_{y,i,j}^-, \phi_{y,i,j}^+) = 0, \quad (3.32)$$

where the numerical Hamiltonian $\hat{H}(u, u; v, v)$ is a Lipschitz continuous monotone flux that is consistent with H . As in the one-dimensional case, we employ the local Lax-Friedrichs numerical Hamiltonian.

4 Numerical results

In this section, we present the numerical results of the proposed scheme for one-dimensional and two-dimensional Hamilton-Jacobi equations using regular and irregular grids discussed in Section 4.1, 4.2 and 4.3, respectively. The performance of proposed scheme is examined by comparing the results from the previous kernel based unconditionally stable scheme [11]. The code implementation in Python with some sample results are available on the web [12]. The parameter $\lambda > 0$ in the exponential basis can be tuned according to the problem, but in this paper, it is selected so that $\lambda\Delta\xi = 1$ in all experiments. While there are examples with a CFL > 1 , unless otherwise stated, the presented numerical results are computed by the third-order scheme (i.e., $k = 3$) using CFL = 0.5 to demonstrate the performance.

4.1 One-dimensional cases

Here, we present convergence results for the schemes on a one-dimensional uniform mesh with $\Delta x = \Delta x_i$ for $i = 0, \dots, N$. The time step is set by

$$\Delta t = \text{CFL} \frac{\Delta x}{\alpha}, \quad (4.1)$$

where α is the maximum wave propagation speed. We see the order of accuracy of proposed scheme for the linear and non-linear problems and present numerical results for several H-J examples.

Example 4.1. We first solve the linear advection equation

$$\phi_t + \phi_x = 0 \quad (4.2)$$

on the spatial domain $[0, 1]$ with periodic boundary conditions. For the initial condition, we use the smooth function

$$\phi(x, 0) = \phi_1(x) := \sin(2\pi x).$$

In Table 4.1, we provide the L_∞ errors at time $T = 1$ and along with the associated order of accuracy. We can see that k th order of accuracy is achieved for $k = 2$ and 3 cases and second order accuracy is observed for the case $k = 1$. Such superconvergence for the first order scheme, with $k = 1$, is expected by observing that the proposed scheme, with $\beta = 2$, applied to the linear problem, is equivalent to the second order Crank-Nicolson scheme [9].

In the second case, we use the following initial function

$$\phi(x, 0) = \phi_2(x) := \begin{cases} 0 & \text{if } 0 \leq x \leq 0.25, \\ \frac{40}{3}(x - \frac{1}{4}) & \text{if } 0.25 < x < 0.4, \\ 2 & \text{if } 0.4 \leq x \leq 0.6, \\ \frac{40}{3}(\frac{3}{4} - x) & \text{if } 0.6 < x < 0.75, \\ 0 & \text{if } 0.75 \leq x \leq 1 \end{cases}$$

which is a continuous and piecewise linear function. We plot the numerical solution and its derivative at time $T = 1$, using $N = 80$ grid points, in Figure 4.3. We see that the proposed scheme improves the accuracy of the approximation near the jump discontinuity in the derivatives when compared to our previous scheme.

Example 4.2. In this example, we consider the Burgers' equation

$$\phi_t + \frac{1}{2}(\phi_x + 1)^2 = 0, \quad (4.3)$$

with the smooth initial function

$$\phi(x, 0) = -\cos(\pi x)$$

on the spatial domain $[-1, 1]$ with periodic boundary conditions at two different final times.

Our first test for this problem stops at time $T = 0.5/\pi^2$, when the solution is still smooth. In Table 4.2, we provide the L_∞ errors and orders of accuracy for schemes with $k = 1, 2$ and 3, which are shown to be

CFL	N	$k = 1. \beta = 2.$		$k = 2. \beta = 1.$		$k = 3. \beta = 1.2.$	
		error	order	error	order	error	order
0.5	20	1.28e-02	—	1.73e-01	—	2.25e-03	—
	40	3.23e-03	1.987	4.48e-02	1.947	1.72e-04	3.707
	80	8.07e-04	1.998	1.13e-02	1.990	1.74e-05	3.310
	160	2.02e-04	2.000	2.82e-03	1.998	2.02e-06	3.104
	320	5.05e-05	2.000	7.06e-04	2.000	2.40e-07	3.076
1	20	5.08e-02	—	5.61e-01	—	3.15e-02	—
	40	1.29e-02	1.981	1.73e-01	1.698	2.35e-03	3.744
	80	3.23e-03	1.996	4.48e-02	1.948	1.86e-04	3.660
	160	8.07e-04	1.999	1.13e-02	1.990	1.80e-05	3.368
	320	2.02e-04	2.000	2.82e-03	1.998	2.06e-06	3.129
2	20	1.94e-01	—	9.92e-01	—	3.08e-01	—
	40	5.09e-02	1.931	5.66e-01	0.810	3.16e-02	3.283
	80	1.29e-02	1.984	1.73e-01	1.710	2.36e-03	3.747
	160	3.23e-03	1.996	4.48e-02	1.948	1.86e-04	3.661
	320	8.07e-04	1.999	1.13e-02	1.990	1.80e-05	3.370

Table 4.1: L_∞ -errors and orders of accuracy for Example 4.1 with $\phi_1(x)$ at $T = 1$.

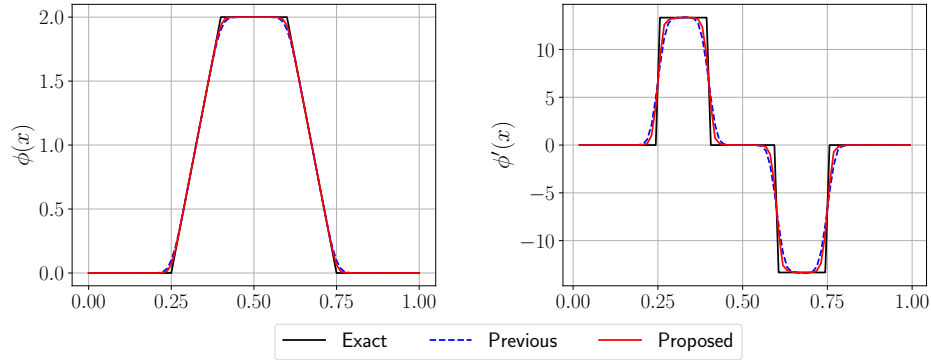


Figure 4.3: Numerical solution ϕ and its derivative ϕ_x for Example 4.1 with $\phi_2(x)$ at $T = 1$.

