

HIGH-ORDER CENTRAL WENO SCHEMES FOR MULTIDIMENSIONAL HAMILTON–JACOBI EQUATIONS*

STEVE BRYSON[†] AND DORON LEVY[‡]

Abstract. We present new third- and fifth-order Godunov-type central schemes for approximating solutions of the Hamilton–Jacobi (HJ) equation in an arbitrary number of space dimensions. These are the first central schemes for approximating solutions of the HJ equations with an order of accuracy that is greater than two. In two space dimensions we present two versions for the third-order scheme: one scheme that is based on a genuinely two-dimensional central weighted ENO reconstruction, and another scheme that is based on a simpler dimension-by-dimension reconstruction. The simpler dimension-by-dimension variant is then extended to a multidimensional fifth-order scheme. Our numerical examples in one, two, and three space dimensions verify the expected order of accuracy of the schemes.

Key words. Hamilton–Jacobi equations, central schemes, high order, WENO, CWENO

AMS subject classifications. Primary, 65M06; Secondary, 35L99

DOI. 10.1137/S0036142902408404

1. Introduction. We are interested in high-order numerical approximations for the solution of multidimensional Hamilton–Jacobi (HJ) equations of the form

$$\phi_t + H(\nabla\phi) = 0, \quad \vec{x} = (x_1, \dots, x_d) \in \mathbb{R}^d,$$

where H is the Hamiltonian, which we assume depends on $\nabla\phi$ and possibly on x and t . In recent years, the HJ equations have attracted a lot of attention from analysts and numerical analysts due to the important role that they play in applications such as optimal control theory, image processing, geometric optics, differential games, calculus of variations, etc. The main difficulty in treating these equations arises from the discontinuous derivatives that develop in finite time even when the initial data is smooth. Vanishing viscosity solutions provide a good tool for defining weak solutions when the Hamiltonian is convex [15]. The celebrated *viscosity solution* provides a suitable extension of weak solutions for more general Hamiltonians [3, 7, 8, 9, 10, 28, 29].

Given the importance of the HJ equations, there has been relatively little activity in developing numerical tools for approximating their solutions. This is surprising, given that most of the numerical ideas are based on the similarity between hyperbolic conservation laws and the HJ equations, and that the field of numerical methods for conservation laws has been flourishing in recent years.

Converging first-order approximations were introduced by Souganidis in [38]. High-order upwind methods were introduced by Osher and Sethian [34] and Osher and Shu [35]. These methods are based on Harten’s essentially nonoscillatory (ENO) reconstruction [13, 37], which is evolved in time with a monotone flux. The weighted

*Received by the editors May 28, 2002; accepted for publication (in revised form) February 16, 2003; published electronically August 22, 2003.

<http://www.siam.org/journals/sinum/41-4/40840.html>

[†]Program in Scientific Computing/Computational Mathematics, Stanford University and the NASA Advanced Supercomputing Division, NASA Ames Research Center, Moffett Field, CA 94035-1000 (bryson@nas.nasa.gov).

[‡]Department of Mathematics, Stanford University, Stanford, CA 94305-2125 (dlevy@math.stanford.edu). This author was supported in part by the National Science Foundation under Career Grant DMS-0133511.

ENO (WENO) interpolant of [18, 32, 36] was used for constructing high-order upwind methods for the HJ equations in [17], and extensions of these methods for triangular meshes were introduced in [1, 40]. We note in passing that there are other approaches for approximating solutions of HJ equations such as discontinuous Galerkin methods [14, 24] and relaxation schemes [20].

A different class of Godunov-type schemes for hyperbolic conservation laws, the so-called central schemes, has recently been applied to the HJ equations. The prototype for these schemes is the Lax–Friedrichs scheme [11]. A second-order staggered central scheme was developed for conservation laws by Nessyahu and Tadmor in [33]. The main advantage of central schemes is their simplicity. Since they do not require any (approximate) Riemann solvers, they are particularly suitable for approximating multidimensional systems of conservation laws. Lin and Tadmor applied these ideas to the HJ equations in [31]. There, first- and second-order staggered schemes versions of [2, 19, 33] were written in one and two space dimensions. An L^1 convergence of order one for this scheme was proved in [30]. After the introduction of a semidiscrete central scheme for hyperbolic conservation laws in [23], a second-order semidiscrete scheme for HJ equations was introduced by the same authors in [22]. While less dissipative, this scheme requires the estimation of the local speed of propagation at every grid-point, a task that is computationally intensive, particularly with problems of high dimensionality. By considering more precise information about the local speed of propagation, an even less dissipative scheme was generated in [21].

Recently we introduced in [5] new and efficient central schemes for multidimensional HJ equations. These nonoscillatory, nonstaggered schemes were first- and second-order accurate and were designed to scale well with an increasing dimension. Efficiency was obtained by carefully choosing the location of the evolution points and by using a one-dimensional projection step. Avoiding staggering by adding an additional projection step is an idea which we already utilized in the framework of conservation laws [16].

In this work we introduce third- and fifth-order accurate schemes for approximating solutions of multidimensional HJ equations. These are the *first* central schemes for such equations of order greater than two. This work is the HJ analogue to the corresponding works in conservation laws: an ENO-based central scheme [4] and the central WENO (CWENO) central schemes [25, 26, 27]. We announced a preliminary version of the one-dimensional results in a recent proceedings publication [6].

The structure of this paper is as follows. We start in section 2 with the derivation of our one-dimensional schemes. A third-order WENO reconstruction scheme is presented in section 2.2. This scheme requires a fourth-order reconstruction of the point-values and a third-order reconstruction of the derivatives at the evolution points. Even though the optimal location of the evolution points in one dimension is in the center of the interval, in order to prepare the grounds for the multidimensional schemes we write a reconstruction for an arbitrary location of the evolution points. A fifth-order method is then presented in section 2.3.

We turn to the multidimensional framework in section 3. Here there is flexibility in the reconstruction step. For simplicity we carry out most of the discussion in two space dimensions. Extensions to more than two space dimensions are presented in section 3.4. First, we provide a brief outline of the general structure of two-dimensional central schemes in section 3.1. The main remaining ingredient, the reconstruction step, is then described in the following two sections. For a two-dimensional third-order scheme we present in section 3.2 two ways to obtain a high-order reconstruction

of the approximate solution at the evolution points. The first option in section 3.2.1 is based on a genuinely two-dimensional reconstruction. An alternative dimension-by-dimension approach is based on a sequence of one-dimensional reconstructions and is presented in section 3.2.2. Our numerical results show that both approaches are essentially equivalent. Hence, the rest of the paper deals with the dimension-by-dimension reconstruction. A fifth-order dimension-by-dimension extension of the one-dimensional scheme in section 2.3 to two dimensions is then presented in section 3.3. Since the solution at the next time step is computed at grid-points that are different from those on which the data is given, we reproject the evolved solution back onto the original grid-points. Different ways to approach this reprojection step are discussed in section 3.2.3.

We conclude in section 4 with several numerical examples in one, two, and three space dimensions that confirm the expected order of accuracy and the high-resolution nature of our scheme. We compare our results with the scheme of Jiang and Peng [17]. We also study the convergence rate after the emergence of the discontinuities in the solution.

2. One-dimensional schemes.

2.1. One-dimensional central schemes. Consider the one-dimensional HJ equation of the form

$$(2.1) \quad \phi_t(x, t) + H(\phi_x) = 0, \quad x \in \mathbb{R}.$$

We are interested in approximating solutions of (2.1) subject to the initial data $\phi(x, t = 0) = \phi_0(x)$. For simplicity we assume a uniform grid in space and time with mesh spacings Δx and Δt , respectively. Denote the grid-points by $x_i = i\Delta x$, $t^n = n\Delta t$, and the fixed mesh ratio by $\lambda = \Delta t/\Delta x$. Let φ_i^n denote the approximate value of $\phi(x_i, t^n)$, and $(\varphi_x)_i^n$ denote the approximate value of the derivative $\phi_x(x_i, t^n)$. We define the forward and backward differencing as $\Delta^+ \varphi_i^n := \varphi_{i+1}^n - \varphi_i^n$ and $\Delta^- \varphi_i^n := \varphi_i^n - \varphi_{i-1}^n$.

Assume that the approximate solution at time t^n , φ_i^n is given. A Godunov-type scheme for approximating the solution of (2.1) starts with a continuous piecewise-polynomial $\tilde{\varphi}(x, t^n)$ that is reconstructed from the data φ_i^n :

$$(2.2) \quad \tilde{\varphi}(x, t^n) = \sum_i P_{i+\frac{1}{2}}(x, t^n) \chi_{i+\frac{1}{2}}(x).$$

Here, $\chi_{i+1/2}(x)$ is the characteristic function of the interval $[x_i, x_{i+1}]$, and $P_{i+1/2}(x, t^n)$ is a polynomial of a suitable degree that satisfies the interpolation requirements

$$P_{i+\frac{1}{2}}(x_{i+\beta}, t^n) = \varphi_{i+\beta}^n, \quad \beta = 0, 1.$$

The reconstruction (2.2) is then evolved from time t^n to time t^{n+1} according to (2.1) and is sampled at the half-integer grid-points $\{x_{i+1/2}\}$, where the reconstruction is smooth (as long as the CFL condition $\lambda |H'(\varphi_x)| \leq 1/2$ is satisfied):

$$(2.3) \quad \varphi_{i+\frac{1}{2}}^{n+1} = \varphi_{i+\frac{1}{2}}^n - \int_{t^n}^{t^{n+1}} H\left(\tilde{\varphi}_x\left(x_{i+\frac{1}{2}}, \tau\right)\right) d\tau.$$

The point-value $\varphi_{i+1/2}^n$ is obtained by sampling (2.2) at $x_{i+1/2}$; i.e., $\varphi_{i+1/2}^n = \tilde{\varphi}(x_{i+1/2}, t^n)$. Since the evolution step (2.3) is done at points where the solution is smooth, we can

approximate the time integral on the right-hand side (RHS) of (2.3) using a sufficiently accurate quadrature rule. For example, for a third- and fourth-order method, this integral can be replaced by a Simpson's quadrature,

$$(2.4) \quad \int_{t^n}^{t^{n+1}} H\left(\tilde{\varphi}_x\left(x_{i+\frac{1}{2}}, \tau\right)\right) d\tau \approx \frac{\Delta t}{6} \left[H\left(\varphi'_{i+\frac{1}{2}}{}^n\right) + 4H\left(\varphi'_{i+\frac{1}{2}}{}^{n+\frac{1}{2}}\right) + H\left(\varphi'_{i+\frac{1}{2}}{}^{n+1}\right) \right].$$

The derivative at time t^n , $\varphi'_{i+\frac{1}{2}}{}^n$, is obtained by sampling the derivative of the reconstruction (2.2), i.e., $\varphi'_{i+\frac{1}{2}}{}^n = \tilde{\varphi}'(x_{i+\frac{1}{2}}, t^n)$. The intermediate values of the derivative in time, $\varphi'_{i+\frac{1}{2}}{}^{n+\frac{1}{2}}$ and $\varphi'_{i+\frac{1}{2}}{}^{n+1}$, which are required in the quadrature (2.4), can be predicted using a Taylor expansion or with a Runge–Kutta (RK) method. Alternatively, (2.1) can be treated as a semidiscrete equation by replacing the spatial derivatives with their numerical approximations and integrating in time via an RK method.

The only remaining ingredient to specify is the reconstruction (2.2). Below we present two reconstructions. The first is a fourth-order reconstruction of the point-values and the derivatives, which leads to a third-order scheme, and the second is a sixth-order reconstruction that results in a fifth-order scheme.

Remarks.

1. In order to return to the original grid, we project $\varphi_{i+\frac{1}{2}}{}^{n+1}$ back onto the integer grid-points $\{x_i\}$ to end up with $\varphi_i{}^{n+1}$. This projection is accomplished with the same reconstruction used to approximate $\varphi_{i+\frac{1}{2}}{}^n$ from $\varphi_i{}^n$.

2. In order to maximize the size of the time step, the evolution points should be taken as far as possible from the singularities in the reconstructed piecewise polynomial. In one dimension the appropriate evolution point is located at $x_{i+\frac{1}{2}}$. In d dimensions with a uniform grid with spacing Δx , the optimal evolution points are located at $x_{i+\alpha} = x_i + \alpha\Delta x$ in each direction, where $\alpha = 1/(d + \sqrt{d})$ (see [5]). One of the multidimensional schemes we present in section 3 is based on one-dimensional reconstructions. Hence, in order to prepare the grounds for the multidimensional setup, we write the one-dimensional reconstruction in this section, assuming that the evolution points are $x_{i\pm\alpha}$. The reader should keep in mind that in one dimension, $\alpha = 1/2$.

3. We would like to point out that one does not need to fully reconstruct the polynomials $P_{i+\frac{1}{2}}(x, t^n)$. The only values that the scheme requires are the approximated point-values $\varphi_{i+\frac{1}{2}}{}^n = \tilde{\varphi}(x_{i+\frac{1}{2}}, t^n)$ and the approximated derivatives $\varphi'_{i+\frac{1}{2}} = \tilde{\varphi}'(x_{i+\frac{1}{2}})$. Hence, in the rest of the paper whenever we refer to reconstruction steps we directly treat the recovery of these two quantities.

2.2. A third-order scheme. A third-order scheme is generated by combining a third-order accurate ODE solver in time, for predicting the intermediate values of the derivatives in (2.4), with a sufficiently high-order reconstruction in space.

Given $\varphi_i{}^n$, in order to invoke (2.3), we should compute two quantities in every time step: the point-values at the evolution points, $\varphi_{i\pm\alpha}$, and the derivatives $\varphi'_{i\pm\alpha}$. In order to obtain a third-order scheme, the approximations of the point-values should be fourth-order accurate, and the approximation of the derivatives should be third-order accurate. In this scheme, the reconstruction of the point-values is done in locations that are staggered with respect to the location of the data. The reconstruction of the derivatives, which is required in every step of the ODE solver, is done at the same points where the data is given. Since we need two types of reconstructions and due to symmetry considerations, we derive a fourth-order approximation of the derivatives.

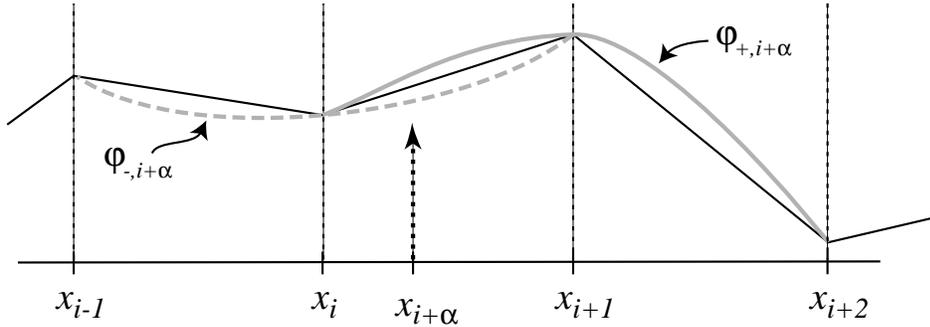


FIG. 2.1. The two interpolants used for the third-order reconstruction at the evolution point at $x_{i+\alpha}$.

Obviously, this more accurate reconstruction of the derivatives does not increase the order of accuracy of the scheme, but it does reduce the error.

