Communications on Applied Mathematics and Computation
https://doi.org/10.1007/s42967-025-00482-6

ORIGINAL PAPER

A CWENO Large Time-Step Scheme for Hamilton-Jacobi Equations

Elisabetta Carlini[1] · Roberto Ferretti[2] · Silvia Preda[3] · Matteo Semplice[3]

Received: 8 June 2024 / Revised: 30 December 2024 / Accepted: 7 January 2025

Abstract
We propose a high-order numerical scheme for time-dependent first-order Hamilton-Jacobi-Bellman (HJB) equations. In particular, we propose to combine a semi-Lagrangian (SL) scheme with a Central Weighted Essentially Non-Oscillatory (CWENO) reconstruction. The CWENO method provides a non-oscillatory, high-order reconstruction polynomial that allows efficient evaluations at multiple reconstruction points, while the SL method ensures stability without any time-step restrictions. Together, they form a particularly effective framework for solving HJB equations. We prove a convergence result in the case of state- and time-independent Hamiltonians. Numerical simulations are presented in space dimensions one and two, also for more general state- and time-dependent Hamiltonians, demonstrating superior performance in terms of CPU time gain compared with a semi-Lagrangian scheme coupled with Weighted Non-Oscillatory reconstructions.

Keywords Semi-Lagrangian (SL) schemes · Hamilton-Jacobi (HJ) equations · Central Weighted Essentially Non-Oscillatory (CWENO) methods

Mathematics Subject Classification 65N12 · 65M10 · 49L25

✉ Roberto Ferretti
roberto.ferretti@uniroma3.it

Elisabetta Carlini
carlini@mat.uniroma1.it

Silvia Preda
silvia.preda@uninsubria.it

Matteo Semplice
matteo.semplice@uninsubria.it

[1] Department of Mathematics "G. Castelnuovo", "Sapienza" University of Rome, Rome, Italy

[2] Department of Mathematics and Physics, Roma Tre University, Rome, Italy

[3] Department of Science and High Technology, University of Insubria, Como, Italy

Published online: 09 April 2025

1 Introduction

In this paper, we address the numerical approximation of the following hyperbolic Hamilton-Jacobi-Bellman (HJB) equation:

$$\begin{cases} v_t(t,x) + H(t,x,\mathrm{D}v(t,x)) = 0, & \text{for } t,x \in (0,T)\times\mathbb{R}^d, \\ v(0,x) = v_0(x), & \text{for } x \in \mathbb{R}^d, \end{cases} \tag{1}$$

where $v:(0,T)\times\mathbb{R}^d \to \mathbb{R}$, $\mathrm{D}v$ stands for the spatial gradient, $v_0:\mathbb{R}^d \to \mathbb{R}$ denotes the initial data, and $H:(0,T)\times\mathbb{R}^d\times\mathbb{R}^d \to \mathbb{R}$ is the Hamiltonian function, which will be assumed to be convex with respect to $\mathrm{D}v$.

Numerous schemes have been proposed to approximate (1), but only a small number are aimed at high-order accuracy. High-order finite difference schemes based on Essentially Non-Oscillatory (ENO) reconstruction, defined in [23], were proposed in [32] and extended to unstructured grids in [1]. Second-order Godunov-type schemes based on global projection operators were discussed in [29]. Weighted ENO (WENO) schemes were introduced in [26, 30] and combined with finite difference schemes for Hamilton-Jacobi (HJ) equations in [25]. Recent high-order accurate finite difference WENO schemes can be found in [22, 27, 33, 34]. A fifth-order central scheme based on WENO reconstruction has been developed in [8].

Over the past 2 decades, semi-Lagrangian (SL) schemes have been employed to discretize the HJ equation. In [15], a high-order semi-discrete SL scheme was proposed to discretize stationary HJB equations, while in [16], SL schemes were applied to evolutionary HJB equations. A high-order SL scheme for HJB equations was presented in [10] by combining the SL technique with WENO reconstructions.

A different approach to achieve the high-order accuracy involves filtered schemes. In [5, 18], filtered schemes were combined with monotone finite difference schemes to discretize first-order evolutionary HJ equations.

For an overview of numerical methods for first-order HJ equations, we also refer to the book [17].

We will develop our theory for the HJB equation related to a finite horizon optimal control. More precisely, given a compact set $A \in \mathbb{R}^m$, denote by $\mathcal{A} = \{\alpha:[0,T]\to A,\ \text{measurable}\}$ the set of admissible controls. Given the running cost $f_C : (0,T)\times\mathbb{R}^d\times A \to \mathbb{R}$ and the dynamics $f_D : (0,T)\times\mathbb{R}^d\times A \to \mathbb{R}^d$ for the system

$$\dot{y}(s) = f_D(t-s, y(s), \alpha(s)), \quad s \in (0,t], \quad y(0) = x, \tag{2}$$

we consider the following Hamiltonian:

$$H(t,x,p) = \max_{a\in A}\{-f_D(t,x,a)\cdot p - f_C(t,x,a)\}. \tag{3}$$

Let us assume that

(H1) f_C, f_D are continuous and bounded; moreover, for every $a \in A$, the functions $f_C(\cdot,\cdot,a)$, $f_D(\cdot,\cdot,a)$ are Lipschitz continuous with Lipschitz constants independent of $a \in A$;

(H2) the initial data v_0 is Lipschitz continuous and bounded.

Under Assumptions **(H1)–(H2)**, problem (1) admits a unique viscosity solution v, which is Lipschitz continuous and bounded. Moreover, the Dynamic Programming Principle holds (see [4, Chapter 3], i.e., for any $h > 0$,

$$v(t,x) = \inf_{\alpha\in\mathcal{A}}\left\{\int_0^h f_C(t-s, y_{x,t}(s), \alpha(s))\mathrm{d}s + v(y_{x,t}(h), t-h)\right\}, \tag{4}$$

where $y_{x,t}$ denotes the solution to (2).

A numerical scheme based on the previous setting involves a minimization procedure over the set of controls; in turn, each function evaluation requires to approximate the cost integral and to interpolate the solution at the previous time step at the foot $y_{x,t}(h)$ of the characteristics. In this respect, WENO reconstructions, especially in higher space dimensions, are not efficient, due to the dependence of the WENO linear weights on the reconstruction point.

A novel paradigm for non-oscillatory reconstruction operators has been introduced in [28], where the authors suggested to blend, in a WENO-like fashion, polynomials of different degrees, allowing us to overcome some difficulties of non-existence, non-positivity, and dependence on the reconstruction point of the WENO linear weights. The idea has been further developed into the so-called Central WENO (CWENO) reconstruction and exploited in multiple spatial dimensions, as well as in the case of adaptive mesh refinement and non-uniform grids [2, 3, 14, 39, 40]. The technique has also been exploited in finite difference schemes for HJ equations on Cartesian meshes via dimensional splitting [38, 41], as well as on general meshes [42]. General results for establishing the convergence order of a CWENO reconstruction have been presented in [12, 13, 36].

One of the main advantages of CWENO over the traditional WENO is that the central approach provides a reconstruction polynomial that is defined everywhere in the reconstruction cell and that can be evaluated later, with no essential extra cost, at many different reconstruction points. This is guaranteed by the independence of the linear weights from the reconstruction point. Furthermore, as shown in [11, 12, 14, 35], the CWENO approach allows to avoid the dimensional splitting procedure and this is advantageous also on Cartesian meshes when the number of reconstruction points per cell is high.

In this paper, we want to exploit the positive features of CWENO to obtain a high-order SL scheme that is more efficient than the one in [10], which is based on WENO. The main novelty is to propose a new numerical method for first-order Hamilton Jacobi equations, based on the combination of an SL scheme with CWENO reconstructions. This approach offers the advantages of CWENO's high-order, non-oscillatory reconstructions while benefiting from the stability of SL schemes, which are unconditionally stable and allow for large time steps without restrictions. This method proves to be particularly robust for solving HJB equations, which in general have no smooth solutions. These equations have a wide range of applications, including optimal control problems, front propagation models, image processing, differential games, and mean field games, among others.

The rest of the paper is organized as follows. In Sect. 2, the general principles of the SL scheme are recalled, whereas in Sect. 3 a CWENO interpolation from point values is constructed in one- and two-space dimensions. Section 4 proves a convergence result in the framework of [20], and finally, Sect. 5 provides an extensive number of numerical examples to validate the scheme.

2 Numerical Scheme

To obtain an approximate version of (1), we need first to discretize the control problem in time. Let $\Delta t > 0$ be a time step, $t_n = n\Delta t$ a uniform time grid with $n = 0, \cdots, N_T$, where $N_T = \lceil \frac{T}{\Delta t} \rceil$, and consider the Dynamic Programming Principle on a single time step $[t_n, t_{n+1}]$, by choosing in (4) $h = \Delta t$ and $t = t_{n+1}$.

When using a ν-stages scheme for (2), we discretize an admissible control $\alpha \in \mathcal{A}$ via a sequence $\underline{a} = (a_1, a_2, \cdots, a_\nu) \in A^\nu$. Then the foot of the characteristic corresponding to the control $\underline{a}$ is

$$y^n(x, \underline{a}) \simeq y_{x,t_{n+1}}(\Delta t)$$

for any $n = 0, \cdots, N_T - 1$, $x \in \mathbb{R}^d$, and $\underline{a} \in A^\nu$.

Moreover, a suitable quadrature formula based on ν quadrature nodes is introduced to discretize the integral term related to the running cost

$$C^n(x, \underline{a}) \simeq \int_0^{\Delta t} f_C(t_{n+1} - s, y_{x,t_{n+1}}(s), \alpha(s))\, \mathrm{d}s,$$

for any $n = 0, \cdots, N_T - 1$, $x \in \mathbb{R}^d$, and $\underline{a} \in A^\nu$.

Let us now introduce the space grid with the space step Δx so that $x_i = i\Delta x = (i_1 \Delta x, \cdots, i_d \Delta x)$ with a multiindex $i \in \mathbb{Z}^d$, and denote $y_i^n(\underline{a}) = y^n(x_i, \underline{a})$ and $C_i^n(\underline{a}) = C^n(x_i, \underline{a})$. We approximate the solution $v(x, t)$ of (1) by a discrete function $u_i^n \simeq v(x_i, t_n)$ defined on the space-time grid, computed by the following iterative scheme, for $n = 0, \cdots, N_T - 1$:

$$\begin{cases} u_i^{n+1} = \min\limits_{\underline{a} \in A^\nu} \{R[u^n](y_i^n(\underline{a})) + C_i^n(\underline{a})\}, & i \in \mathbb{Z}^d, \\ u_i^0 = v_0(x_i), & i \in \mathbb{Z}^d, \end{cases} \tag{5}$$

where $R[u^n](x)$ denotes the spatial reconstruction at $x \in \mathbb{R}^d$ of the numerical solution $u^n = (u_i^n)_{i \in \mathbb{Z}^d}$.

For a given $\underline{a} \in A^\nu$, $y_i^n(\underline{a})$ is computed via a ν-stage Runge-Kutta (RK) method as follows:

$$\begin{cases} y_i^n(\underline{a}) = x_i + \Delta t \sum\limits_{k=1}^{\nu} b_k K_k(\underline{a}), \\ K_k(\underline{a}) = f_D\left(t_n + (1 - c_k)\Delta t, X_k(\underline{a}), a_k\right), \\ X_k(\underline{a}) = x_i + \Delta t \sum\limits_{j=1}^{k-1} A_{kj} K_j(\underline{a}), \end{cases} \tag{6}$$

where b_k, c_k, and A_{kj} are the coefficients of the Butcher tableau defining the RK method used to solve (2). For our purposes, we resort to the forward Euler method for first-order schemes, to the Heun method

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{7}$$

to get the second-order accuracy, and to the RK method with tableau

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \tag{8}$$

to get the third-order accuracy.

To compute $C_i^n(\underline{a})$, we employ a suitable quadrature rule and replace the function evaluations at the nodes with the numerical solution of (2):

$$\begin{aligned}
&\int_0^{\Delta t} f_C(t_{n+1}-s, y_{x_i,t_{n+1}}(s), \alpha(s))\mathrm{d}s \\
&= \Delta t \sum_k w_k f_C(t_{n+1}-\xi_k\Delta t, y_{x_i,t_{n+1}}(\xi_k\Delta t), \alpha(\xi_k\Delta t)) + \mathcal{O}(\Delta t^{Q+1}) \\
&= \Delta t \sum_k w_k f_C(t_{n+1}-\xi_k\Delta t, \tilde{y}_{x_i,t_{n+1}}(\xi_k\Delta t), \alpha(\xi_k\Delta t)) + \mathcal{O}(\Delta t^{Q+1}+\Delta t^{R+1}), \quad (9)
\end{aligned}$$

where ξ_k, w_k are the nodes and the weights of the quadrature rule and $\tilde{y}_{x_i,t_{n+1}}(\xi_k\Delta t)$ is a suitable approximation of the numerical solution of $y_{x_i,t_{n+1}}(s)$ for $s \in [0, \Delta t]$; Q is the accuracy of the quadrature rule and R describes the extra error introduced by the approximation of $y_{x_i,t_{n+1}}(\xi_k\Delta t)$ with $\tilde{y}_{x_i,t_{n+1}}(\xi_k\Delta t)$ within the quadrature rule. It is tempting to choose a quadrature rule whose nodes coincide with the abscissae $\{c_k\}$ of the RK scheme and replace, in the last approximation of (9), $\tilde{y}_{x_i,t_{n+1}}(\xi_k\Delta t)$ with the stage values of the RK scheme X_k and $\alpha(\xi_k\Delta t)$ with a_k, for $k = 1, \cdots, \nu$.

For the Heun method, the numerical computation of $C_i^n(\underline{a})$ would then be performed with the trapezoidal rule, whose nodes are at $\xi = 0, 1$; for this method $\tilde{y}_{x_i,t_{n+1}}(0) = x_i$ and $\tilde{y}_{x_i,t_{n+1}}(\Delta t) = X_2$ are computed with the correct accuracy by the RK scheme (7).

