SIAM J. SCI. COMPUT.
Vol. 00, No. 0, pp. 000–000

A WEIGHTED ESSENTIALLY NONOSCILLATORY, LARGE TIME-STEP SCHEME FOR HAMILTON–JACOBI EQUATIONS*

E. CARLINI[†], R. FERRETTI[‡], AND G. RUSSO[§]

Abstract. We investigate the application of weighted essentially nonoscillatory (WENO) reconstructions to a class of semi-Lagrangian schemes for first order time-dependent Hamilton–Jacobi equations. In particular, we derive a general form of the scheme, study sufficient conditions for its convergence with high-order reconstructions, and perform numerical tests to study its efficiency. In addition, we prove that the weights of the WENO interpolants are positive for any order.

Key words. semi-Lagrangian schemes, Hamilton–Jacobi equations, WENO methods

AMS subject classifications. Primary, 65N12, 65M10; Secondary, 49L25

DOI. 10.1137/040608787

1. Introduction.

In this paper we treat a class of high-order, large time-step Godunov schemes for time-dependent Hamilton–Jacobi equation of first order. Although several extensions are possible, we will focus on the model problem

$$\begin{cases} v_t + H(\nabla v) = 0 & \text{in } \mathbb{R}^N \times \mathbb{R}, \\ v(x,0) = v_0(x) & \text{in } \mathbb{R}^N \end{cases} \tag{1.1}$$

and assume that the function H is smooth and strictly convex.

Many approximation schemes have been proposed for (1.1), but only a relatively recent part of the existing literature is devoted to high-order schemes. Within this rapidly growing research line, we quote the pioneering works by Osher and Sethian [23], Osher and Shu [24], Abgrall [1], [2] and, among more recent contributions, Lin and Tadmor [20], and Bryson and Levy [6]. Weighted essentially nonoscillatory (WENO) schemes were introduced by Liu, Osher, and Chan in [22], extensively studied and improved in [18], [16], and applied to Hamilton–Jacobi (HJ) equations in Jiang and Peng [17], Zhang and Shu [29], and Bryson and Levy [7].

On the other hand, large time-step, characteristics-based schemes for hyperbolic PDEs have been first proposed by Courant, Isaacson, and Rees in [8]. At the moment, they have gained a considerable popularity in the Numerical Weather Prediction community (where they are also known as *semi-Lagrangian (SL) schemes*). In the field of HJ equations, such schemes have been first applied to stationary Bellman equations related to optimal control problems (see [10] and the review [9]) and later to time-dependent equations for which the schemes under consideration might be seen (see [11]) as discrete counterparts of the Lax–Hopf–Oleinik representation formula.

It is well known that solutions of (1.1) should be understood in the viscosity sense (see [21], [4], [5]), which does not require v to be everywhere differentiable. Moreover,

[*] Received by the editors November 12, 2004; accepted for publication (in revised form) March 29, 2005; published electronically DATE. This work has been partly supported by MURST, Progetto Nazionale “Metologie Numeriche Avanzate per il Calcolo Scientifico.”

http://www.siam.org/journals/sisc/x-x/60878.html

[†] Dipartimento di Matematica, Università di Roma “La Sapienza,” p.le Aldo Moro, 2, 00185 Roma (carlini@mat.uniroma1.it).

[‡] Dipartimento di Matematica, Università di Roma Tre, L.go S. L. Murialdo, 1, 00146 Roma (ferretti@mat.uniroma3.it).

[§] Dipartimento di Matematica ed Informatica, Università di Catania, v.le A. Doria, 6, 95125 Catania (russo@dmi.unict.it).

singularities in the gradient ∇v usually bring important information in HJ equations (as an example, in dynamic programming equations, singularities are typically associated to switching curves for the optimal control). Therefore, it would be crucial for the scheme to be accurate in smooth regions, while retaining low numerical dissipation and a nonoscillatory behavior in the neighborhood of singularities.

A further argument leads us to consider the use of WENO reconstructions in SL schemes. In fact, the scheme performs at any node and any time step a minimization which involves several computations of the numerical reconstruction. Its efficiency is, therefore, critical. Piecewise smooth functions can be effectively reconstructed using ENO or WENO methods (see [14], [15], [25]). Both techniques tend to choose interpolation points on the smooth side of the function; however, ENO schemes compute twice the required number of divided differences: at each step (that is, at any increase of the polynomial degree) the choice between the two new interpolation points is made according to the magnitude of the two divided differences obtained with the two candidate stencils (see [25]). On the other hand, in WENO schemes this process is performed by weighting the different polynomials, and this allows information from both sides to be used whenever possible.

The paper is organized as follows. Section 2 reviews the construction and basic convergence theory of SL schemes for (1.1), section 3 presents a general framework for the implementation of WENO interpolation as required by the SL scheme, with the explicit form of the schemes of order 2/3 and 3/5. In section 4 we prove positivity of the linear and nonlinear weights for any order of the polynomial and give their explicit form for evenly spaced nodes. In section 5 we derive from the positivity of the weights a sufficient condition for the convergence of the scheme, whereas section 6 is devoted to numerical tests.

2. Construction of the scheme and basic convergence theory. As it has been already pointed out, the SL schemes for HJ equations might be interpreted (see [11]) as discrete versions of the Lax–Hopf–Oleinik representation formula for the solution, which reads

$$v(x,t) = \inf_{\alpha \in \mathbb{R}^N} \{tH^*(\alpha) + v_0(x + \alpha t)\} \tag{2.1}$$

(where $H^*(\alpha)$ is the Legendre transform of $H(p)$). The general form of the scheme is a discretization of (2.1) on a single time step, namely,

$$v_j^{n+1} = \min_{a \in \mathbb{R}^N} \left(\Delta t H^*(a) + I[V^n](x_j + a\Delta t)\right), \tag{2.2}$$

where $V^n = \{v_j^n\}_{j\in\mathbb{Z}}$, $I[V^n](x)$ is a numerical reconstruction of the approximate solution at time $t_n = n\Delta t$ for any $x \in \mathbb{R}^N$, and v_j^{n+1} approximates $v(x_j, t_{n+1})$. If the reconstruction $I[V^n]$ is of degree r, then the consistency error of the scheme is estimated by

$$E_{\Delta t,\Delta x} \leq C\frac{\Delta x^r}{\Delta t} \tag{2.3}$$

(note that in this situation, the best accuracy of the scheme would, in principle, be achieved by performing a single time step). If the Hamiltonian H depends also on x, then the representation formula (2.1) takes a more general form (see [4]) and in the consistency analysis, there arises a further error term related to the approximation of characteristics (see [11] for a complete analysis). More precisely, if characteristics are approximated with order p, the full consistency estimate reads

$$E_{\Delta t,\Delta x} \leq C_1 \Delta t^p + C_2 \frac{\Delta x^r}{\Delta t}. \tag{2.4}$$

In this case, both the term Δt^p and the need for a global minimum search in (2.2) make it unfeasible to perform a single time step when approximating the solution at a fixed time T. Note that according to (2.4), the scheme is consistent provided $\Delta x^r = o(\Delta t)$ and that the highest consistency rate is achieved for

$$\Delta t \sim \Delta x^{\frac{r}{p+1}}.$$

We briefly review here the convergence theory developed in [13] for the scheme (2.2). Assume that (1.1) is posed on $\mathbb{R}$ and discretized on an infinite uniform grid with nodes $x_j = j\Delta x$ ($j = 0, \pm 1, \pm 2, \ldots$). Also assume that the Hamiltonian function H is a $W^{2,\infty}$ function and that

$$H''(p) \geq m_H > 0. \tag{2.5}$$

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 $I[V]$ satisfies, for some constant $C < 1$,

$$I[V](x_k) = v(x_k) \ , \quad |I[V](x) - I_1[V](x)| \leq C \max_{x_k \in U(x)} |v_{k+1} - 2v_k + v_{k-1}|, \tag{2.6}$$

where by $I_1[V]$ we denote the P_1, i.e., piecewise linear, interpolation on the sequence V, and by $U(x) = (x - h_- \Delta x, x + h_+ \Delta x)$ the stencil of the reconstruction $I[V](x)$. For example, a quadratic Lagrange reconstruction can be performed taking one node on the left and two nodes on the right of the point x (in this case, $h_- = 1$, $h_+ = 2$), or two nodes on the left and one on the right (and in this case, $h_- = 2$, $h_+ = 1$). In a second order ENO reconstruction (see [25]), both cases are possible depending on the solution (and thus, $h_- = h_+ = 2$). A third order Lagrange reconstruction is typically performed using two nodes on the left and two on the right, so that $h_- = h_+ = 2$. In the third order ENO case, $h_- = h_+ = 3$ and so forth. It is proved in [13] that condition (2.6) is satisfied for Lagrange and ENO reconstructions up to the fifth order if the reconstruction stencil includes the interval $[x_j, x_{j+1}]$.