2.2.1. The reconstruction of $\varphi_{i\pm\alpha}$ from φ_i . A fourth-order reconstruction of $\varphi_{i+\alpha}$ can be obtained by considering a convex combination of two quadratic polynomials, each of which requires the evaluation of φ on a three-point stencil. One quadratic polynomial $\varphi_-(x)$ is constructed on a stencil that is left-biased with respect to $x_{i+\alpha}$, $\{x_{i-1}, x_i, x_{i+1}\}$, while the other polynomial $\varphi_+(x)$ is constructed on a right-biased stencil, $\{x_i, x_{i+1}, x_{i+2}\}$; see Figure 2.1. We set

$$(2.5) \quad \begin{aligned} \varphi_{-,i+\alpha} &= \left(\frac{-\alpha + \alpha^2}{2}\right) \varphi_{i-1} + (1 - \alpha^2) \varphi_i + \left(\frac{\alpha + \alpha^2}{2}\right) \varphi_{i+1}, \\ \varphi_{+,i+\alpha} &= \left(\frac{2 - 3\alpha + \alpha^2}{2}\right) \varphi_i + (2\alpha - \alpha^2) \varphi_{i+1} + \left(\frac{-\alpha + \alpha^2}{2}\right) \varphi_{i+2}. \end{aligned}$$

For smooth φ , a straightforward computation shows that $\varphi_{\pm,i+\alpha} = \varphi(x_{i+\alpha}) + O(\Delta x^3)$ and

$$\frac{1}{3}(2 - \alpha) \varphi_{-,i+\alpha} + \frac{1}{3}(1 + \alpha) \varphi_{+,i+\alpha} = \varphi(x_{i+\alpha}) + O(\Delta x^4).$$

Similarly, the reconstruction of $\varphi_{i-\alpha}$ is obtained using the quadratic polynomials $\varphi_-(x)$ based on the left-biased stencil enclosing $x_{i-\alpha}$, $\{x_{i-2}, x_{i-1}, x_i\}$, and $\varphi_+(x)$ based on the right-biased stencil $\{x_{i-1}, x_i, x_{i+1}\}$:

$$(2.6) \quad \begin{aligned} \varphi_{-,i-\alpha} &= \left(\frac{-\alpha + \alpha^2}{2}\right) \varphi_{i-2} + (2\alpha - \alpha^2) \varphi_{i-1} + \left(\frac{2 - 3\alpha + \alpha^2}{2}\right) \varphi_i, \\ \varphi_{+,i-\alpha} &= \left(\frac{\alpha + \alpha^2}{2}\right) \varphi_{i-1} + (1 - \alpha^2) \varphi_i + \left(\frac{-\alpha + \alpha^2}{2}\right) \varphi_{i+1}. \end{aligned}$$

This time, $\varphi_{\pm,i-\alpha} = \varphi(x_{i-\alpha}) + O(\Delta x^3)$ and

$$\frac{1}{3}(1 + \alpha) \varphi_{-,i-\alpha} + \frac{1}{3}(2 - \alpha) \varphi_{+,i-\alpha} = \varphi(x_{i-\alpha}) + O(\Delta x^4).$$

A fourth-order WENO estimate of $\varphi_{i\pm\alpha}$ is therefore given by the convex combination

$$(2.7) \quad \varphi_{i\pm\alpha} = w_{i\pm\alpha}^- \varphi_{-,i\pm\alpha} + w_{i\pm\alpha}^+ \varphi_{+,i\pm\alpha},$$

where the weights satisfy $w_{i\pm\alpha}^- + w_{i\pm\alpha}^+ = 1$, $w_{i\pm\alpha}^\pm \geq 0$, $\forall i$. In smooth regions we would like to satisfy $w_{i+\alpha}^- = w_{i-\alpha}^+ \approx (2 - \alpha)/3$ and $w_{i+\alpha}^+ = w_{i-\alpha}^- \approx (1 + \alpha)/3$ to attain an $O(\Delta x^4)$ error. When the stencil supporting $\varphi_{i\pm\alpha}$ contains a discontinuity, the weight of the more oscillatory polynomial should vanish. Following [18, 32], these requirements are met by setting

$$(2.8) \quad w_{i\pm\alpha}^k = \frac{\alpha_{i\pm\alpha}^k}{\sum_l \alpha_{i\pm\alpha}^l}, \quad \alpha_{i\pm\alpha}^k = \frac{c_{i\pm\alpha}^k}{(\epsilon + S_{i\pm\alpha}^k)^p},$$

where $k, l \in \{+, -\}$. The constants are independent of the grid index i and are given by $c_{i+\alpha}^- = c_{i-\alpha}^+ = (2 - \alpha)/3$, $c_{i+\alpha}^+ = c_{i-\alpha}^- = (1 + \alpha)/3$. We choose ϵ as 10^{-6} to prevent the denominator in (2.8) from vanishing, and set $p = 2$ (see [18]). The smoothness measures S_i^\pm should be large when φ is nearly singular. Following [18], we take $S_{i\pm\alpha}$ to be the sum of the squares of the L^2 -norms of the derivatives on the stencil supporting φ_\pm . If we approximate the first derivative at x_i by $\Delta^+ \varphi_i / \Delta x$, the second derivative by $\Delta^+ \Delta^- \varphi_i / (\Delta x)^2$, and define the smoothness measure

$$(2.9) \quad S_i[r, s] = \Delta x \sum_{j=r}^s \left(\frac{1}{\Delta x} \Delta^+ \varphi_{i+j} \right)^2 + \Delta x \sum_{j=r+1}^s \left(\frac{1}{\Delta x^2} \Delta^+ \Delta^- \varphi_{i+j} \right)^2,$$

then we have $S_{i+\alpha}^- = S_i[-1, 0]$, $S_{i+\alpha}^+ = S_i[0, 1]$, $S_{i-\alpha}^- = S_i[-2, -1]$, and $S_{i-\alpha}^+ = S_i[-1, 0]$.

For future reference we label the reconstruction in this section with the procedural form

$$(2.10) \quad \varphi_{i\pm\alpha} = \text{reconstruct_}\varphi\text{-1D-3}(i, \pm\alpha, \varphi),$$

where φ is the one-dimensional array $(\varphi_1, \dots, \varphi_N)$. This notation will be used in the dimension-by-dimension reconstructions in section 3.

2.2.2. The reconstruction of $\varphi'_{i\pm\alpha}$ from $\varphi_{i\pm\alpha}$. The values of φ that we recovered in the previous step at the regularly spaced locations $\{x_{i\pm\alpha}\}$ can be used to recover the derivative $\varphi'_{i\pm\alpha}$ via a (noncentral) WENO reconstruction. To obtain a fourth-order WENO approximation of $\varphi'_{i\pm\alpha}$, we write a convex combination of three quadratic interpolants: $\varphi'_{-,i\pm\alpha}$ on the stencil $\{x_{i-2\pm\alpha}, x_{i-1\pm\alpha}, x_{i\pm\alpha}\}$, $\varphi'_{0,i\pm\alpha}$ on $\{x_{i-1\pm\alpha}, x_{i\pm\alpha}, x_{i+1\pm\alpha}\}$, and $\varphi'_{+,i\pm\alpha}$ on $\{x_{i\pm\alpha}, x_{i+1\pm\alpha}, x_{i+2\pm\alpha}\}$. For smooth φ ,

$$(2.11) \quad \begin{aligned} \varphi'_{-,i\pm\alpha} &= \frac{1}{2\Delta x}(\varphi_{i-2\pm\alpha} - 4\varphi_{i-1\pm\alpha} + 3\varphi_{i\pm\alpha}) = \varphi'(x_{i\pm\alpha}) + O(\Delta x^2), \\ \varphi'_{0,i\pm\alpha} &= \frac{1}{2\Delta x}(\varphi_{i+1\pm\alpha} - \varphi_{i-1\pm\alpha}) = \varphi'(x_{i\pm\alpha}) + O(\Delta x^2), \\ \varphi'_{+,i\pm\alpha} &= \frac{1}{2\Delta x}(-3\varphi_{i\pm\alpha} + 4\varphi_{i+1\pm\alpha} - \varphi_{i+2\pm\alpha}) = \varphi'(x_{i\pm\alpha}) + O(\Delta x^2). \end{aligned}$$

A straightforward computation yields

$$\frac{1}{6}\varphi'_{-,i\pm\alpha} + \frac{2}{3}\varphi'_{0,i\pm\alpha} + \frac{1}{6}\varphi'_{+,i\pm\alpha} = \varphi'(x_{i\pm\alpha}) + O(\Delta x^4).$$

The fourth-order WENO estimate of $\varphi'_{i\pm\alpha}$ from $\varphi_{i\pm\alpha}$ is therefore

$$(2.12) \quad \varphi'_{i\pm\alpha} = w_{i\pm\alpha}^- \varphi'_{-,i\pm\alpha} + w_{i\pm\alpha}^0 \varphi'_{0,i\pm\alpha} + w_{i\pm\alpha}^+ \varphi'_{+,i\pm\alpha},$$

where the weights w are of the form (2.8), with $k, l \in \{+, 0, -\}$, $c^- = c^+ = 1/6$, $c^0 = 2/3$, and the oscillatory indicators are $S_{i\pm\alpha}^- = S_{i\pm\alpha}[-2, -1]$, $S_{i\pm\alpha}^0 = S_{i\pm\alpha}[-1, 0]$, and $S_{i\pm\alpha}^+ = S_{i\pm\alpha}[0, 1]$.

For future reference we label the above reconstruction of $\varphi'_{i\pm\alpha}$ with the procedural form

$$(2.13) \quad \varphi'_{i\pm\alpha} = \text{reconstruct_}\varphi'_{-1D-3}(i, \pm\alpha, \varphi_{\pm\alpha}),$$

where $\varphi_{\pm\alpha}$ is the one-dimensional array $(\varphi_{1\pm\alpha}, \dots, \varphi_{N\pm\alpha})$.

We would like to summarize the one-dimensional third-order algorithm in the following, where $\text{RK}(\varphi_{i\pm\alpha}^n, \varphi'_{i\pm\alpha}^n, \Delta t)$ is the third-order Runge–Kutta method that integrates (2.1) and is used to predict the intermediate values of the derivatives. Each internal step of the RK method will require additional reconstructions of $\varphi'_{i\pm\alpha}$ from that step's $\varphi_{i\pm\alpha}$.

ALGORITHM 2.1. Assume that $\{\varphi_i^n\}$ are given.

(a) *Reconstruct:*

$$\begin{aligned} \varphi_{i\pm\alpha}^n &= \text{reconstruct_}\varphi_{-1D-3}(i, \pm\alpha, \varphi^n), \\ \varphi'_{i\pm\alpha}^n &= \text{reconstruct_}\varphi'_{-1D-3}(i, \pm\alpha, \varphi_{i\pm\alpha}^n). \end{aligned}$$

(b) *Integrate:*

$$\begin{aligned} \varphi_{i\pm\alpha}^{n+\frac{1}{2}} &= \text{RK}(\varphi_{i\pm\alpha}^n, \varphi'_{i\pm\alpha}^n, \Delta t/2), \\ \varphi'_{i\pm\alpha}^{n+\frac{1}{2}} &= \text{reconstruct_}\varphi'_{-1D-3}(i, \pm\alpha, \varphi_{i\pm\alpha}^{n+\frac{1}{2}}), \\ \varphi_{i\pm\alpha}^{n+1} &= \text{RK}(\varphi_{i\pm\alpha}^n, \varphi'_{i\pm\alpha}^n, \Delta t), \\ \varphi'_{i\pm\alpha}^{n+1} &= \text{reconstruct_}\varphi'_{-1D-3}(i, \pm\alpha, \varphi_{i\pm\alpha}^{n+1}), \\ \varphi_{i\pm\alpha}^{n+1} &= \varphi_{i\pm\alpha}^n + \frac{\Delta t}{6} \left[H(\varphi'_{i\pm\alpha}^n) + 4H(\varphi'_{i\pm\alpha}^{n+\frac{1}{2}}) + H(\varphi'_{i\pm\alpha}^{n+1}) \right]. \end{aligned}$$

(c) *Reproject:*

$$\varphi_i^{n+1} = \text{reconstruct_}\varphi_{-1D-3}(i, \mp\alpha, \varphi_{i\pm\alpha}^{n+1}).$$

Remark. It is possible to replace the Simpson’s quadrature in the integration step with a single RK time step, $\varphi_{i\pm\alpha}^{n+1} = \text{RK}(\varphi_{i\pm\alpha}^n, \varphi'_{i\pm\alpha}^n, \Delta t)$. Our simulations show that this choice reduces the complexity of the computation but also reduces its accuracy.

2.3. A fifth-order scheme. In order to obtain a fifth-order scheme, we need a sixth-order approximation of the point-values of φ , a fifth-order approximation of the derivative φ' , and a higher-order prediction of the intermediate derivatives which appear in the quadrature formula. Due to arguments similar to those given in section 2.2, we again derive a more accurate reconstruction of the derivatives, which in this case is sixth-order.