For the third-order RK scheme (8), whose nodes are at $\xi = 0, \frac{1}{2}, 1$, one can choose the approximations $\tilde{y}_{x_i,t_{n+1}}(0) = x_i$, $\tilde{y}_{x_i,t_{n+1}}(\frac{1}{2}\Delta t) = X_2$ and $\tilde{y}_{x_i,t_{n+1}}(\Delta t) = X_3$. In this special case, even if both $y_{x_i,t_{n+1}}(\xi_2\Delta t) - X_2$ and $y_{x_i,t_{n+1}}(\xi_3\Delta t) - X_3$ are $\mathcal{O}(\Delta x^2)$, these errors cancel each other out when computing the linear combination of Simpson's quadrature rule, resulting in a third-order accuracy for the approximation of $C_i^n(\underline{a})$. For more general RK processes, this issue can be overcome by resorting to the continuous extension [37] to compute $\tilde{y}_{x_i,t_{n+1}}(\xi)$ with the proper accuracy.

Summarizing, denote with $\Phi(\underline{a}) = R[u^n](y_i^n(\underline{a})) + C_i^n(\underline{a})$ the function to be minimized in (5). To compute the solution of (5), one has to find $a^* = \arg\min \Phi(\underline{a})$ and set $u_i^{n+1} = \Phi(a^*)$. To solve the minimization problem, we employ tabulation on a coarse grid in A, followed by a Nelder-Mead algorithm [21, 31], adjusted so that no vertex of the simplices could exit the compact set A. At each step of the minimization, we need to evaluate the reconstruction at the feet of the characteristics $y_i^n(\underline{a})$ for each node x_i and for each discrete control $\underline{a}$. Thus an efficient and accurate reconstruction operator is crucial to obtain a robust and fast numerical scheme. Since (5) has in general nonsmooth solutions, standard high-order interpolation gives rise to oscillations in the numerical solution. In [10], a WENO reconstruction has been applied to overcome this problem. On the other hand, the CWENO approach is more efficient when many evaluations of the reconstruction in each given cell are required. For this reason, we propose to apply CWENO interpolation within our SL scheme (5).

3 CWENO Reconstruction

We recall here the definitions of the CWENO and CWENOZ operators, following the presentation of [12, 13]; in particular, despite the present finite-difference setting in place of the finite-volume one, we will still be able to exploit the general theorems for the accuracy of the reconstructions on smooth solutions proven in the aforementioned papers.

For a point $x \in \mathbb{R}^d$, let Ω be the grid cell containing it and $\mathcal{S}_\Omega$ be the set of its vertices. To achieve better than first-order accuracy, we need to consider stencils $\mathcal{S} \supset \mathcal{S}_\Omega$. We associate to any $\mathcal{S}$ a polynomial $P_{\mathcal{S}}^{(r)}(x) \in \mathbb{P}_d^r$ which interpolates the data in $\mathcal{S}$, i.e., such that $P_{\mathcal{S}}^{(r)}(x_j) = u_j$ for any $j \in \mathcal{S}$. Larger symmetric stencils will define highly accurate interpolators, which however would be very oscillatory when the data in $\mathcal{S}$ represent a nonsmooth function. Instead, polynomials associated with smaller stencils, when biased in a specific direction, could avoid the discontinuities in the data.

The general idea of CWENO and CWENOZ reconstruction is to blend, in a nonlinear and data-dependent fashion, a high-order accurate interpolating polynomial with a set of lower-order ones to produce an essentially non-oscillatory reconstruction polynomial for the cell Ω. This can be later evaluated at any point in $x \in \Omega$ with negligible cost.

The nonlinear selection or blending of polynomials relies on oscillation indicators $\mathrm{I}[P]$, which are in general scalar quantities associated with a polynomial P, designed in such a way that $\mathrm{I}[P] \to 0$ under grid refinement, if P is associated with smooth data, and $\mathrm{I}[P] \asymp 1$ (i.e., the indicator is asymptotically a nonzero constant), in the presence of a discontinuity within the stencil $\mathcal{S}$. In this work, we rely on the classical Jiang-Shu oscillation indicators [26], suitably modified to evaluate the regularity of the solution of HJB problems under consideration [19, 25].

We recall now the definition of the CWENO and CWENOZ reconstructions given in [13].

Definition 1 Given a stencil $\mathcal{S}_{\mathrm{opt}}$, including $\mathcal{S}_\Omega$, let $P_{\mathrm{opt}} \in \mathbb{P}_n^G$ (optimal polynomial) be the polynomial of degree G, associated with $\mathcal{S}_{\mathrm{opt}}$. Further, let $P_1, P_2, \cdots, P_m$ be a set of $m \geqslant 1$ polynomials of degree g with $g < G$, associated with the substencil $\mathcal{S}_k$ such that $\mathcal{S}_\Omega \subset \mathcal{S}_k \subset \mathcal{S}_{\mathrm{opt}}$ for any $k = 1, \cdots, m$. Let also $\{d_k\}_{k=0}^m$ be a set of strictly positive real coefficients such that $\sum_{k=0}^m d_k = 1$.

The CWENO and CWENOZ operators compute a reconstruction polynomial

$$\begin{cases} P_{\mathrm{rec}}^{\mathrm{CW}} = \mathsf{CWENO}(P_{\mathrm{opt}}, P_1, \cdots, P_m) \in \mathbb{P}_n^G, \\ P_{\mathrm{rec}}^{\mathrm{CWZ}} = \mathsf{CWENOZ}(P_{\mathrm{opt}}, P_1, \cdots, P_m) \in \mathbb{P}_n^G \end{cases} \tag{10}$$

as follows:

(i) first, introduce the polynomial P_0 defined as

$$P_0(x) = \frac{1}{d_0}\left(P_{\mathrm{opt}}(x) - \sum_{k=1}^m d_k P_k(x)\right) \in \mathbb{P}_n^G; \tag{11}$$

(ii) compute suitable regularity indicators

$$\mathrm{I}_0 = \mathrm{I}[P_{\mathrm{opt}}], \quad \mathrm{I}_k = \mathrm{I}[P_k], \ k \geqslant 1; \tag{12}$$

(iii) compute the nonlinear coefficients $\{\omega_k\}_{k=0}^m$ or $\{\omega_k^Z\}_{k=0}^m$ as

(a) CWENO operator: for $k = 0, \cdots, m$,

$$\alpha_k = \frac{d_k}{(\mathrm{I}_k + \epsilon)^l}, \quad \omega_k^{\mathrm{CW}} = \frac{\alpha_k}{\sum_{i=0}^m \alpha_i}, \tag{13}$$

(b) CWENOZ operator: for $k = 0, \cdots, m$,

$$\alpha_k^Z = d_k\left(1 + \left(\frac{\tau}{\mathrm{I}_k + \epsilon}\right)^l\right), \quad \omega_k^{\mathrm{CWZ}} = \frac{\alpha_k^Z}{\sum_{i=0}^m \alpha_i^Z}, \tag{14}$$

where ϵ is a small positive quantity, $l \geqslant 1$, and, in the case of CWENOZ, τ is a global smoothness indicator;

(iv) finally, define the reconstruction polynomial as

$$P_{\text{rec}}^{\text{CW}}(x) = \sum_{k=0}^{m} \omega_k^{\text{CW}} P_k(x) \in \mathbb{P}_n^G, \tag{15}$$

$$P_{\text{rec}}^{\text{CWZ}}(x) = \sum_{k=0}^{m} \omega_k^{\text{CWZ}} P_k(x) \in \mathbb{P}_n^G. \tag{16}$$

Note that the reconstruction polynomial defined in (15) and (16) can be evaluated at any point in the computational cell Ω at a very low computational cost and this happens because the reconstruction procedure, in both cases, starts with the definition of the linear coefficients $\{d_k\}_{k=0}^m$ that do not depend on the reconstruction point. Therefore, the nonlinear coefficients given by (13) and (14) are computed once per cell and not once per reconstruction point, as in the standard WENO. We also remark that, since every polynomial considered in the reconstruction procedure is required to satisfy the interpolation constraint on $\mathcal{S}_\Omega$, also $P_{\text{rec}}^{\text{CW}}$ and $P_{\text{rec}}^{\text{CWZ}}$ satisfy the same constraint on the vertices of Ω.

The accuracy and non-oscillatory properties of CWENO and CWENOZ schemes are guaranteed by the dependence of their nonlinear weights (13) and (14) on the regularity indicators I_k. On smooth data, the nonlinear weights are driven sufficiently close to the optimal ones, so that $P_{\text{rec}} \approx P_{\text{opt}}$ and the reconstruction reaches the optimal order of accuracy $G+1$. On the other hand, when a discontinuity is present in $\mathcal{S}_{\text{opt}}$, both $\mathrm{I}_0 \asymp 1$ and at least one $\mathrm{I}_{\hat{k}} \asymp 1$ for some $\hat{k} \in \{1, \cdots, m\}$. Then the formulas (13) and (14) for the nonlinear weights will ensure that $\omega_0 \approx 0$ and $\omega_{\hat{k}} \approx 0$ for all $\hat{k}$ such that $P_{\hat{k}}$ would bring oscillations in the reconstruction. The reconstruction polynomial will then be a linear combination of all polynomials of degree g that are not affected by the discontinuity; the accuracy of the reconstruction thus reduces to $g+1$, but spurious oscillations would be tamed.

3.1 One Spatial Dimension

Let us describe the reconstruction for any point x in the cell $\Omega = [x_j, x_{j+1}]$, so that $\mathcal{S}_\Omega = \{j, j+1\}$. We consider $\mathcal{S}_{\text{opt}} = \{j-1, j, j+1, j+2\}$ and introduce the optimal cubic polynomial $P_{\text{opt}}(x) = Q(x) = \sum_{i=0}^{3} z_i[(x-x_j)/\Delta x]^i$ that interpolates the data $u_{j-1}, u_j, u_{j+1}, u_{j+2}$ at the nodes $x_{j-1}, x_j, x_{j+1}, x_{j+2}$. Next, we consider two parabolas $P_L(x)$ and $P_R(x)$ that interpolate only the data u_{j-1}, u_j, u_{j+1} and, respectively, u_j, u_{j+1}, u_{j+2}.

The reconstruction operators thus compute

$$\begin{cases} P_{\text{rec}}^{\text{CW}} = \text{CWENO}(Q, P_L, P_R) \in \mathbb{P}_1^3, \\ P_{\text{rec}}^{\text{CWZ}} = \text{CWENOZ}(Q, P_L, P_R) \in \mathbb{P}_1^3. \end{cases} \tag{17}$$

For any polynomial, its oscillation indicator is defined as

$$\mathrm{I}[P] = \sum_{\alpha \geqslant 2} \Delta x^{2\alpha-3} \int_{x_j}^{x_{j+1}} \left(\frac{\mathrm{d}^{(\alpha)} P}{\mathrm{d}x^\alpha} \right)^2 \mathrm{d}x. \tag{18}$$

The above definition is similar to the classical definition of the oscillation indicators for the WENO reconstruction as given in [26], except that the first derivative is not included in the

sum. This choice is the appropriate one for HJ equations, whose solution can be at worst continuous with kinks, see [25], and ensures that at worst $\mathrm{I}[P] = \mathcal{O}(1)$.

Let $P(x) = \sum_{i=0}^{3} z_i[(x - x_j)/\Delta x]^i$ be a polynomial of degree up to 3. Its indicator (18) can be written as a quadratic form of its coefficients given by

$$\mathrm{I}[P] = \frac{1}{\Delta x^2}(4z_2^2 + 12z_2z_3 + 48z_3^2)$$

or equivalently, denoting with $\boldsymbol{z}$ the vector of coefficients,

$$\mathrm{I}[P] = \frac{1}{\Delta x^2}\boldsymbol{z}^{\mathrm{T}}M\boldsymbol{z} \quad \text{with } M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 \\ 0 & 0 & 6 & 48 \end{pmatrix}. \tag{19}$$

Denoting by U the vector of data $(u_{j-1}, u_j, u_{j+1}, u_{j+2})^{\mathrm{T}}$, since the coefficients of the polynomial linearly depends on U, we can express the regularity indicators also as a quadratic form on the data being interpolated. More precisely, the vector of coefficients $\boldsymbol{z}$ can be expressed as $\boldsymbol{z} = V^{-1}U$, where V^{-1} is the inverse of the Vandermonde matrix. Thus, in the one-dimensional case, one obtains that the matrices of the quadratic forms expressed in terms of U scale again globally as $1/\Delta x^2$:

$$\mathrm{I}[Q] = \frac{1}{\Delta x^2}U^{\mathrm{T}}A_QU, \quad \mathrm{I}[P_k] = \frac{1}{\Delta x^2}U^{\mathrm{T}}A_kU \quad (k = L, R). \tag{20}$$

In our case, we get

$$A_Q = \begin{pmatrix} 4/3 & -7/2 & 3 & -5/6 \\ -7/2 & 10 & -19/2 & 3 \\ 3 & -19/2 & 10 & -7/2 \\ -5/6 & 3 & -7/2 & 4/3 \end{pmatrix},$$

$$A_L = \begin{pmatrix} 1 & -2 & 1 & 0 \\ -2 & 4 & -2 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & -2 & 4 & -2 \\ 0 & 1 & -2 & 1 \end{pmatrix}.$$

The CWENO reconstruction applied in the numerical tests of this paper is then defined by choosing linear coefficients $d_L = d_R = \frac{1}{8}$, $d_0 = \frac{3}{4}$, $l = 2$, and $\epsilon = \Delta x^2$ in the above construction.