Under such assumptions it is possible (see [13]) to prove the following.

THEOREM 2.1. *Consider the scheme* (2.2) *applied to* (1.1) *with $N = 1$. Assume that* (2.5), (2.6) *hold, that $\Delta x = O(\Delta t^2)$, and that v_0 is Lipschitz continuous. Then, the numerical solution $V^n = \{v_j^n\}_j$ (with v_j^n defined by* (2.2)*) satisfies*

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

(where v is the solution of (1.1)*) for $0 \leq n \leq T/\Delta t$ as $\Delta t \to 0$.*

3. A general form of WENO interpolation.

Although the construction of a WENO interpolation is a well-established matter, we will give here some details of the procedure, since it is usually treated and used in a different framework. In this section, we will *not* assume a uniform mesh spacing.

To construct a WENO interpolation of degree $2n - 1$ on the interval $[x_j, x_{j+1}]$, we start from the Lagrange polynomial built on the stencil $S = \{x_{j-n+1}, \ldots, x_{j+n}\}$

and written in the form

$$Q(x) = \sum_{k=1}^{n} C_k(x) P_k(x), \tag{3.1}$$

where the "linear weights" C_k are polynomials of degree $n-1$ and the P_k are polynomials of degree n interpolating V on the stencil $S_k = \{x_{j-n+k}, \ldots, x_{j+k}\}$, $k = 1, \ldots, n$ (note that all the stencils S_k overlap on the interval $[x_j, x_{j+1}]$, and that the dependence on j has been dropped in this general expression). The nonlinear weights are then constructed so as to obtain an approximation of the highest degree if suitable smoothness indicators β_k give the same result on all stencils. This leads us to define

$$\alpha_k(x) = \frac{C_k(x)}{(\beta_k + \varepsilon)^2} \tag{3.2}$$

(with ε a properly small parameter, usually of the order of 10^{-6}), and then the nonlinear weights as

$$w_k(x) = \frac{\alpha_k(x)}{\sum_l \alpha_l(x)}, \tag{3.3}$$

where, following [25], the smoothness indicators are typically computed as

$$\beta_k = \sum_{l=1}^{n} \int_{x_j}^{x_{j+1}} \Delta x^{2l-1} (P_k^{(l)})^2 \, dx. \tag{3.4}$$

In the section on numerical tests we shall consider some different expressions for the smoothness indicators, such as the one proposed in [17], which does not depend on the first derivative of the function.

Last, the WENO interpolation reads

$$I[V](x) = \sum_{k=1}^{n} w_k(x) P_k(x). \tag{3.5}$$

Note that usually one is interested in computing the reconstruction in specific points in the cell. In this case the polynomials C_k take some specific value.

Let us now turn back to the linear weights C_k, which have not been given a precise expression. Due to (3.1) and to the definition of the polynomials P_k, the C_k are characterized by the fact that $Q(x)$ should be the interpolating polynomial of $V(x)$ on the stencil S. Since P_k interpolates V on the stencil S_k, it would be natural to require that C_k should vanish at the nodes outside S_k, and that in the nodes of S the nonzero weights should have unit sum. In this way, if P_k is the polynomial that interpolates the function on S_k, then (3.1) is the polynomial that interpolates V on S.

Taking into account that the number of nodes belonging to S but not to S_k is precisely n, we thus infer that the linear weights must necessarily have the form

$$C_k(x) = \gamma_k \prod_{x_l \in S \setminus S_k} (x - x_l) = \gamma_k \tilde{C}_k(x) \tag{3.6}$$

with γ_k to be determined in order to have unit sum. Note that the conditions

$$\sum_{k=1}^{n} C_k(x_i) = 1 \tag{3.7}$$

for each $x_i \in S$ apparently constitute a set of $2n$ equations in n unknowns. Actually, the correct perspective to look at these conditions is the following. The left-hand side of (3.7) is a polynomial of degree $n-1$ (computed at x_i), and on the right-hand side we have the polynomial $p(x) \equiv 1$. Therefore, imposing that the two polynomials coincide on more than $n-1$ points is equivalent to impose that they are identical. Now, it is easy to show that the monic polynomials $\tilde{C}_k$, $k = 1, \dots, n$, are independent and form a basis of the space $\mathbb{P}_{n-1}$ of polynomials of degree $n-1$. Condition

$$\sum_{k=1}^{n} C_k(x) = 1 \quad \forall x \in \mathbb{R} \tag{3.8}$$

uniquely determines the constants γ_k as a polynomial identity.

Condition (3.8) can be conveniently satisfied by imposing it on a suitable set of points. Consider first the node x_{j-n+1}. Then, using (3.6) into (3.1), we note that, among all the linear weights computed at x_{j-n+1}, the only nonzero weight is C_1, so that

$$Q(x_{j-n+1}) = C_1(x_{j-n+1})P_1(x_{j-n+1}) = C_1(x_{j-n+1})v_{j-n+1} \tag{3.9}$$

and this implies that $C_1(x_{j-n+1}) = 1$ and allows us to compute γ_1. On the other hand, in the following node x_{j-n+2}, the nonzero weights are C_1 and C_2, and with a similar argument we obtain the condition

$$\begin{aligned} Q(x_{j-n+2}) &= C_1(x_{j-n+2})P_1(x_{j-n+2}) + C_2(x_{j-n+2})P_2(x_{j-n+2}) \\ &= [C_1(x_{j-n+2}) + C_2(x_{j-n+2})]v_{j-n+2}, \end{aligned} \tag{3.10}$$

which implies that $C_1(x_{j-n+2}) + C_2(x_{j-n+2}) = 1$, whence γ_2 can be computed. Following this guideline, we finally obtain the set of conditions

$$\sum_{i=1}^{k} C_i(x_{j-n+k}) = 1 \quad (k = 1, \dots, n), \tag{3.11}$$

that is, more explicitly,

$$\sum_{i=1}^{k} \gamma_i \prod_{x_l \in S \setminus S_i} (x_{j-n+k} - x_l) = 1 \quad (k = 1, \dots, n), \tag{3.12}$$

which is a linear triangular system in the unknowns γ_i.

Although the construction outlined in this section does not require a constant space step, the problem of giving an explicit expression to the coefficients γ_i is considerably easier in the case of constant Δx. This case will be treated in detail in section 4, along with the study of sign for the linear weights. In the next subsections we give two examples of this construction (on a uniform mesh), including the expressions for the smoothness indicators (3.4). The resulting schemes will be used in the section on numerical tests.

3.1. Second–third order WENO interpolation. To construct a third order interpolation we start from two polynomials of second degree, so that

$$I[V^n](x) = w_L P_L(x) + w_R P_R(x), \tag{3.13}$$

where $P_L(x)$ and $P_R(x)$ are second order polynomials constructed, respectively, on the nodes x_{j-1}, x_j, x_{j+1} and on the nodes x_j, x_{j+1}, x_{j+2}. The two linear weights C_L and C_R are first degree polynomials in x, and according to the general theory outlined so far, they read

$$C_L = \frac{x_{j+2} - x}{3\Delta x}, \quad C_R = \frac{x - x_{j-1}}{3\Delta x}, \tag{3.14}$$

and the expressions of α_L, α_R, w_L, and w_R may be easily recovered from the general form.