We start with the reconstruction of $\varphi_{i+\alpha}$ from φ_i . We write sixth-order interpolants as a convex combination of three cubic interpolants, each of which requires the evaluation of φ on a four-point stencil. We use the polynomials $\varphi_-(x)$ defined on the left-biased stencil $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$, $\varphi_0(x)$ defined on the centered stencil $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, and $\varphi_+(x)$ defined on the right-biased stencil $\{x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$; see Figure 2.2. For smooth φ ,

$$(2.14) \quad \begin{aligned} \varphi_{-,i+\alpha} &= a_1\varphi_{i-2} + a_2\varphi_{i-1} + a_3\varphi_i + a_4\varphi_{i+1} = \varphi(x_{i+\alpha}) + O(\Delta x^4), \\ \varphi_{0,i+\alpha} &= a_5\varphi_{i-1} + a_6\varphi_i + a_7\varphi_{i+1} + a_8\varphi_{i+2} = \varphi(x_{i+\alpha}) + O(\Delta x^4), \\ \varphi_{+,i+\alpha} &= a_9\varphi_i + a_{10}\varphi_{i+1} + a_{11}\varphi_{i+2} + a_{12}\varphi_{i+3} = \varphi(x_{i+\alpha}) + O(\Delta x^4), \end{aligned}$$

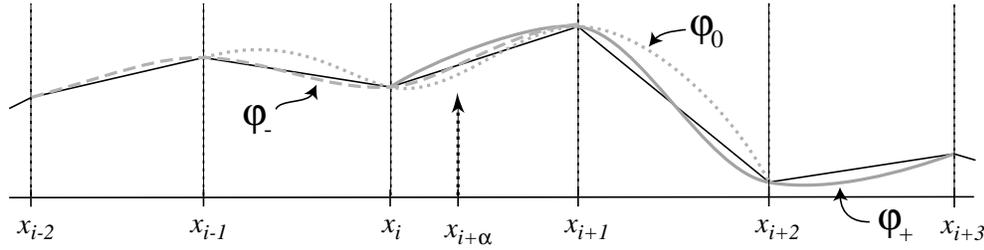


FIG. 2.2. The three interpolants used for the fifth-order reconstruction $\varphi_{i+\alpha}$ at the evolution point at $x_{i+\alpha}$. In this example, because of the large gradient between x_{i+1} and x_{i+2} , the interpolant φ_- will have the strongest contribution to the CWENO reconstruction at $x_{i+\alpha}$.

where the constants are given by

$$\begin{aligned} a_1 &= \frac{1}{6}\alpha - \frac{1}{6}\alpha^3, & a_2 &= -\alpha + \frac{1}{2}\alpha^2 + \frac{1}{2}\alpha^3, \\ a_3 &= 1 + \frac{1}{2}\alpha - \alpha^2 - \frac{1}{2}\alpha^3, & a_4 &= \frac{1}{3}\alpha + \frac{1}{2}\alpha^2 + \frac{1}{6}\alpha^3, \\ a_5 &= -\frac{1}{3}\alpha + \frac{1}{2}\alpha^2 - \frac{1}{6}\alpha^3, & a_6 &= 1 - \frac{1}{2}\alpha - \alpha^2 + \frac{1}{2}\alpha^3, \\ a_7 &= \alpha + \frac{1}{2}\alpha^2 - \frac{1}{2}\alpha^3, & a_8 &= -\frac{1}{6}\alpha + \frac{1}{6}\alpha^3 = -a_1, \\ a_9 &= 1 - \frac{11}{6}\alpha + \alpha^2 - \frac{1}{6}\alpha^3, & a_{10} &= 3\alpha - \frac{5}{2}\alpha^2 + \frac{1}{2}\alpha^3, \\ a_{11} &= -\frac{3}{2}\alpha + 2\alpha^2 - \frac{1}{2}\alpha^3, & a_{12} &= \frac{1}{3}\alpha - \frac{1}{2}\alpha^2 + \frac{1}{6}\alpha^3. \end{aligned}$$

At $x_{i-\alpha}$ we have

$$\begin{aligned} (2.15) \quad \varphi_{-,i-\alpha} &= a_{12}\varphi_{i-3} + a_{11}\varphi_{i-2} + a_{10}\varphi_{i-1} + a_9\varphi_i = \varphi(x_{i-\alpha}) + O(\Delta x^4), \\ \varphi_{0,i-\alpha} &= a_8\varphi_{i-2} + a_7\varphi_{i-1} + a_6\varphi_i + a_5\varphi_{i+1} = \varphi(x_{i-\alpha}) + O(\Delta x^4), \\ \varphi_{+,i-\alpha} &= a_4\varphi_{i-1} + a_3\varphi_i + a_2\varphi_{i+1} + a_1\varphi_{i+2} = \varphi(x_{i-\alpha}) + O(\Delta x^4). \end{aligned}$$

A straightforward computation yields

$$c_{i\pm\alpha}^- \varphi_{-,i\pm\alpha} + c_{i\pm\alpha}^0 \varphi_{0,i\pm\alpha} + c_{i\pm\alpha}^+ \varphi_{+,i\pm\alpha} = \varphi(x_{i\pm\alpha}) + O(\Delta x^6),$$

where

$$\begin{aligned} (2.16) \quad c_{i+\alpha}^- &= c_{i-\alpha}^+ = \frac{1}{20}\alpha^2 - \frac{1}{4}\alpha + \frac{3}{10}, \\ c_{i\pm\alpha}^0 &= -\frac{1}{10}\alpha^2 + \frac{1}{10}\alpha + \frac{3}{5}, \\ c_{i+\alpha}^+ &= c_{i-\alpha}^- = \frac{1}{20}\alpha^2 + \frac{3}{20}\alpha + \frac{1}{10}. \end{aligned}$$

A sixth-order reconstruction of $\varphi_{i\pm\alpha}$ is therefore given by

$$(2.17) \quad \varphi_{i\pm\alpha} = w_{i\pm\alpha}^- \varphi_{-,i\pm\alpha} + w_{i\pm\alpha}^0 \varphi_{0,i\pm\alpha} + w_{i\pm\alpha}^+ \varphi_{+,i\pm\alpha},$$

where the weights w^k are given by (2.8) with $k, l \in \{+, 0, -\}$, and the constants c^k are given by (2.16). The oscillatory indicators are given via (2.9) by $S_{i\pm\alpha}^- = S_i[-2, 0]$, $S_{i\pm\alpha}^0 = S_i[-1, 1]$, and $S_{i\pm\alpha}^+ = S_i[0, 2]$.

A sixth-order approximation of $\varphi'_{i\pm\alpha}$ from $\varphi_{i\pm\alpha}$ is written as a convex combination of four cubic interpolants. This reconstruction is similar to the third-order case and is based on a noncentral WENO reconstruction. We skip the details and summarize the result:

$$(2.18) \quad \varphi'_{i\pm\alpha} = w_{i\pm\alpha}^1 \varphi'_{1,i\pm\alpha} + w_{i\pm\alpha}^2 \varphi'_{2,i\pm\alpha} + w_{i\pm\alpha}^3 \varphi'_{3,i\pm\alpha} + w_{i\pm\alpha}^4 \varphi'_{4,i\pm\alpha},$$

where

$$\begin{aligned} \varphi'_{1,i\pm\alpha} &= \frac{1}{6\Delta x} (-2\varphi_{i-3\pm\alpha} + 9\varphi_{i-2\pm\alpha} - 18\varphi_{i-1\pm\alpha} + 11\varphi_{i\pm\alpha}), \\ \varphi'_{2,i\pm\alpha} &= \frac{1}{6\Delta x} (\varphi_{i-2\pm\alpha} - 6\varphi_{i-1\pm\alpha} + 3\varphi_{i\pm\alpha} + 2\varphi_{i+1\pm\alpha}), \\ \varphi'_{3,i\pm\alpha} &= \frac{1}{6\Delta x} (-2\varphi_{i-1\pm\alpha} - 3\varphi_{i\pm\alpha} + 6\varphi_{i+1\pm\alpha} - \varphi_{i+2\pm\alpha}), \\ \varphi'_{4,i\pm\alpha} &= \frac{1}{6\Delta x} (-11\varphi_{i\pm\alpha} + 18\varphi_{i+1\pm\alpha} - 9\varphi_{i+2\pm\alpha} + 2\varphi_{i+3\pm\alpha}). \end{aligned}$$

Here the weights w^k are given by (2.8) with $c_1 = c_4 = 1/20$, $c_2 = c_3 = 9/20$, $S_{i\pm\alpha}^1 = S_{i\pm\alpha}[-3, -1]$, $S_{i\pm\alpha}^2 = S_{i\pm\alpha}[-2, 0]$, $S_{i\pm\alpha}^3 = S_{i\pm\alpha}[-1, 1]$, and $S_{i\pm\alpha}^4 = S_{i\pm\alpha}[0, 2]$.

Notation.

1. We label the reconstruction of the point-values, (2.17), as

$$(2.19) \quad \varphi_{i\pm\alpha} = \text{reconstruct_}\varphi\text{-1D-5}(i, \pm\alpha, \varphi),$$

where φ is the one-dimensional array $(\varphi_1, \dots, \varphi_N)$.

2. We label the reconstruction of $\varphi'_{i\pm\alpha}$, (2.18), as

$$(2.20) \quad \varphi'_{i\pm\alpha} = \text{reconstruct_}\varphi'\text{-1D-5}(i, \pm\alpha, \varphi_{\pm\alpha}),$$

where $\varphi_{\pm\alpha}$ is the one-dimensional array $(\varphi_{1\pm\alpha}, \dots, \varphi_{N\pm\alpha})$.

Remarks.

1. To conclude, the fifth-order method is given by Algorithm 2.1, where the fourth-order reconstructions are replaced by the sixth-order reconstructions (2.19)–(2.20). As is, this scheme is only fourth-order in time. A higher-order method in time can be easily obtained by replacing Simpson’s quadrature with a more accurate quadrature and computing the sixth-order approximations for the point-values and the derivatives at the new quadrature points.

2. We choose to predict the intermediate values of the derivatives in time using the fourth-order strong stability preserving (SSP) RK scheme of [12]. For $s \in \{\frac{1}{2}, 1\}$, the SSP-RK scheme is given by

$$\begin{aligned} \varphi^{(1)} &= \varphi^n - \frac{1}{2}s\Delta tH(\varphi_x^n), \\ \varphi^{(2)} &= \frac{649}{1600}\varphi^n + \frac{10890423}{25193600}s\Delta tH(\varphi_x^n) + \frac{951}{1600}\varphi^{(1)} - \frac{5000}{7873}s\Delta tH(\varphi_x^{(1)}), \end{aligned}$$

$$\begin{aligned} \varphi^{(3)} &= \frac{53989}{2500000} \varphi^n + \frac{102261}{5000000} s \Delta t H(\varphi_x^n) + \frac{4806213}{20000000} \varphi^{(1)} \\ &\quad + \frac{5121}{20000} s \Delta t H(\varphi_x^{(1)}) + \frac{23619}{32000} \varphi^{(2)} + \frac{7873}{10000} s \Delta t H(\varphi_x^{(2)}), \\ \varphi^{n+s} &= \frac{1}{5} \varphi^n - \frac{1}{10} s \Delta t H(\varphi_x^n) + \frac{6127}{30000} \varphi^{(1)} + \frac{1}{6} s \Delta t H(\varphi_x^{(1)}) + \frac{7873}{30000} \varphi^{(2)} \\ &\quad + \frac{1}{3} \varphi^{(3)} - \frac{1}{6} s \Delta t H(\varphi_x^{(3)}). \end{aligned}$$

Alternatively, the natural continuous extension of the RK method [39] can be used to produce the intermediate values $\varphi'^{n+\frac{1}{2}}$ and φ'^{n+1} with a single RK step, though we observe that errors are somewhat larger in this case.

3. Multidimensional schemes.

3.1. Two-dimensional central schemes. Consider the two-dimensional HJ equation of the form

$$(3.1) \quad \phi_t + H(\nabla \phi) = 0, \quad \vec{x} = (x_1, x_2) \in \mathbb{R}^2,$$

subject to the initial data $\phi(\vec{x}, t = 0) = \phi_0(\vec{x})$. Define $x_{i,j} := (x_1 + i\Delta x_1, x_2 + j\Delta x_2)$. Similarly to the one-dimensional setup, $\varphi_{i,j}$ will denote the approximation of ϕ at $x_{i,j}$. We define the two sets of grid-points, $I_+ = \{x_{i,j}, x_{i+1,j}, x_{i,j+1}\}$ and $I_- = \{x_{i,j}, x_{i-1,j}, x_{i,j-1}\}$, and denote by T_+, T_- the triangles with vertices I_+ and I_- , respectively. For simplicity we assume a uniform grid $\Delta x_1 = \Delta x_2 = \Delta x$.

Assume that the approximate solution at time t^n , $\varphi_{i,j}^n$, is given. Similarly to the one-dimensional setup in section 2.1, a Godunov-type scheme for approximating the solution of (3.1) starts with a continuous piecewise polynomial $\tilde{\varphi}(\vec{x}, t^n)$ that is reconstructed from the data $\varphi_{i,j}^n$,

$$(3.2) \quad \tilde{\varphi}(\vec{x}, t^n) = \sum_{i,j} P_{i,j}^{T_{\pm}}(\vec{x}, t^n) \chi_{T_{\pm}}(\vec{x}).$$

As usual, $\chi_{T_{\pm}}(\vec{x})$ is the characteristic function of the triangle T_{\pm} , and $P_{i,j}^{T_{\pm}}(\vec{x}, t^n)$ is a polynomial of a suitable degree that satisfies the interpolation requirements

$$P_{i,j}^{T_{\pm}}(\vec{x}_l, t^n) = \varphi(\vec{x}_l, t^n), \quad \vec{x}_l \in I_{\pm}$$

(see Figure 3.1). The reconstruction (3.2) is then evolved from time t^n to time t^{n+1} by (3.1) and sampled at the evolution points $\{x_{i\pm\alpha, j\pm\alpha}\}$. In two dimensions the choice $\alpha = 1/(2 + \sqrt{2})$ guarantees that the solution remains smooth at the evolution point as long as the CFL condition $\frac{\Delta t}{\Delta x} |H'(\nabla \varphi)| < \alpha$ is satisfied. The evolved solution now reads

$$(3.3) \quad \varphi_{i\pm\alpha, j\pm\alpha}^{n+1} = \varphi_{i\pm\alpha, j\pm\alpha}^n - \int_{t^n}^{t^{n+1}} H(\nabla \tilde{\varphi}(x_{i\pm\alpha, j\pm\alpha}, \tau)) d\tau.$$

The point-values $\varphi_{i\pm\alpha, j\pm\alpha}^n$ are obtained by sampling (3.2) at $x_{i\pm\alpha, j\pm\alpha}$, i.e., $\varphi_{i\pm\alpha, j\pm\alpha}^n = \tilde{\varphi}(x_{i\pm\alpha, j\pm\alpha}, t^n)$. As in the one-dimensional case, the evolution points are in smooth regions, and therefore the integral on the RHS of (3.3) can be replaced with a sufficiently accurate quadrature such as the Simpson rule (2.4), which leads to a scheme that is

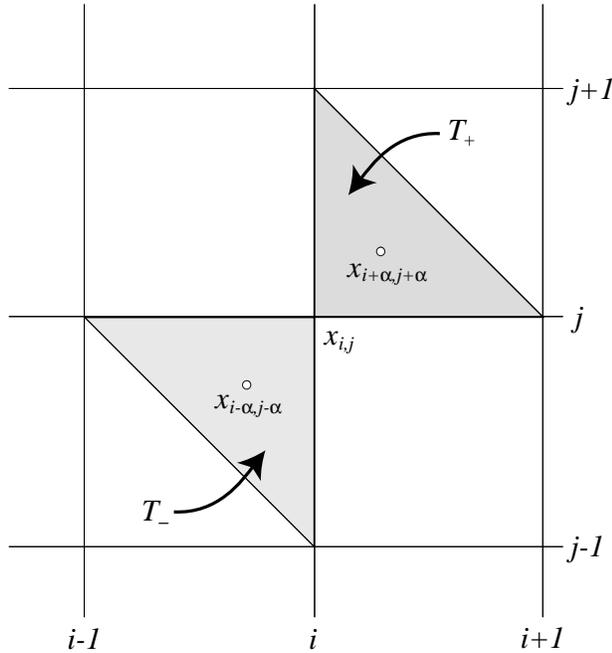


FIG. 3.1. The location of the evolution points $x_{i\pm\alpha, j\pm\alpha}$ and the domain of definition of the interpolants $\varphi_{i\pm\alpha, j\pm\alpha}$ in two dimensions.

fourth-order accurate in time. The derivatives at time t^n , $\varphi'_{i\pm\alpha, j\pm\alpha}$, are obtained by sampling the derivative of the reconstruction (3.2), i.e., $\varphi'_{i\pm\alpha, j\pm\alpha} = \tilde{\varphi}'(x_{i\pm\alpha, j\pm\alpha}, t^n)$. The other intermediate values of the derivative in time that are required in the quadrature can be predicted using a Taylor expansion or with a RK method in a way analogous to that for the one-dimensional case.

Remarks.

1. We present two different algorithms for constructing $\varphi_{i\pm\alpha, j\pm\alpha}$: two-dimensional interpolants defined on two-dimensional stencils and a dimension-by-dimension approach. We present both algorithms for the third-order scheme and extend the simpler dimension-by-dimension approach to fifth-order. Our numerical simulations in section 4 indicate that both reconstructions of $\varphi_{i\pm\alpha, j\pm\alpha}$ are of a comparable quality. In both approaches, the reconstruction of the derivatives $\nabla\varphi_{i\pm\alpha, j\pm\alpha}$ is done dimension-by-dimension.