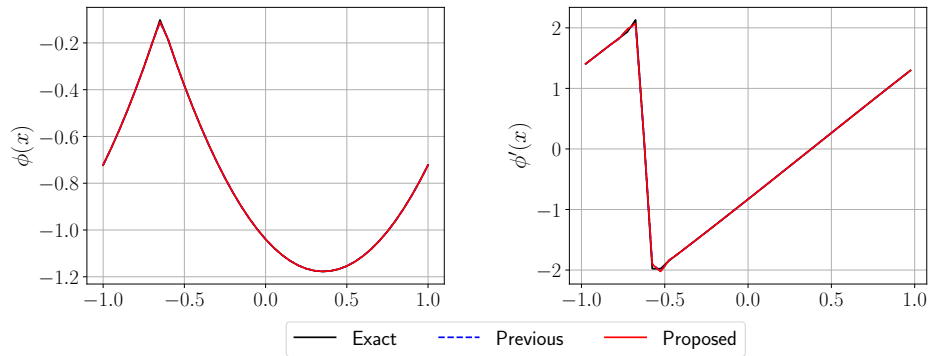


Figure 4.4: Numerical solution ϕ and its derivative ϕ_x for Example 4.2 at $T = 3.5/\pi^2$.

CFL	N	$k = 1. \beta = 2.$		$k = 2. \beta = 1.$		$k = 3. \beta = 1.2.$	
		error	order	error	order	error	order
0.5	20	1.57e-02	—	1.69e-02	—	3.30e-04	—
	40	7.94e-03	0.985	4.73e-03	1.839	3.63e-05	3.186
	80	4.09e-03	0.958	1.29e-03	1.881	4.58e-06	2.986
	160	2.07e-03	0.980	3.31e-04	1.958	5.20e-07	3.137
	320	1.05e-03	0.987	8.47e-05	1.966	6.00e-08	3.115
1	20	3.19e-02	—	5.48e-02	—	4.06e-03	—
	40	1.57e-02	1.025	1.70e-02	1.691	5.08e-04	2.998
	80	7.95e-03	0.980	4.76e-03	1.833	4.98e-05	3.351
	160	4.12e-03	0.947	1.29e-03	1.890	5.00e-06	3.316
	320	2.07e-03	0.992	3.31e-04	1.959	5.32e-07	3.234
2	20	9.14e-02	—	1.92e-01	—	3.36e-02	—
	40	3.23e-02	1.501	5.67e-02	1.757	4.51e-03	2.898
	80	1.57e-02	1.042	1.71e-02	1.731	5.12e-04	3.139
	160	8.03e-03	0.966	4.76e-03	1.842	5.00e-05	3.356
	320	4.12e-03	0.961	1.29e-03	1.890	5.01e-06	3.320

Table 4.2: L_∞ -errors and orders of accuracy for Example 4.2 at $T = 0.5/\pi^2$.

k -th order. Provided in Figure 4.4 is a plot of the numerical solution at $T = 3.5/\pi^2$, when the shock occurs. Here, $N = 40$ grid points are used to compute the solution. It seems that the proposed scheme is as effective as the previous result.

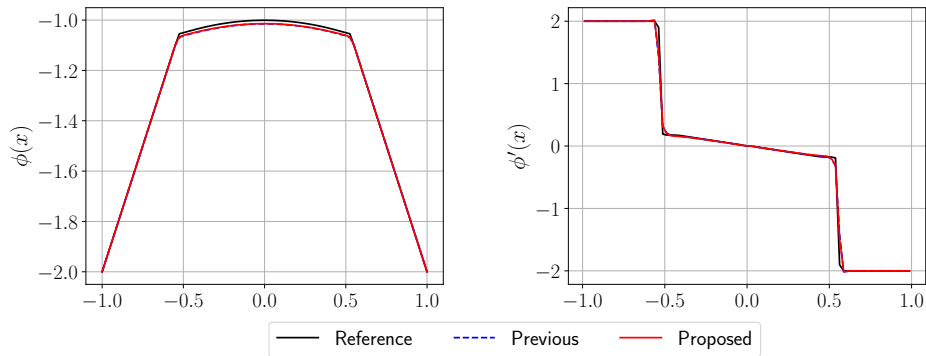
Example 4.3. We now solve the one-dimensional Riemann problem with a non-convex Hamiltonian

$$\phi_t + \frac{1}{4} (\phi_x^2 - 1) (\phi_x^2 - 4) = 0. \quad (4.4)$$

Here, we use the initial condition

$$\phi(x, 0) = -2|x|,$$

on the fixed spatial domain $[-1, 1]$ with the inflow Dirichlet boundary conditions $\phi(\pm 1, t) = -2$. In Figure 4.5, we show plots of the numerical solution, which is computed up to time $T = 1$ using $N = 80$ grid points. We measure convergence relative to a reference solution that is computed with $N = 1600$ grid points. Both the previous and proposed methods are effective at resolving the reference solution.

Figure 4.5: Numerical solution ϕ and its derivative ϕ_x for Example 4.3 at $T = 1$.

4.2 Two-dimensional cases

For the two-dimensional cases with uniformly distributed grids, the time step is chosen as

$$\Delta t = \frac{\text{CFL}}{(\alpha_x/\Delta x + \alpha_y/\Delta y)}, \quad (4.5)$$

where α_x and α_y are the maximum wave propagation speeds in the x and y directions, respectively. To demonstrate the performance of proposed scheme, we present the numerical solutions using $\text{CFL} = 0.5$ as well as $\text{CFL} = 2$.

Example 4.4. For our first example, we consider the linear advection equation

$$\phi_t + (\phi_x + \phi_y + 1) = 0, \quad (4.6)$$

and we apply a periodic boundary condition in each direction. We first consider the smooth initial data given by

$$\phi(x, y, 0) = \phi_3(x, y) := -\cos(\pi(x + y)/2),$$

which is defined on the spatial domain $[-2, 2] \times [-2, 2]$. We report the L_∞ error and orders of accuracy at time $T = 2$ in Table 4.3. It is observed that the proposed scheme achieves the appropriate convergence

CFL	$N_x \times N_y$	$k = 1. \beta = 1.$		$k = 2. \beta = 0.5.$		$k = 3. \beta = 0.6.$	
		error	order	error	order	error	order
0.5	20×20	1.28e-02	—	1.73e-01	—	2.25e-03	—
	40×40	3.22e-03	1.987	4.48e-02	1.947	1.69e-04	3.736
	80×80	8.07e-04	1.998	1.13e-02	1.990	1.65e-05	3.356
	160×160	2.02e-04	2.000	2.82e-03	1.998	1.90e-06	3.115
	320×320	5.05e-05	2.000	7.06e-04	2.000	2.33e-07	3.032
1	20×20	5.08e-02	—	5.61e-01	—	3.15e-02	—
	40×40	1.29e-02	1.981	1.73e-01	1.698	2.35e-03	3.744
	80×80	3.23e-03	1.996	4.48e-02	1.948	1.86e-04	3.662
	160×160	8.07e-04	1.999	1.13e-02	1.990	1.80e-05	3.370
	320×320	2.02e-04	2.000	2.82e-03	1.998	2.04e-06	3.135
2	20×20	1.94e-01	—	9.92e-01	—	3.08e-01	—
	40×40	5.09e-02	1.931	5.66e-01	0.810	3.16e-02	3.283
	80×80	1.29e-02	1.984	1.73e-01	1.710	2.36e-03	3.747
	160×160	3.23e-03	1.996	4.48e-02	1.948	1.86e-04	3.661
	320×320	8.07e-04	1.999	1.13e-02	1.990	1.80e-05	3.370

Table 4.3: L_∞ -errors and orders of accuracy for Example 4.4 with $\phi_3(x, y)$ at $T = 2$.