According to (3.4), the smoothness indicators have the explicit expressions

$$\beta_L = \frac{13}{12}v_{j-1}^2 + \frac{16}{3}v_j^2 + \frac{25}{12}v_{j+1}^2 - \frac{13}{3}v_{j-1}v_j + \frac{7}{6}v_{j-1}v_{j+1} - \frac{19}{3}v_j v_{j+1}, \tag{3.15}$$

$$\beta_R = \frac{13}{12}v_{j+2}^2 + \frac{16}{3}v_{j+1}^2 + \frac{25}{12}v_j^2 - \frac{13}{3}v_{j+2}v_{j+1} + \frac{7}{6}v_{j+2}v_j - \frac{19}{3}v_j v_{j+1}. \tag{3.16}$$

3.2. Third–fifth order WENO interpolation. To construct a fifth order interpolation we start from three polynomials of third degree:

$$I[V^n](x) = w_L P_L(x) + w_C P_C(x) + w_R P_R(x), \tag{3.17}$$

where the third order polynomials $P_L(x)$, $P_C(x)$, and $P_R(x)$ are constructed, respectively, on $x_{j-2}, x_{j-1}, x_j, x_{j+1}$, on $x_{j-1}, x_j, x_{j+1}, x_{j+2}$, and on $x_j, x_{j+1}, x_{j+2}, x_{j+3}$. The weights C_L, C_C, and C_R are second degree polynomials in x, and have the form

$$\begin{aligned} C_L &= \frac{(x - x_{j+2})(x - x_{j+3})}{20\Delta x^2}, \quad C_C = \frac{(x - x_{j-2})(x - x_{j+3})}{10\Delta x^2}, \\ C_R &= \frac{(x - x_{j-2})(x - x_{j-1})}{20\Delta x^2}, \end{aligned} \tag{3.18}$$

while the smoothness indicators β_C and β_R have the expressions

$$\begin{aligned} \beta_C = {}& \frac{61}{45}v_{j-1}^2 + \frac{331}{30}v_j^2 + \frac{331}{30}v_{j+1}^2 + \frac{61}{45}v_{j+2}^2 - \frac{141}{20}v_{j-1}v_j + \frac{179}{30}v_{j-1}v_{j+1} \\ & + \frac{293}{180}v_{j-1}v_{j+2} - \frac{1259}{60}v_j v_{j+1} + \frac{179}{30}v_j v_{j+2} - \frac{141}{20}v_{j+1}v_{j+2}, \end{aligned} \tag{3.19}$$

$$\begin{aligned} \beta_R = {}& \frac{407}{90}v_j^2 + \frac{721}{30}v_{j+1}^2 + \frac{248}{15}v_{j+2}^2 + \frac{61}{45}v_{j+3}^2 - \frac{1193}{60}v_j v_{j+1} + \frac{439}{30}v_j v_{j+2} \\ & - \frac{683}{180}v_j v_{j+3} - \frac{2309}{60}v_{j+1}v_{j+2} + \frac{309}{30}v_{j+1}v_{j+3} - \frac{553}{60}v_{j+2}v_{j+3}, \end{aligned} \tag{3.20}$$

and β_L can be obtained using the same set of coefficients of β_R in a symmetric way (that is, replacing the indices $j-2, \ldots, j+3$ with $j+3, \ldots, j-2$).

It is interesting to check that, due to their structure, the linear weights are always positive in the interval $[x_j, x_{j+1}]$ for $n = 2, 3$. This also means that pointwise the linear combination (3.5) is in fact a convex combination, and this has important consequences at the level of convergence analysis. We will prove in the next section that this property holds for linear weights up to an arbitrary order.

4. On the positivity of the weights in WENO reconstruction. We prove in this section that the positivity result for the weights in WENO interpolation, obtained in the previous section by direct computation up to third–fifth order, actually holds in general. In addition, we give an explicit expression for the linear weights in the case of evenly spaced nodes.

It is worth pointing out that this result of positive sign for the linear and nonlinear weights is not in contrast with the well-known situation in which the weights may change sign (see [26]). In fact, this latter situation may occur in the case of reconstructions based on cell averages instead of pointwise values such as the interpolation procedure above.

We first give a general proof of positivity for the linear weights $C_k(x)$.

Theorem 4.1. *Let $\{x_i\}$ be a family of consecutively numbered points of $\mathbb{R}$. Then, once the Lagrange polynomial $Q(x)$ of a given function $f(x)$ built on the stencil $S = \{x_{j-n+1}, \dots, x_{j+n}\}$ is written in the form* (3.1)*, the linear weights $C_k(x)$ $(k = 1, \dots, n)$ are nonnegative for any $x \in (x_j, x_{j+1})$. Moreover, $\sum_k C_k(x) \equiv 1$.*

Proof. Let us denote by $Q^{(l,m)}(x)$ (with $j-n+1 \le l \le j < j+1 \le m \le j+n$ and $m-l \ge n$) the interpolating polynomial constructed on the stencil $\{x_l, \dots, x_m\}$, so that

$$Q(x) = Q^{(j-n+1,j+n)}(x) \tag{4.1}$$

and, moreover,

$$P_k(x) = Q^{(j-n+k,j+k)}(x). \tag{4.2}$$

We will also write a generic $Q^{(l,m)}(x)$ as

$$Q^{(l,m)}(x) = \sum_k C_k^{(l,m)}(x) P_k(x), \tag{4.3}$$

where the summation may be extended to all $k = 1, \dots, n$ by setting $C_k^{(l,m)} = 0$ whenever the stencil of P_k is not included in $\{x_l, \dots, x_m\}$. Therefore, the final linear weights will be

$$C_k(x) = C_k^{(j-n+1,j+n)}(x). \tag{4.4}$$

We proceed by induction on the degree of $Q^{(l,m)}$, given by $i = \deg Q^{(l,m)} = m-l$, starting with $i = n$ up to $i = 2n-1$. First we prove that the claim of the theorem extends to all polynomials $Q^{(l,m)}(x)$ defined in (4.3). The claim is obviously true for $Q^{(j-n+k,j+k)}(x)$ by (4.2). In this case the coefficients $C_k^{(l,m)} = 1$, $C_s^{(l,m)} \equiv 0$, $s \ne k$. Then, we assume the claim is true for the set of linear weights $C_k^{(l,m)}(x)$ of a generic $Q^{(l,m)}(x)$ such that $m-l = i$ and add a node to the right (adding a node to the left leads to an analogous computation). By inductive assumption we have, for any l and m such that $l \le j < j+1 \le m$ and $m-l = i$, that (4.3) holds with $C_k^{(l,m)}(x) \ge 0$. On the other hand, by elementary interpolation theory arguments (Neville's recursive form of the interpolating polynomial),

$$Q^{(l,m+1)}(x) = \frac{x_{m+1}-x}{x_{m+1}-x_l} Q^{(l,m)}(x) + \frac{x-x_l}{x_{m+1}-x_l} Q^{(l+1,m+1)}(x). \tag{4.5}$$

Note that $\deg Q^{(l,m)} = \deg Q^{(l+1,m+1)} = i$, and that both fractions which multiply $Q^{(l,m)}(x)$ and $Q^{(l+1,m+1)}(x)$ are positive as long as $x \in (x_j, x_{j+1})$. Therefore, using

(4.3) into (4.5), we get

$$Q^{(l,m+1)}(x) = \sum_k \left[\frac{x_{m+1}-x}{x_{m+1}-x_l} C_k^{(l,m)}(x) + \frac{x-x_l}{x_{m+1}-x_l} C_k^{(l+1,m+1)}(x) \right] P_k(x). \tag{4.6}$$

The term in square brackets is $C_k^{(l,m+1)}(x)$ and it is immediate to check that it is nonnegative. Iterating this argument up to $Q^{(j-n+1,j+n)}(x)$ completes the proof of nonnegativity for all the linear weights $C_k(x)$.