For the optimal definition of the CWENOZ reconstruction, in order to exploit the results of [13], we need to study the Taylor expansions of the indicators centered at the point $x_0 = (x_j + x_{j+1})/2$, namely the center of the reconstruction cell Ω. They are given by

$$\begin{cases} \mathrm{I}[P_L] = B - u''(x_0)u'''(x_0)\Delta x^3 + \frac{1}{4}u'''(x_0)^2\Delta x^4 + \mathcal{O}(\Delta x^5), \\ \mathrm{I}[P_R] = B + u''(x_0)u'''(x_0)\Delta x^3 + \frac{1}{4}u'''(x_0)^2\Delta x^4 + \mathcal{O}(\Delta x^5), \\ \mathrm{I}[P_L] = B + \frac{13}{12}u'''(x_0)^2\Delta x^4 + \mathcal{O}(\Delta x^5), \end{cases} \tag{21}$$

where $B = u''(x_0)^2\Delta x^2$. Thus, defining

$$\tau = \Big|2\mathrm{I}[Q] - \mathrm{I}[P_L] - \mathrm{I}[P_R]\Big|, \tag{22}$$

all terms up to $\mathcal{O}(\Delta x^3)$ cancel. Note that this is the optimal definition of τ since it is never possible to cancel all the $\mathcal{O}(\Delta x^4)$ terms in a convex combination of the three indicators. The

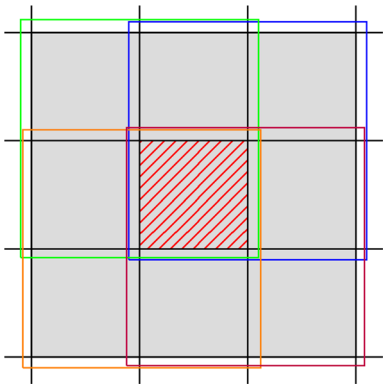

Fig. 1 Stencils of the two-dimensional CWENO and CWENOZ reconstructions. The red-hatched region represents the cell $\Omega_{i,j}$ in which we compute the reconstruction. The vertices of $\mathcal{S}_{\text{opt}}$ are enclosed in the grey shaded region, while the stencils for the low-degree polynomials are enclosed in the coloured squares: blue, green, orange, and purple, respectively for the north-east, north-west, south-west, and south-east polynomials

hypotheses of Theorem 24 in [13] hold and we can guarantee the optimal order of accuracy of the CWENOZ reconstruction for $l = 2$.

3.2 Two Spatial Dimensions

In two-space dimensions, we will not rely on dimensional splitting. Such an approach would in fact nullify the advantages of a fast evaluation of the reconstruction polynomial on a large number of arbitrary points in the reconstruction cell. Our reconstruction will instead generate a polynomial of two variables, like it is done in [11, 35] for Cartesian meshes or in general mesh settings [3, 14, 42].

Let Ω be the cell of the Cartesian grid containing the reconstruction point x. The setup of the stencils for the reconstruction is illustrated in Fig. 1: the cell Ω, with corners (x_i, y_j) and (x_{i+1}, y_{j+1}), is hatched in red and the reconstruction stencil is shaded in grey.

The reconstruction operators compute

$$\begin{cases} P_{\text{rec}}^{\text{CW}} = \text{CWENO}(Q^{(3)}; Q_{\text{ne}}^{(2)}, Q_{\text{se}}^{(2)}, Q_{\text{sw}}^{(2)}, Q_{\text{nw}}^{(2)}) \in \mathbb{P}_1^3 \otimes \mathbb{P}_1^3, \\ P_{\text{rec}}^{\text{CWZ}} = \text{CWENOZ}(Q^{(3)}; Q_{\text{ne}}^{(2)}, Q_{\text{se}}^{(2)}, Q_{\text{sw}}^{(2)}, Q_{\text{nw}}^{(2)}) \in \mathbb{P}_1^3 \otimes \mathbb{P}_1^3, \end{cases} \tag{23}$$

with the optimal polynomial set to the bicubic polynomial interpolating the 16 data points in the square with corners (x_{i-1}, y_{j-1}) and (x_{i+2}, y_{j+2}) enclosed in the grey shaded region of Fig. 1. The other four polynomials are the biquadratic polynomials that interpolate the four 3×3 substencils in the north-east, south-east, south-west, and north-west directions, as illustrated in Fig. 1, named $\mathcal{S}_{\text{ne}}$, $\mathcal{S}_{\text{se}}$, $\mathcal{S}_{\text{sw}}$, and $\mathcal{S}_{\text{nw}}$, respectively. In particular, the blue box with vertices (x_i, y_j) and (x_{i+2}, y_{j+2}) encloses $\mathcal{S}_{\text{ne}}$, the green box with vertices (x_{i-1}, y_j) and (x_{i+1}, y_{j+2}) encloses $\mathcal{S}_{\text{nw}}$, the orange box with vertices (x_{i-1}, y_{j-1}) and (x_{i+1}, y_{j+1}) encloses $\mathcal{S}_{\text{sw}}$, and the purple box with vertices (x_i, y_{j-1}) and (x_{i+2}, y_{j+1}) encloses $\mathcal{S}_{\text{se}}$. The basis for bicubic and biquadratic polynomials are defined by tensorization and are thus respectively given by

$$\mathcal{B}_3 = \{1, \hat{x}, \hat{y}, \hat{x}^2, \hat{x}\hat{y}, \hat{y}^2, \hat{x}^3, \hat{x}^2\hat{y}, \hat{x}\hat{y}^2, \hat{y}^3, \hat{x}^3\hat{y}, \hat{x}^2\hat{y}^2, \hat{x}\hat{y}^3, \hat{x}^3\hat{y}^2, \hat{x}^2\hat{y}^3, \hat{x}^3\hat{y}^3\},$$
$$\mathcal{B}_2 = \{1, \hat{x}, \hat{y}, \hat{x}^2, \hat{x}\hat{y}, \hat{y}^2\},$$

where $\hat{x} = (x - x_i)/\Delta x$ and $\hat{y} = (y - y_j)/\Delta y$.

The oscillation indicators are defined as

$$\mathrm{I}[P] = \sum_{|\vec{\alpha}|\geqslant 2} \Delta^{2|\boldsymbol{\alpha}|-4} \int_{\Omega_{i,j}} \left[\frac{\partial P}{\partial x^{\alpha_1} \partial y^{\alpha_2}}\right]^2 \mathrm{d}x\mathrm{d}y, \tag{24}$$

where Δ is the diameter of the cell $\Omega_{i,j}$. These indicators are inspired by the classical ones of [24], but the powers of Δ appearing in (24) have been adjusted for the expected regularity of the solution to HJ equations. In particular, the scaling $\Delta^{2|\boldsymbol{\alpha}|-4}$ is a generalization to two-space dimensions of the choice of [25]; it is introduced so that $\mathrm{I}[P] \asymp 1$ for functions with discontinuous first derivative and $\mathrm{I}[P] = \mathcal{O}(\Delta^2)$ on regular solutions. A detailed discussion of this scaling can be found in [19].

Denoting with z the vector of coefficients along the basis $\mathcal{B}_3$ of a generic polynomial in two variables of degree up to 3, and considering a uniform grid of size Δx, its oscillation indicator can be expressed as

$$\mathrm{I}[P] = \frac{1}{\Delta x^2} z^{\mathrm{T}} M z \tag{25}$$

for a 16×16 matrix M, which is reported in Appendix B. We point out that in the case of Cartesian but uneven grids, there would still be a global factor inversely proportional to the local grid size, but matrix M would also depend on the aspect ratios and on the ratios of sizes of nearby cells.

Consider now the stencil of the reconstruction given by $\{(x_{i-1+k}, y_{j-1+l})\}_{\substack{l=0,\cdots,3\\k=0,\cdots,3}}$ and denote by U the corresponding vector of values $\{(u_{i-1+k}, y_{j-1+l})\}_{\substack{l=0,\cdots,3\\k=0,\cdots,3}}$ with a prescribed ordering, e.g., lexicographic. Thanks to the linear relation among the coefficients z and the data vector U, the regularity indicators can be expressed in the form

$$\mathrm{I}[Q] = \frac{1}{\Delta x^2} U^{\mathrm{T}} A_Q U, \quad \mathrm{I}[Q_k] = \frac{1}{\Delta x^2} U^{\mathrm{T}} A_k U \quad (k = \mathrm{ne}, \mathrm{se}, \mathrm{sw}, \mathrm{nw}). \tag{26}$$

The matrices A_Q and A_k will in general depend on the local aspect ratio of cells and on the neighbours size ratio. Their entries are reported in the appendix for the case of a uniform Cartesian grid.

The CWENO reconstruction is then defined by choosing linear coefficients $d_k = 1/16$ and $d_0 = 1 - \sum_k d_k$ with $k \in \{\mathrm{ne}, \mathrm{se}, \mathrm{sw}, \mathrm{nw}\}$, $l = 2$, and $\epsilon = \Delta x^2$.

The CWENOZ reconstruction is computed choosing $l = 2$ and

$$\tau = \left|4\mathrm{I}[Q^{(3)}] - \mathrm{I}[Q^{(2)}_{\mathrm{ne}}] - \mathrm{I}[Q^{(2)}_{\mathrm{se}}] - \mathrm{I}[Q^{(2)}_{\mathrm{sw}}] - \mathrm{I}[Q^{(2)}_{\mathrm{nw}}]\right| \tag{27}$$

to guarantee the optimal order of accuracy of the CWENOZ reconstruction, see [13, Theorem 24]. In fact, computing the Taylor expansions of the indicators centered at the center of the cell Ω, one gets, for smooth enough data,

$$\begin{cases} \mathrm{I}[Q^{(2)}_{\mathrm{ne}}] = B + 2u_{xx}u_{xxx}\Delta x^3 + 2u_{yy}u_{yyy}\Delta x^3 + \mathcal{O}(\Delta x^4), \\ \mathrm{I}[Q^{(2)}_{\mathrm{nw}}] = B - 2u_{xx}u_{xxx}\Delta x^3 + 2u_{yy}u_{yyy}\Delta x^3 + \mathcal{O}(\Delta x^4), \\ \mathrm{I}[Q^{(2)}_{\mathrm{sw}}] = B - 2u_{xx}u_{xxx}\Delta x^3 - 2u_{yy}u_{yyy}\Delta x^3 + \mathcal{O}(\Delta x^4), \\ \mathrm{I}[Q^{(2)}_{\mathrm{se}}] = B + 2u_{xx}u_{xxx}\Delta x^3 - 2u_{yy}u_{yyy}\Delta x^3 + \mathcal{O}(\Delta x^4), \\ \mathrm{I}[Q^{(3)}] = B + \mathcal{O}(\Delta x^4), \end{cases} \tag{28}$$

where $B = (u_{xx}^2 + u_{xy}^2 + u_{yy}^2)\Delta x^2$. Therefore, the coefficients chosen for the definition of τ in (27), cancel all terms up to $\mathcal{O}(\Delta x^3)$, thus ensuring that the hypotheses of [13, Theorem 24] hold.

4 Convergence

The convergence analysis will be carried out in one-space dimension, and follow the guidelines of [10, 20], which will be briefly reviewed here. Assume that (1) is posed on $\mathbb{R}$ in the simplified form

$$\begin{cases} v_t(t,x) + H(v_x(t,x)) = 0, & t,x \in (0,T)\times\mathbb{R}, \\ v(0,x) = v_0(x), & x \in \mathbb{R}. \end{cases} \tag{29}$$

Also assume that the Hamiltonian function H is a $W^{2,\infty}$ function and that

$$H''(p) \geqslant m_H > 0. \tag{30}$$

The convexity assumption (30) implies, by the Fenchel duality formula, that $H(\cdot)$ can always be written in the form

$$H(p) = \sup_{a\in\mathbb{R}}\{a\,p - H^*(a)\},$$

where H^* denotes the Legendre transform of H. Moreover, the supremum is a maximum, and its computation can be limited to a suitable compact set A, so that problem (29) can be recast in the case of a Hamiltonian of the form (3), by setting $f_D(t,x,a) = -a$ and $f_C(t,x,a) = H^*(a)$.

The key assumption is that for any Lipschitz continuous function $v(x)$, once defined the sequence $V = \{v_j\}_j = \{v(x_j)\}_j$, the interpolation operator $R[V]$ satisfies $R[V](x_k) = v(x_k)$ and, for some constant $C < 1$,

$$|R[V](x) - R_1[V](x)| \leqslant C_r \max_{x_k\in U(x)} |v_{k+1} - 2v_k + v_{k-1}|, \tag{31}$$

where by $R_1[V]$ we denote the $\mathbb{P}_1$ (i.e., piecewise linear) interpolation on the sequence V, and by $U(x) = (x - h_-\Delta x, x + h_+\Delta x)$ the interval containing the stencil of the reconstruction $R[V](x)$. For example, in our case the reconstruction is performed by taking two nodes on the left and two nodes on the right of the point x, so that $h_- = h_+ = 2$.

Under such assumptions it is possible (see [20]) to prove the following theorem.

Theorem 1 *Consider the scheme* (5) *applied to* (29), *and assume that* (30) *and* (31) *hold*, $\Delta x = \mathcal{O}(\Delta t^2)$, *and* v_0 *is Lipschitz continuous. Then, the numerical solution* $U^n = \{u_i^n\}_i$ (*with* u_i^n *defined by* (5)) *satisfies*

$$\|R[U^n] - v(n\Delta t)\|_\infty \to 0$$

(*where* v *is the solution of* (29)) *for* $0 \leqslant n \leqslant T/\Delta t$, *as* $\Delta t \to 0$.

It is proved in [20] that condition (31) is satisfied for Lagrange reconstructions up to the fifth order, if the reconstruction stencil includes the interval $[x_j, x_{j+1}]$. Since the WENO interpolation is performed by taking a convex combination of polynomials which satisfy (31) up the degree 5 for the partial polynomials (and therefore, up to the degree 9 for the global interpolant), this fact is used in [10] to obtain convergence via Theorem 1. This essentially boils down to proving that all the functions appearing in the convex combination defining R satisfy (31), and we plan to apply the same principle here.

In the case of CWENO, once set $d_L = d_R = d$ and recast P_0 as

$$P_0(x) = \frac{1}{1-2d}(Q(x) - d\,P_L(x) - d\,P_R(x)),$$

we estimate the left-hand side of (31) as

$$\begin{aligned} |R - R_1| &= |\omega_0 P_0 + \omega_L P_L + \omega_R P_R - R_1| \\ &= |\omega_0 (P_0 - R_1) + \omega_L (P_L - R_1) + \omega_R (P_R - R_1)| \\ &\leqslant \max\{|P_0 - R_1|, |P_L - R_1|, |P_R - R_1|\}, \end{aligned}$$

where the identity $\omega_0 + \omega_L + \omega_R = 1$ has been used twice. As it has been proved in [20], both P_L and P_R satisfy (31). It remains then to check that the same property holds for P_0. We have

$$\begin{aligned} |P_0 - R_1| &= \left| \frac{1}{1-2d}(Q - dP_L - dP_R) - R_1 \right| \\ &= \left| \frac{1}{1-2d}(Q - dP_L - dP_R + (2d-1)R_1) \right| \\ &= \left| \frac{1}{1-2d}\big((Q - R_1) - d(P_L - R_1) - d(P_R - R_1)\big) \right| \\ &\leqslant \frac{1}{1-2d}|Q - R_1| + \frac{d}{1-2d}|P_L - R_1| + \frac{d}{1-2d}|P_R - R_1|. \end{aligned}$$

Denoting now by C_r the constant appearing in (31) for the specific case of an interpolation of degree r, we finally obtain an estimate in the form

$$|P_0 - R_1| \leqslant \left(\frac{1}{1-2d} C_3 + \frac{2d}{1-2d} C_2 \right) \max_{x_k \in U(x)} |v_{k+1} - 2v_k + v_{k-1}|.$$

As proved in [20], $C_2 = 1/8$ and $C_3 \approx 0.253\,3$; then, it turns out that (31) is satisfied for $d \lesssim 0.332$, and therefore, with this additional condition, all the assumptions of Theorem 1 are satisfied, and the scheme (5) converges to the viscosity solution of (29).