2. We reproject $\varphi_{i+\alpha, j+\alpha}^{n+1}$ and $\varphi_{i-\alpha, j-\alpha}^{n+1}$ back onto the integer grid-points, obtaining $\varphi_{i,j}^{n+1}$. We present several ways to carry out this reprojectioin: a genuinely two-dimensional approach, a dimension-by-dimension strategy, and a reprojectioin along the diagonal line through $x_{i-\alpha, j-\alpha}$ and $x_{i+\alpha, j+\alpha}$.

3.2. Two-dimensional third-order schemes. In order to obtain a third-order scheme, we need a fourth-order reconstruction of the point-values at the evolution points $x_{i\pm\alpha, j\pm\alpha}$.

3.2.1. A two-dimensional reconstruction of $\varphi_{i\pm\alpha, j\pm\alpha}$. In this section we present a two-dimensional fourth-order reconstruction of the point-values $\varphi_{i\pm\alpha, j\pm\alpha}$. In principle, a two-dimensional cubic interpolant would provide a reconstruction with the desired accuracy. Such an interpolant is based on a ten-point stencil. As usual,

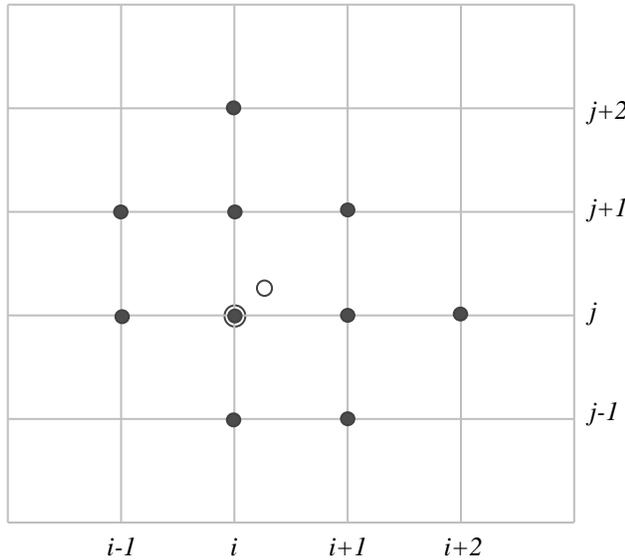


FIG. 3.2. The ten-point stencil for the two-dimensional reconstruction of $\varphi_{i+\alpha, j+\alpha}$. The open circle shows the location of the evolution point at $x_{i+\alpha, j+\alpha}$.

solving such a direct interpolation problem is unsatisfactory because spurious oscillations might develop as a result of the lack of smoothness in the solution. Instead, we generate a two-dimensional fourth-order reconstruction as a convex combination of four quadratic interpolants, each of which is based on a six-point stencil. We choose compact quadratic interpolants such that the union of all the six-point stencils is a compact ten-point stencil. Similarly to any WENO-type reconstruction, when singularities are present the six-point stencils containing the singularities are suppressed. In any case, we implicitly assume that the solution is sufficiently resolved such that the singularities in the solution are isolated in the sense that they do not occur along neighboring parallel cell edges. Singularities will in general occur along adjacent cell edges. There is a lot of flexibility in choosing the ten-point stencil as well as the different six-point stencils. Here, for the evolution point $x_{i+\alpha, j+\alpha}$ we choose the ten-point stencil shown in Figure 3.2. We also choose to use the four six-point stencils that are shown in Figure 3.3; obviously, the union of these stencils is the ten-point stencil in Figure 3.2. Furthermore, they all enclose the cell containing the evolution point, and they all cross different edges of the enclosing cell. A singularity along an edge will suppress two of these stencils, while a singularity in a corner will suppress three of these stencils.

Remarks.

1. The stencils for the evolution point at $x_{i-\alpha, j-\alpha}$ are obtained by a rotation of 180 degrees of the stencils in Figures 3.2–3.3.

2. We could use fewer than four stencils and still generate a scheme that will have the desired order of accuracy.

Given the four six-point stencils in Figure 3.3, a straightforward computation shows that third-order approximations for smooth φ at the evolution points $x_{i\pm\alpha, j\pm\alpha}$, $\varphi_{i\pm\alpha, j\pm\alpha}^k = \varphi(x_{i\pm\alpha}, y_{j\pm\alpha}) + O(\Delta x^3, \Delta y^3) \forall k \in \{1, 2, 3, 4\}$ are obtained with

$$(3.4) \quad \varphi_{i\pm\alpha, j\pm\alpha}^1 = a_1\varphi_{i, j} + a_2\varphi_{i\pm 1, j} + a_2\varphi_{i, j\pm 1} + a_3\varphi_{i\pm 1, j\pm 1} + a_4\varphi_{i\pm 2, j} + a_4\varphi_{i, j\pm 2},$$

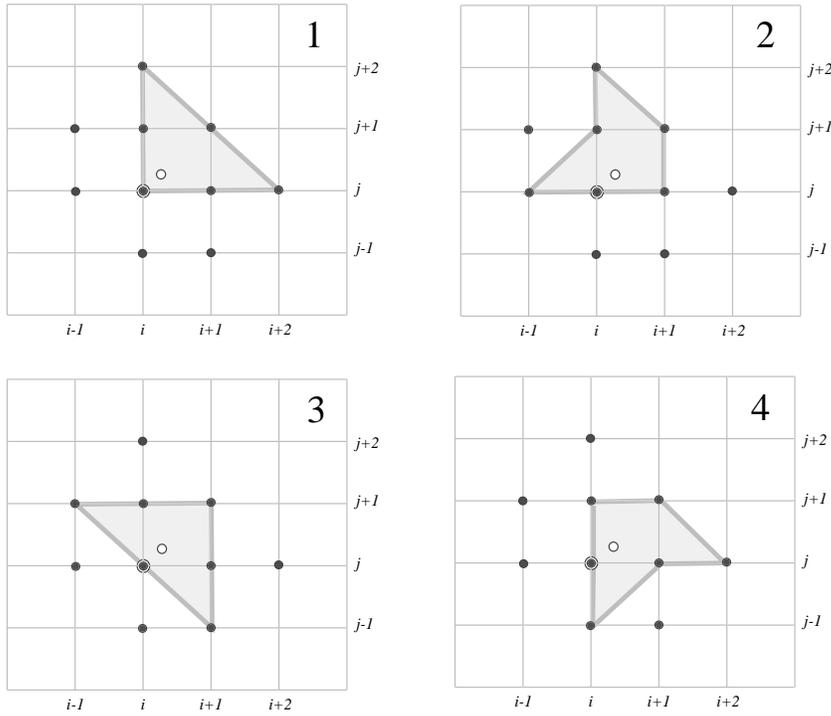


FIG. 3.3. The four six-point stencils that cover the ten-point stencil for the two-dimensional reconstruction.

$$\begin{aligned} \varphi_{i\pm\alpha, j\pm\alpha}^2 &= a_5\varphi_{i,j} + a_6\varphi_{i\pm 1,j} + a_2\varphi_{i,j\pm 1} + a_3\varphi_{i\pm 1,j\pm 1} + a_4\varphi_{i,j\pm 2} + a_4\varphi_{i\mp 1,j}, \\ \varphi_{i\pm\alpha, j\pm\alpha}^3 &= a_7\varphi_{i,j} + a_2\varphi_{i\pm 1,j} + a_2\varphi_{i,j\pm 1} + a_8\varphi_{i\pm 1,j\pm 1} + a_4\varphi_{i\pm 1,j\mp 1} + a_4\varphi_{i\mp 1,j\pm 1}, \\ \varphi_{i\pm\alpha, j\pm\alpha}^4 &= a_5\varphi_{i,j} + a_2\varphi_{i\pm 1,j} + a_6\varphi_{i,j\pm 1} + a_3\varphi_{i\pm 1,j\pm 1} + a_4\varphi_{i\pm 2,j} + a_4\varphi_{i,j\mp 1}, \end{aligned}$$

where

$$\begin{aligned} (3.5) \quad a_1 &= 1 - 3\alpha + 2\alpha^2, & a_2 &= 2\alpha - 2\alpha^2, & a_3 &= \alpha^2, \\ a_4 &= -\frac{1}{2}\alpha + \frac{1}{2}\alpha^2, & a_5 &= 1 - \frac{3}{2}\alpha + \frac{1}{2}\alpha^2, & a_6 &= \frac{1}{2}\alpha - \frac{1}{2}\alpha^2, \\ a_7 &= 1 - 2\alpha + \alpha^2, & a_8 &= -\alpha + 2\alpha^2. \end{aligned}$$

The linear combination

$$\sum_{k=1}^4 c_k \varphi_{i\pm\alpha, j\pm\alpha}^k = \varphi(x_{i\pm\alpha}, y_{j\pm\alpha}) + O(\Delta x^4, \Delta y^4)$$

is fourth-order accurate, provided that the constants c_i are taken as

$$(3.6) \quad c_1 = \frac{1}{3}(5\alpha - 1), \quad c_2 = c_4 = \frac{2}{3}(-2\alpha + 1), \quad c_3 = \alpha.$$

A two-dimensional CWENO reconstruction is a straightforward generalization of

the one-dimensional case (compare with (2.7), (2.8)):

$$\varphi_{i\pm\alpha,j\pm\alpha} = \sum_{k=1}^4 w_{i\pm\alpha,j\pm\alpha}^k \varphi_{i\pm\alpha,j\pm\alpha}^k.$$

Here

$$w_{i\pm\alpha,j\pm\alpha}^k = \frac{\alpha_{i\pm\alpha,j\pm\alpha}^k}{\sum_{l=1}^4 \alpha_{i\pm\alpha,j\pm\alpha}^l}, \quad \alpha_{i\pm\alpha,j\pm\alpha}^k = \frac{c_k}{(\epsilon + S_{i\pm\alpha,j\pm\alpha}^k)^p},$$

with the constants c_k given by (3.6). As usual, the smoothness measure for every stencil is taken as a normalized sum of the discrete L^2 -norms of the derivatives. If we define the forward and backward differences $\Delta_x^+ \varphi_{i,j} = \varphi_{i+1,j} - \varphi_{i,j}$, $\Delta_x^- \varphi_{i,j} = \varphi_{i,j} - \varphi_{i-1,j}$, $\Delta_y^+ \varphi_{i,j} = \varphi_{i,j+1} - \varphi_{i,j}$, $\Delta_y^- \varphi_{i,j} = \varphi_{i,j} - \varphi_{i,j-1}$, then the smoothness measures for the evolution point $x_{i+\alpha,j+\alpha}$ are given by

$$\begin{aligned} S_{i+\alpha,j+\alpha}^1 &= (\Delta_x^+ \varphi_{i,j})^2 + (\Delta_x^+ \varphi_{i+1,j})^2 + (\Delta_x^+ \varphi_{i,j+1})^2 + (\Delta_y^+ \varphi_{i,j})^2 + (\Delta_y^+ \varphi_{i,j+1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i+1,j})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i+1,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j+1})^2 \right], \\ S_{i+\alpha,j+\alpha}^2 &= (\Delta_x^+ \varphi_{i,j})^2 + (\Delta_x^+ \varphi_{i-1,j})^2 + (\Delta_x^+ \varphi_{i,j+1})^2 + (\Delta_y^+ \varphi_{i,j})^2 + (\Delta_y^+ \varphi_{i,j+1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i+1,j})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j+1})^2 \right], \\ S_{i+\alpha,j+\alpha}^3 &= (\Delta_x^+ \varphi_{i,j})^2 + (\Delta_x^+ \varphi_{i,j+1})^2 + (\Delta_x^+ \varphi_{i-1,j+1})^2 + (\Delta_y^+ \varphi_{i,j})^2 + (\Delta_y^+ \varphi_{i+1,j})^2 \\ &\quad + (\Delta_y^+ \varphi_{i+1,j-1})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i,j+1})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i+1,j})^2 \right], \\ S_{i+\alpha,j+\alpha}^4 &= (\Delta_x^+ \varphi_{i,j})^2 + (\Delta_x^+ \varphi_{i+1,j})^2 + (\Delta_x^+ \varphi_{i,j+1})^2 + (\Delta_y^+ \varphi_{i,j})^2 + (\Delta_y^+ \varphi_{i,j-1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i+1,j})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i+1,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j})^2 \right]. \end{aligned}$$

The smoothness measures for the evolution point $x_{i-\alpha,j-\alpha}$ are

$$\begin{aligned} S_{i-\alpha,j-\alpha}^1 &= (\Delta_x^+ \varphi_{i-2,j})^2 + (\Delta_x^+ \varphi_{i-1,j})^2 + (\Delta_x^+ \varphi_{i-1,j-1})^2 + (\Delta_y^+ \varphi_{i,j-2})^2 + (\Delta_y^+ \varphi_{i,j-1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i-1,j-1})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i-1,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j-1})^2 \right], \\ S_{i-\alpha,j-\alpha}^2 &= (\Delta_x^+ \varphi_{i,j})^2 + (\Delta_x^+ \varphi_{i-1,j})^2 + (\Delta_x^+ \varphi_{i-1,j-1})^2 + (\Delta_y^+ \varphi_{i-1,j})^2 + (\Delta_y^+ \varphi_{i-1,j-1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i,j-2})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j-1})^2 \right], \\ S_{i-\alpha,j-\alpha}^3 &= (\Delta_x^+ \varphi_{i-1,j})^2 + (\Delta_x^+ \varphi_{i,j-1})^2 + (\Delta_x^+ \varphi_{i-1,j-1})^2 + (\Delta_y^+ \varphi_{i,j-1})^2 + (\Delta_y^+ \varphi_{i-1,j})^2 \\ &\quad + (\Delta_y^+ \varphi_{i-1,j-1})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i,j-1})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i-1,j})^2 \right], \\ S_{i-\alpha,j-\alpha}^4 &= (\Delta_x^+ \varphi_{i-2,j})^2 + (\Delta_x^+ \varphi_{i-1,j})^2 + (\Delta_x^+ \varphi_{i-1,j-1})^2 + (\Delta_y^+ \varphi_{i,j})^2 + (\Delta_y^+ \varphi_{i,j-1})^2 \\ &\quad + (\Delta_y^+ \varphi_{i-1,j-1})^2 + \frac{1}{\Delta x^2} \left[(\Delta_x^+ \Delta_x^- \varphi_{i-1,j})^2 + (\Delta_y^+ \Delta_y^- \varphi_{i,j})^2 \right]. \end{aligned}$$

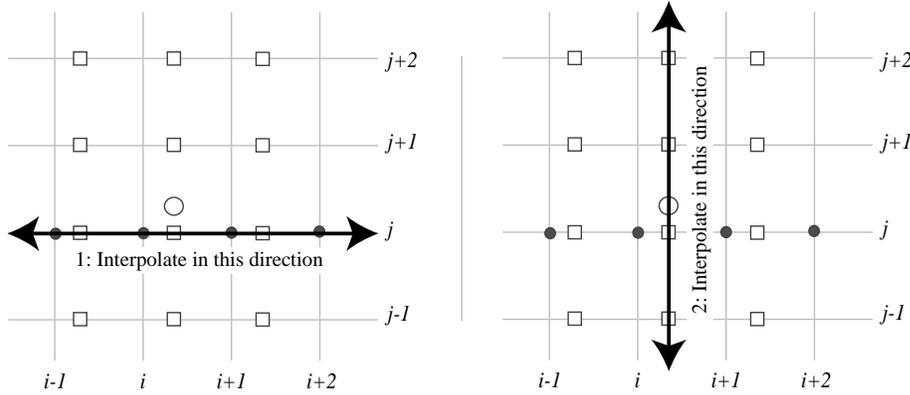


FIG. 3.4. The dimension-by-dimension reconstruction process in two dimensions. Left: the first step, where the intermediate interpolants $\varphi_{i+\alpha,j}$ at $x_{i+\alpha,j}$ (open squares) are computed using the data $\varphi_{i,j}$ (black dots). Right: the second step, where $\varphi_{i+\alpha,j}$ is interpolated in the j direction, giving $\varphi_{i+\alpha,j+\alpha}$ at $x_{i+\alpha,j+\alpha}$ (open circle).