Last, setting $f(x) \equiv 1$, we also have $Q(x) \equiv 1$ and, for any k, $P_k(x) \equiv 1$. Plugging these identities into (3.1) we obtain $\sum_k C_k(x) \equiv 1$. □

We turn now to the problem of giving an explicit expression to the linear weights in the situation of evenly spaced nodes. The expression of the weights for the linear scheme is

$$C_k = \gamma_k \prod_{x_l \in S \setminus S_k} (x - x_l).$$

First observe that the polynomial $C_k(x)$ does not change sign in the interval $[x_j, x_{j+1}]$. More precisely, the sign of it, in this interval, is given by $\mathrm{sign}(\gamma_k)(-1)^{n+k}$. Therefore,

$$C_k(x) > 0 \quad \forall x \in [x_j, x_{j+1}] \quad \forall k, n \quad \Leftrightarrow \quad \mathrm{sign}\gamma_k = (-1)^{n+k}. \tag{4.7}$$

For simplicity, set $j = n$ in the definition of the stencil S, so that $S = \{x_1, \dots, x_{2n}\}$. Let us introduce the integer stencils

$$\tilde{S} = S/\Delta x, \quad \tilde{S}_k = S_k/\Delta x.$$

Then the polynomials C_k can be written as

$$C_k(\Delta x \xi) = \tilde{\gamma}_k \prod_{l \in \tilde{S} \setminus \tilde{S}_k} (\xi - l),$$

where $\tilde{\gamma}_k = \gamma_k \Delta x^{n-1}$. The nondimensional constants $\tilde{\gamma}_k$ satisfy the condition

$$\sum_{i=1}^{k} \tilde{\gamma}_i \prod_{\substack{l=1 \\ l \notin (i,\dots,i+n)}}^{2n} (i - l) = 1.$$

Introducing now the matrix

$$a_{ki} = \frac{\prod_{l=1}^{2n}(k-l)}{\prod_{l=i}^{i+n}(k-l)} = \frac{(k-1)!(2n-k)!}{(n+i-k)!(k-i)!}(-1)^{n+i}, \tag{4.8}$$

the system can be written as

$$\sum_{i=1}^{k} a_{ki} \tilde{\gamma}_i = 1.$$

Let $\mu_i \equiv (-1)^{n+i}\tilde{\gamma}_i$. Then, such constants satisfy the triangular system

$$\sum_{i=1}^{k} |a_{ki}| \mu_i = 1, \tag{4.9}$$

and according to (4.7) $C_i(x) > 0$, $x \in [x_n, x_{n+1}]$ if and only if $\mu_i > 0$.

The solution to system (4.9) can be explicitly given. In fact, we will prove now that

$$\mu_i = \frac{n!(n-1)!}{(n-i)!(i-1)!(2n-1)!} \quad (i = 1, \ldots, n). \tag{4.10}$$

Actually, plugging (4.10) and (4.8) into (4.9) we obtain the set of conditions that we want to check:

$$\sum_{i=1}^{k} \frac{(k-1)!(2n-k)!n!(n-1)!}{(n+i-k)!(k-i)!(n-i)!(i-1)!(2n-1)!} = 1.$$

These relations can be rearranged in the form

$$\frac{(k-1)!(2n-k)!}{(2n-1)!} \sum_{i=1}^{k} \begin{pmatrix} n \\ k-i \end{pmatrix} \begin{pmatrix} n-1 \\ i-1 \end{pmatrix} = 1,$$

and, hence, rewriting the first term and shifting the summation index as $j = i-1$,

$$\sum_{j=0}^{k-1} \begin{pmatrix} n \\ k-1-j \end{pmatrix} \begin{pmatrix} n-1 \\ j \end{pmatrix} = \begin{pmatrix} 2n-1 \\ k-1 \end{pmatrix},$$

which is in turn a special case of the identity[1] (see [3])

$$\sum_{j=0}^{N} \begin{pmatrix} n \\ N-j \end{pmatrix} \begin{pmatrix} m \\ j \end{pmatrix} = \begin{pmatrix} n+m \\ N \end{pmatrix}$$

with $N = k-1$ and $m = n-1$.

We also prove here that the μ_i are inverse of integers. In fact, note first that by (4.10), $\mu_i = \mu_{n-i+1}$, so that it suffices to prove the claim for $i \le (n+1)/2$. Then, we rewrite the μ_i as

$$\mu_i = \begin{cases} \dfrac{n!}{(2n-1)!} & \text{if } i = 1, \\ \dfrac{(n-i+1)\cdots(n-1)n(n-1)!}{(i-1)!(2n-1)!} & \text{if } i > 1. \end{cases} \tag{4.11}$$

While the claim is obvious if $i = 1$, if $i > 1$, we derive from (4.11)

$$\mu_i = \frac{(2n-2i+2)\cdots(2n-2)}{2^{i-1}(i-1)!(n+1)\cdots(2n-1)}.$$

Now, if $i \le (n+1)/2$, then $2n-2i+2 \ge n+1$ and, therefore, any term of the product $(2n-2i+2)\cdots(2n-2)$ can be simplified with a term of the product $(n+1)\cdots(2n-1)$. This completes the proof of the claim.

We report in Table 1 the value of $1/\mu_i$ for values of n up to $n = 8$.

[1] We thank professor Corrado Falcolini for bringing this result to our attention.

Table 1
Values of $1/\mu_i$ for the first 8 values of n.

n	$1/\mu_i$
1	1
2	3, 3
3	20, 10, 20
4	210, 70, 70, 210
5	3024, 756, 504, 756, 3024
6	55440, 11088, 5544, 5544, 11088, 55440
7	1235520, 205920, 82368, 61776, 82368, 205920, 1235520
8	32432400, 4633200, 1544400, 926640, 926640, 1544400, 4633200, 32432400

5. A sufficient condition for convergence. We briefly check here that the convergence theory developed in [13] may be applied to the scheme under consideration. To this end, we work in the framework of Theorem 2.1 and prove the following.

Theorem 5.1. *Assume that $V = \{v_j\}_j = \{v(x_j)\}_j$ with $v(x)$ a Lipschitz continuous function. Then, if $I[V]$ is computed by* (3.5), *assumption* (2.6) *is satisfied for $n \leq 5$.*

Proof. First, the condition

$$I[V](x_k) = v(x_k) \tag{5.1}$$

is clearly satisfied since (3.5) is built as an interpolation. Concerning the second part of (2.6), namely,

$$|I[V](x) - I_1[V](x)| \leq C \max_{x_k \in U(x)} |v_{k+1} - 2v_k + v_{k-1}| \tag{5.2}$$

with $C < 1$, this condition is satisfied for all the polynomials $P_k(x)$ if $n \leq 5$, as proved in [13], since all the stencils S_k include the interval $[x_j, x_{j+1}]$. Although the structure under consideration for $I[V]$ is not globally polynomial, it is easy to see from (3.2), (3.3) that once x is given, since all the linear weights C_k are nonnegative by Theorem 4.1, then the right-hand side of (3.5) is in fact a convex combination of the polynomials P_k. Therefore,

$$\min_k P_k(x) \leq I[V](x) \leq \max_k P_k(x) \tag{5.3}$$

and, therefore, the constant C appearing in (2.6) is the same as obtained in [13] for polynomials of degree n. That is, (2.6) is satisfied with $C < 1$ up to WENO interpolations of fifth–ninth order. □

Remark. The positivity of the linear weights, obtained in the framework of WENO reconstructions, also allows to prove a further result related to Lagrange reconstructions. Actually, the bound (5.3) does not require $I[V]$ to be in the form (3.5), but it also applies to the form (3.1). Therefore, (2.6) is also satisfied for Lagrange reconstructions up to the ninth order, at least if the reconstruction stencil is symmetrical, which corresponds to odd orders of interpolation. Moreover, in the case of even (up to the eighth) orders, an analogous result can be obtained. More precisely, if a sixth order interpolation is performed using the nodes $x_{j-3}, \ldots, x_{j+3}$, the linear weights have the expressions

$$C_1(x) = \frac{(x - x_{j+2})(x - x_{j+3})}{30\Delta x^2},$$

$$C_2(x) = -\frac{(x - x_{j-3})(x - x_{j+3})}{15\Delta x^2},$$

$$C_3(x) = \frac{(x - x_{j-3})(x - x_{j-2})}{30\Delta x^2},$$

whereas if an eighth order interpolation is performed using the nodes $x_{j-4}, \dots, x_{j+4}$, the linear weights read

$$C_1(x) = -\frac{(x - x_{j+2})(x - x_{j+3})(x - x_{j+4})}{336\Delta x^3},$$

$$C_2(x) = \frac{(x - x_{j-4})(x - x_{j+3})(x - x_{j+4})}{112\Delta x^3},$$

$$C_3(x) = -\frac{(x - x_{j-4})(x - x_{j-3})(x - x_{j+4})}{112\Delta x^3},$$

$$C_4(x) = \frac{(x - x_{j-4})(x - x_{j-3})(x - x_{j-2})}{336\Delta x^3}.$$

Again, a direct computation shows that $C_k(x) > 0$ for $x \in [x_j, \dots, x_{j+1}]$, so that (5.3) (and hence (2.6)) still holds.