We point out that our choice of $d_0 = 3/4$, i.e., $d = d_L = d_R = 1/8$ in the CWENO and CWENOZ reconstructions employed in this paper fully satisfies the above requirement.

5 Numerical Tests

Here, we present some numerical tests to assess the performance of the schemes proposed in this paper. In particular, we focus on the expected accuracy of the schemes, considering the L^1-norm of the error, and the computational times, reported in seconds. In particular, we will compare the SL schemes using the CWENO and CWENOZ reconstructions of Sect. 3 and those employing the dimensionally-split WENO approach of [10].

Although the convergence analysis requires a relation $\Delta x = \mathcal{O}(\Delta t^2)$ between the time and space steps, in practice, smaller time steps are also allowed. Heuristically, assuming exact minimization, and supposing a third-order time discretization is used, then the consistency error is given by

$$\mathcal{T}(\Delta t, \Delta x) = \mathcal{O}\left(\Delta t^3 + \frac{\Delta x^q}{\Delta t} \right),$$

where q is the order of accuracy of the interpolation in space. For smooth solutions, the choice $\Delta t = \mathcal{O}\left(\Delta x^{q/4}\right)$ optimizes this error and leads to a numerical convergence rate of $3q/4$.

In particular, choosing $q = 4$ (i.e., a cubic reconstruction), we obtain

$$\mathcal{T}(\Delta t, \Delta x) = \mathcal{O}\left(\Delta t^3 + \frac{\Delta x^4}{\Delta t}\right),$$

which results in an overall order of 3 for the choice $\Delta t = \mathcal{O}(\Delta x)$. In the particular case where the Hamiltonian is independent of (t, x), the characteristics solving problem (2) are affine, and even a first-order time discretization computes them exactly, reducing the consistency error to

$$\mathcal{T}(\Delta t, \Delta x) = \mathcal{O}\left(\frac{\Delta x^q}{\Delta t}\right).$$

In this case, the larger the time step, the smaller the consistency errors.

In what follows N indicates the number of grid points per direction, so that $\Delta x \sim 1/N$. The tests have been performed on the cluster Galileo 100 hosted at CINECA.[1]

Boundary Condition

Boundary conditions are challenging to handle for SL schemes because the characteristics may exit the domain. In such cases, they need to be properly truncated or reflected back inside the domain. In our paper, we consider either periodic boundary conditions or Dirichlet boundary conditions. Periodic boundary conditions are implemented by treating the computational domain as a torus, which allows for mapping the characteristics back into the domain whenever they exit its boundaries. Dirichlet boundary conditions are handled using an extrapolation technique. When characteristics exit the domain, their values are determined by extrapolating from the cell where the characteristics leave the domain. It is noteworthy that in all the numerical tests where extrapolation is applied, the solution has a compact support, or the boundary of the numerical domain is entirely of outflow type. In both cases, these conditions ensure no loss in accuracy in the numerical treatment of the boundary. In all tests, unless otherwise stated, we employ an extrapolation technique. For more general boundary conditions, we refer to [9] for approximations of Neumann-type boundary conditions and to [7] for a second-order accurate treatment of Dirichlet boundary conditions.

Test 1: Passive advection.

As a first benchmark problem, we consider the uniform rotation of a scalar field in two-space dimensions, expressed by the linear transport equation

$$v_t - f_D \cdot Dv = 0,$$

which is included in our setting by choosing in (3) $A = \varnothing$ and $f_C = 0$. The speed of propagation is given by the vector field $f_D = (-2\,\pi(x_2 - 0.5), 2\,\pi(x_1 - 0.5))$ in the square domain $[0, 1] \times [0, 1]$. We choose as the initial condition the global $\mathcal{C}^2$ function

$$v(0, x) = v_0(r) = M\left(1 + \frac{r^3}{R^3}\left(-1 + 3\,\frac{r - R}{R}\left(1 - 2\,\frac{r - R}{R}\right)\right)\right)$$

with $r = |x - (0.3, 0.7)|$ and parameters $M = 0.15$, $R = 0.15$. Numerical solutions are computed at the final time $T = 1$, using a relationship of $\Delta t = 3\Delta x$ among the discretization steps. Since the advection is rigid, the exact solution at the final time coincides with the initial condition. We point out that the characteristic lines are not affine, therefore we use the third-order RK scheme and a third-order reconstruction scheme as described in Sects. 2 and 3, respectively. Plots of the initial data and of the numerical solution are shown in Fig. 2.

As expected, since the transport is rigid and the initial data is globally $\mathcal{C}^2$, all the schemes (WENO, CWENO, and CWENOZ) yield solutions converging with the same order, see Table 1.

[1] https://www.hpc.cineca.it/systems/hardware/galileo100/.

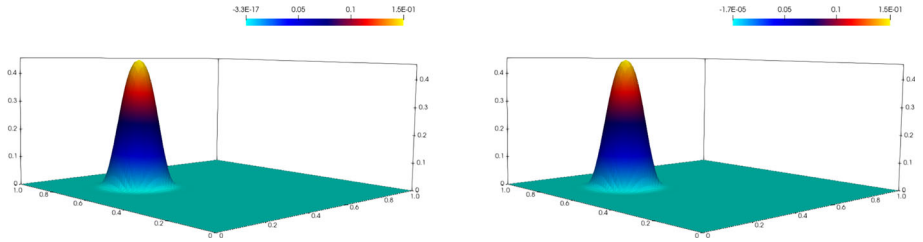

Fig. 2 Initial condition (left) and numerical solution at $T = 1$ (right) for Test 1 computed with CWENOZ reconstruction and $N = 81$

Table 1 Errors at time $T = 1$ for Test 1, WENO, CWENO, and CWENOZ schemes

N	WENO		CWENO		CWENOZ	
	L^1-err	L^1-ord	L^1-err	L^1-ord	L^1-err	L^1-ord
21×21	2.97E−03		2.98E−03		2.98E−03	
41×41	5.46E−04	2.44	4.88E−04	2.61	4.84E−04	2.62
81×81	1.24E−04	2.13	8.18E−05	2.57	7.85E−05	2.62
161×161	2.34E−05	2.41	1.22E−05	2.75	1.12E−05	2.80
321×321	3.92E−06	2.58	1.72E−06	2.82	1.49E−06	2.92
641×641	6.11E−07	2.68	2.27E−07	2.92	1.79E−07	3.05
$1\,281 \times 1\,281$	8.87E−08	2.78	2.92E−08	2.96	2.06E−08	3.12

Table 2 CPU times for Test 1, WENO and CWENO schemes

Grid	WENO	CWENO		CWENOZ	
	CPU time	CPU time	% gain	CPU time	% gain
21×21	1.05E+00	7.22E−01	31.21	1.01E−01	90.36
41×41	1.10E+00	1.06E+00	3.39	7.20E−01	34.25
81×81	7.84E+00	5.42E+00	30.87	5.46E+00	30.39
161×161	6.19E+01	4.36E+01	29.51	4.37E+01	29.35
321×321	4.86E+02	3.38E+02	30.47	3.36E+02	30.95
641×641	3.87E+03	2.70E+03	30.31	2.67E+03	30.99
$1\,281 \times 1\,281$	3.09E+04	2.12E+04	31.25	2.12E+04	31.32

However, central reconstructions are more accurate with respect to the traditional WENO one; in particular, CWENOZ achieves the best results: its errors are about half the errors of WENO. The fundamental difference among the schemes is in their computational costs, which are shown in Table 2. The high number of reconstructions performed in each cell (see Fig. 3) allows the CWENO and CWENOZ to save about 30% of the time. This happens because CWENO reconstructions compute their nonlinear coefficients only once per cell instead of once per reconstruction point. Looking at Fig. 3, note in particular that the high number of reconstructions performed near the boundary is related to the boundedness of the domain: in fact, except for the case of periodic boundary conditions, each time a foot of a characteristic falls outside the domain, it is projected back in the first cell near the boundary to compute the reconstruction.

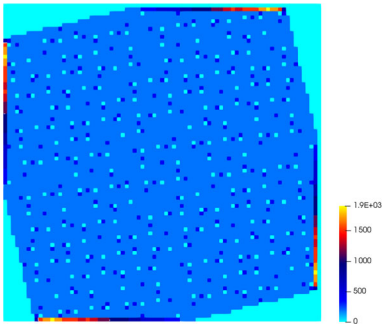

Fig. 3 Reconstructions count for Test 1 in the final time step on a grid of 81×81 points

Test 2: One-dimensional semi-concave data.

This test deals with the HJ equation

$$\begin{cases} v_t(t,x) + \frac{1}{2}|\partial_x v(t,x)|^2 = 0, \\ v(0,x) = v_0(x) = \min(-\cos(\pi x/2), 0) \end{cases} \tag{32}$$

in the domain $[-2, 2]$ with homogeneous boundary conditions, which are treated using the extrapolation technique. The approximate solution is computed, at $T = 1$, with $\Delta t = 10\Delta x$, while the exact solution $v(t,x)$ is defined as follows:

$$v(t,x) = \min\left(0, \frac{1}{2}[a^*(t,x)]^2 + v_0(x - ta^*(t,x))\right), \tag{33}$$

where, for a given (t,x), $a^*(t,x)$ represents the optimal control, which is constant along the characteristics, and it can be computed as the solution of

$$a^*(t,x) = g(t,x,a^*(t,x)) \tag{34}$$

with the function g defined by

$$g(t,x,a^*(t,x)) = \begin{cases} -\frac{2}{\pi t}\arcsin\frac{2a^*(t,x)}{\pi} + \frac{x}{t} & \text{if } |a^*(t,x)| \leqslant \frac{\pi}{2}, \\ \frac{\pi}{2} & \text{if } a^*(t,x) > \frac{\pi}{2}, \\ -\frac{\pi}{2} & \text{if } a^*(t,x) < -\frac{\pi}{2}. \end{cases} \tag{35}$$

For our purposes, since the exact solution of (34) is not available, we compute an approximate solution by using a fixed point algorithm with the initial guess $a_0^*(t,x) = \frac{1}{2}x^2 t$.

Since the characteristics are affine, in this test we have combined the third-order spatial reconstructions with first-order RK time stepping, which computes these curves exactly. Plots of the numerical and exact solutions are shown in the left panel of Fig. 4 and L^1 errors are listed in Table 3. In this case too, all schemes compute approximations affected by errors of the same order but the central reconstructions turn out to be more accurate (errors are about 2/3 than the WENO case). In fact, the feet of the characteristics fall in regular regions of v at each time step, and therefore all the reconstructions are close to the optimal cubic one. Table 4 reports the computational costs of each scheme, showing an average gain of $16\% - 19\%$ for both the CWENO reconstructions. This is again the effect of multiple reconstructions performed in each cell (see the right panel of Fig. 4, in which the number of reconstructions performed in each cell during the last time step is depicted). We point out that, as in Test 1, the high number of reconstructions computed in the first and in the last cell is related to the boundary conditions.

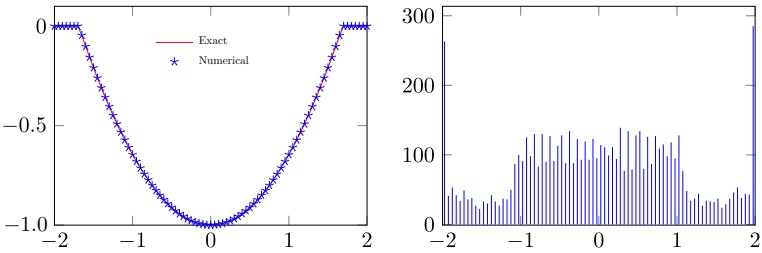

Fig. 4 Left: the exact solution of Test 2 in red and the numerical one in blue computed with a third-order CWENOZ reconstruction and $N = 81$. Right: reconstructions count in the last time step

Table 3 Errors at time $T = 1$ for Test 2, WENO, CWENO, and CWENOZ schemes

N	WENO		CWENO		CWENOZ	
	L^1-err	L^1-ord	L^1-err	L^1-ord	L^1-err	L^1-ord
81	3.56E−06		2.24E−06		1.78E−06	
161	2.83E−07	3.65	1.80E−07	3.64	1.44E−07	3.63
321	2.45E−08	3.53	1.59E−08	3.50	1.30E−08	3.47
641	2.61E−09	3.23	1.96E−09	3.02	1.75E−09	2.90

Table 4 CPU times for Test 2, WENO and CWENO schemes

Grid	WENO	CWENO		CWENOZ	
	CPU time	CPU time	% gain	CPU time	% gain
81	6.19E−03	4.83E−03	21.95	4.86E−03	21.48
161	9.79E−03	8.88E−03	9.31	8.64E−03	11.76
321	2.74E−02	2.39E−02	13.03	2.39E−02	12.84
641	9.42E−02	7.65E−02	18.81	7.68E−02	18.41
1 281	3.48E−01	2.83E−01	18.65	2.90E−01	16.72
2 561	1.33E+00	1.10E+00	17.15	1.10E+00	16.83
5 121	5.28E+00	4.30E+00	18.51	4.29E+00	18.77
10 241	2.09E+01	1.68E+01	19.42	1.68E+01	19.46

Test 3: One-dimensional eikonal equation.

In this test, we consider the HJ equation

$$\begin{cases} v_t(t,x) + \frac{1}{2}|\partial_x v(t,x)|^2 - f(t,x) = 0, \\ v(0,x) = v_0(x) \end{cases} \tag{36}$$

with $f(t,x) = -\sin(x) + (\frac{9}{8} + \frac{t^2-3t}{2})\cos^2(x)$ and $v_0(x) = \frac{3}{2}\sin(x)$. We consider the problem in $[0, 2\pi]$ with periodic boundary conditions and compute the numerical solution at time $T = 0.5$ with $\Delta t = \Delta x$. This problem has a known exact solution given by $v(t,x) = (\frac{3}{2} - t)\sin(x)$.