3.2.2. A dimension-by-dimension reconstruction of $\varphi_{i\pm\alpha,j\pm\alpha}$. A different way to obtain high-order approximations for the values of $\varphi_{i\pm\alpha,j\pm\alpha}$ is by carrying out a sequence of one-dimensional reconstructions from section 2.2. This dimension-by-dimension approach for the reconstruction step is similar in spirit to that of [17], but here, in order to generate a Godunov-type scheme (unlike [17]), we are forced to use evolution points that are not positioned in the same locations as the data $x_{i,j}$. An appropriately chosen sequence of one-dimensional reconstructions addresses this problem.

We use the subscript “*” to denote the full range of an array, such that $\varphi_{*,j}$ and $\varphi_{i,*}$ denote the one-dimensional arrays $\varphi_{*,j} = (\varphi_{1,j}, \dots, \varphi_{N,j})$ and $\varphi_{i,*} = (\varphi_{i,1}, \dots, \varphi_{i,N})$. With the notation for the one-dimensional third-order reconstruction, (2.10), we can express the dimension-by-dimension reconstruction at $x_{i+\alpha,j+\alpha}$ as

1. for each i, j : $\varphi_{i+\alpha,j} = \text{reconstruct_}\varphi\text{-1D-3}(i, \alpha, \varphi_{*,j})$;
2. for each i, j : $\varphi_{i+\alpha,j+\alpha} = \text{reconstruct_}\varphi\text{-1D-3}(j, \alpha, \varphi_{i+\alpha,*})$.

Here, we first interpolate along the first coordinate axis and reconstruct φ at $x_{i+\alpha,j}$. The data at $x_{i+\alpha,j}$ is then interpolated along the second coordinate axis to the location $x_{i+\alpha,j+\alpha}$ to give $\varphi_{i+\alpha,j+\alpha}$ (see Figure 3.4). Obviously, the order in which the steps are performed is not important. In a similar way, a dimension-by-dimension reconstruction at $x_{i-\alpha,j-\alpha}$ is given by

1. for each i, j : $\varphi_{i-\alpha,j} = \text{reconstruct_}\varphi\text{-1D-3}(i, -\alpha, \varphi_{*,j})$;
2. for each i, j : $\varphi_{i-\alpha,j-\alpha} = \text{reconstruct_}\varphi\text{-1D-3}(j, -\alpha, \varphi_{i-\alpha,*})$.

3.2.3. The reprojection step. After evolving the solution to the next time step at the evolution points $x_{i\pm\alpha,j\pm\alpha}$, we would like to reproject $\varphi_{i\pm\alpha,j\pm\alpha}^{n+1}$ back onto the integer grid-points $x_{i,j}$ to end up with $\varphi_{i,j}^{n+1}$. There are several different ways to perform this task, out of which we choose to present the following: a two-dimensional reprojection using the two-dimensional reconstruction of section 3.2.1 or the dimension-by-dimension reconstruction of section 3.2.2, and a one-dimensional projection along the diagonal.

I. *A 2D reprojection.* The evolution points at $x_{i\pm\alpha,j\pm\alpha}$ have the same geometrical relationship to $x_{i,j}$ as $x_{i,j}$ has to $x_{i-\alpha,j-\alpha}$. Hence, in order to reconstruct $\varphi_{i,j}^{n+1}$ from $\varphi_{i\pm\alpha,j\pm\alpha}$, we can directly utilize the projections from section 3.2.1 or section 3.2.2,

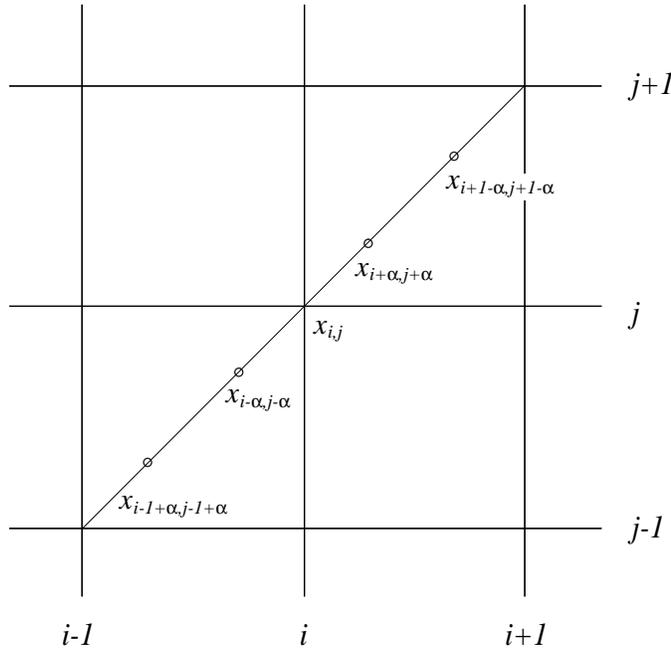


FIG. 3.5. The evolution points used for the diagonal reconstruction of $\varphi_{i,j}$.

taking $\varphi_{i\pm\alpha, j\pm\alpha}$ as the input data and reversing the sign of the parameter from $\pm\alpha$ to $\mp\alpha$. The final value $\varphi_{i,j}^{n+1}$ is then taken as the average of the projections of $\varphi_{i+\alpha, j+\alpha}$ and $\varphi_{i-\alpha, j-\alpha}$. Hence, if we denote either the two-dimensional or the dimension-by-dimension reconstruction described in section 3.2.1 or section 3.2.2 as

$$(3.7) \quad \varphi_{i\pm\alpha, j\pm\alpha} = \text{reconstruct_}\varphi\text{-2D-3}(i, j, \pm\alpha, \varphi),$$

where φ is now the two-dimensional array $\{\varphi_{i,j}\}$, then the reprojection step is

- (i) for each i, j : $\varphi_{i,j}^+ = \text{reconstruct_}\varphi\text{-2D-3}(i, -\alpha, \varphi_{i+\alpha, j+\alpha})$;
- (ii) for each i, j : $\varphi_{i,j}^- = \text{reconstruct_}\varphi\text{-2D-3}(i, \alpha, \varphi_{i-\alpha, j-\alpha})$;
- (iii) for each i, j : $\varphi_{i,j}^{n+1} = \frac{1}{2}(\varphi_{i,j}^+ + \varphi_{i,j}^-)$.

II. *A diagonal reprojection.* In this case we use one-dimensional data along the diagonal, $\{\varphi_{i-1+\alpha, j-1+\alpha}, \varphi_{i-\alpha, j-\alpha}, \varphi_{i+\alpha, j+\alpha}, \varphi_{i+1-\alpha, j+1-\alpha}\}$, to construct a third-order WENO approximation of $\varphi_{i,j}^{n+1}$ (see Figure 3.5).

Define

$$(3.8) \quad \begin{aligned} \varphi_{i,j}^- &:= \frac{\alpha^2}{2\alpha-1}\varphi_{i-1+\alpha, j-1+\alpha} + \frac{\alpha-1}{2(2\alpha-1)}\varphi_{i-\alpha, j-\alpha} \\ &\quad + \frac{1-\alpha}{2}\varphi_{i+\alpha, j+\alpha} = \varphi(x_{i,j}) + O(\Delta x^3, \Delta y^3), \\ \varphi_{i,j}^+ &:= \frac{1-\alpha}{2}\varphi_{i-\alpha, j-\alpha} + \frac{\alpha-1}{2(2\alpha-1)}\varphi_{i+\alpha, j+\alpha} \\ &\quad + \frac{\alpha^2}{2\alpha-1}\varphi_{i+1-\alpha, j+1-\alpha} = \varphi(x_{i,j}) + O(\Delta x^3, \Delta y^3). \end{aligned}$$

Since $(\varphi_{i,j}^- + \varphi_{i,j}^+)/2 = \varphi(x_{i,j}) + O(\Delta x^4, \Delta y^4)$, we can obtain $\varphi_{i,j}^{n+1}$ as

$$(3.9) \quad \varphi_{i,j}^{n+1} = w_{i,j}^- \varphi_{i,j}^- + w_{i,j}^+ \varphi_{i,j}^+,$$

where as usual $w_{i,j}^\pm = \alpha_{i,j}^\pm / (\alpha_{i,j}^+ + \alpha_{i,j}^-)$ and $\alpha_{i,j}^\pm = (2(\epsilon + S_{i,j}^\pm)^p)^{-1}$. The smoothness measures are again taken as the sum of the discrete L^2 -norm of the derivatives, which in this case is more complicated due to the uneven spacing of the data:

$$\begin{aligned} S_{i,j}^- &= \frac{1}{\Delta x} \left[\left(\frac{\varphi_{i-\alpha,j-\alpha} - \varphi_{i-1+\alpha,j-1+\alpha}}{1-2\alpha} \right)^2 + \left(\frac{\varphi_{i+\alpha,j+\alpha} - \varphi_{i-\alpha,j-\alpha}}{2\alpha} \right)^2 \right] \\ &\quad + \frac{4}{\Delta x^3} \left(\frac{\varphi_{i-\alpha,j-\alpha} - \varphi_{i-1+\alpha,j-1+\alpha}}{1-2\alpha} - \frac{\varphi_{i+\alpha,j+\alpha} - \varphi_{i-\alpha,j-\alpha}}{2\alpha} \right)^2, \\ S_{i,j}^+ &= \frac{1}{\Delta x} \left[\left(\frac{\varphi_{i+\alpha,j+\alpha} - \varphi_{i-\alpha,j-\alpha}}{2\alpha} \right)^2 + \left(\frac{\varphi_{i+1-\alpha,j+1-\alpha} - \varphi_{i+\alpha,j+\alpha}}{1-2\alpha} \right)^2 \right] \\ &\quad + \frac{4}{\Delta x^3} \left(\frac{\varphi_{i+\alpha,j+\alpha} - \varphi_{i-\alpha,j-\alpha}}{2\alpha} - \frac{\varphi_{i+1-\alpha,j+1-\alpha} - \varphi_{i+\alpha,j+\alpha}}{1-2\alpha} \right)^2. \end{aligned}$$

Remark. Our numerical simulations in section 4.3 indicate that there is little difference between the quality of the two-dimensional reconstruction and the dimension-by-dimension reconstruction of sections 3.2.1 and 3.2.2. We will use this fact when extending our methods to fifth order and higher dimensions. We note that the diagonal reprojection significantly reduces the CFL number (see section 4.4).

3.3. A two-dimensional fifth-order scheme. Using the dimension-by-dimension approach, it is easy to extend the above scheme to fifth order: simply replace the one-dimensional third-order interpolations by the fifth-order interpolation in section 3.2.2. Using the one-dimensional notation, (2.19), we obtain a fifth-order reconstruction at $x_{i+\alpha,j+\alpha}$ as

1. for each i, j : $\varphi_{i+\alpha,j} = \text{reconstruct_}\varphi\text{-1D.5}(i, \alpha, \varphi_{*,j})$;
2. for each i, j : $\varphi_{i+\alpha,j+\alpha} = \text{reconstruct_}\varphi\text{-1D.5}(j, \alpha, \varphi_{i+\alpha,*})$.

Similarly, at $x_{i-\alpha,j-\alpha}$ we have

1. for each i, j : $\varphi_{i-\alpha,j} = \text{reconstruct_}\varphi\text{-1D.5}(i, -\alpha, \varphi_{*,j})$;
2. for each i, j : $\varphi_{i-\alpha,j-\alpha} = \text{reconstruct_}\varphi\text{-1D.5}(j, -\alpha, \varphi_{i-\alpha,*})$.

We denote this reconstruction as

$$(3.10) \quad \varphi_{i\pm\alpha,j\pm\alpha} = \text{reconstruct_}\varphi\text{-2D.5}(i, j, \pm\alpha, \varphi).$$

For the derivatives we have

1. for each i, j : $\varphi'_{i\pm\alpha,j} = \text{reconstruct_}\varphi'\text{-1D.5}(i, \pm\alpha, \varphi_{*,j})$,
2. for each i, j : $\varphi'_{i\pm\alpha,j\pm\alpha} = \text{reconstruct_}\varphi'\text{-1D.5}(j, \pm\alpha, \varphi_{i\pm\alpha,*})$,

which we denote as

$$(3.11) \quad \varphi'_{i\pm\alpha,j\pm\alpha} = \text{reconstruct_}\varphi'\text{-2D.5}(i, j, \pm\alpha, \varphi).$$

Reprojection onto the original grid-points $x_{i,j}$ is performed using the two-dimensional dimension-by-dimension reprojection option described in section 3.2.3.

Remarks.

1. Due to the reduced stability resulting from the use of diagonal reprojection, which is demonstrated in section 4.4, we do not develop a fifth-order analogue to the third-order diagonal reprojection.

2. It is straightforward to develop a fifth-order two-dimensional method involving two-dimensional stencils, extending section 3.2.1. Such a method would involve four interpolants defined on ten-point stencils that cover a 21-point stencil.

We summarize the two-dimensional fifth-order algorithm in the following, where $RK(\varphi_{i\pm\alpha}^n, \varphi'_{i\pm\alpha}, \Delta t)$ is now the fourth-order RK method which integrates (2.1). As in Algorithm 2.1, each internal step of the RK method will require additional reconstructions of $\varphi'_{i\pm\alpha}$ from that step's $\varphi_{i\pm\alpha}$.