6. Numerical tests. In this section we present the results of several numerical tests in which we compare the WENO scheme described with P_1, P_2, cubic Lagrange, and third order ENO schemes. The WENO schemes considered are of second–third order and third–fifth order, and the two-dimensional reconstruction has been implemented by a tensor product of one-dimensional interpolations. The minimization step has been performed by bisection in the one-dimensional case, and by the Powell-like routine `PRAXIS` (taken from the NETLIB repository at ftp.netlib.org) in the two-dimensional examples. In order to evaluate the accuracy of the reconstruction, the tolerance in the minimization phase has been set to an order of magnitude far lower than the truncation error, although this may have somewhat slowed down the scheme. The upwinding along characteristics has been performed by means of a plain Euler method as in (2.2) (in the situation of H depending on the gradient alone, this choice has the best accuracy anyway).

Errors have been expressed in terms of the relative discrete L^∞ norm, defined as

$$\epsilon_{\infty,rel} = \frac{\max_j |v_j - v(x_j)|}{\max_j |v_j|}.$$

The computation of relative discrete L^1 norm of the errors has given similar indications and is therefore omitted.

Unless otherwise stated, figures showing approximate solutions have been obtained with the WENO 2/3 scheme.

In addition to the usual choice (3.4) as proposed by Shu, which has been denoted by S in the tables, some different possibility has been considered for the smoothness indicator. An indicator based on the second derivative alone, that is,

$$\beta_k = \int_{x_j}^{x_{j+1}} \Delta x^3 (P_k'')^2 dx, \tag{6.1}$$

has been denoted by D2 in the tables, whereas an indicator which only takes into account the highest order derivative, namely,

$$\beta_k = \int_{x_j}^{x_{j+1}} \Delta x^{2n-1} (P_k^{(n)})^2 dx, \tag{6.2}$$

has been implemented—it clearly coincides with the indicator D2 for the second–third order scheme, and has been denoted by D3 for the third–fifth order scheme. Last, as proposed by Jiang and Peng (see [17]), a smoothness indicator which takes into account all derivatives but the first, that is,

$$\beta_k = \sum_{l=2}^{n} \int_{x_j}^{x_{j+1}} \Delta x^{2l-1} (P_k^{(l)})^2 dx, \tag{6.3}$$

has been implemented for the third-fifth order scheme and denoted by D2+D3. Although the creation of singularities in HJ equations is at the level of the gradient, and therefore it could seem natural to drop first derivatives in the smoothness indicator, nevertheless, we will show that the use of the full expression (3.4) seems to be the most robust choice, although it might not be the most accurate one in any situation. We have summarized in Table 7 a comparison of the WENO schemes (with various smoothness indicators) with P_1, P_2, cubic, and third order ENO implementations of (2.2) on all numerical tests, in the standard setting of a 50×50 grid (50 nodes for test 1), and after the onset of the singularity for tests 1, 2, and 4.

Test 1: One-dimensional periodic solution. This test (also considered in [6], [20], [17]) deals with the HJ equation

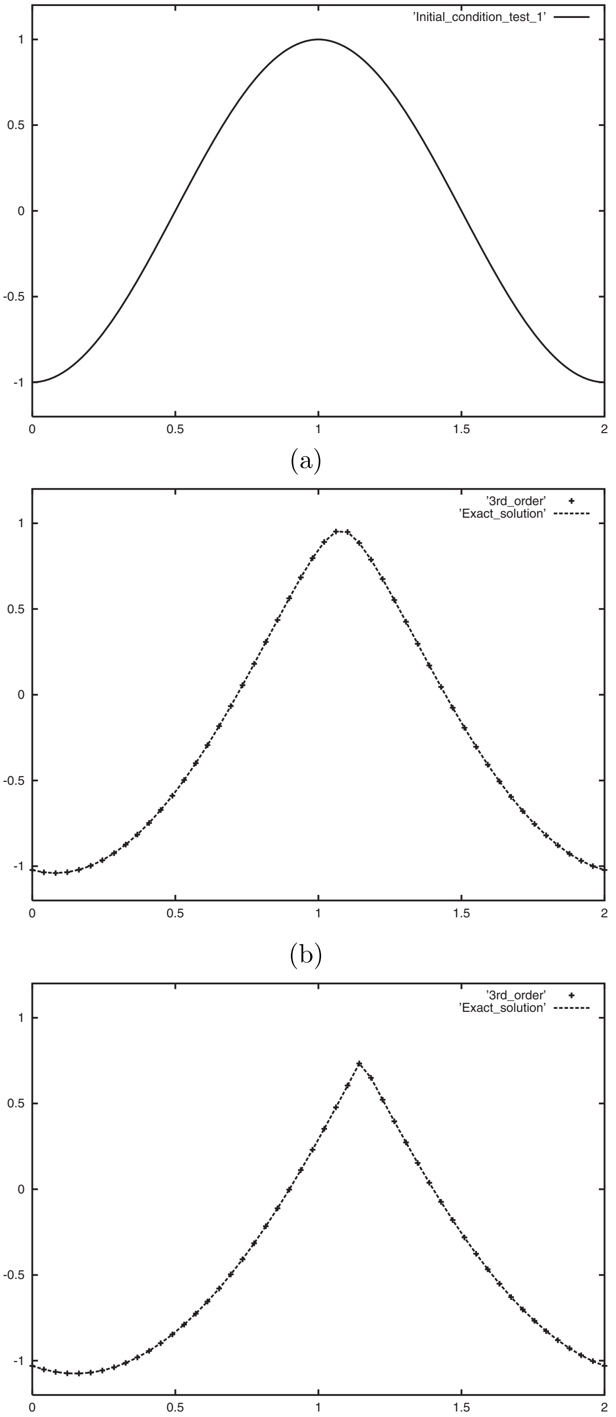
$$\begin{cases} v_t(x,t) + \frac{1}{2}(v_x(x,t)+1)^2 = 0, \\ v(x,0) = v_0(x) = -\cos(\pi x), \end{cases} \tag{6.4}$$

considered in $[0,2]$, with periodic boundary conditions. The approximate solution is computed before the singularity, at $T = 0.8/\pi^2$, with 4 time steps, and after the singularity, at $T = 1.5/\pi^2$, with 5 time steps. Plots of exact vs. approximate solutions are shown in Figure 1 and relative errors obtained with the Shu smoothness indicator are compared in Tables 2–3.

Note that according to the consistency estimate (2.3), the convergence rates should be $p = 4$ for the WENO3 scheme and $p = 6$ for the WENO5 scheme, since the time step has been kept constant. Roughly speaking, this behavior is confirmed by numerical tests, although with a slight decrease in the convergence rate after the onset of the singularity. However, note that even in this latter case the computation of the approximate solution is highly accurate. If the exact solution v is semiconcave (roughly speaking, this means that it can be written as the sum of a concave and a $W^{2,\infty}$ function), this fact can be heuristically explained as follows. At a time step n and for a given node x_j, the reconstruction is performed at an "upwind" point which in the Lax–Hopf formula would be a minimum of the function

$$F_{n,j}(a) = \Delta t H^*(a) + v(t_{n-1}, x_j + a\Delta t).$$

Now, the function to be minimized contains two terms. The first is smooth, due to the assumptions on H, whereas the second is semiconcave. Their sum is therefore semiconcave and it is known (see [4]) that minima of semiconcave functions can occur only at points where the function is at least differentiable. Although the scheme in



(a)

(b)

(c)

FIG. 1. *Approximate vs. exact solution for test* 1. *$N = 50$ points have been used.* (a) *Initial condition.* (b) *Solution before singularity.* (c) *Solution after singularity.*

Table 2

Errors before and after the singularity for test 1, WENO3 scheme.

WENO 3rd order				
	Before singularity		After singularity	
N	Relative L^∞-error	L^∞ order	Relative L^∞-error	L^∞ order
25	2.52×10^{-3}		2.88×10^{-3}	
50	8.77×10^{-5}	4.1	5.12×10^{-5}	5.8
100	1.53×10^{-5}	2.5	2.19×10^{-6}	4.5
200	9.63×10^{-7}	3.9	2.39×10^{-7}	3.1

Table 3

Errors before and after the singularity for test 1, WENO5 scheme.