In this case, characteristics are not affine and both the dynamic and the cost function explicitly depend on x and t. Therefore, this is a specific test for the third-order accuracy

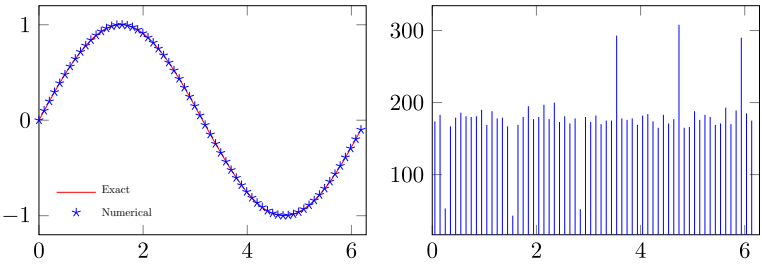

Fig. 5 Left: the exact solution of Test 3 in red and the numerical one in blue computed with a third-order CWENOZ scheme and $N = 63$. Right: reconstructions count in the last time step

Table 5 Errors at time $T = 0.5$ for Test 3, WENO, CWENO, and CWENOZ schemes

N	WENO		CWENO		CWENOZ	
	L^1-err	L^1-ord	L^1-err	L^1-ord	L^1-err	L^1-ord
126	1.28E−05		1.38E−05		1.43E−05	
252	1.54E−06	3.05	1.65E−06	3.07	1.70E−06	3.07
503	1.93E−07	3.00	2.03E−07	3.02	2.10E−07	3.02
1 006	2.56E−08	2.91	2.63E−08	2.95	2.71E−08	2.95

Table 6 CPU times for Test 3, WENO and CWENO schemes

Grid	WENO	CWENO		CWENOZ	
	CPU time	CPU time	% gain	CPU time	% gain
126	5.92E−02	5.42E−02	8.43	5.64E−02	4.81
252	2.11E−01	1.94E−01	8.04	1.94E−01	8.05
503	7.97E−01	7.37E−01	7.53	7.48E−01	6.05
1 006	3.13E+00	2.83E+00	9.75	2.87E+00	8.45
2 011	1.19E+01	1.11E+01	6.67	1.11E+01	6.79
4 022	4.73E+01	4.27E+01	9.58	4.21E+01	11.03
8 043	1.80E+02	1.65E+02	8.63	1.62E+02	10.02
16 085	6.96E+02	6.41E+02	7.96	6.38E+02	8.35

of our scheme, which depends on the use of the third-order RK scheme (8) and on the approximation of the integral of the cost function.

Plots of the exact and the numerical solution are shown in the left panel of Fig. 5, while in the right panel the reconstruction count in the last time step is depicted. We point out that in this case, with periodic boundary conditions, there is no artificial increase in the first and the last cells near the boundary.

Errors and CPU times are listed in Tables 5 and 6. As in the previous tests, due to the regularity of the exact solution, all schemes perform almost equally when compared in terms of the error. Looking at the computational costs, the CWENO reconstructions allow for a saving of about 6%−10%.

Test 4: Two-dimensional semi-convex data.

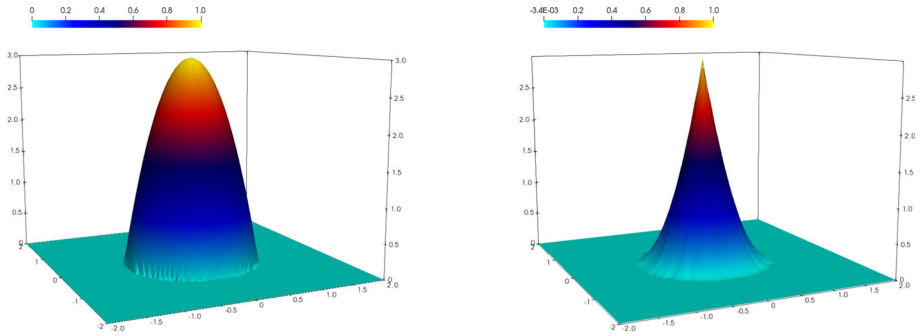

Fig. 6 Test 4. Initial condition (left) and numerical solution computed with CWENO at $T = 0.5$ (right) with $N = 81$

Table 7 Errors at time $T = 0.5$ for Test 4, WENO, CWENO, and CWENOZ schemes

N	WENO		CWENO		CWENOZ	
	L^1-err	L^1-ord	L^1-err	L^1-ord	L^1-err	L^1-ord
41×41	4.02E−02		3.38E−02		3.38E−02	
81×81	2.25E−02	0.84	1.82E−02	0.90	1.81E−02	0.90
161×161	1.15E−02	0.96	9.01E−03	1.01	8.99E−03	1.01
321×321	5.56E−03	1.05	4.10E−03	1.14	4.09E−03	1.14
641×641	2.85E−03	0.97	2.03E−03	1.02	2.02E−03	1.02

In this test, the HJ equation

$$\begin{cases} v_t(t,x) + \frac{1}{2}|\mathrm{D}v(t,x)|^2 = 0, \\ v(0,x) = v_0(x) = \max(1 - |x|^2, 0) \end{cases} \tag{37}$$

is considered in $[-2, 2]^2$, using extrapolation techniques to treat the homogeneous boundary condition. The exact solution of this problem is known and is given, for $t \geqslant 1/2$, by

$$v(t,x) = \begin{cases} \frac{(|x|-1)^2}{2t} & \text{if } |x| \leqslant 1, \\ 0 & \text{if } |x| > 1. \end{cases} \tag{38}$$

We set the final time $T = 0.5$ and compute the approximate solution with $\Delta t = \frac{5}{4}\Delta x$. The initial condition and final numerical solution are shown in Fig. 6, while L^1 errors and CPU times are reported respectively in Tables 7 and 8.

Since in this case the feet of characteristics always fall in the singular region of the solution, we observe, as expected, that all numerical schemes are degraded to the first-order accuracy. From Table 7 one can see that the L^1-norm of the error is about 30% lower for the central reconstructions, as in the previous cases. From Table 9 and Fig. 7 we can see that near the kink in the solution, the central reconstructions produce larger undershoots in the flat region, while being closer to the exact solution in the nonzero region. Finally, from Table 8 one can observe that central reconstruction schemes are about 30% faster (Table 10).

Test 5: Front propagation with obstacles.

Table 8 CPU times for Test 4, WENO and CWENO schemes

Grid	WENO	CWENO		CWENOZ	
	CPU time	CPU time	% gain	CPU time	% gain
41×41	2.88E+00	1.98E+00	31.37	1.96E+00	32.07
81×81	2.19E+01	1.54E+01	29.70	1.53E+01	30.07
161×161	1.72E+02	1.19E+02	31.10	1.18E+02	31.35
321×321	1.37E+03	9.38E+02	31.56	9.34E+02	31.86
641×641	1.09E+04	7.44E+03	31.59	7.51E+03	30.93

Table 9 Overshoots and undershoots observed in Test 4, WENO, CWENO, and CWENOZ schemes

N	WENO		CWENO		CWENOZ	
	min	max	min	max	min	max
21×21	−5.67E−05	9.76E−01	−6.08E−03	9.80e−01	−6.08E−03	9.80E−01
41×41	−4.62E−05	9.89E−01	−5.19E−03	9.92E−01	−5.19E−03	9.92E−01
81×81	−1.21E−05	9.95E−01	−3.39E−03	9.96E−01	−3.39E−03	9.96E−01
161×161	−3.45E−06	9.98E−01	−1.97E−03	9.98e−01	−1.97E−03	9.98E−01

Table 10 CPU times for Test 5, WENO and CWENO schemes

Grid	WENO	CWENO		CWENOZ	
	CPU time	CPU time	% gain	CPU time	% gain
126×126	6.28E+00	5.34E+00	14.94	5.30E+00	15.66
251×251	4.56E+01	3.96E+01	13.04	3.96E+01	13.06
501×501	3.61E+02	3.11E+02	13.87	3.20E+02	11.50
$1\,001 \times 1\,001$	2.81E+03	2.43E+03	13.64	2.40E+03	14.59
$2\,001 \times 2\,001$	2.20E+04	1.92E+04	13.01	1.89E+04	14.00

Last, we consider a state-constrained Zermelo problem, proposed in [6], for a swimmer that has to reach a circular island at $(1, 0)$, facing a non-constant current flowing in the horizontal direction. Two obstacles are placed in the domain, near the target.

More precisely, to consider the minimum time problem the running cost $f_C(t, x, a)$ is set to zero, and the nonlinear dynamics is given by $f_D(t, x, a) = (2 - 0.5x_2^2 + a_1, a_2)$ with $A := \{a \in \mathbb{R}^2 \mid \|a\| = (a_1^2 + a_2^2)^{1/2} = 1\}$. The swimmer has unit speed and can move in any direction $a \in A$; the current has speed $2 - 0.5x_2^2$ and flows in the direction $(1, 0)$. The target is the set $\mathcal{X} = \{x \mid v_0(x) \leqslant 0\}$, where $v_0(x) = C \min\big(\|(x_1 - 1, x_2)\| - r, r\big)$ with $r = 0.25$ and $C = 20$. To take into account the presence of the two obstacles, we follow the level-set approach, as proposed in [6], and represent the set of constrained states as $\mathcal{K} = \{x \in \mathbb{R}^2 \mid g(x) \leqslant 0\}$ where

$$g(x) = \max\Big(-\gamma, C\big(\gamma - \max(|x_1 - 0.3|, |x_2 - 0.4|)\big), C\big(\gamma - \max(|x_1 + 1|, |x_2 + 1.5|)\big)\Big)$$

with $C = 20$, $\gamma = 0.2$.

For every $t \in [0, T]$, the set of points from which the swimmer can reach the target before time t (Backward Reachable set) is represented by the level set $\mathcal{R}[0, t] = \{x \in \mathbb{R}^2 \mid \exists\, s \in$

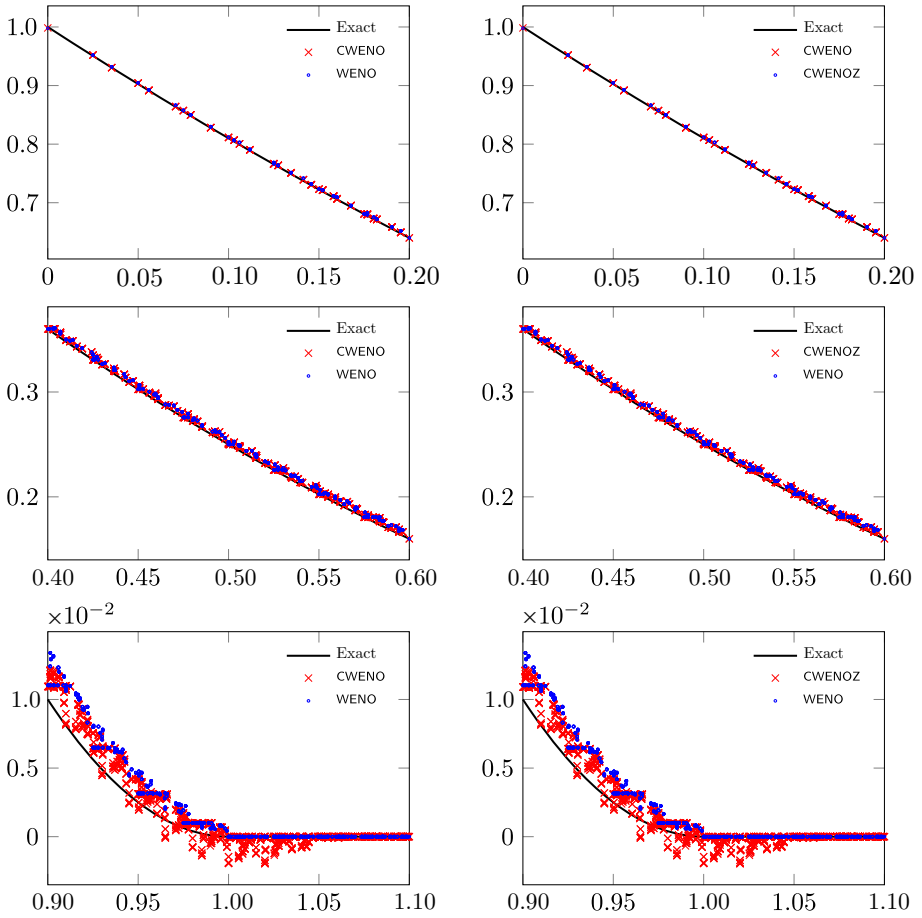

Fig. 7 Test 4. Comparison of overshoots and undershoots of the numerical solutions computed with the traditional WENO scheme and with the CWENO ones on a 161×161 grid. In the left panels WENO and CWENO are compared, while in the right panels similar plots are shown for WENO and CWENOZ. The exact solution is always represented with a thick black line. First row: scatter plots of all data points with $r \in [0, 0.2]$ corresponding to the spike of the solution. Second row: scatter plots of all data points with $r \in [0.4, 0.6]$ corresponding to a regular region of the solution. Third row: scatter plots of all data points with $r \in [0.9, 1.1]$ corresponding to the singular region of the initial data

$[0, t],\ v(x, s) \leqslant 0\}$ where v is the solution to a constrained version of (1)

$$\min(v_t + H(t, x, \mathrm{D}v), v - g(x)) = 0 \tag{39}$$

with the initial condition $v(x, 0) = \max(v_0(x), g(x))$ (see [6, Remark 2]). The space domain is defined as $[-3, 2] \times [-2, 2]$ and the final time to $T = 3$. By using scheme (5), we discretize (39) as

$$\min\left\{u_i^{n+1} - \min_{\underline{a} \in A^v}\{R[u^n](y_i^n(\underline{a}))\}, u^{n+1} - g(x_i)\right\} = 0,$$

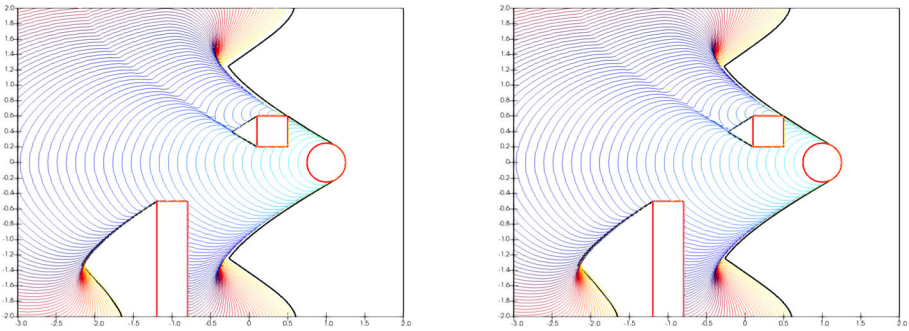

Fig. 8 Reachable sets (40) computed at each time step for Test 5 on a grid 101×81 and $T = 3$, using WENO (left) and CWENO (right) reconstructions. In both panels the black line represents a reference solution computed on a grid $1\,001 \times 801$, the red circle represents the target and the red rectangles represent the obstacles

resulting in the following time-marching scheme:

$$u_i^{n+1} = \max\left\{\min_{\underline{a}\in A^v}\{R[u^n](y_i^n(\underline{a}))\}, g(x_i)\right\},$$

from which we define the numerical approximation of the reachable set $\mathcal{R}[0, t_n]$ as

$$\mathcal{R}^n = \{x_i \mid \exists\, k \in \{0, \cdots, n\}, u_i^k \leqslant 0\}. \tag{40}$$

The solution is computed with RK3 and $\Delta t = \Delta x$. In Fig. 8, we show that, even in this very complex situation, no relevant difference in the accuracy of the computed solution can be observed between the WENO and the CWENO approaches. On the other hand, central reconstruction schemes provide a reduction of about $13\%-14\%$ of the computational time (see Table 10).