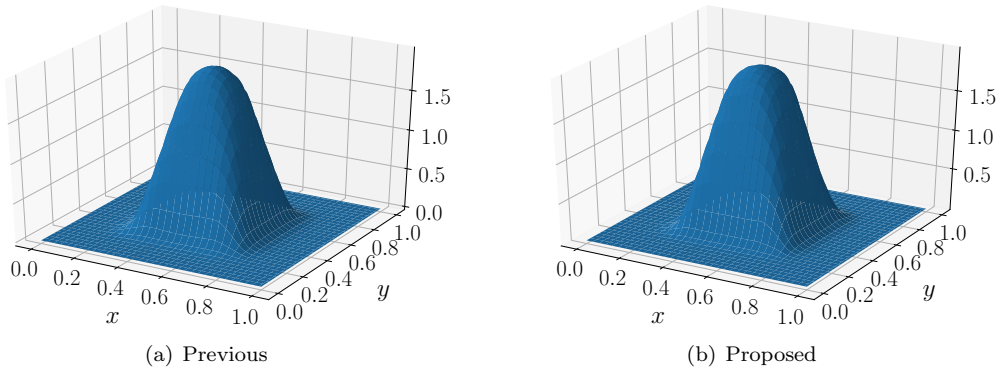


Figure 4.6: Numerical solutions for Example 4.4 with $\phi_4(x, y)$ at $T = 1$.

orders as it does for the one-dimensional case in Example 4.1. As before, we have the superconvergence for $k = 1$ case.

In the second test problem, we test our method on a piece-wise continuous, i.e., C^0 , function

$$\phi(x, y, 0) = \phi_4(x, y) := \begin{cases} 0 & \text{if } x \leq 0.2 \text{ or } y \leq 0.2, \\ 2 & \text{if } 0.4 \leq x, y \leq 0.6, \\ 0 & \text{if } x \geq 0.8 \text{ or } y \geq 0.8, \\ f(x, y) & \text{otherwise} \end{cases}$$

defined on the domain $[0, 1] \times [0, 1]$, where $f(x, y)$ is a real-valued piece-wise linear function defined so that ϕ_4 is continuous. We display the result obtained with the proposed scheme, as well as the previous approach using a 100×100 grid of points. Plots of the numerical solutions at the time $T = 1$ are provided in Figure 4.6. As with Example 4.1, we can see the resolution improvements of the proposed scheme around sharp edges, compared with the previous scheme.

Example 4.5. In this example, we solve the Burgers' equation

$$\phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0 \quad (4.7)$$

with a smooth initial condition

$$\phi(x, y, 0) = -\cos(\pi(x + y)/2), \quad (4.8)$$

CFL	$N_x \times N_y$	$k = 1. \beta = 1.$		$k = 2. \beta = 0.5.$		$k = 3. \beta = 0.6.$	
		error	order	error	order	error	order
0.5	20×20	5.48e-02	—	1.68e-02	—	6.36e-04	—
	40×40	2.98e-02	0.877	4.72e-03	1.836	5.11e-05	3.637
	80×80	1.63e-02	0.874	1.28e-03	1.880	4.29e-06	3.574
	160×160	8.37e-03	0.961	3.30e-04	1.956	5.13e-07	3.065
	320×320	4.27e-03	0.970	8.46e-05	1.965	6.03e-08	3.091
1	20×20	9.84e-02	—	5.89e-02	—	7.60e-03	—
	40×40	5.54e-02	0.829	1.69e-02	1.801	9.22e-04	3.043
	80×80	3.05e-02	0.860	4.74e-03	1.835	6.16e-05	3.904
	160×160	1.63e-02	0.906	1.28e-03	1.887	4.75e-06	3.697
	320×320	8.37e-03	0.961	3.30e-04	1.956	5.25e-07	3.177
2	20×20	1.66e-01	—	2.41e-01	—	5.63e-02	—
	40×40	9.94e-02	0.739	5.96e-02	2.017	8.39e-03	2.745
	80×80	5.64e-02	0.817	1.70e-02	1.810	9.15e-04	3.197
	160×160	3.05e-02	0.886	4.74e-03	1.841	6.26e-05	3.871
	320×320	1.63e-02	0.906	1.28e-03	1.886	4.76e-06	3.717

Table 4.4: L_∞ -errors and orders of accuracy for Example 4.5 at $T = 0.5/\pi^2$.

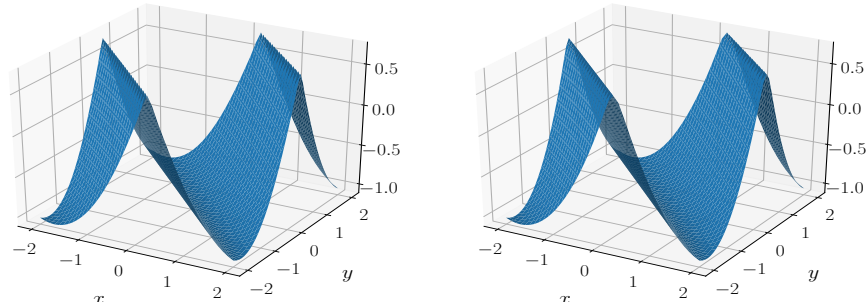


Figure 4.7: Numerical solutions for Example 4.5 with CFL = 0.5 (left) and CFL = 2 (right) at $T = 1.5/\pi^2$.

on the periodic spatial domain $[-2, 2] \times [-2, 2]$. We compute the L_∞ errors at time $T = 0.5/\pi^2$, when the solution is smooth. Table 4.4 shows the associated orders of accuracy measured for each of the schemes, which match the expected orders. Also in Figure 4.7, is a plot of the numerical solutions using 40×40 grid points at time $T = 1.5/\pi^2$, when the solution has developed a discontinuous derivative.

Example 4.6. We now consider a Hamiltonian which is neither convex nor concave:

$$\phi_t - \cos(\phi_x + \phi_y + 1) = 0. \quad (4.9)$$

Here the spatial domain $[-2, 2] \times [-2, 2]$ is defined with periodic boundary conditions in both x and y directions. In Figure 4.8, we plot the numerical solutions at time $T = 1.5/\pi^2$ using a 40×40 grid with the smooth initial data (4.8). We observe that the proposed schemes maintain good resolution when a larger CFL number is used, i.e., $\text{CFL} = 2$, relative to $\text{CFL} = 0.5$.