ALGORITHM 3.1. *Let $\alpha = 1/(2 + \sqrt{2})$. Assume that $\{\varphi_{i,j}^n\}$ are given.*

(a) *Reconstruct:*

$$\begin{aligned} \varphi_{i\pm\alpha, j\pm\alpha} &= \text{reconstruct_}\varphi\text{-2D_5}(i, j, \pm\alpha, \varphi), \\ \varphi'_{i\pm\alpha, j\pm\alpha} &= \text{reconstruct_}\varphi'\text{-2D_5}(i, j, \pm\alpha, \varphi). \end{aligned}$$

(b) *Integrate:*

$$\begin{aligned} \varphi_{i\pm\alpha, j\pm\alpha}^{n+\frac{1}{2}} &= RK(\varphi_{i\pm\alpha, j\pm\alpha}^n, \varphi'_{i\pm\alpha, j\pm\alpha}, \Delta t/2), \\ \varphi'_{i\pm\alpha, j\pm\alpha}^{n+\frac{1}{2}} &= \text{reconstruct_}\varphi'\text{-2D_5}(i, \pm\alpha, \varphi_{\pm\alpha, \pm\alpha}^{n+\frac{1}{2}}), \\ \varphi_{i\pm\alpha, j\pm\alpha}^{n+1} &= RK(\varphi_{i\pm\alpha, j\pm\alpha}^n, \varphi'_{i\pm\alpha, j\pm\alpha}, \Delta t), \\ \varphi'_{i\pm\alpha, j\pm\alpha}^{n+1} &= \text{reconstruct_}\varphi'\text{-2D_5}(i, \pm\alpha, \varphi_{\pm\alpha, \pm\alpha}^{n+1}), \\ \varphi_{i\pm\alpha, j\pm\alpha}^{n+1} &= \varphi_{i\pm\alpha, j\pm\alpha}^n + \frac{\Delta t}{6} \left[H(\varphi'_{i\pm\alpha, j\pm\alpha}) + 4H(\varphi'_{i\pm\alpha, j\pm\alpha}^{n+\frac{1}{2}}) + H(\varphi'_{i\pm\alpha, j\pm\alpha}^{n+1}) \right]. \end{aligned}$$

(c) *Reproject:*

$$\varphi_{i,j}^{n+1} = \text{reconstruct_}\varphi\text{-2D_5}(i, j, \mp\alpha, \varphi_{\pm\alpha, \pm\alpha}^{n+1}).$$

3.4. Multidimensional extensions. The extension of the dimension-by-dimension approach to more than two space dimensions is straightforward. For example, using the notation of section 3.3, a three-dimensional fifth-order reconstruction is

1. for each i, j, k : $\varphi_{i+\alpha, j, k} = \text{reconstruct_}\varphi\text{-1D_5}(i, \alpha, \varphi_{*, j, k})$;
2. for each i, j, k : $\varphi_{i+\alpha, j+\alpha, k} = \text{reconstruct_}\varphi\text{-1D_5}(j, \alpha, \varphi_{i+\alpha, *, k})$;
3. for each i, j, k : $\varphi_{i+\alpha, j+\alpha, k+\alpha} = \text{reconstruct_}\varphi\text{-1D_5}(k, \alpha, \varphi_{i+\alpha, j+\alpha, *})$.

The reconstruction at $x_{i-\alpha, j-\alpha, k-\alpha}$ is handled similarly, and the same for the reconstruction of $\varphi'_{i+\alpha, j+\alpha, k+\alpha}$. In three dimensions, $\alpha = 1/(3 + \sqrt{3})$.

A d -dimensional reconstruction based on d -dimensional stencils quickly becomes very large. It is readily apparent that the dimension-by-dimension approach will scale to high dimensions better than d -dimensional interpolants.

4. Numerical simulations. In this section we present simulations that test the schemes we developed in this paper. In section 4.1 we demonstrate the third- and fifth-order methods in one dimension. Section 4.2 focuses on the fifth-order method in two and three space dimensions. In section 4.3 we compare the two-dimensional third-order method based on two-dimensional stencils with the dimension-by-dimension approach. In section 4.4 we examine, in detail, stability issues in two dimensions, including comparisons with [17]. Some of these examples are standard test cases that can be found, e.g., in [22, 31, 35].

We do not follow the practice in [17] of masking singular regions from our error measurements.

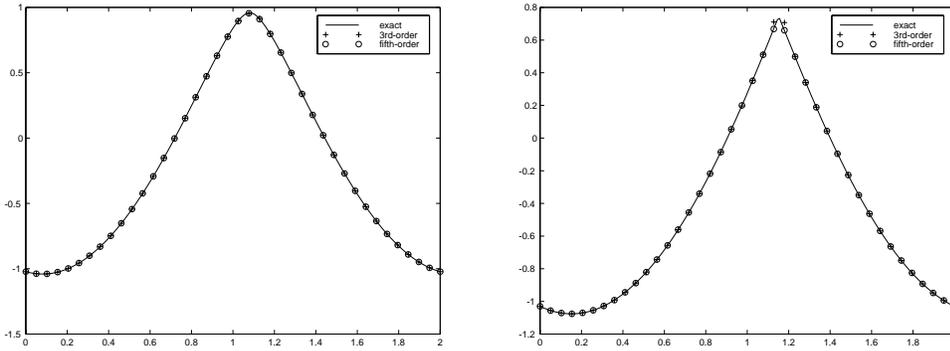


FIG. 4.1. One-dimensional convex Hamiltonian (4.1). Left: the solution before the singularity formation, $T = 0.8/\pi^2$. Right: the solution after the singularity formation, $T = 1.5/\pi^2$. In both panels $N = 40$. Shown are the third- and fifth-order approximations and the exact solution.

4.1. One-dimensional examples.

A convex Hamiltonian. We start by testing the performance of our schemes on a convex Hamiltonian. We approximate solutions of the one-dimensional equation

$$(4.1) \quad \phi_t + \frac{1}{2}(\phi_x + 1)^2 = 0,$$

subject to the initial data $\phi(x, 0) = -\cos(\pi x)$ with periodic boundary conditions on $[0, 2]$. The change of variables $u(x, t) = \phi(x, t) + 1$ transforms the equation into the Burgers equation $u_t + \frac{1}{2}(u^2)_x = 0$, which can be easily solved via the method of characteristics [35]. As is well known, the Burgers equation generally develops discontinuous solutions even with smooth initial data, and hence we expect the solutions of (4.1) to have discontinuous derivatives. In our case, the solution develops a singularity at time $t = \pi^{-2}$.

The results of our simulations are shown in Figure 4.1. The order of accuracy of these methods is determined from the relative L^1 error (see [30]), defined as the L^1 -norm of the error divided by the L^1 -norm of the exact solution. These results along with the relative L^∞ -norm before the singularity, at $T = 0.8/\pi^2$, are given in Table 4.1, and after the singularity, at $T = 1.5/\pi^2$, in Table 4.2.

A nonconvex Hamiltonian. In this example we deal with nonconvex HJ equations. In one dimension we solve

$$(4.2) \quad \phi_t - \cos(\phi_x + 1) = 0,$$

subject to the initial data $\phi(x, 0) = -\cos(\pi x)$ with periodic boundary conditions on $[0, 2]$. In this case (4.2) has a smooth solution for $t \lesssim 1.049/\pi^2$, after which a singularity forms. A second singularity forms at $t \approx 1.29/\pi^2$. The results are shown in Figure 4.2. The convergence results before and after the singularity formation are given in Tables 4.3–4.4.

A linear advection equation. In this example (from [17], with a misprint corrected in [40]) we solve the one-dimensional linear advection equation, i.e., $H(\phi_x) = \phi_x$. We assume periodic boundary conditions on $[-1, 1]$ and take the initial data as

TABLE 4.1

Relative L^1 errors for the one-dimensional convex HJ problem (4.1) before the singularity formation. $T = 0.8/\pi^2$.

N	Third-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	9.41×10^{-5}	–	1.77×10^{-5}	–
200	1.13×10^{-5}	3.06	1.33×10^{-6}	3.73
400	1.39×10^{-6}	3.02	9.35×10^{-8}	3.83
800	1.74×10^{-7}	3.00	5.94×10^{-9}	3.00
N	Fifth-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	1.41×10^{-5}	–	2.61×10^{-6}	–
200	4.21×10^{-7}	5.07	4.03×10^{-8}	6.02
400	3.31×10^{-8}	5.00	6.53×10^{-10}	5.95
800	4.03×10^{-10}	5.03	1.00×10^{-11}	6.03

TABLE 4.2

Relative L^1 errors for the one-dimensional convex HJ problem (4.1) after the singularity formation. $T = 1.5/\pi^2$.

N	Third-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	9.10×10^{-4}	–	2.77×10^{-4}	–
200	2.16×10^{-4}	2.07	7.63×10^{-5}	1.86
400	6.84×10^{-5}	1.66	2.68×10^{-5}	1.51
800	2.75×10^{-5}	1.31	2.08×10^{-5}	0.37
N	Fifth-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	7.85×10^{-4}	–	5.78×10^{-4}	–
200	1.61×10^{-4}	2.29	8.29×10^{-5}	2.29
400	6.71×10^{-5}	1.26	5.09×10^{-5}	1.26
800	3.44×10^{-5}	0.96	3.44×10^{-5}	0.96

$\phi(x, 0) = g(x - 0.5)$ on $[-1, 1]$, where

$$g(x) = - \left(\frac{\sqrt{3}}{2} + \frac{9}{2} + \frac{2\pi}{3} \right) (x + 1) + h(x),$$

$$(4.3) \quad h(x) = \begin{cases} 2 \cos\left(\frac{3\pi}{2}x^2\right) - \sqrt{3}, & -1 < x < -\frac{1}{3}, \\ 3/2 + 3 \cos(2\pi x), & -\frac{1}{3} < x < 0, \\ 15/2 - 3 \cos(2\pi x), & 0 < x < \frac{1}{3}, \\ (28 + 4\pi + \cos(3\pi x))/3 + 6\pi x(x - 1), & \frac{1}{3} < x < 1. \end{cases}$$

The results of the fifth-order method are shown in Figure 4.3, where it is compared with the fifth-order method of [17]. The reduced dissipation effects of our method are visible in the reduced round-off of the corners.

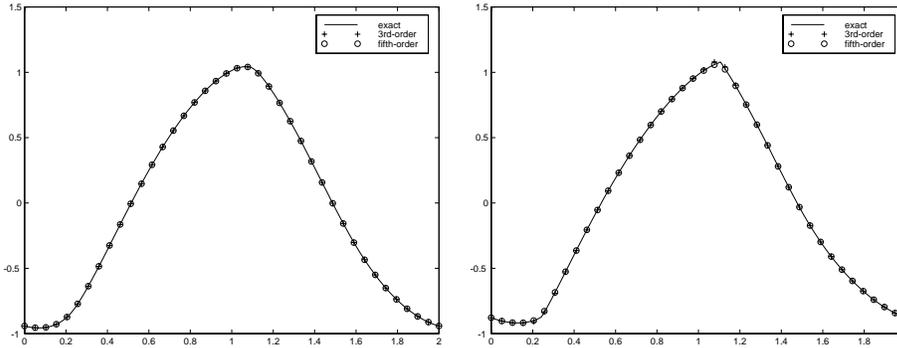


FIG. 4.2. One-dimensional nonconvex Hamiltonian (4.2). Left: The solution before the singularity formation, $T = 0.8/\pi^2$. Right: The solution after the singularity formation, $T = 1.5/\pi^2$. In both panels $N = 40$. Shown are the third- and fifth-order approximations and the exact solution.

TABLE 4.3

Relative L^1 errors for the one-dimensional nonconvex HJ problem (4.2) before the singularity formation. $T = 0.8/\pi^2$.

N	Third-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	6.47×10^{-5}	–	9.05×10^{-6}	–
200	7.78×10^{-6}	3.06	1.11×10^{-6}	3.03
400	8.77×10^{-7}	3.15	9.27×10^{-8}	3.58
800	9.87×10^{-8}	3.15	6.12×10^{-9}	3.92
N	Fifth-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	1.29×10^{-5}	–	4.97×10^{-6}	–
200	6.52×10^{-7}	4.31	2.38×10^{-7}	4.38
400	2.10×10^{-8}	4.95	6.13×10^{-9}	5.28
800	5.96×10^{-10}	5.14	1.03×10^{-10}	5.90

TABLE 4.4

Relative L^1 errors for the one-dimensional nonconvex HJ problem (4.2) after the singularity formation. $T = 1.5/\pi^2$.

N	Third-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	2.81×10^{-4}	–	9.64×10^{-5}	–
200	1.32×10^{-4}	1.08	5.05×10^{-5}	0.93
400	2.31×10^{-5}	2.52	6.00×10^{-6}	3.07
800	8.43×10^{-6}	1.46	3.30×10^{-6}	0.86
N	Fifth-order method			
	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
100	1.57×10^{-4}	–	1.12×10^{-4}	–
200	8.34×10^{-5}	0.91	6.60×10^{-5}	0.77
400	1.22×10^{-5}	2.78	8.64×10^{-6}	2.93
800	6.67×10^{-5}	0.87	5.23×10^{-6}	.072

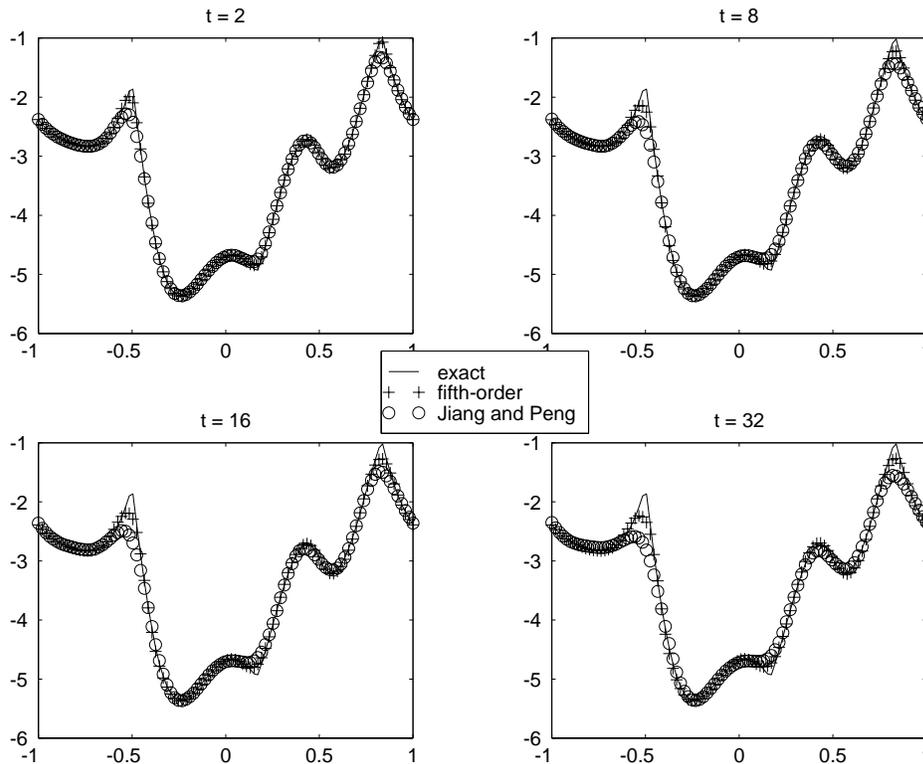


FIG. 4.3. One-dimensional linear advection, (4.3). $T = 2, 8, 16, 32$; $N = 100$. Crosses: our fifth-order method. Circles: the fifth-order method of [17] with a local Lax-Friedrichs flux. Solid line: the exact solution.

4.2. Two-dimensional examples.

A convex Hamiltonian. In two dimensions we solve a problem similar to (4.1),

$$(4.4) \quad \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0,$$

which can be reduced to a one-dimensional problem via the coordinate transformation $\begin{pmatrix} \xi \\ \eta \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$. The results of the fifth-order calculations for the initial data $\phi(x, y, 0) = -\cos(\pi(x+y)/2) = -\cos(\pi\xi)$ are shown in Figure 4.4. The convergence rates for the two-dimensional fifth-order scheme before and after the singularity are shown in Table 4.5.