WENO 5th order				
	Before singularity		After singularity	
N	Relative L^∞-error	L^∞ order	Relative L^∞-error	L^∞ order
25	1.29×10^{-3}		3.05×10^{-3}	
50	1.87×10^{-5}	6.1	5.83×10^{-6}	9.0
100	9.13×10^{-7}	4.3	7.25×10^{-8}	6.3
200	2.01×10^{-8}	5.5	1.89×10^{-9}	5.2

fact uses a discretization of the function $F_{n,j}$, this analysis seems to give a convincing explanation of numerical results.

In the summarizing Table 7 we note that the performance of the third order schemes is similar and that there is an apparent loss of accuracy in the D2 version of the WENO3 scheme, as well as in the D2+D3 version of the WENO5 scheme.

Test 2: Generation of a two-dimensional singularity. This test refers to the HJ equation

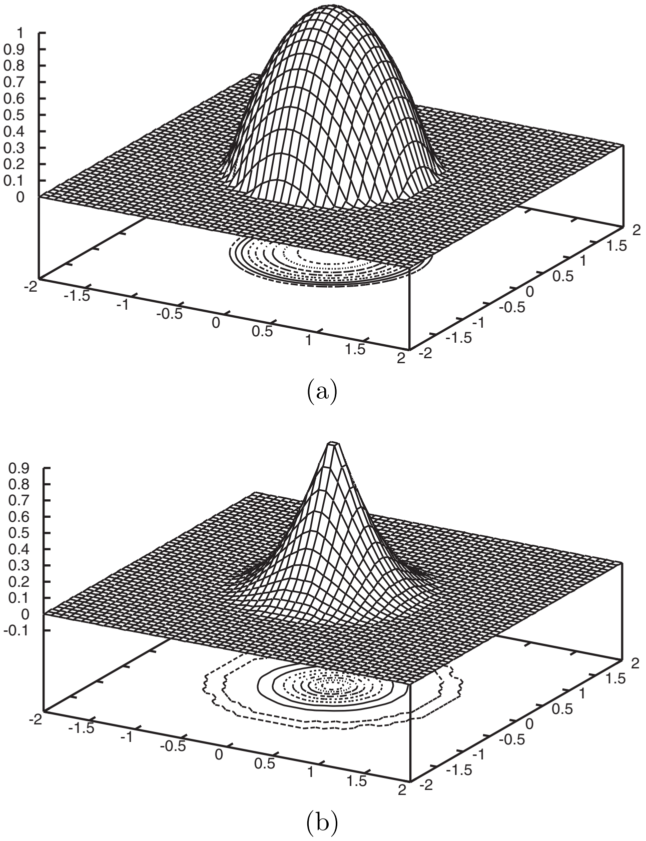
$$\begin{cases} v_t(x,t) + \frac{1}{2}|\nabla v(x,t)|^2 = 0, \\ v(x,0) = v_0(x) = \max(0, 1 - |x|^2). \end{cases} \tag{6.5}$$

The problem is considered in $[-2,2]^2$. For any positive time, the solution of this test problem has discontinuous second derivatives along the circle $|x| = 1$ and generates a further singularity in the gradient at the origin for $t \geq 1/2$.

The exact solution of (6.5) can be explicitly computed and for $t \geq 1/2$ reads

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

The solution, computed with $\Delta t = 0.1$, is shown in Figure 2 at $t = 0$ and $t = 0.5$. Table 4 shows execution times (on a SUN E 420 R platform) at $t = T = 0.5$ for different reconstructions and different smoothness indicators, on a 50×50 grid. The error Table 7 shows that the solution is computed with nearly the same accuracy by all the schemes, and the use of nonoscillatory interpolation does not seem to give any advantage (this is related to the lack of uniform semiconcavity of the solution; see the above discussion of this point). We point out, in any case, that the singularity in the gradient is well resolved, and that the scheme does not introduce instabilities. However, looking at Figure 3, in which we zoom the solution close to the x_1-x_2 plane for the third order linear, ENO, and WENO schemes, we may see that the plain cubic



(a)

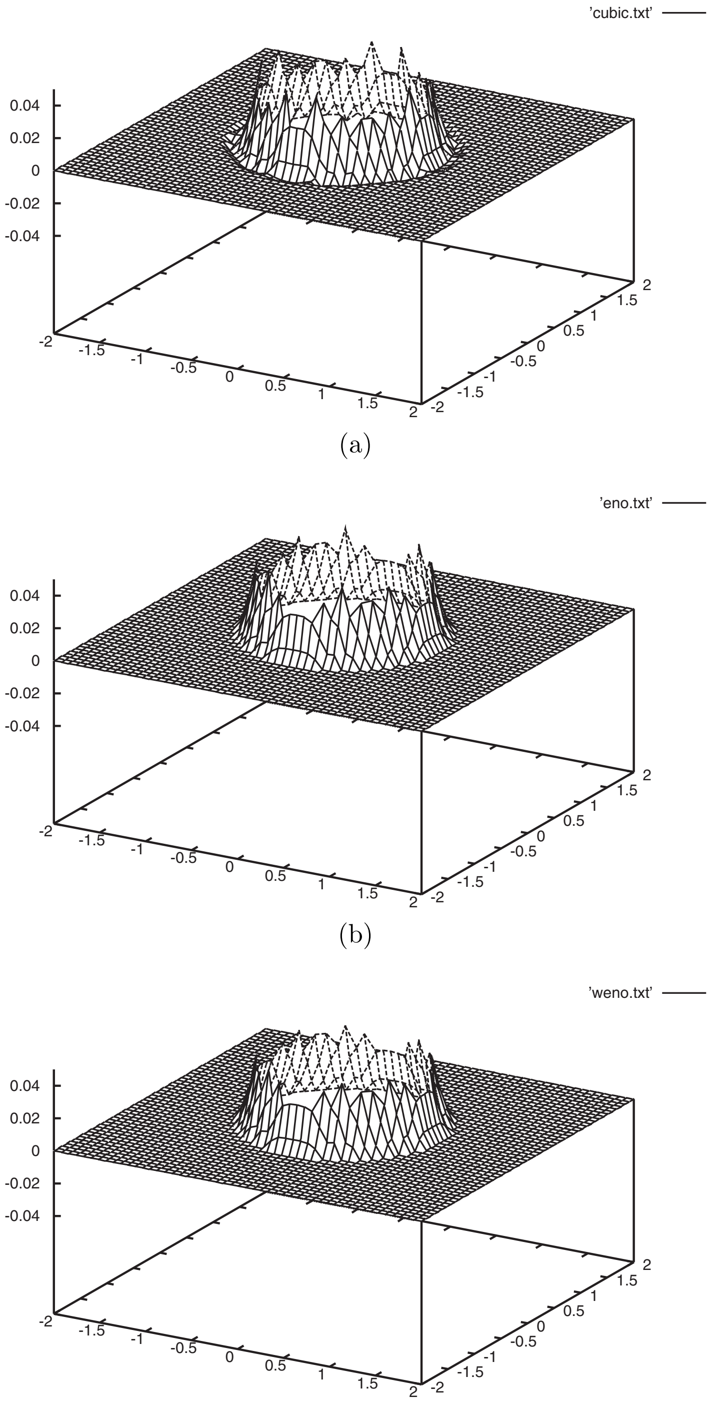
(b)

Fig. 2. *Approximate solution for test* 2. (a) *Initial condition.* (b) *Solution at* $t = 0.5$.

Table 4
Comparison of CPU times for different schemes on test 2.