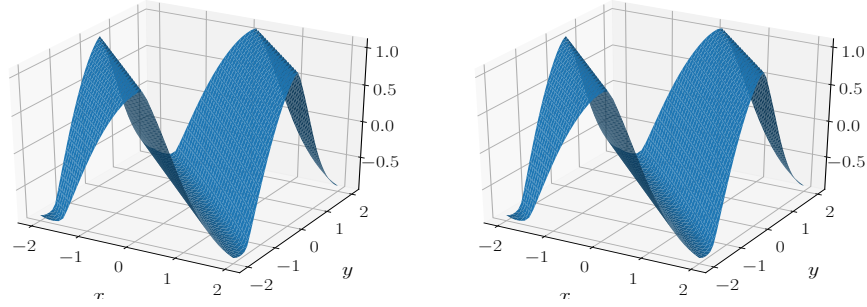


Figure 4.8: Numerical solutions for Example 4.6 with $\text{CFL}=0.5$ (left) and $\text{CFL}=2$ (right) at $T = 1.5/\pi^2$.

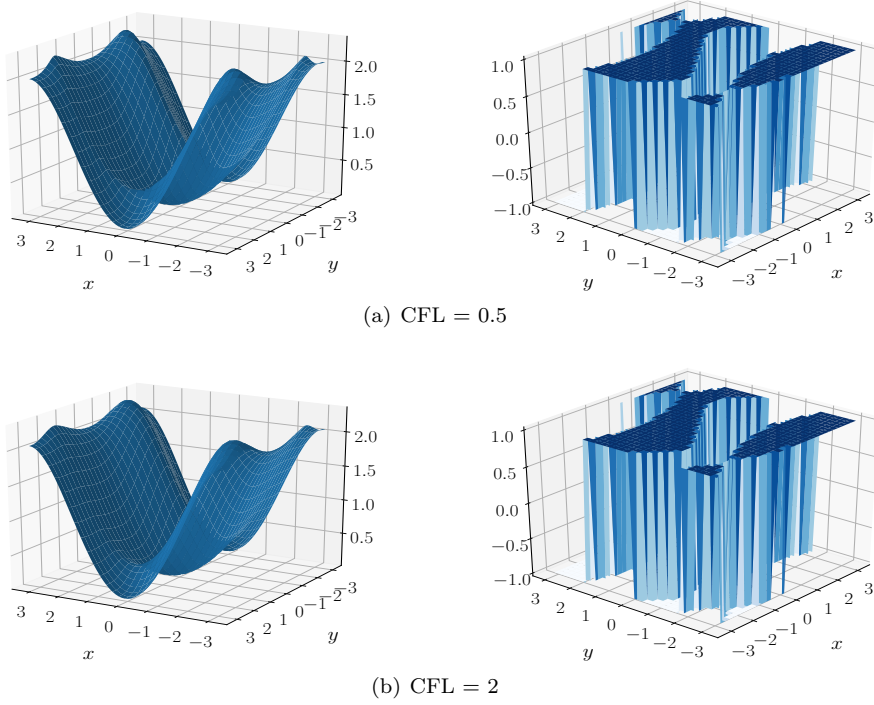


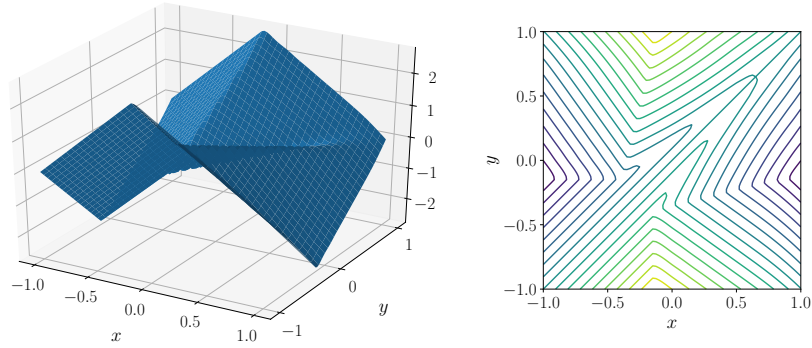
Figure 4.9: Numerical solutions' surfaces (left) and optimal controls (right) for Example 4.7 at $T = 1$.

Example 4.7. We solve the following optimal control problem related to cost determination:

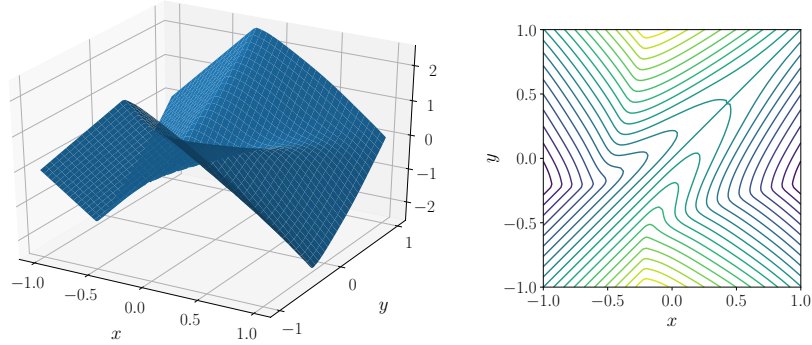
$$\begin{aligned} \phi_t + \sin(y)\phi_x + (\sin(x) + \text{sign}(\phi_y))\phi_y - \frac{1}{2}\sin^2(y) - 1 + \cos(x) &= 0, \\ \phi(x, y, 0) &= 0, \end{aligned} \quad (4.10)$$

on the periodic spatial domain $[-\pi, \pi] \times [-\pi, \pi]$. We compute the numerical solutions up to $T = 1$ using grids of size 60×60 and provide plots of the numerical solution and the optimal control $\text{sign}(\phi_y)$ in Figure 4.9. Proposed schemes capture the non-smooth structures of the solutions with both $\text{CFL} = 0.5$ and 2.

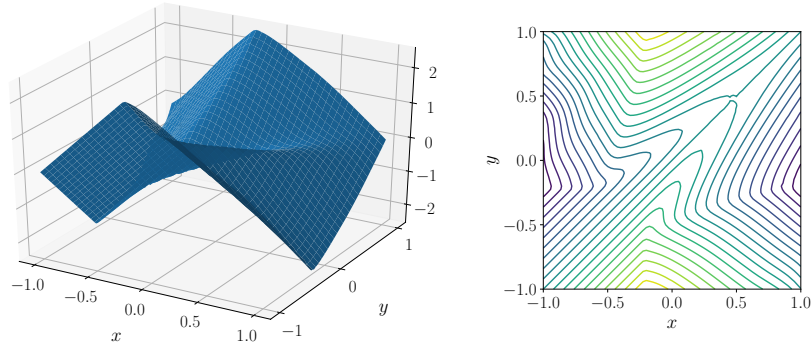
Example 4.8. In this example, we consider the two-dimensional Riemann problem with a non-convex



(a) Proposed ($\text{CFL} = 0.5$)



(b) Proposed ($\text{CFL} = 2$)



(c) Previous ($\text{CFL} = 2$)

Figure 4.10: Numerical solutions' surfaces (left) and contours (right) for Example 4.8 at $T = 1$.