6 Conclusion

In this study, a high-order SL scheme has been developed to approximate first-order HJB equations. The primary challenge lies in the nonsmooth nature of the solutions, which would lead to spurious oscillations when using the unlimited high-order polynomial interpolation.

To address this issue, the reconstruction of the solution at the previous time step at the foot of characteristics has to be performed with a non-oscillatory scheme. Since one has to perform very many interpolations during the minimization procedure, a CWENO technique was proposed to be coupled with the SL approach. Our study demonstrates that, in terms of errors, our new scheme maintains the favorable behavior of WENO schemes of [10], producing about 30% more accurate results in smooth regions at the price of some extra over/undershoots. However, its computational cost is significantly lower, by $10\%-30\%$ depending on the specific test.

Additionally, we have established a convergence result in a simpler case, and provided several numerical simulations to further validate the effectiveness of our proposed scheme.

Future Directions

Future research directions could explore the inclusion of high-order treatments for boundary conditions and extensions of the convergence analysis to more general Hamiltonians. Investigating the scheme's performance in more complex scenarios would also contribute to a comprehensive understanding of its capabilities.

Appendix A Sketch of the Convergence Proof

This appendix provides some ideas on the proof of Theorem 1. The proof refers to the one-dimensional case, to a strictly convex Hamiltonian which depends only on v_x. It is given in slightly different forms in [17, 20], and we will use the latter form here.

We rewrite the scheme in the specific form used here as

$$\begin{cases} u_i^{n+1} = \min\limits_{a\in\mathbb{R}}\{R[u^n](x_i - a\Delta t) + H^*(a)\}, \\ u_i^0 = v_0(x_i). \end{cases} \tag{A1}$$

We will denote the minimizer in (A1) as $\bar{a}_i$.

The convergence theorem is proved via the following steps.

Estimate on the Second Incremental Ratio

Setting ourselves in the basic framework of Sect. 4, this estimate is based on the following lemma from [20].

Lemma A1 *If* (30) *holds, then, for any* $j \in \mathbb{Z}$ *and* $n \geqslant 1$,

$$u_{j+1}^n - 2u_j^n + u_{j-1}^n \leqslant \frac{\Delta x^2}{m_H \Delta t}. \tag{A2}$$

Moreover, assuming, in addition, that (31) *holds, then*

$$\max_{x_{i-1},x_i,x_{i+1}\in U(z_j^n)} |u_{i+1}^n - 2u_i^n + u_{i-1}^n| \leqslant C\frac{\Delta x^2}{\Delta t} \tag{A3}$$

(with z_j^n *denoting the foot of the characteristic ending at* x_j*) for some positive constant* C *depending on* C_r, $h_\pm$, *and* m_H.

The lemma basically states that the unilateral bound (A2) becomes bilateral at the feet of characteristics: in a way, this parallels the property of semiconcave functions to have minima only at regular points.

The proof of Lemma A1, and especially of (A3), is very technical and will be omitted; it relies on the fact that a minimum in (A1) cannot occur at points where (A3) is violated. To this end, all the specific features of problem (29) (one-space dimension, uniformly convex Hamiltonian with no dependence on x) have a crucial role. Therefore, in addition to the difficulty of an increase of dimension, it appears hard to recast the proof in a more general setting, even for the dynamic programming Hamiltonian (3) which, in particular, may fail to be strictly convex.

Uniform Lipschitz Continuity of Numerical Solutions

We can now state the result of the Lipschitz stability.

Theorem A1 *Let u^n be defined by the scheme (A1). Assume that (30), (31) hold, $\Delta x = O(\Delta t^2)$, and u^0 is Lipschitz continuous with the Lipschitz constant L_0. Then, u^n satisfy, for any i and j, the discrete Lipschitz estimate*

$$\frac{|u_i^n - u_j^n|}{|x_i - x_j|} \leqslant L$$

for a constant L independent of Δx and Δt, and for $0 \leqslant n \leqslant T/\Delta t$, as $\Delta t \to 0$.

Sketch of the Proof With no loss in generality, we set $i = j \pm 1$, and assume that at the step $n-1$ the discrete solution satisfies

$$\frac{|u_i^{n-1} - u_j^{n-1}|}{\Delta x} \leqslant L_{n-1}.$$

Making the argmin explicit and using (A3), we have

$$\begin{aligned} u_j^n &= \Delta t H^*(\bar{a}_j) + R\left[u^{n-1}\right](x_j + \bar{a}_j \Delta t) \\ &\geqslant \Delta t H^*(\bar{a}_j) + R_1\left[u^{n-1}\right](x_j + \bar{a}_j \Delta t) + O\left(\frac{\Delta x^2}{\Delta t}\right). \end{aligned} \tag{A4}$$

On the other hand, u_i^n can be bounded as

$$\begin{aligned} u_i^n &= \Delta t H^*(\bar{a}_i) + R\left[u^{n-1}\right](x_i + \bar{a}_i \Delta t) \\ &\leqslant \Delta t H^*(\bar{a}_j) + R\left[u^{n-1}\right](x_i + \bar{a}_j \Delta t) \\ &\leqslant \Delta t H^*(\bar{a}_j) + R_1\left[u^{n-1}\right](x_i + \bar{a}_j \Delta t) + O\left(\frac{\Delta x^2}{\Delta t}\right), \end{aligned} \tag{A5}$$

which results from both the optimality of $\bar{a}_i$ and (A3). From (A4) and (A5) we obtain the unilateral estimate

$$\begin{aligned} \frac{u_i^n - u_j^n}{\Delta x} &\leqslant \frac{1}{\Delta x}\left(R_1\left[u^{n-1}\right](x_i + \bar{a}_j \Delta t) - R_1\left[u^{n-1}\right](x_j + \bar{a}_j \Delta t)\right) + O\left(\frac{\Delta x}{\Delta t}\right) \\ &\leqslant L_{n-1} + O\left(\frac{\Delta x}{\Delta t}\right), \end{aligned} \tag{A6}$$

in which we have used the fact that the first-order reconstruction R_1 at step $n-1$ also has a Lipschitz constant L_{n-1}. Interchanging the roles of $\bar{a}_j$ and $\bar{a}_i$, we obtain the reverse estimate, and ultimately

$$\frac{|u_j^n - u_i^n|}{\Delta x} \leqslant L_{n-1} + O\left(\frac{\Delta x}{\Delta t}\right). \tag{A7}$$

Iterating back, we obtain a finite bound as $\Delta x, \Delta t \to 0$ if and only if $\Delta x/\Delta t = O(\Delta t)$, that is, if $\Delta x = O(\Delta t^2)$.

Quasi-monotonicity and Convergence of the Scheme

Last, once recalled that the scheme is consistent and invariant for the addition of constants, we note that, for a Lipschitz stable family of numerical solutions u,

$$|R[u](x) - R_1[u](x)| \leqslant O(\Delta x). \tag{A8}$$

Using the space reconstruction R_1 the scheme is monotone, and (A8) implies that the scheme (A1) is monotone up to $O(\Delta x) = O(\Delta t^2)$. Therefore, the monotonicity defect is $o(\Delta t)$, and this suffices to obtain the convergence via the Barles-Souganidis theorem in its generalized formulation (see [17]).

Appendix B Smoothness Indicators for the Two-Dimensional Case

We report here the matrix involved in the definition (25),

$$M = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 6 & 2 & 0 & 0 & 3 & \frac{4}{3} & 0 & 2 & 1 & \frac{3}{2} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 2 & 6 & 0 & \frac{4}{3} & 3 & 1 & 2 & \frac{3}{2} \\
0 & 0 & 0 & 6 & 0 & 0 & 84 & 3 & 0 & 0 & 42 & 2 & 0 & 28 & \frac{3}{2} & 21 \\
0 & 0 & 0 & 2 & 1 & 0 & 3 & \frac{32}{3} & 1 & 0 & \frac{31}{2} & \frac{31}{3} & 1 & 15 & \frac{152}{15} & \frac{147}{10} \\
0 & 0 & 0 & 0 & 1 & 2 & 0 & 1 & \frac{32}{3} & 3 & 1 & \frac{31}{3} & \frac{31}{2} & \frac{152}{15} & 15 & \frac{147}{10} \\
0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 3 & 84 & 0 & 2 & 42 & \frac{3}{2} & 28 & 21 \\
0 & 0 & 0 & 3 & 1 & 0 & 42 & \frac{31}{2} & 1 & 0 & \frac{989}{5} & 15 & 1 & \frac{954}{5} & \frac{147}{10} & \frac{933}{5} \\
0 & 0 & 0 & \frac{4}{3} & 1 & \frac{4}{3} & 2 & \frac{31}{3} & \frac{31}{3} & 2 & 15 & \frac{3\,992}{45} & 15 & \frac{1\,918}{15} & \frac{1\,918}{15} & \frac{737}{4} \\
0 & 0 & 0 & 0 & 1 & 3 & 0 & 1 & \frac{31}{2} & 42 & 1 & 15 & \frac{989}{5} & \frac{147}{10} & \frac{954}{5} & \frac{933}{5} \\
0 & 0 & 0 & 2 & 1 & 1 & 28 & 15 & \frac{152}{15} & \frac{3}{2} & \frac{954}{5} & \frac{1\,918}{15} & \frac{147}{10} & \frac{56\,076}{35} & \frac{737}{4} & \frac{161\,501}{70} \\
0 & 0 & 0 & 1 & 1 & 2 & \frac{3}{2} & \frac{152}{15} & 15 & 28 & \frac{147}{10} & \frac{1\,918}{15} & \frac{954}{5} & \frac{737}{4} & \frac{56\,076}{35} & \frac{161\,501}{70} \\
0 & 0 & 0 & \frac{3}{2} & 1 & \frac{3}{2} & 21 & \frac{147}{10} & \frac{147}{10} & 21 & \frac{933}{5} & \frac{737}{4} & \frac{933}{5} & \frac{161\,501}{70} & \frac{161\,501}{70} & \frac{721\,401}{25}
\end{pmatrix}.$$

In addition, we report the matrices involved in the definitions (26). We assume that the vector U of data in the 4×4 stencil is ordered lexicographically, i.e., the x direction faster and the y direction slower,

$$A_{\mathrm{ne}} = \begin{pmatrix}
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0 & \frac{3\,497}{720} & \frac{-623}{90} & \frac{1\,787}{720} & 0 & \frac{-623}{90} & \frac{1\,667}{180} & \frac{-571}{180} & 0 & \frac{1\,787}{720} & \frac{-571}{180} & \frac{797}{720} \\
0&0&0&0&0 & \frac{-623}{90} & \frac{1\,141}{90} & \frac{-229}{45} & 0 & \frac{1\,667}{180} & \frac{-1\,547}{90} & \frac{1\,187}{180} & 0 & \frac{-571}{180} & \frac{278}{45} & \frac{-421}{180} \\
0&0&0&0&0 & \frac{1\,787}{720} & \frac{-229}{45} & \frac{1\,817}{720} & 0 & \frac{-571}{180} & \frac{1\,187}{180} & \frac{-293}{90} & 0 & \frac{797}{720} & \frac{-421}{180} & \frac{827}{720} \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0 & \frac{-623}{90} & \frac{1\,667}{180} & \frac{-571}{180} & 0 & \frac{1\,141}{90} & \frac{-1\,547}{90} & \frac{278}{45} & 0 & \frac{-229}{45} & \frac{1\,187}{180} & \frac{-421}{180} \\
0&0&0&0&0 & \frac{1\,667}{180} & \frac{-1\,547}{90} & \frac{1\,187}{180} & 0 & \frac{-1\,547}{90} & \frac{1\,472}{45} & \frac{-1\,157}{90} & 0 & \frac{1\,187}{180} & \frac{-1\,157}{90} & \frac{887}{180} \\
0&0&0&0&0 & \frac{-571}{180} & \frac{1\,187}{180} & \frac{-293}{90} & 0 & \frac{278}{45} & \frac{-1\,157}{90} & \frac{571}{90} & 0 & \frac{-421}{180} & \frac{887}{180} & \frac{-109}{45} \\
0&0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0&0 & 0 & 0 \\
0&0&0&0&0 & \frac{1\,787}{720} & \frac{-571}{180} & \frac{797}{720} & 0 & \frac{-229}{45} & \frac{1\,187}{180} & \frac{-421}{180} & 0 & \frac{1\,817}{720} & \frac{-293}{90} & \frac{827}{720} \\
0&0&0&0&0 & \frac{-571}{180} & \frac{278}{45} & \frac{-421}{180} & 0 & \frac{1\,187}{180} & \frac{-1\,157}{90} & \frac{887}{180} & 0 & \frac{-293}{90} & \frac{571}{90} & \frac{-109}{45} \\
0&0&0&0&0 & \frac{797}{720} & \frac{-421}{180} & \frac{827}{720} & 0 & \frac{-421}{180} & \frac{887}{180} & \frac{-109}{45} & 0 & \frac{827}{720} & \frac{-109}{45} & \frac{857}{720}
\end{pmatrix},$$