Scheme	CPU
P_1	34s
P_2	41s
cubic	52s
ENO 3	3m36s
WENO 2/3 S	1m43s
WENO 2/3 D2	1m55s
WENO 3/5 S	3m55s
WENO 3/5 D2	3m55s
WENO 3/5 D3	4m02s
WENO 3/5 D2+D3	4m05s

reconstruction, as expected, introduces a zone of undershoot in the neighborhood of the circle $|x| = 1$, where a discontinuity in the second derivative appears. This artifact is avoided by nonoscillatory schemes, among which, however, the WENO scheme performs apparently better in terms of CPU time. Also, the behavior of the scheme is remarkably isotropic, as shown in the scatter plot of Figure 4, in which at



(a)

(b)

(c)

FIG. 3. *Zoom of the approximate solutions for test* 2. (a) *Linear third order scheme.* (b) *Third order ENO scheme.* (c) *Third order WENO scheme.*

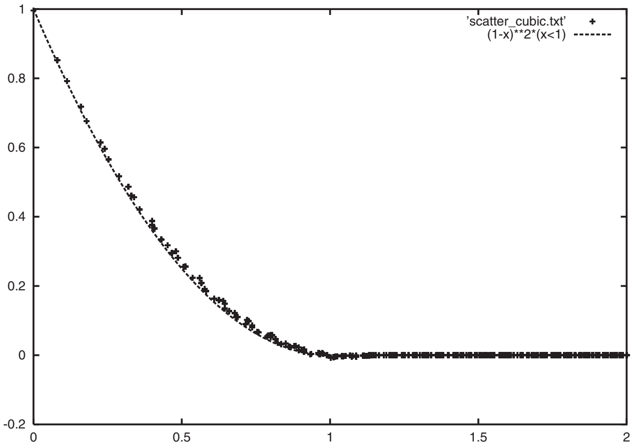


FIG. 4. *Scatter plot of the approximate solution for test* 2.

any node the numerical solution is plotted as a function of $|x|$ for the cubic Lagrange reconstruction (ENO and WENO reconstructions give similar results).

Test 3: Uniformly semiconcave solution. This test refers again to the HJ equation

$$\begin{cases} v_t(x,t) \ + \ \frac{1}{2}|\nabla v(x,t)|^2 = 0, \\ v(x,0) \ = v_0(x) = \min(0, |x|^2 - 1), \end{cases} \tag{6.6}$$

considered in $[-2,2]^2$, in which the initial condition v_0 has been changed to a semiconcave function.

The exact solution of (6.6) is now uniformly semiconcave for any positive time; more precisely, it has the explicit form

$$v(t,x) = \begin{cases} \frac{|x|^2}{2t+1} - 1 & \text{if} |x| \le 2t+1, \\ 0 & \text{if} |x| \ge 2t+1. \end{cases}$$

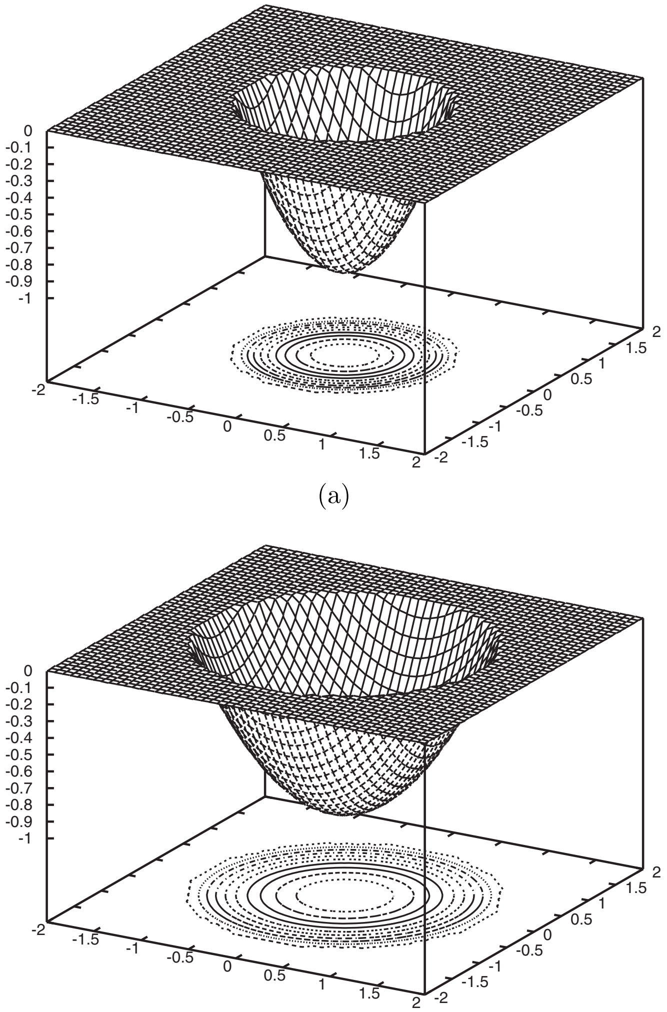
The approximate solution is shown in Figure 5 at $t = 0$ and $t = 0.5$ with a 50×50 nodes computational grid. Table 7 reports a strong reduction of the error with respect to the previous test. This shows that in the case of semiconcave solutions, numerical schemes can take advantage of higher consistency rates, although the piecewise quadratic structure of the solution is too simple to be really discriminating in this respect. Among different smoothness indicators, the best accuracy is achieved by D2+D3 indicator, whereas other choices yield similar results.

Test 4: Two-dimensional periodic solution. This test (taken again from [6], [17], [20]) deals with the HJ equation

$$\begin{cases} v_t(x,t) + \frac{(v_{x_1}(x,t) + v_{x_2}(x,t)+1)^2}{2} = 0, \\ v(x,0) = v_0(x) = -\cos\frac{\pi(x_1+x_2)}{2}, \end{cases} \tag{6.7}$$

considered in $[-2,2]^2$, with doubly periodic boundary conditions.

In this situation, the structure of the solution is more complex, and we have tested the various schemes in terms of convergence rate using computational meshes



(a)

(b)

FIG. 5. *Approximate solutions for test* 3. (a) *Initial condition.* (b) *Solution at* $t = 0.5$.

of 25 × 25, 50 × 50, 100 × 100, and 200 × 200 nodes. The approximate solution has been computed before the singularity, at $T = 0.8/\pi^2$, with 4 time steps, and after the singularity, at $T = 1.5/\pi^2$, with 5 time steps. The Euler method has been used to approximate characteristics and reconstruction has been performed by the WENO Shu version. Relative errors are reported in Tables 5–6, whereas Figure 6 shows the approximate solution plotted at $t = 0$ and $t = T = 1.5/\pi^2$.

Error tables confirm the theoretical convergence rate of the schemes in the presence of a smooth solution, whereas, after the singularity, convergence rate of the WENO5 scheme is heavily underestimated by the theory (we suppose this might be due to particular circumstances of the test).

The comparison among the different schemes, in Table 7, shows (at fixed 50 × 50 grid and after the singularity) similar performances except for a loss in accuracy of the WENO3 D2 scheme. At this level the fifth order scheme has no better performance,

Table 5
Errors before and after the singularity for test 4*, WENO*3 *scheme.*

WENO 3rd order				
	Before singularity		After singularity	
N	Relative L^∞-error	L^∞ order	Relative L^∞-error	L^∞ order
25	3.00×10^{-3}		1.00×10^{-2}	
50	4.66×10^{-4}	2.6	8.68×10^{-5}	6.8
100	2.68×10^{-5}	4.1	9.35×10^{-6}	3.2
200	1.48×10^{-6}	4.1	2.97×10^{-7}	4.4

Table 6
Errors before and after the singularity for test 4*, WENO*3 *scheme.*

WENO 5th order				
	Before singularity		After singularity	
N	Relative L^∞-error	L^∞ order	Relative L^∞-error	L^∞ order
25	1.28×10^{-3}		8.55×10^{-3}	
50	4.89×10^{-5}	4.3	8.53×10^{-4}	3.6
100	2.07×10^{-6}	4.5	2.08×10^{-6}	8.6
200	2.37×10^{-8}	6.4	5.34×10^{-9}	8.6