Hamiltonian:

$$\phi_t + \sin(\phi_x + \phi_y) = 0, \quad (4.11)$$

on the spatial domain $[-1, 1] \times [-1, 1]$, where outflow boundary conditions are imposed. Our initial data is represented by the non-smooth function

$$\phi(x, y, 0) = \pi(|y| - |x|).$$

We run the simulation using 80×80 grid points and track the solution up to time $T = 1$. Plots of the numerical solutions are provided in Figure 4.10. Note that when a CFL number of 2 is used, the proposed scheme reduces dissipation encountered by the previous scheme near the boundaries.

Example 4.9. The next problem is a prototypical model in geometric optics, which is a Cauchy problem for H-J equation with a non-convex Hamiltonian:

$$\begin{aligned} \phi_t + \sqrt{\phi_x^2 + \phi_y^2 + 1} &= 0, \\ \phi(x, y, 0) &= \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1) - 1, \end{aligned} \quad (4.12)$$

on a periodic domain $[0, 1] \times [0, 1]$. We approximate the solution using 60×60 grids up to a final time $T = 0.6$, during which the characteristics intersect. The surfaces and contour lines of the numerical solution are shown in Figure 4.11. We note that sharp and symmetric regions are well-maintained in the solutions.

Example 4.10. In this test problem, we change the sign of Hamiltonian and use the same initial function as the previous example to simulate a propagating surface:

$$\begin{aligned} \phi_t - \sqrt{\phi_x^2 + \phi_y^2 + 1} &= 0, \\ \phi(x, y, 0) &= 1 - \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1), \end{aligned} \quad (4.13)$$

on the periodic domain $[0, 1] \times [0, 1]$. As above, 60×60 grid points are used and the snapshots of numerical solutions at $t = 0, 0.3, 0.6$ and 0.9 are given in Figures 4.12. We have also included plots of the solutions, which do not use the nonlinear filters (see Figures 4.12(c)) to demonstrate their effect.

4.3 Examples with irregular grids

In this section, we apply the proposed scheme to several examples defined on irregular grids x and y in order to demonstrate the capabilities of the coordinate transformation $x = x(\xi), y = y(\eta)$ using uniform computational grids ξ and η . Here, we present the results with the time step

$$\Delta t = \frac{\text{CFL}}{(\alpha_x/\Delta\xi + \alpha_y/\Delta\eta)}, \quad (4.14)$$

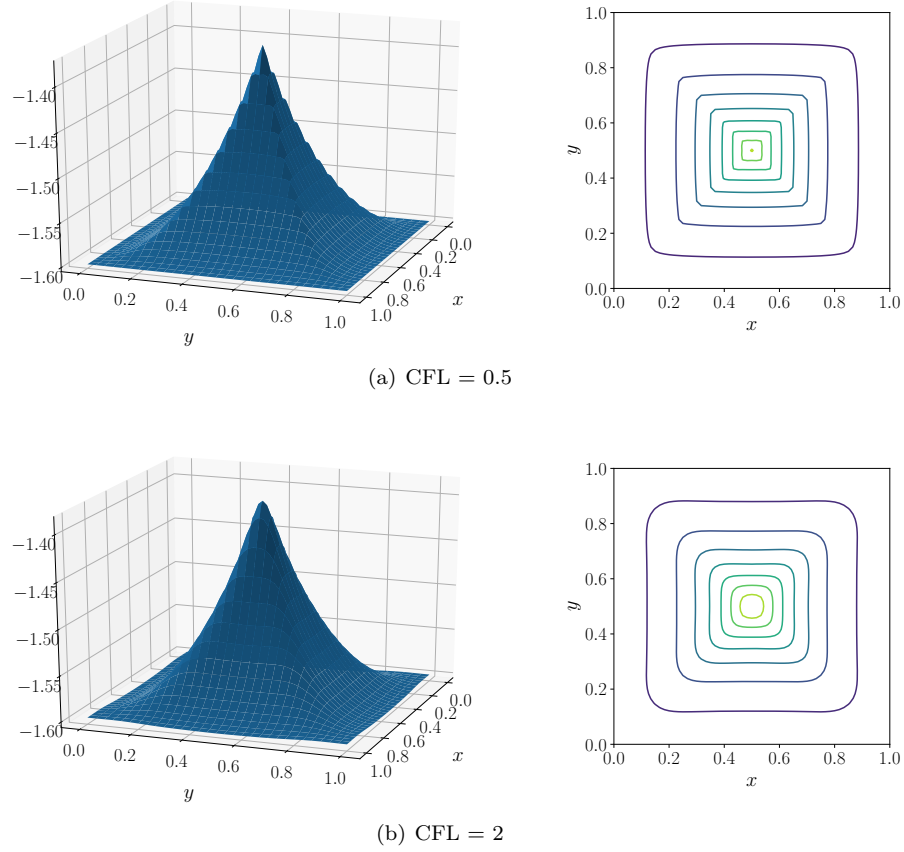
where α_x and α_y are the maximum wave propagation speeds in the x and y directions, respectively.

Example 4.11. We first consider the two-dimensional Burgers' equation

$$\begin{aligned} \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 &= 0, \\ \phi(x, y, 0) &= -\cos(\pi(x + y)/2), \end{aligned} \quad (4.15)$$

on the periodic domain $[-2, 2] \times [-2, 2]$. We generate a nonuniform mesh using random perturbations and compute the L_∞ errors and orders of accuracy at $T = 0.5/\pi^2$ while the solution is still smooth. In Table 4.5, we confirm the convergence rates of the mapped scheme on nonuniform meshes.