A nonconvex Hamiltonian. The two-dimensional nonconvex problem, which is analogous to the one-dimensional problem (4.2), is

$$(4.5) \quad \phi_t - \cos(\phi_x + \phi_y + 1) = 0.$$

Here we assume initial data, given by $\phi(x, y, 0) = -\cos(\pi(x+y)/2)$, and periodic boundary conditions. The results are shown in Figure 4.5. The convergence results for the two-dimensional fifth-order scheme before and after the singularity formation are given in Table 4.6.

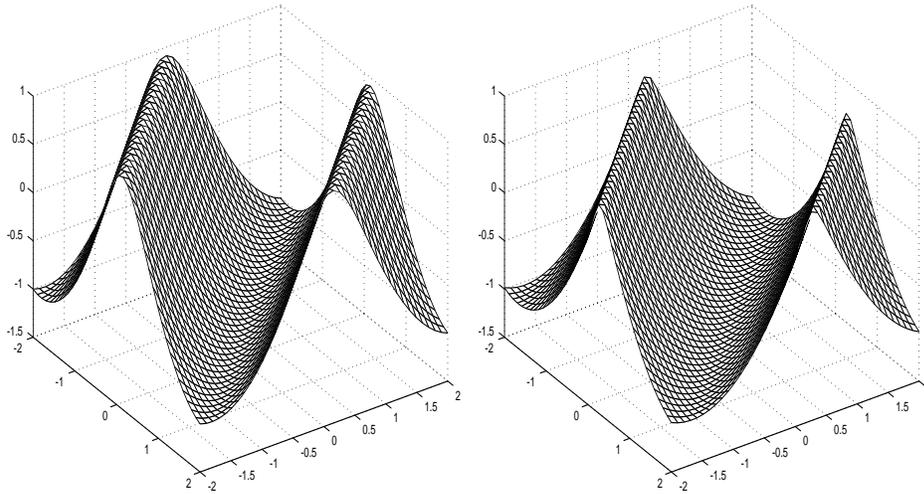


FIG. 4.4. Two-dimensional convex Hamiltonian, (4.4). Left: the solution before the singularity formation, $T = 0.8/\pi^2$. Right: the solution after the singularity formation, $T = 1.5/\pi^2$. In both panels $N = 40 \times 40$. The solution is computed with the fifth-order method.

TABLE 4.5

Relative L^1 and L^∞ errors for the two-dimensional convex HJ problem (4.4) before and after singularity formation, computed via the fifth-order method.

Before singularity $T = 0.8/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
50	1.19×10^{-4}	–	7.78×10^{-7}	–
100	6.80×10^{-6}	4.13	1.64×10^{-8}	5.56
200	1.73×10^{-7}	5.30	1.12×10^{-10}	7.20
After singularity $T = 1.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
50	1.32×10^{-3}	–	2.07×10^{-5}	–
100	3.89×10^{-4}	1.76	3.60×10^{-6}	2.52
200	4.86×10^{-5}	3.00	1.69×10^{-7}	4.41

A fully two-dimensional example. The above two-dimensional examples are actually one-dimensional along the diagonal. To check the performance of our methods on fully two-dimensional problems, we solve

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

on $[-\pi, \pi] \times [-\pi, \pi]$, subject to the initial data $\phi(x, y, 0) = \sin(x) + \cos(y)$ with periodic boundary conditions. The exact solution for this problem is given implicitly by $\phi(x, y, t) = -\cos(q)\sin(r) + \sin(q) + \cos(r)$, where $x = q - t \sin(r)$ and $y = r + t \cos(q)$. This solution is smooth for $t < 1$, continuous $\forall t$, and has discontinuous derivatives for $t \geq 1$. The results of our simulations at times $T = 0.8, 1.5$ are shown in Figure 4.6. The convergence results for the fifth-order two-dimensional schemes before the singularity formation are given in Table 4.7 and confirm the expected order of accuracy of our methods.

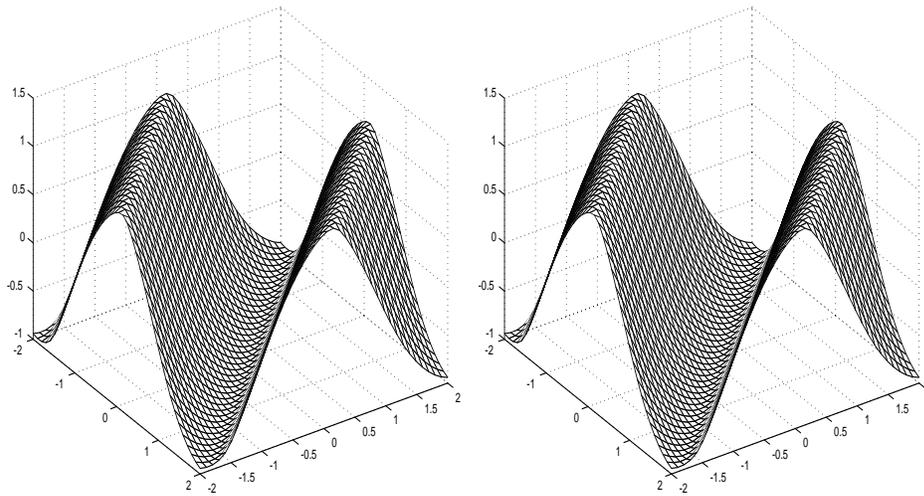


FIG. 4.5. Two-dimensional nonconvex Hamiltonian, (4.5). Left: the solution before the singularity formation, $T = 0.8/\pi^2$. Right: the solution after the singularity formation, $T = 1.5/\pi^2$. $N = 40 \times 40$. The solution is computed with the fifth-order method.

TABLE 4.6

Relative L^1 and L^∞ errors for the two-dimensional nonconvex HJ problem (4.5) before and after the singularity formation, computed with the fifth-order method.

Before singularity $T = 0.8/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
50	1.11×10^{-4}	—	1.26×10^{-6}	—
100	6.91×10^{-6}	4.00	2.42×10^{-8}	5.70
200	3.85×10^{-7}	4.17	6.27×10^{-10}	5.27
After singularity $T = 1.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
50	1.47×10^{-3}	—	8.58×10^{-6}	—
100	1.93×10^{-4}	2.93	9.27×10^{-7}	3.21
200	8.87×10^{-5}	1.12	3.09×10^{-7}	1.58

An eikonal equation in geometric optics. We consider a two-dimensional nonconvex problem that arises in geometric optics [20]:

$$(4.7) \quad \begin{cases} \phi_t + \sqrt{\phi_x^2 + \phi_y^2 + 1} = 0, \\ \phi(x, y, 0) = \frac{1}{4} (\cos(2\pi x) - 1) (\cos(2\pi y) - 1) - 1. \end{cases}$$

The results of our fifth-order method at time $T = 0.6$ are shown in Figure 4.7, where we see the sharp corners that develop in this problem.

An optimal control problem. We solve an optimal control problem related to cost determination [35]. Here the Hamiltonian is of the form $H(x, y, \nabla\phi)$:

$$(4.8) \quad \begin{cases} \phi_t - \sin(y) \phi_x + \sin(x) \phi_y + |\phi_y| - \frac{1}{2} \sin^2(y) - 1 + \cos(x) = 0, \\ \phi(x, y, 0) = 0. \end{cases}$$

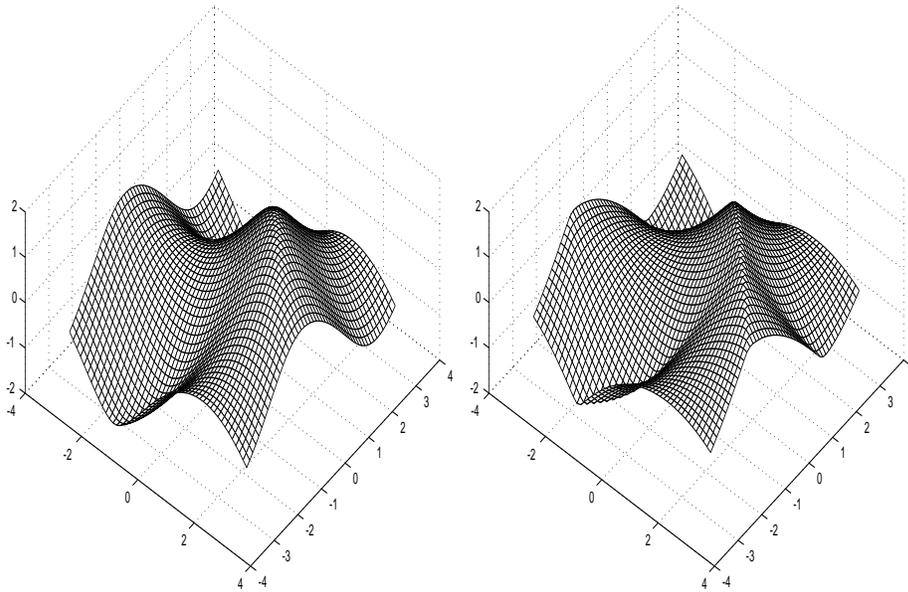


FIG. 4.6. Fully two-dimensional Hamiltonian, (4.6). Left: the solution before the singularity formation, $T = 0.8$. Right: the solution after the singularity formation, $T = 1.5$. In both panels $N = 50 \times 50$. The solution is computed with the fifth-order method.

TABLE 4.7

Relative L^1 errors for the two-dimensional HJ problem (4.6) before singularity formation. $T = 0.8$. The solution is computed with the fifth-order method.

Before singularity $T = 0.8$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
50	6.10×10^{-6}	—	8.15×10^{-8}	—
100	2.10×10^{-7}	4.86	7.35×10^{-10}	6.79
200	7.53×10^{-9}	4.80	5.59×10^{-12}	7.04

The result of our fifth-order scheme is presented in Figure 4.8 and is in qualitative agreement with [31].

4.3. A comparison of two-dimensional third-order interpolants. In this section we use the examples (4.4), (4.5), and (4.6) to compare the third-order method of section 3.2.1, based on interpolation via two-dimensional stencils, with that of section 3.2.2, where we used a dimension-by-dimension approach. The results are shown in Table 4.8. The dimension-by-dimension method produces errors that are approximately twice as large as those for the genuinely two-dimensional reconstruction. However, the convergence rate is qualitatively the same in both methods. These results motivated us to base our fifth-order scheme on the much simpler dimension-by-dimension reconstruction.

4.4. A stability study. In this section we present a couple of stability studies that we obtained in our simulations. We start by checking the stability properties of the third-order scheme with different reprojection steps. The reconstruction step

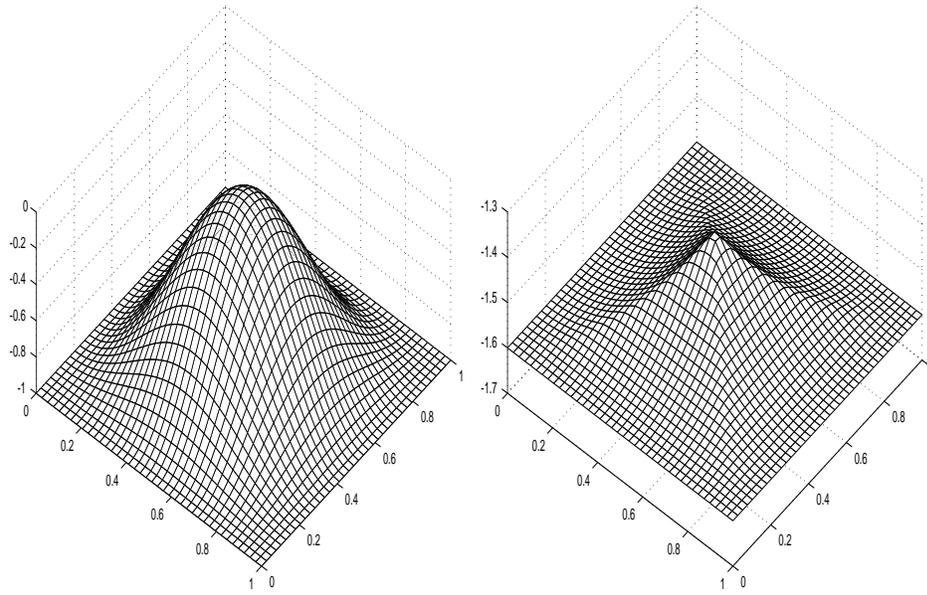


FIG. 4.7. *Two-dimensional eikonal equation, (4.7). $N = 40 \times 40$. Left: the initial data. Right: the fifth-order approximation at $T = 0.6$.*

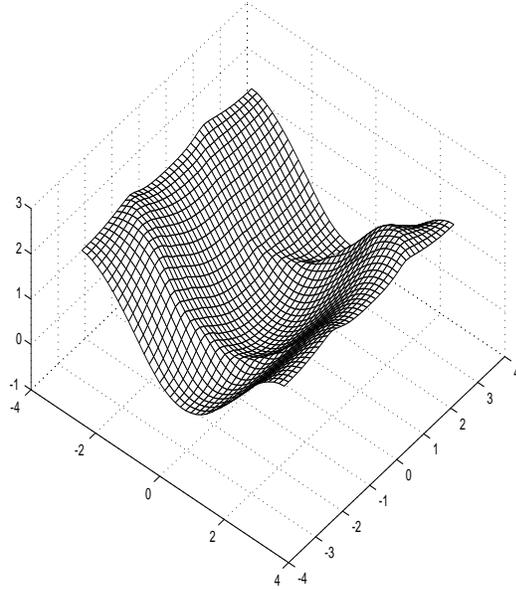


FIG. 4.8. *Two-dimensional optimal control problem, (4.8). An approximation with the fifth-order method is shown at $T = 1$ and $N = 40 \times 40$.*

TABLE 4.8

Comparison of the third-order method of section 3.2.1, using an interpolation via two-dimensional stencils, and that of section 3.2.2, using the dimension-by-dimension approach.

N	2D stencils		Dimension-by-dimension	
	Relative L^1 error	L^1 -order	Relative L^1 error	L^1 -order
Convex Hamiltonian at $T = 0.8/\pi^2$				
50	4.70×10^{-4}	–	6.13×10^{-4}	–
100	7.54×10^{-5}	2.64	9.43×10^{-5}	2.70
200	8.07×10^{-6}	3.23	1.02×10^{-5}	3.21
Convex Hamiltonian at $T = 1.5/\pi^2$				
50	1.23×10^{-3}	–	2.61×10^{-3}	–
100	4.56×10^{-4}	1.44	8.19×10^{-4}	1.67
200	3.70×10^{-5}	3.62	1.22×10^{-4}	2.74
Nonconvex Hamiltonian at $T = 0.8/\pi^2$				
50	2.27×10^{-4}	–	3.92×10^{-4}	–
100	3.75×10^{-5}	2.60	6.97×10^{-5}	2.49
200	3.99×10^{-6}	3.23	7.22×10^{-6}	3.27
Nonconvex Hamiltonian at $T = 1.5/\pi^2$				
50	1.23×10^{-3}	–	1.94×10^{-3}	–
100	2.50×10^{-4}	2.30	4.16×10^{-4}	2.22
200	7.63×10^{-5}	1.71	1.20×10^{-4}	1.79
Fully 2D example at $T = 0.8$				
50	2.01×10^{-4}	–	1.48×10^{-4}	–
100	2.42×10^{-5}	3.05	1.65×10^{-5}	3.16
200	2.95×10^{-6}	3.04	1.95×10^{-6}	3.08

is done in all cases using the dimension-by-dimension interpolant. We compare the dimension-by-dimension reprojection and the diagonal reprojection (of section 3.2.3). In Figure 4.9 we plot the L^1 error as a function of the CFL number. The test problem is (4.6) with the fully two-dimensional Hamiltonian. The solution is computed at $T = 0.8$. We see that the use of a diagonal reprojection significantly reduces the maximum allowed CFL number.