Table 7
Comparison of relative L^∞-error for different schemes on tests 1–4*.*

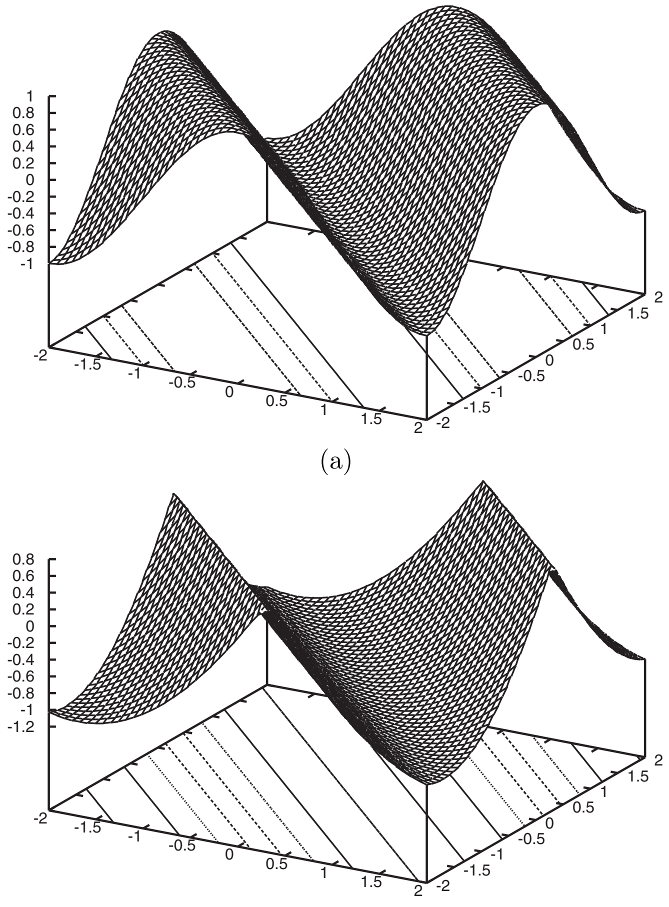
Scheme	Test 1	Test 2	Test 3	Test 4
P_1	5.93×10^{-3}	3.79×10^{-2}	1.20×10^{-2}	1.23×10^{-2}
P_2	2.47×10^{-4}	1.69×10^{-2}	5.82×10^{-3}	7.24×10^{-4}
Cubic	5.12×10^{-5}	2.54×10^{-2}	1.20×10^{-2}	3.18×10^{-4}
ENO 3	3.51×10^{-5}	2.75×10^{-2}	5.80×10^{-4}	1.43×10^{-4}
WENO 2/3 S	5.12×10^{-5}	2.69×10^{-2}	1.20×10^{-2}	8.68×10^{-5}
WENO 2/3 D2	6.81×10^{-4}	3.15×10^{-2}	1.20×10^{-2}	1.29×10^{-3}
WENO 3/5 S	5.83×10^{-6}	2.52×10^{-2}	3.17×10^{-3}	8.53×10^{-4}
WENO 3/5 D2	7.52×10^{-6}	2.58×10^{-2}	1.14×10^{-3}	1.45×10^{-4}
WENO 3/5 D3	3.00×10^{-6}	2.97×10^{-2}	3.34×10^{-3}	4.13×10^{-5}
WENO 3/5 D2+D3	6.17×10^{-5}	2.68×10^{-2}	8.73×10^{-5}	1.78×10^{-4}

although looking at Table 5 we see that it would have a great improvement passing from 50 to 100 nodes.

Conclusions. We have built a class of WENO large time-step schemes based on an SL formulation. Under suitable assumptions, convergence of the schemes can be proved up to the ninth order. Theoretical consistency analysis is generally confirmed by numerical experiments, even when singular solutions are taken into account. Among all the smoothness indicators considered, there has been now and then a considerably better or worse accuracy for some of them. The original Shu indicator, however, seems to be a robust and accurate choice in all the cases tested.

While this work has been focused on the special case of (1.1), this class of schemes can be immediately adapted (see the appendix in [4]) to treat dynamic programming equations of the Bellman type

$$v + \max_{a \in A}\{ -b(x,a) \cdot Dv - f(x,a)\} = 0 \tag{6.8}$$



(a)

(b)

FIG. 6. *Approximate solutions for test* 4. (a) *Initial condition.* (b) *Solution at* $t = 1.5/\pi^2$.

(which leads to a convex Hamiltonian and includes (1.1) as a special case) or, in even greater generality, of the nonconvex Isaacs type

$$v + \min_{a_2 \in A_2} \max_{a_1 \in A_1} \{ -b(x, a_1, a_2) \cdot Dv - f(x, a_1, a_2)\} = 0, \tag{6.9}$$

although in such settings no complete theory exists on the high-order case.

The advantage of WENO over ENO reconstructions is apparent in terms of CPU time, while the accuracy in both cases seems to be comparable with the accuracy of linear Lagrange reconstructions. WENO techniques can be valuable, however, when special information associated to singularities (e.g., switching curves for dynamic programming equations) requires a nonoscillatory behavior of the approximation scheme.

REFERENCES

[1] R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation,* J. Comput. Phys., 114 (1994), pp. 45–58.

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

[3] M. Abramowitz and I.A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables,* National Bureau of Standards, Washington, D.C., 1964.

[4] M. Bardi and I. Capuzzo Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman Equations,* Birkhäuser, Boston, 1997.

[5] G. Barles, *Solutions de viscositè des equations d'Hamilton–Jacobi,* Springer-Verlag, Berlin, 1998.

[6] S. Bryson and D. Levy, *Central schemes for multidimensional Hamilton–Jacobi equations,* SIAM J. Sci. Comput., 25 (2003), pp. 767–791.

[7] S. Bryson and D. Levy, *High-order central schemes for multidimensional Hamilton–Jacobi equations,* SIAM J. Numer. Anal., 41 (2004), pp. 1339–1369.

[8] R. Courant, E. Isaacson, and M. Rees, *On the solution of nonlinear hyperbolic differential equations by finite differences,* Comm. Pure Appl. Math., 5 (1952), pp. 243–255.

[9] M. Falcone, *Numerical solution of dynamic programming equations,* in Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman Equations, Appendix A, Birkhäuser, Boston, 1997.

[10] M. Falcone and R. Ferretti, *Discrete-time high-order schemes for viscosity solutions of Hamilton–Jacobi equations,* Numer. Math., 67 (1994), pp. 315–344.

[11] M. Falcone and R. Ferretti, *Semi-Lagrangian schemes for Hamilton–Jacobi equations, discrete representation formulae and Godunov methods,* J. Comput. Phys., 175 (2002), pp. 559–575.

[12] M. Falcone and T. Giorgi, *An approximation scheme for evolutive Hamilton–Jacobi equations,* in Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin, and Q. Zhang, eds., Birkhäuser, Boston, 1998.

[13] R. Ferretti, *Convergence of semi-Lagrangian approximations to convex Hamilton–Jacobi equations under (very) large Courant numbers,* SIAM J. Numer. Anal., 40 (2003), pp. 2240–2253.

[14] A. Harten and S. Osher, *Uniformly high-order accurate nonoscillatory schemes.* I, SIAM J. Numer. Anal., 24 (1987), pp. 279–309.

[15] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes.* III, J. Comput. Phys., 71 (1987), pp. 231–303.

[16] C. Hu and C.-W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes,* J. Comput. Phys., 150 (1999), pp. 97–127.

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

[18] G.S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes,* J. Comput. Phys., 126 (1996), pp. 202–228.

[19] R.J. LeVeque, *Numerical Methods for Conservation Laws,* Birkhäuser, Basel, 1992.

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

[21] P.L. Lions, *Generalized Solutions of Hamilton–Jacobi Equations,* Pitman, London, 1982.

[22] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes,* J. Comput. Phys., 115 (1994), pp. 200–212.

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

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

[25] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws,* in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Math. 1697, Papers from the C.I.M.E. Summer School held in Cetraro, June 23–28, 1997, A. Quarteroni, ed., Springer-Verlag, Berlin; Centro Internazionale Matematico Estivo (C.I.M.E.), Florence, 1998.

[26] J. Shi, C. Hu, and C.-W. Shu, *A technique of treating negative weights in WENO schemes,* J. Comput. Phys., 175 (2002), pp. 108–127.

[27] A.N. Staniforth and J. Côtè, *Semi-Lagrangian integration schemes for atmospheric models: A review,* Mon. Weather Rev., 119 (1991), pp. 2206–2223.

[28] J. Strain, *Semi-Lagrangian methods for level set equations,* J. Comput. Phys., 151 (1999), pp. 498–533.

[29] Y.-T. Zhang and C.-W. Shu, *High-order WENO schemes for Hamilton–Jacobi equations on triangular meshes,* SIAM J. Sci. Comput., 24 (2002), pp. 1005–1030.