SIAM J. NUMER. ANAL.
Vol. 45, No. 5, pp. 1979–2011
© 2007 Society for Industrial and Applied Mathematics

FAST SEMI-LAGRANGIAN SCHEMES FOR THE EIKONAL EQUATION AND APPLICATIONS*

EMILIANO CRISTIANI† AND MAURIZIO FALCONE‡

Abstract. We introduce and analyze a fast version of the semi-Lagrangian algorithm for front propagation originally proposed in [M. Falcone, "The minimum time problem and its applications to front propagation," in *Motion by Mean Curvature and Related Topics*, A. Visintin and G. Buttazzo, eds., de Gruyter, Berlin, 1994, pp. 70–88]. The new algorithm is obtained using the local definition of the approximate solution typical of semi-Lagrangian schemes and redefining the set of "neighboring nodes" necessary for fast marching schemes. A new proof of convergence is needed since that definition produces a new *narrow band* centered at the interphase which is larger than the one used in fast marching methods based on finite differences. We show that the new algorithm converges to the viscosity solution of the problem and that its complexity is $O(N \log N_{nb})$, as it is for the fast marching method based on finite difference (N and N_{nb} being, respectively, the total number of nodes and the number of nodes in the *narrow band*). A new sufficient condition for the convergence of the standard finite difference fast marching method is also given. We present several tests comparing the two algorithms and other fast methods (e.g., fast sweeping) on a series of benchmarks which include the minimum time problem and the shape-from-shading problem.

Key words. front propagation, eikonal equation, semi-Lagrangian schemes, finite differences, fast marching methods

AMS subject classifications. Primary, 65N12; Secondary, 49L25, 49L20

DOI. 10.1137/050637625

1. Introduction.

The level set method is a clever and rather simple way to describe an interface separating two or more regions with different physical phases. As is well known, the method describes the evolution of the front by a continuous representation function $u(x,t)$ which is negative in the domain Ω_t corresponding to one of the phases, positive outside that domain, and changes sign across the interfaces. A comprehensive introduction to the level set method as well as to several applications and references can be found in [19] and [32].

The level set method leads to a nonlinear first order PDE whenever the interface evolution is simply driven by a normal velocity and (possibly) a given advection term. More complicated types of evolution consider the normal velocity as a function of the curvature and/or of other geometric parameters of the interface, and this leads to second order nonlinear PDEs (or integrodifferential equations).

The typical model problem for an interface which evolves in the normal direction driven by a given scalar velocity $c(x) : \mathbb{R}^n \to \mathbb{R}$ leads to the first order Hamilton–Jacobi equation

$$\begin{cases} u_t(x,t) + c(x)|\nabla u(x,t)| = 0 & \text{for } x \in \mathbb{R}^n\,, \quad t \in (0,+\infty), \\ u(x,0) = u_0(x) & \text{for } x \in \mathbb{R}^n, \end{cases} \tag{1}$$

*Received by the editors August 5, 2005; accepted for publication (in revised form) January 18, 2007; published electronically September 7, 2007. This research was partially supported by the MIUR Project 2003 "Modellistica Numerica per il Calcolo Scientifico ed Applicazioni Avanzate."
http://www.siam.org/journals/sinum/45-5/63762.html
†Dipartimento di "Metodi e Modelli Matematici per le Scienze Applicate," SAPIENZA - Università di Roma, Via A. Scarpa, 16 - 00161, Roma, Italy (cristiani@dmmm.uniroma1.it).
‡Dipartimento di Matematica, SAPIENZA - Università di Roma, P.le Aldo Moro, 5 - 00185, Roma, Italy (falcone@mat.uniroma1.it).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where the initial condition u_0 must be a representation function of the initial position of the front $\Gamma_0 = \partial\Omega_0$. Note that u_0 is unknown since Γ_0 is the only initial datum, so that the first step is usually to compute u_0. The above problem can be simplified when the evolution is monotone (increasing or decreasing), i.e., when either $\Omega_t \subset \Omega_{t+s}$ or the reverse inclusion are satisfied for any $t, s > 0$. For monotone types of evolution, it has been proved in [17] (see also [26]) that (1) can be replaced by the stationary equation