We now turn to checking the stability properties of the two-dimensional fifth-order method of section 3.3 by computing the L^1 errors for various examples while varying the CFL number. In Figure 4.10 we compare the results obtained with our fifth-order scheme with the fifth-order method of [17], for which we used a local Lax–Friedrichs flux. The numerical tests indicate that larger CFL numbers can be used with our method.

4.5. Three-dimensional examples. We proceed with a three-dimensional generalization of the convex Hamiltonian (4.4),

$$(4.9) \quad \phi_t + \frac{1}{2} (\phi_x + \phi_y + \phi_z + 1)^2 = 0,$$

subject to the initial data $\phi(x, y, z, 0) = -\cos(\pi(x + y + z)/3)$. The convergence results for the three-dimensional fifth-order scheme before and after the singularity formation are given in Table 4.9. We also approximate the solution of the nonconvex

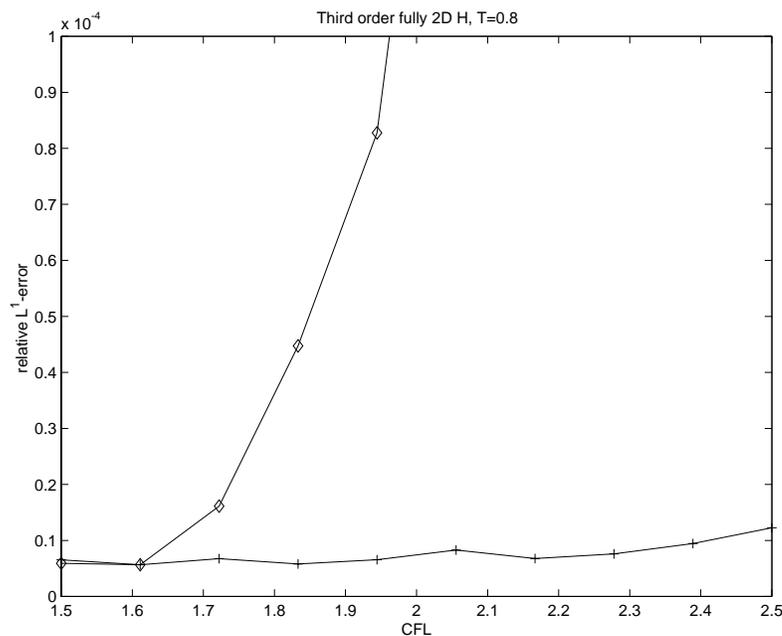


FIG. 4.9. Stability of the two-dimensional third-order method with a dimension-by-dimension reprojection (crosses) vs. a diagonal reprojection (diamonds). Fully two-dimensional Hamiltonian (4.6). $T = 0.8$ (before singularity), $N = 100 \times 100$.

TABLE 4.9

Relative L^1 and L^∞ errors for the three-dimensional convex HJ problem (4.9) before and after the singularity formation, computed with the fifth-order method.

Before singularity $T = 0.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
25	2.61×10^{-4}	–	1.07×10^{-7}	–
50	6.40×10^{-6}	5.35	3.16×10^{-10}	8.41
100	1.50×10^{-7}	5.42	9.18×10^{-13}	8.43
After singularity $T = 1.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
25	6.95×10^{-3}	–	1.80×10^{-5}	–
50	1.40×10^{-3}	2.31	4.15×10^{-6}	2.12
100	5.33×10^{-4}	1.39	6.94×10^{-7}	2.58

problem

$$(4.10) \quad \phi_t - \cos(\phi_x + \phi_y + \phi_z + 1) = 0,$$

with the same initial data. The convergence rates for the three-dimensional fifth-order schemes are given in Table 4.10.

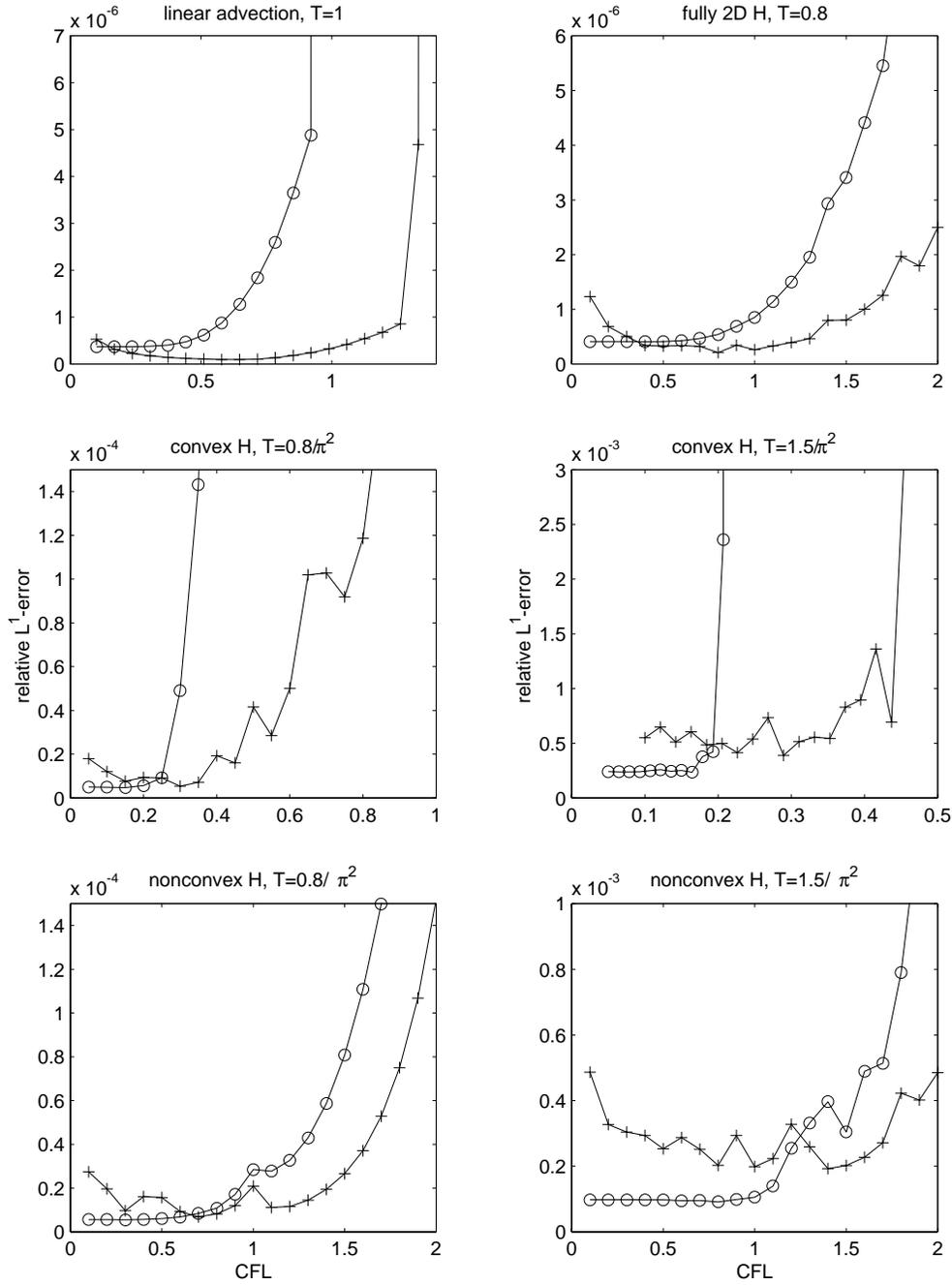


FIG. 4.10. Stability of the two-dimensional fifth-order method. $N = 100 \times 100$. Crosses: our fifth-order method. Circles: the fifth-order method of [17] with a local Lax-Friedrichs flux. Upper left: linear advection ($H(\nabla\varphi) = \nabla\varphi$) with initial condition $\phi(x, y, 0) = -\cos(\pi(x+y)/2)$. Upper right: fully 2D Hamiltonian (4.6). Middle row: convex Hamiltonian (4.4), before the singularity (left) and after the singularity (right). Bottom row: nonconvex Hamiltonian (4.5), before the singularity (left) and after the singularity (right).

TABLE 4.10

Relative L^1 and L^∞ errors for the three-dimensional nonconvex HJ problem (4.10) before and after the singularity formation, computed with the fifth-order method.

Before singularity $T = 0.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
25	7.28×10^{-4}	–	3.70×10^{-7}	–
50	3.71×10^{-5}	4.29	4.06×10^{-9}	6.51
100	1.05×10^{-6}	5.14	2.18×10^{-11}	7.54
After singularity $T = 1.5/\pi^2$				
N	Relative L^1 error	L^1 -order	Relative L^∞ error	L^∞ -order
25	6.74×10^{-3}	–	3.27×10^{-6}	–
50	1.26×10^{-3}	2.42	6.90×10^{-7}	2.25
100	4.21×10^{-4}	1.59	6.84×10^{-8}	3.33

Acknowledgment. We would like to thank Volker Elling for helpful discussions throughout the early stages of this work.

REFERENCES

- [1] R. ABGRALL, *Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes*, Comm. Pure Appl. Math., 49 (1996), pp. 1339–1373.
- [2] P. ARMINJON AND M.-C. VIALON, *Généralisation du schéma de Nessyahu-Tadmor pour une équation hyperbolique à deux dimensions d'espace*, C. R. Acad. Sci. Paris Sér. I, 320 (1995), pp. 85–88.
- [3] G. BARLES, *Solution de viscosité des équations de Hamilton-Jacobi*, Springer-Verlag, Berlin, 1994.
- [4] F. BIANCO, G. PUPPO, AND G. RUSSO, *High-order central schemes for hyperbolic systems of conservation laws*, SIAM J. Sci. Comput., 21 (1999), pp. 294–322.
- [5] S. BRYSON AND D. LEVY, *Central schemes for multidimensional Hamilton-Jacobi equations*, SIAM J. Sci. Comput., to appear.
- [6] S. BRYSON AND D. LEVY, *High-order central WENO schemes for 1D Hamilton-Jacobi equations*, in Numerical Mathematics and Advanced Applications (Proceedings of ENUMATH 2001, Ischia, Italy), F. Brezzi, A. Buffa, S. Cosaro, and A. Murli, eds., Springer-Verlag, 2003.
- [7] M.G. CRANDALL, L.C. EVANS, AND P.-L. LIONS, *Some properties of viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 282 (1984), pp. 487–502.
- [8] M.G. CRANDALL, H. ISHII, 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.
- [9] M.G. CRANDALL AND P.-L. LIONS, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 277 (1983), pp. 1–42.
- [10] M.G. CRANDALL AND P.-L. LIONS, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19.
- [11] K.O. FRIEDRICHS AND P.D. LAX, *Systems of conservation equations with a convex extension*, Proc. Natl. Acad. Sci. USA, 68 (1971), pp. 1686–1688.
- [12] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
- [13] A. HARTEN, B. ENGQUIST, S. OSHER, AND S. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes III*, J. Comput. Phys., 71 (1987), pp. 231–303.
- [14] C. HU AND C.-W. SHU, *A discontinuous Galerkin finite element method for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 21 (1999), pp. 666–690.
- [15] S.N. KRUKOV, *The Cauchy problem in the large for nonlinear equations and for certain quasilinear systems of the first order with several variables*, Soviet Math. Dokl., 5 (1964), pp. 493–496.
- [16] G.-S. JIANG, D. LEVY, C.-T. LIN, S. OSHER, AND E. TADMOR, *High-resolution nonoscillatory central schemes with nonstaggered grids for hyperbolic conservation laws*, SIAM J. Numer. Anal., 35 (1998), pp. 2147–2168.

- [17] G.-S. JIANG AND D. PENG, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.
- [18] G.-S. JIANG AND D.-W. SHU, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
- [19] G.-S. JIANG AND E. TADMOR, *Nonoscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., 19 (1998), pp. 1892–1917.
- [20] S. JIN AND Z. XIN, *Numerical passage from systems of conservation laws to Hamilton–Jacobi equations, and relaxation schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 2385–2404.
- [21] A. KURGANOV, S. NOELLE, AND G. PETROVA, *Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 23 (2001), pp. 707–740.
- [22] A. KURGANOV AND E. TADMOR, *New high-resolution semi-discrete central schemes for Hamilton–Jacobi equations*, J. Comput. Phys., 160 (2000), pp. 720–724.
- [23] A. KURGANOV AND E. TADMOR, *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282.
- [24] O. LEPSKY, C. HU, AND C.-W. SHU, *Analysis of the discontinuous Galerkin method for Hamilton–Jacobi equations*, Appl. Numer. Math., 33 (2000), pp. 423–434.
- [25] D. LEVY, G. PUPPO, AND G. RUSSO, *A fourth order central WENO scheme for multi-dimensional hyperbolic systems of conservation laws*, SIAM J. Sci. Comput., 24 (2002), pp. 480–506.
- [26] D. LEVY, G. PUPPO, AND G. RUSSO, *Central WENO schemes for hyperbolic systems of conservation laws*, Math. Model. Numer. Anal., 33 (1999), pp. 547–571.
- [27] D. LEVY, G. PUPPO, AND G. RUSSO, *Compact central WENO schemes for multidimensional conservation laws*, SIAM J. Sci. Comput., 22 (2000), pp. 656–672.
- [28] P.L. LIONS, *Generalized Solutions of Hamilton–Jacobi Equations*, Pitman, London, 1982.
- [29] P.L. LIONS AND P.E. SOUGANIDIS, *Convergence of MUSCL and filtered schemes for scalar conservation laws and Hamilton–Jacobi equations*, Numer. Math., 69 (1995), pp. 441–470.
- [30] C.-T. LIN AND E. TADMOR, *L^1 -stability and error estimates for approximate Hamilton–Jacobi solutions*, Numer. Math., 87 (2001), pp. 701–735.
- [31] C.-T. LIN AND E. TADMOR, *High-resolution nonoscillatory central schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2163–2186.
- [32] X.-D. LIU, S. OSHER, AND T. CHAN, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
- [33] H. NESSYAHU AND E. TADMOR, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 408–463.
- [34] 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.
- [35] S. OSHER AND C.-W. SHU, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 28 (1991), pp. 907–922.
- [36] J. SHI, C. HU, AND C.-W. SHU, *A technique of treating negative weights in WENO schemes*, J. Comput. Phys., 175 (2002), pp. 108–127.
- [37] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, J. Comput. Phys., 83 (1989), pp. 32–78.
- [38] P.E. SOUGANIDIS, *Approximation schemes for viscosity solutions of Hamilton–Jacobi equations*, J. Differential Equations, 59 (1985), pp. 1–43.
- [39] M. ZENNARO, *Natural continuous extensions of Runge–Kutta methods*, Math. Comp., 46 (1986), pp. 119–133.
- [40] 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.