$$A_{\mathrm{nw}} = \begin{pmatrix}
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0 & \frac{1\,817}{720} & \frac{-229}{45} & \frac{1\,787}{720} & 0 & \frac{-293}{90} & \frac{1\,187}{180} & \frac{-571}{180} & 0 & \frac{827}{720} & \frac{-421}{180} & \frac{797}{720} & 0 \\
0&0&0&0 & \frac{-229}{45} & \frac{1\,141}{90} & \frac{-623}{90} & 0 & \frac{1\,187}{180} & \frac{-1\,547}{90} & \frac{1\,667}{180} & 0 & \frac{-421}{180} & \frac{278}{45} & \frac{-571}{180} & 0 \\
0&0&0&0 & \frac{1\,787}{720} & \frac{-623}{90} & \frac{3\,497}{720} & 0 & \frac{-571}{180} & \frac{1\,667}{180} & \frac{-623}{90} & 0 & \frac{797}{720} & \frac{-571}{180} & \frac{1\,787}{720} & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0 & \frac{-293}{90} & \frac{1\,187}{180} & \frac{-571}{180} & 0 & \frac{571}{90} & \frac{-1\,157}{90} & \frac{278}{45} & 0 & \frac{-109}{45} & \frac{887}{180} & \frac{-421}{180} & 0 \\
0&0&0&0 & \frac{1\,187}{180} & \frac{-1\,547}{90} & \frac{1\,667}{180} & 0 & \frac{-1\,157}{90} & \frac{1\,472}{45} & \frac{-1\,547}{90} & 0 & \frac{887}{180} & \frac{-1\,157}{90} & \frac{1\,187}{180} & 0 \\
0&0&0&0 & \frac{-571}{180} & \frac{1\,667}{180} & \frac{-623}{90} & 0 & \frac{278}{45} & \frac{-1\,547}{90} & \frac{1\,141}{90} & 0 & \frac{-421}{180} & \frac{1\,187}{180} & \frac{-229}{45} & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0 \\
0&0&0&0 & \frac{827}{720} & \frac{-421}{180} & \frac{797}{720} & 0 & \frac{-109}{45} & \frac{887}{180} & \frac{-421}{180} & 0 & \frac{857}{720} & \frac{-109}{45} & \frac{827}{720} & 0 \\
0&0&0&0 & \frac{-421}{180} & \frac{278}{45} & \frac{-571}{180} & 0 & \frac{887}{180} & \frac{-1\,157}{90} & \frac{1\,187}{180} & 0 & \frac{-109}{45} & \frac{571}{90} & \frac{-293}{90} & 0 \\
0&0&0&0 & \frac{797}{720} & \frac{-571}{180} & \frac{1\,787}{720} & 0 & \frac{-421}{180} & \frac{1\,187}{180} & \frac{-229}{45} & 0 & \frac{827}{720} & \frac{-293}{90} & \frac{1\,817}{720} & 0 \\
0&0&0&0&0 & 0 & 0 & 0&0 & 0 & 0 & 0&0 & 0 & 0 & 0
\end{pmatrix},$$

$$A_{\mathrm{sw}}=\begin{pmatrix}
\frac{857}{720} & \frac{-109}{45} & \frac{827}{720} & 0 & \frac{-109}{45} & \frac{887}{180} & \frac{-421}{180} & 0 & \frac{827}{720} & \frac{-421}{180} & \frac{797}{720} & 0 & 0 & 0 & 0 & 0\\
\frac{-109}{45} & \frac{571}{90} & \frac{-293}{90} & 0 & \frac{887}{180} & \frac{-1\,157}{90} & \frac{1\,187}{180} & 0 & \frac{-421}{180} & \frac{278}{45} & \frac{-571}{180} & 0 & 0 & 0 & 0 & 0\\
\frac{827}{720} & \frac{-293}{90} & \frac{1\,817}{720} & 0 & \frac{-421}{180} & \frac{1\,187}{180} & \frac{-229}{45} & 0 & \frac{797}{720} & \frac{-571}{180} & \frac{1\,787}{720} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{-109}{45} & \frac{887}{180} & \frac{-421}{180} & 0 & \frac{571}{90} & \frac{-1\,157}{90} & \frac{278}{45} & 0 & \frac{-293}{90} & \frac{1\,187}{180} & \frac{-571}{180} & 0 & 0 & 0 & 0 & 0\\
\frac{887}{180} & \frac{-1\,157}{90} & \frac{1\,187}{180} & 0 & \frac{-1\,157}{90} & \frac{1\,472}{45} & \frac{-1\,547}{90} & 0 & \frac{1\,187}{180} & \frac{-1\,547}{90} & \frac{1\,667}{180} & 0 & 0 & 0 & 0 & 0\\
\frac{-421}{180} & \frac{1\,187}{180} & \frac{-229}{45} & 0 & \frac{278}{45} & \frac{-1\,547}{90} & \frac{1\,141}{90} & 0 & \frac{-571}{180} & \frac{1\,667}{180} & \frac{-623}{90} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{827}{720} & \frac{-421}{180} & \frac{797}{720} & 0 & \frac{-293}{90} & \frac{1\,187}{180} & \frac{-571}{180} & 0 & \frac{1\,817}{720} & \frac{-229}{45} & \frac{1\,787}{720} & 0 & 0 & 0 & 0 & 0\\
\frac{-421}{180} & \frac{278}{45} & \frac{-571}{180} & 0 & \frac{1\,187}{180} & \frac{-1\,547}{90} & \frac{1\,667}{180} & 0 & \frac{-229}{45} & \frac{1\,141}{90} & \frac{-623}{90} & 0 & 0 & 0 & 0 & 0\\
\frac{797}{720} & \frac{-571}{180} & \frac{1\,787}{720} & 0 & \frac{-571}{180} & \frac{1\,667}{180} & \frac{-623}{90} & 0 & \frac{1\,787}{720} & \frac{-623}{90} & \frac{3\,497}{720} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

$$A_{\mathrm{se}}=\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \frac{1\,817}{720} & \frac{-293}{90} & \frac{827}{720} & 0 & \frac{-229}{45} & \frac{1\,187}{180} & \frac{-421}{180} & 0 & \frac{1\,787}{720} & \frac{-571}{180} & \frac{797}{720} & 0 & 0 & 0 & 0\\
0 & \frac{-293}{90} & \frac{571}{90} & \frac{-109}{45} & 0 & \frac{1\,187}{180} & \frac{-1\,157}{90} & \frac{887}{180} & 0 & \frac{-571}{180} & \frac{278}{45} & \frac{-421}{180} & 0 & 0 & 0 & 0\\
0 & \frac{827}{720} & \frac{-109}{45} & \frac{857}{720} & 0 & \frac{-421}{180} & \frac{887}{180} & \frac{-109}{45} & 0 & \frac{797}{720} & \frac{-421}{180} & \frac{827}{720} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \frac{-229}{45} & \frac{1\,187}{180} & \frac{-421}{180} & 0 & \frac{1\,141}{90} & \frac{-1\,547}{90} & \frac{278}{45} & 0 & \frac{-623}{90} & \frac{1\,667}{180} & \frac{-571}{180} & 0 & 0 & 0 & 0\\
0 & \frac{1\,187}{180} & \frac{-1\,157}{90} & \frac{887}{180} & 0 & \frac{-1\,547}{90} & \frac{1\,472}{45} & \frac{-1\,157}{90} & 0 & \frac{1\,667}{180} & \frac{-1\,547}{90} & \frac{1\,187}{180} & 0 & 0 & 0 & 0\\
0 & \frac{-421}{180} & \frac{887}{180} & \frac{-109}{45} & 0 & \frac{278}{45} & \frac{-1\,157}{90} & \frac{571}{90} & 0 & \frac{-571}{180} & \frac{1\,187}{180} & \frac{-293}{90} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \frac{1\,787}{720} & \frac{-571}{180} & \frac{797}{720} & 0 & \frac{-623}{90} & \frac{1\,667}{180} & \frac{-571}{180} & 0 & \frac{3\,497}{720} & \frac{-623}{90} & \frac{1\,787}{720} & 0 & 0 & 0 & 0\\
0 & \frac{-571}{180} & \frac{278}{45} & \frac{-421}{180} & 0 & \frac{1\,667}{180} & \frac{-1\,547}{90} & \frac{1\,187}{180} & 0 & \frac{-623}{90} & \frac{1\,141}{90} & \frac{-229}{45} & 0 & 0 & 0 & 0\\
0 & \frac{797}{720} & \frac{-421}{180} & \frac{827}{720} & 0 & \frac{-571}{180} & \frac{1\,187}{180} & \frac{-293}{90} & 0 & \frac{1\,787}{720} & \frac{-229}{45} & \frac{1\,817}{720} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

The matrix A_{opt} appearing in (26) is splitted into two matrices $A^{(1)}_{\text{opt}}$ and $A^{(2)}_{\text{opt}}$ reporting the first eight columns and the second eight columns of A_{opt},

$$A^{(1)}_{\text{opt}} = \begin{pmatrix} \frac{2\,903}{1\,575} & \frac{-182\,131}{37\,800} & \frac{10\,859}{2\,700} & \frac{-13\,889}{12\,600} & \frac{-182\,131}{37\,800} & \frac{60\,463}{4\,800} & \frac{-58\,881}{5\,600} & \frac{871\,553}{302\,400} \\ \frac{-182\,131}{37\,800} & \frac{48\,799}{3\,150} & \frac{-58\,687}{4\,200} & \frac{10\,859}{2\,700} & \frac{60\,463}{4\,800} & \frac{-340\,087}{8\,400} & \frac{1\,226\,693}{33\,600} & \frac{-58\,881}{5\,600} \\ \frac{10\,859}{2\,700} & \frac{-58\,687}{4\,200} & \frac{48\,799}{3\,150} & \frac{-182\,131}{37\,800} & \frac{-58\,881}{5\,600} & \frac{1\,226\,693}{33\,600} & \frac{-340\,087}{8\,400} & \frac{60\,463}{4\,800} \\ \frac{-13\,889}{12\,600} & \frac{10\,859}{2\,700} & \frac{-182\,131}{37\,800} & \frac{2\,903}{1\,575} & \frac{871\,553}{302\,400} & \frac{-58\,881}{5\,600} & \frac{60\,463}{4\,800} & \frac{-182\,131}{37\,800} \\ \frac{-182\,131}{37\,800} & \frac{60\,463}{4\,800} & \frac{-58\,881}{5\,600} & \frac{871\,553}{302\,400} & \frac{48\,799}{3\,150} & \frac{-340\,087}{8\,400} & \frac{426\,623}{12\,600} & \frac{-11\,137}{1\,200} \\ \frac{60\,463}{4\,800} & \frac{-340\,087}{8\,400} & \frac{1\,226\,693}{33\,600} & \frac{-58\,881}{5\,600} & \frac{-340\,087}{8\,400} & \frac{815\,527}{6\,300} & \frac{-2\,958\,593}{25\,200} & \frac{426\,623}{12\,600} \\ \frac{-58\,881}{5\,600} & \frac{1\,226\,693}{33\,600} & \frac{-340\,087}{8\,400} & \frac{60\,463}{4\,800} & \frac{426\,623}{12\,600} & \frac{-2\,958\,593}{25\,200} & \frac{815\,527}{6\,300} & \frac{-340\,087}{8\,400} \\ \frac{871\,553}{302\,400} & \frac{-58\,881}{5\,600} & \frac{60\,463}{4\,800} & \frac{-182\,131}{37\,800} & \frac{-11\,137}{1\,200} & \frac{426\,623}{12\,600} & \frac{-340\,087}{8\,400} & \frac{48\,799}{3\,150} \\ \frac{10\,859}{2\,700} & \frac{-58\,881}{5\,600} & \frac{221\,047}{25\,200} & \frac{-363\,499}{151\,200} & \frac{-58\,687}{4\,200} & \frac{1\,226\,693}{33\,600} & \frac{-1\,536\,757}{50\,400} & \frac{841\,823}{100\,800} \\ \frac{-58\,881}{5\,600} & \frac{426\,623}{12\,600} & \frac{-1\,536\,757}{50\,400} & \frac{221\,047}{25\,200} & \frac{1\,226\,693}{33\,600} & \frac{-2\,958\,593}{25\,200} & \frac{10\,709\,107}{100\,800} & \frac{-1\,536\,757}{50\,400} \\ \frac{221\,047}{25\,200} & \frac{-1\,536\,757}{50\,400} & \frac{426\,623}{12\,600} & \frac{-58\,881}{5\,600} & \frac{-1\,536\,757}{50\,400} & \frac{10\,709\,107}{100\,800} & \frac{-2\,958\,593}{25\,200} & \frac{1\,226\,693}{33\,600} \\ \frac{-363\,499}{151\,200} & \frac{221\,047}{25\,200} & \frac{-58\,881}{5\,600} & \frac{10\,859}{2\,700} & \frac{841\,823}{100\,800} & \frac{-1\,536\,757}{50\,400} & \frac{1\,226\,693}{33\,600} & \frac{-58\,687}{4\,200} \\ \frac{-13\,889}{12\,600} & \frac{871\,553}{302\,400} & \frac{-363\,499}{151\,200} & \frac{66\,427}{100\,800} & \frac{10\,859}{2\,700} & \frac{-58\,881}{5\,600} & \frac{22\,1047}{25\,200} & \frac{-363\,499}{151\,200} \\ \frac{871\,553}{302\,400} & \frac{-11\,137}{1\,200} & \frac{841\,823}{100\,800} & \frac{-363\,499}{151\,200} & \frac{-58\,881}{5\,600} & \frac{426\,623}{12\,600} & \frac{-1\,536\,757}{50\,400} & \frac{221\,047}{25\,200} \\ \frac{-363\,499}{151\,200} & \frac{841\,823}{100\,800} & \frac{-11\,137}{1\,200} & \frac{871\,553}{302\,400} & \frac{221\,047}{25\,200} & \frac{-58\,881}{5\,600} & \frac{426\,623}{12\,600} & \frac{-58\,881}{5\,600} \\ \frac{66\,427}{100\,800} & \frac{-363\,499}{151\,200} & \frac{871\,553}{302\,400} & \frac{-13\,889}{12\,600} & \frac{-363\,499}{151\,200} & \frac{221\,047}{25\,200} & \frac{-58\,881}{5\,600} & \frac{10\,859}{2\,700} \end{pmatrix},$$