CFL	$N_x \times N_y$	$k = 1. \ \beta = 1.$		$k = 2. \ \beta = 0.5.$		$k = 3. \ \beta = 0.6.$	
		error	order	error	order	error	order
0.5	40×40	1.09e-01	—	6.43e-02	—	7.76e-03	—
	80×80	6.14e-02	0.835	1.66e-02	1.951	1.00e-03	2.952
	160×160	3.30e-02	0.895	4.47e-03	1.894	7.88e-05	3.669
	320×320	1.71e-02	0.945	1.19e-03	1.916	4.69e-06	4.069
	640×640	8.72e-03	0.975	3.05e-04	1.959	3.36e-07	3.802
1	40×40	1.73e-01	—	1.98e-01	—	4.67e-02	—
	80×80	1.09e-01	0.657	6.45e-02	1.621	7.72e-03	2.598
	160×160	6.15e-02	0.831	1.66e-02	1.961	9.92e-04	2.960
	320×320	3.30e-02	0.898	4.46e-03	1.893	7.13e-05	3.799
	640×640	1.71e-02	0.945	1.18e-03	1.915	6.97e-06	3.354
2	40×40	2.16e-01	—	3.21e-01	—	1.73e-01	—
	80×80	1.72e-01	0.327	1.98e-01	0.696	4.68e-02	1.888
	160×160	1.10e-01	0.653	6.45e-02	1.621	7.72e-03	2.600
	320×320	6.15e-02	0.835	1.66e-02	1.962	1.00e-03	2.948
	640×640	3.30e-02	0.897	4.46e-03	1.894	7.62e-05	3.716

Table 4.5: L_∞ -errors and orders of accuracy for Example 4.11 at $T = 0.5/\pi^2$.Figure 4.11: Numerical solutions' surfaces (left) and contours (right) for Example 4.9 at $T = 0.6$.

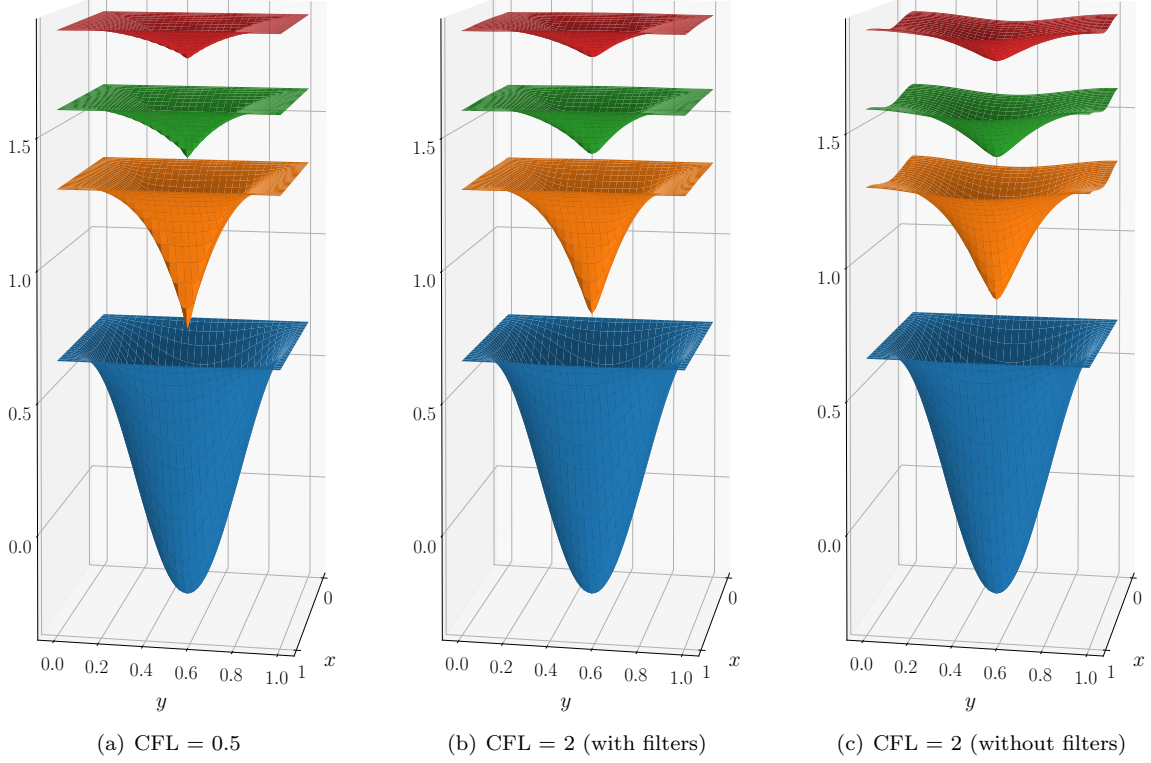


Figure 4.12: Propagating numerical solutions' surfaces for Example 4.10 at $T = 0, 0.3, 0.6$, and 0.9

Example 4.12. We solve the two-dimensional Riemann problem with a non-convex Hamiltonian

$$\begin{aligned} \phi_t + \sin(\phi_x + \phi_y) &= 0, \quad -1 \leq x, y \leq 1 \\ \phi(x, y, 0) &= \pi(|y| - |x|) \end{aligned} \quad (4.16)$$

which we considered in Example 4.8. To see the efficiency of the scheme, we construct a nonuniform mesh consisting of 60×60 grid points using a geometric series, selecting the ratio between the smallest cell size and the biggest cell size to be $1 : 7$. The resulting mesh is displayed in Figure 4.13(a). On this nontrivial grid, we plot the numerical solutions' surfaces and contour lines at time $T = 1$ in Figure 4.13.

Example 4.13. We consider the same problem of a propagating surface (4.13) in Example 4.10 with the initial condition

$$\phi(x, y, 0) = \sin\left(\frac{\pi}{2}(x^2 + y^2)\right)$$

defined on the unit disk $x^2 + y^2 \leq 1$ where the Dirichlet boundary condition

$$\phi(x, y, t) = 1 + t \quad \text{for all } x^2 + y^2 = 1$$

is imposed. The domain is discretized by embedding the boundary in a regular 60×60 Cartesian mesh so that the irregular spacing occurs only near the boundary. The discretization of the domain is illustrated in Figure 4.14(a). Blue and Red dots in the Figure indicate x and y directional boundary points, respectively. Snapshots of the propagating surface taken at $T = 0, 0.6$ and 1.2 are given in Figure 4.14.

Example 4.14. As our last example on nonuniform meshes, we consider the “level set reinitialization” equation [33]

$$\phi_t + \text{sign}(\phi_0)(\sqrt{\phi_x^2 + \phi_y^2} - 1) = 0, \quad (4.17)$$

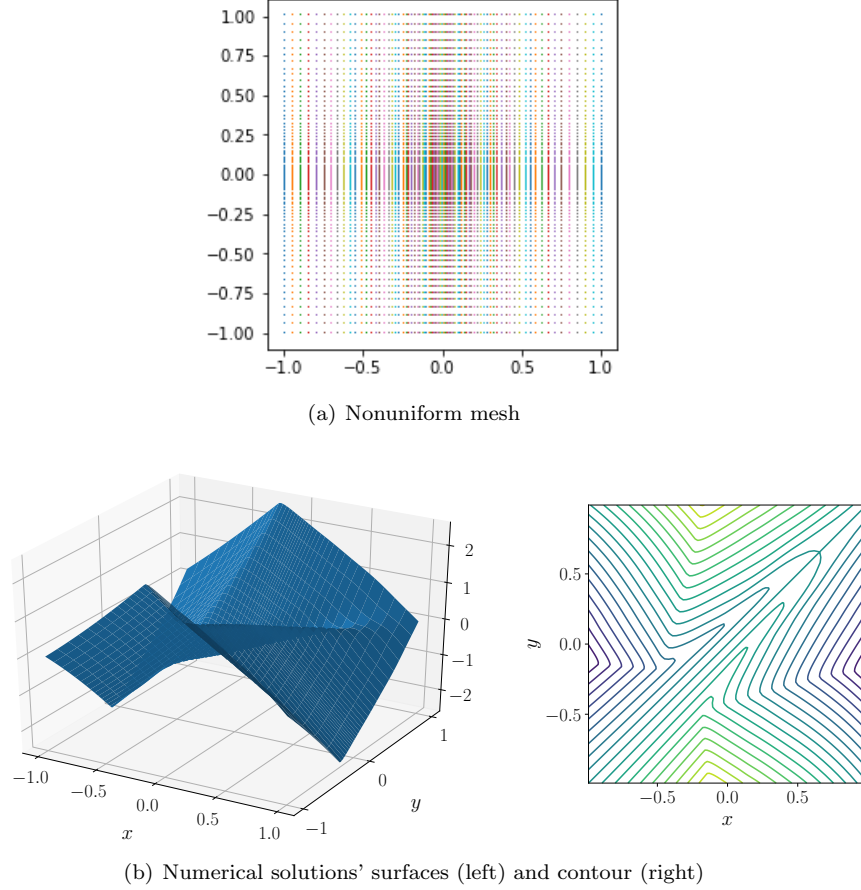


Figure 4.13: Numerical solutions on nonuniform meshes for Example 4.12 at $T = 1$.

on the circular domain $\frac{1}{2} < \sqrt{x^2 + y^2} < 1$. We choose an initial function with the signed distance function to the circle centered at the origin:

$$\phi(x, y, 0) = \phi_0(x, y) = \sqrt{x^2 + y^2} - \frac{1}{2}. \quad (4.18)$$

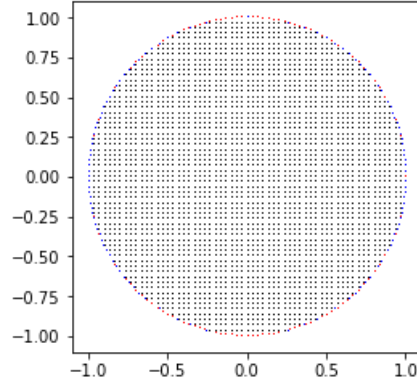
The hole in the circular domain is discretized by, again, embedding the boundary in a regular Cartesian mesh with 60×60 grid points. In Figure 4.15, we plot the resulting surface of the numerical solution at time $T = 1$ on the mesh.

5 Conclusion

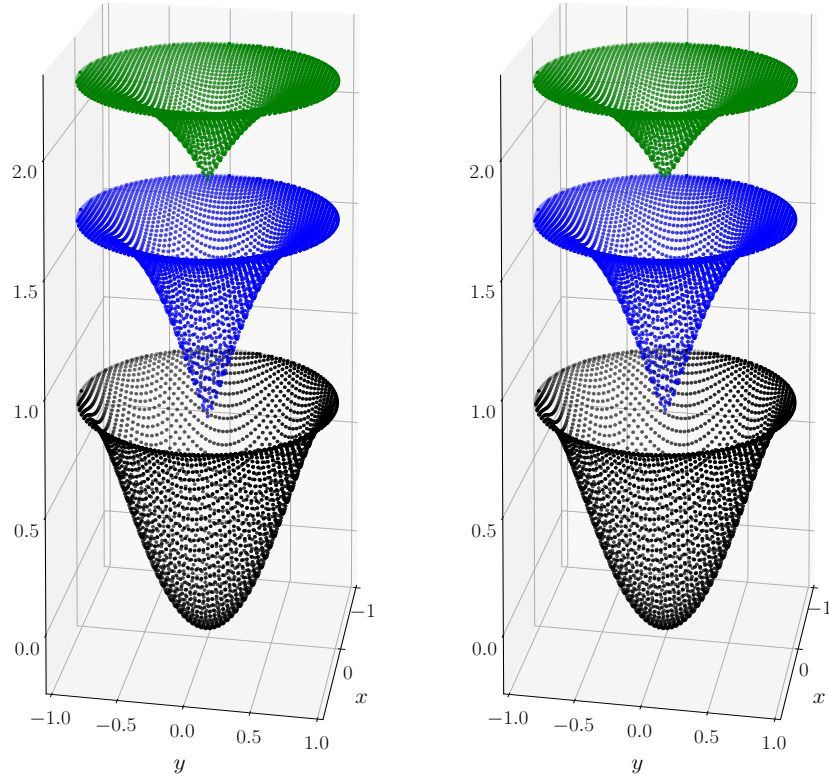
In this paper, we proposed a class of high order unconditionally stable schemes to solve the Hamilton-Jacobi equation. By leveraging a coordinate transformation, we implemented these schemes on two and three-dimensional nonuniform meshes to compute numerical solutions on domains containing non-trivial geometry. Our proposed scheme also makes use of an exponential basis to construct a novel WENO quadrature with improved shock-capturing capabilities. The proposed schemes show significant improvement compared to our previous work [10, 11] on all examples tried in this work.

As previously mentioned, one of the advantages of using exponential polynomials (e.g., $e^{\lambda x}$) is that they can be tuned according to characteristics of the given data by changing the “tension” parameter λ ; however, a proper selection of this parameter is not the focus of this paper. For our future study, we would like to

develop a WENO scheme capable of choosing the optimal parameters using local properties of the underlying data.



(a) The discretization of the domain



(b) Propagating numerical solutions with $CFL = 0.5$ (Left) and $CFL = 2$ (Right)

Figure 4.14: Propagating numerical solutions at $T=0, 0.6$ and 1.2 on nontrivial meshes for Example 4.13

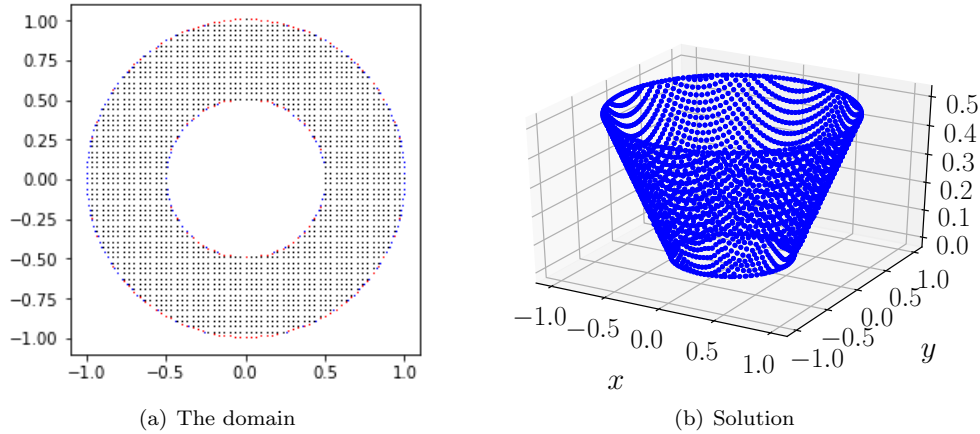


Figure 4.15: (a) The discretization of the domain which is embedded in a Cartesian grid. Blue and Red dots indicate x and y directional boundary points, respectively. (b) Surface of the solution for Example 4.14.