$$A^{(2)}_{\text{opt}} = \begin{pmatrix} \frac{10\,859}{2700} & \frac{-58\,881}{5\,600} & \frac{221\,047}{25\,200} & \frac{-363\,499}{151\,200} & \frac{-13\,889}{12\,600} & \frac{871\,553}{302\,400} & \frac{-363\,499}{151\,200} & \frac{66\,427}{100\,800} \\ \frac{-58\,881}{5\,600} & \frac{426\,623}{12\,600} & \frac{-1\,536\,757}{50\,400} & \frac{221\,047}{25\,200} & \frac{871\,553}{302\,400} & \frac{-11\,137}{1\,200} & \frac{841\,823}{100\,800} & \frac{-363\,499}{151\,200} \\ \frac{221\,047}{25\,200} & \frac{-1\,536\,757}{50\,400} & \frac{426\,623}{12\,600} & \frac{-58\,881}{5\,600} & \frac{-363\,499}{151\,200} & \frac{841\,823}{100\,800} & \frac{-11\,137}{1\,200} & \frac{871\,553}{302\,400} \\ \frac{-363\,499}{151\,200} & \frac{221\,047}{25\,200} & \frac{-58\,881}{5\,600} & \frac{10\,859}{2\,700} & \frac{66\,427}{100\,800} & \frac{-363\,499}{151\,200} & \frac{871\,553}{302\,400} & \frac{-13\,889}{12\,600} \\ \frac{-58\,687}{4\,200} & \frac{1\,226\,693}{33\,600} & \frac{-1\,536\,757}{50\,400} & \frac{841\,823}{100\,800} & \frac{10\,859}{2\,700} & \frac{-58\,881}{5\,600} & \frac{221\,047}{25\,200} & \frac{-363\,499}{151\,200} \\ \frac{1\,226\,693}{33\,600} & \frac{-2\,958\,593}{25\,200} & \frac{10\,709\,107}{100\,800} & \frac{-1\,536\,757}{50\,400} & \frac{-58\,881}{5\,600} & \frac{426\,623}{12\,600} & \frac{-1\,536\,757}{50\,400} & \frac{221\,047}{25\,200} \\ \frac{-1\,536\,757}{50\,400} & \frac{10\,709\,107}{100\,800} & \frac{-2\,958\,593}{25\,200} & \frac{1\,226\,693}{33\,600} & \frac{221\,047}{25\,200} & \frac{-363\,499}{151\,200} & \frac{22\,1047}{25\,200} & \frac{-58\,881}{5\,600} \\ \frac{841\,823}{100\,800} & \frac{-1\,536\,757}{50\,400} & \frac{1\,226\,693}{33\,600} & \frac{-58\,687}{4\,200} & \frac{-11\,137}{1\,200} & \frac{-182\,131}{37\,800} & \frac{60\,463}{4\,800} & \frac{-58\,881}{5\,600} \\ \frac{48\,799}{3\,150} & \frac{-34\,0087}{8\,400} & \frac{426\,623}{12\,600} & \frac{-11\,137}{1\,200} & \frac{426\,623}{60\,463} & \frac{-182\,131}{37\,800} & \frac{60\,463}{4\,800} & \frac{-340\,087}{8\,400} \\ \frac{-340\,087}{8\,400} & \frac{815\,527}{6\,300} & \frac{-2\,958\,593}{25\,200} & \frac{815\,527}{6\,300} & \frac{12\,600}{4\,800} & \frac{-58\,881}{5\,600} & \frac{1\,226\,693}{33\,600} & \frac{-340\,087}{8\,400} \\ \frac{426\,623}{12\,600} & \frac{-2\,958\,593}{25\,200} & \frac{815\,527}{6\,300} & \frac{-340\,087}{8\,400} & \frac{-58\,881}{5\,600} & \frac{1\,226\,693}{33\,600} & \frac{-340\,087}{8\,400} & \frac{60\,463}{4\,800} \\ \frac{-11\,137}{1\,200} & \frac{426\,623}{12\,600} & \frac{-340\,087}{8\,400} & \frac{48\,799}{3\,150} & \frac{871\,553}{302\,400} & \frac{-58\,881}{5\,600} & \frac{60\,463}{4\,800} & \frac{-182\,131}{37\,800} \\ \frac{-182\,131}{37\,800} & \frac{60\,463}{4\,800} & \frac{-58\,881}{5\,600} & \frac{871\,553}{302\,400} & \frac{2\,903}{1\,575} & \frac{-182\,131}{37\,800} & \frac{10\,859}{2\,700} & \frac{-13\,889}{12\,600} \\ \frac{60\,463}{4\,800} & \frac{-340\,087}{8\,400} & \frac{1\,226\,693}{33\,600} & \frac{-58\,881}{5\,600} & \frac{-182\,131}{37\,800} & \frac{48\,799}{3\,150} & \frac{-58\,687}{4\,200} & \frac{10\,859}{2\,700} \\ \frac{-58\,881}{5\,600} & \frac{1\,226\,693}{33\,600} & \frac{-340\,087}{8\,400} & \frac{60\,463}{4\,800} & \frac{2\,700}{-13\,889} & \frac{10\,859}{2\,700} & \frac{-182\,131}{37\,800} & \frac{48\,799}{3\,150} \\ \frac{871\,553}{302\,400} & \frac{-58\,881}{5\,600} & \frac{60\,463}{4\,800} & \frac{-182\,131}{37\,800} & \frac{12\,600}{-13\,889} & \frac{2\,700}{10\,859} & \frac{37\,800}{-182\,131} & \frac{4\,200}{2\,903} \\ & & & & & & & \frac{37\,800}{1\,575} \end{pmatrix}.$$

Acknowledgements Elisabetta Carlini and Roberto Ferretti thank the Italian Ministry of University and Research (MUR) for supporting this research with funds from the PRIN Project 2022 (2022238YY5, entitled "Optimal control problems: analysis, approximation"). All authors are members of the GNCS-INdAM, Gruppo Nazionale per il Calcolo Scientifico of Istituto Nazionale di Alta Matematica "Francesco Severi".

Funding Open access funding provided by Università degli Studi Roma Tre within the CRUI-CARE Agreement.

Data Statement No external data have been used for his work.

Declarations

Conflict of Interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Abgrall, R.: On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. J. Comput. Phys. **114**(6), 45–48 (1994)
2. Baeza, A., Bürger, R., Mulet, P., Zorío, D.: Central WENO schemes through a global average weight. J. Sci. Comput. **78**(1), 499–530 (2019). https://doi.org/10.1007/s10915-018-0773-z
3. Balsara, D.S., Garain, S., Florinski, V., Boscheri, W.: An efficient class of WENO schemes with adaptive order for unstructured meshes. J. Comput. Phys. **404**, 109062 (2020). https://doi.org/10.1016/j.jcp.2019.109062
4. Bardi, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Birkhäuser Boston, Inc., Boston (1997)
5. Bokanowski, O., Falcone, M., Sahu, S.: An efficient filtered scheme for some first order time-dependent Hamilton-Jacobi equations. SIAM J. Sci. Comput. **38**(1), 171–195 (2016)
6. Bokanowski, O., Forcadel, N., Zidani, H.: Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. SIAM J. Control Optim. **48**(7), 4292–4316 (2010). https://doi.org/10.1137/090762075
7. Bonaventura, L., Calzola, E., Carlini, E., Ferretti, R.: Second order fully semi-Lagrangian discretizations of advection-diffusion-reaction systems. J. Sci. Comput. **88**, 23 (2021). https://doi.org/10.1007/s10915-021-01518-8
8. Bryson, S., Levy, D.: High-order schemes for multi-dimensional Hamilton-Jacobi equations. In: Hou, T.Y., Tadmor, E. (eds.) Hyperbolic Problems: Theory, Numerics, Applications, pp. 387–396. Springer, Berlin (2003)
9. Calzola, E., Carlini, E., Dupuis, X., Silva, F.J.: A semi-Lagrangian scheme for Hamilton-Jacobi-Bellman equations with oblique derivatives boundary conditions. Numer. Math. **153**(1), 49–84 (2023)
10. Carlini, E., Ferretti, R., Russo, G.: A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **27**(3), 1071–1091 (2006). https://doi.org/10.1137/040608787
11. Castro, M.J., Semplice, M.: Third- and fourth-order well-balanced schemes for the shallow water equations based on the CWENO reconstruction. Int. J. Numer. Methods Fluid **89**(8), 304–325 (2019). https://doi.org/10.1002/fld.4700
12. Cravero, I., Puppo, G., Semplice, M., Visconti, G.: CWENO: uniformly accurate reconstructions for balance laws. Math. Comput. **87**(312), 1689–1719 (2018). https://doi.org/10.1090/mcom/3273
13. Cravero, I., Semplice, M., Visconti, G.: Optimal definition of the nonlinear weights in multidimensional Central WENOZ reconstructions. SIAM J. Numer. Anal. **57**(5), 2328–2358 (2019)
14. Dumbser, M., Boscheri, W., Semplice, M., Russo, G.: Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes. SIAM J. Sci. Comput. **39**(6), 2564–2591 (2017)
15. Falcone, M., Ferretti, R.: Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations. Numer. Math. **67**(3), 315–344 (1994)
16. Falcone, M., Ferretti, R.: Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods. J. Comput. Phys. **175**(2), 559–575 (2002)
17. Falcone, M., Ferretti, R.: Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2014)

18. Falcone, M., Paolucci, G., Tozza, S.: Convergence of adaptive filtered schemes for first order evolutionary Hamilton-Jacobi equations. Numer. Math. **145**(2), 271–311 (2020)
19. Falcone, M., Paolucci, G., Tozza, S.: Multidimensional smoothness indicators for first-order Hamilton-Jacobi equations. J. Comput. Phys. **409**, 109360 (2020). https://doi.org/10.1016/j.jcp.2020.109360
20. Ferretti, R.: Convergence of semi-Lagrangian approximations to convex Hamilton-Jacobi equations under (very) large Courant numbers. SIAM J. Numer. Anal. **40**(6), 2240–2253 (2002). https://doi.org/10.1137/S0036142901388378
21. Galántai, A.: The Nelder-Mead simplex algorithm is sixty years old: new convergence results and open questions. Algorithms **17**(11), 523 (2024). https://doi.org/10.3390/a17110523
22. Ha, Y., Kim, C.H., Yang, H., Yoon, J.: A sixth-order weighted essentially non-oscillatory schemes based on exponential polynomials for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **75**, 1675–1700 (2018). https://doi.org/10.1007/s10915-017-0603-8
23. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high-order accurate essentially nonoscillatory schemes III. J. Comput. Phys. **71**(2), 231–303 (1987). https://doi.org/10.1016/0021-9991(87)90031-3
24. Hu, C., Shu, C.-W.: Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys. **150**(1), 97–127 (1999). https://doi.org/10.1006/jcph.1998.6165
25. Jiang, G.-S., Peng, D.: Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **21**(6), 2126–2143 (2000). https://doi.org/10.1137/S106482759732455X
26. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
27. Kim, C.H., Ha, Y., Yang, H., Yoon, J.: A third-order WENO scheme based on exponential polynomials for Hamilton-Jacobi equations. Appl. Numer. Math. **165**, 167–183 (2021). https://doi.org/10.1016/j.apnum.2021.01.020
28. Levy, D., Puppo, G., Russo, G.: Compact central WENO schemes for multidimensional conservation laws. SIAM J. Sci. Comput. **22**(2), 656–672 (2000). https://doi.org/10.1137/S1064827599359461
29. Lin, C.T., Tadmor, E.: High-resolution nonoscillatory central schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **21**(6), 2163–2186 (2000)
30. Liu, X.D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**(1), 200–212 (1994)
31. Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal **7**(4), 308–313 (1965)
32. Osher, S., Shu, C.-W.: High-order essentially non-oscillatory schemes for Hamilton-Jacobi equation. SIAM J. Numer. Anal. **28**, 907–922 (1991)
33. Samala, R.: L^1-type smoothness indicators based weighted essentially nonoscillatory scheme for Hamilton-Jacobi equations. Int. J. Numer. Methods Fluid **92**(12), 1927–1947 (2020). https://doi.org/10.1002/fld.4855
34. Samala, R., Biswas, B.: Arc length-based WENO scheme for Hamilton-Jacobi equations. Commun. Appl. Math. Comput. **3**, 481–496 (2021). https://doi.org/10.1007/s42967-020-00091-5
35. Semplice, M., Coco, A., Russo, G.: Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction. J. Sci. Comput. **66**, 692–724 (2016). https://doi.org/10.1007/s10915-015-0038-z
36. Semplice, M., Visconti, G.: Efficient implementation of adaptive order reconstructions. J. Sci. Comput. **83**, 6 (2020). https://doi.org/10.1007/s10915-020-01156-6
37. Zennaro, M.: Natural continuous extensions of Runge-Kutta methods. Math. Comput. **46**(173), 119–133 (1986). https://doi.org/10.1090/S0025-5718-1986-0815835-1
38. Zheng, F., Shu, C.-W., Qiu, J.: High order finite difference Hermite WENO schemes for the Hamilton-Jacobi equations on unstructured meshes. Comput. Fluids **183**, 53–65 (2019). https://doi.org/10.1016/j.compfluid.2019.02.010
39. Zhou, J., Cai, L., Zhou, F.-Q.: New high-resolution scheme for three-dimensional nonlinear hyperbolic conservation laws. Appl. Math. Comput. **198**(2), 770–786 (2008). https://doi.org/10.1016/j.amc.2007.09.017
40. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. J. Comput. Phys. **318**, 110–121 (2016). https://doi.org/10.1016/j.jcp.2016.05.010
41. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for Hamilton-Jacobi equations. Numer. Methods PDEs **33**(4), 1095–1113 (2017). https://doi.org/10.1002/num.22133
42. Zhu, J., Qiu, J.: A new type of high-order WENO schemes for Hamilton-Jacobi equations on triangular meshes. Commun. Comput. Phys. **27**(3), 897–920 (2020). https://doi.org/10.4208/cicp.OA-2018-0156