References

- [1] O. P. Bruno and M. Lyon. High-order unconditionally stable fc-ad solvers for general smooth domains i. basic elements. *Journal of Computational Physics*, 229:2003–2033, 2010.
- [2] M. Causley, H. Cho, and A. Christlieb. Method of lines transpose: Energy gradient flows using direct operator inversion for phase-field models. *SIAM Journal on Scientific Computing*, 39(5):B968–B992, 2017.
- [3] M. Causley, A. Christlieb, B. Ong, and L. Van Groningen. Method of lines transpose: An implicit solution to the wave equation. *Mathematics of Computation*, 83(290):2763–2786, 2014.
- [4] M. F. Causley, H. Cho, A. J. Christlieb, and D. C. Seal. Method of lines transpose: High order L-stable $\mathcal{O}(N)$ schemes for parabolic equations using successive convolution. *SIAM Journal on Numerical Analysis*, 54(3):1635–1652, 2016.
- [5] M. F. Causley, A. J. Christlieb, Y. Guclu, and E. Wolf. Method of lines transpose: A fast implicit wave propagator. *arXiv preprint arXiv:1306.6902*, 2013.
- [6] Y. Cheng, A. J. Christlieb, W. Guo, and B. Ong. An asymptotic preserving Maxwell solver resulting in the darwin limit of electrodynamics. *Journal of Scientific Computing*, 71(3):959–993, 2017.
- [7] Y. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for directly solving the Hamilton–Jacobi equations. *Journal of Computational Physics*, 223(1):398–415, 2007.
- [8] Y. Cheng and Z. Wang. A new discontinuous Galerkin finite element method for directly solving the Hamilton–Jacobi equations. *Journal of Computational Physics*, 268:134–153, 2014.
- [9] A. Christlieb, W. Guo, and Y. Jiang. A WENO-based Method of Lines Transpose approach for Vlasov simulations. *Journal of Computational Physics*, 327:337–367, 2016.
- [10] A. Christlieb, W. Guo, and Y. Jiang. Kernel based high order “explicit” unconditionally-stable scheme for nonlinear degenerate advection-diffusion equations. *arXiv preprint arXiv:1707.09294*, 2017.
- [11] A. Christlieb, W. Guo, and Y. Jiang. A kernel based high order explicit unconditionally stable scheme for time dependent Hamilton–Jacobi equations. *Journal of Computational Physics*, 379:214–236, 2019.
- [12] A. Christlieb, W. Sands, and H. Yang. <https://github.com/hyoseonyang/molt-tutorial.git>, May 2019.

- [13] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502, 1984.
- [14] M. G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.
- [15] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43(167):1–19, 1984.
- [16] Z. Gimbutas and V. Rokhlin. Generalized fast multipole method for nonoscillatory kernels. *SIAM Journal on Scientific Computing*, pages=796–817, year=2002, publisher=SIAM, 24.
- [17] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.
- [18] Y. Ha, C. H. Kim, H. Yang, and J. Yoon. Sixth-order weighted essentially nonoscillatory schemes based on exponential polynomials. *SIAM Journal on Scientific Computing*, 38:1987–2017, 2016.
- [19] Y. Ha, C. H. Kim, H. Yang, and J. Yoon. A sixth-order weighted essentially non-oscillatory schemes based on exponential polynomials for hamilton-jacobi equations. *Journal of Scientific Computing*, 75:16751700, 2018.
- [20] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM Journal on Scientific computing*, 21(2):666–690, 1999.
- [21] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM Journal on Scientific computing*, 21(6):2126–2143, 2000.
- [22] S. Karlin and W. Studden. *Tchebycheff Systems: With Applications in Analysis and Statistics*. Interscience Publishers, 1966.
- [23] M. C. A. Kropinski and B. D. Quaife. Fast integral equation methods for the modified helmholtz equation. *Journal of Computational Physics*, 230:425–434, 2011.
- [24] O. Lepsky, C. Hu, and C.-W. Shu. Analysis of the discontinuous Galerkin method for Hamilton–Jacobi equations. *Applied Numerical Mathematics*, 33(1-4):423–434, 2000.
- [25] M. Lyon and O. P. Bruno. High-order unconditionally stable fc-ad solvers for general smooth domains ii. elliptic, parabolic and hyperbolic pdes; theoretical considerations. *Journal of Computational Physics*, 229:3358–3381, 2010.
- [26] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [27] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM Journal on numerical analysis*, 28(4):907–922, 1991.
- [28] J. Qiu. Hermite WENO schemes with lax-wendroff type time discretizations for Hamilton–Jacobi equations. *Journal of Computational Mathematics*, pages 131–144, 2007.
- [29] J. Qiu and C.-W. Shu. Hermite WENO schemes for Hamilton–Jacobi equations. *Journal of Computational Physics*, 204(1):82–99, 2005.
- [30] A. Salazar, M. Raydan, and A. Campo. Theoretical analysis of the exponential transversal method of lines for the diffusion equation. *Numerical Methods for Partial Differential Equations*, 16(1):30–41, 2000.
- [31] M. Schemann and F. A. Bornemann. An adaptive rothe method for the wave equation. *Computing and Visualization in Science*, 1(3):137–144, 1998.

- [32] P. E. Souganidis. Approximation schemes for viscosity solutions of hamilton-jacobi equations. *Journal of Differential Equations*, 59:1–43, 1985.
- [33] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
- [34] W. Xu and W. Wu. An improved third-order weno-z scheme. *Journal of Scientific Computing*, 75(3):1808–1841, 2018.
- [35] J. Yan and S. Osher. A local discontinuous Galerkin method for directly solving Hamilton–Jacobi equations. *Journal of Computational Physics*, 230(1):232–244, 2011.
- [36] Y.-T. Zhang and C.-W. Shu. High-order WENO schemes for Hamilton–Jacobi equations on triangular meshes. *SIAM Journal on Scientific Computing*, 24(3):1005–1030, 2003.
- [37] F. Zheng, C.-W. Shu, and J. Qiu. Finite difference Hermite WENO schemes for the Hamilton–Jacobi equations. *Journal of Computational Physics*, 337:27–41, 2017.
- [38] J. Zhu and J. Qiu. Hermite WENO schemes for Hamilton–Jacobi equations on unstructured meshes. *Journal of Computational Physics*, 254:76–92, 2013.