$$\begin{cases} c(x)|\nabla T(x)| = 1 & \text{for } x \in \mathbb{R}^n \backslash \Omega_0, \\ T(x) = 0 & \text{for } x \in \partial\Omega_0, \end{cases} \tag{2}$$

where we assume

$$c > 0 \tag{3}$$

and T represents the time needed to transfer a point $x \in \mathbb{R}^n \backslash \Omega_0$ to Ω_0 by appropriate dynamics (see below). In fact, the link between the two equations is simple: if T is the viscosity solution of (2), then $u(x,t) = T(x) - t$ is the viscosity solution of (1). It is worth noting that the second problem is easier to solve since it does not require the additional computation of u_0, which requires the solution of another Hamilton–Jacobi equation of type (2) to compute the (signed) distance function to Ω_0. Moreover, the knowledge of T gives a description of the interface for every time t using the fact that $\Gamma_t = \partial\Omega_t = \{x \in \mathbb{R}^n : T(x) = t\}$. On the other hand, (1) is preferable to its stationary version whenever it is needed to derive a high order scheme that has the same efficiency as the formally first order one (see, e.g., [1]). However, some results on high order methods for stationary first order Hamilton–Jacobi equations including (2) are available, e.g., in [15].

Note that the above stationary approach relies on the link between the propagations of fronts and the minimum time problem of control theory. In fact, as shown in [17], by the change of variable (Kružkov transform)

$$v(x) = 1 - e^{-T(x)} \tag{4}$$

we can transform (2) into the equation

$$\begin{cases} v(x) + \max\limits_{a \in B(0,1)} \{c(x)a \cdot \nabla v(x)\} = 1 & \text{for } x \in \mathbb{R}^n \backslash \Omega_0, \\ v(x) = 0 & \text{for } x \in \partial\Omega_0, \end{cases} \tag{5}$$

where $B(0,1)$ is the unit ball centered in 0. This is the Hamilton–Jacobi–Bellman equation of a minimum time problem for the dynamics

$$\begin{cases} \dot{y}(t) = -c(y)\alpha(t), & t \in (0, +\infty), \\ y(0) = x, \end{cases} \tag{6}$$

where $\alpha(\cdot) \in \mathcal{A} = \big\{\alpha(\cdot) : [0, +\infty) \to B(0,1) \subset \mathbb{R}^n, \text{ measurable}\big\}$. We will denote by $y(t; \alpha, x)$ the solution of the system corresponding to the control α and to the initial condition x. The usual requirement in order to have existence and uniqueness for the trajectories under the Carathéodory conditions is

$$c(x) \text{ Lipschitz continuous and bounded.} \tag{7}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Let us define the cost functional

$$J_x(\alpha(\cdot)) = \inf\left\{t : y(t;\alpha,x) \in \Omega_0\right\} \leq +\infty.$$

It is well known that the minimum time function

$$T(x) = \inf_{\alpha(\cdot)\in\mathcal{A}} J_x(\alpha(\cdot)) \tag{8}$$

is the unique viscosity solution of (2) (see, e.g., [2, 4]).

We will focus our attention on the numerical solution of (2). It should be noted that a fast marching method has a lower cost with respect to the corresponding classical iterative method (or fixed point method), which computes the solution on the whole grid at every iteration. The classical fast marching method based on finite differences (FM-FD) was proposed in [33] as an acceleration method for a monotone first order iterative finite difference scheme (see [7] for a second order version of the scheme and [8] for a general convergence result). Since semi-Lagrangian schemes have shown to be more accurate than the finite difference schemes corresponding to the same order, it is natural to extend the ideas behind the FM-FD method to this class of schemes. In the framework of semi-Lagrangian schemes several convergence results and a priori error estimates have been obtained via control arguments since these schemes correspond to a discrete version of the dynamic programming principle; see [2] and [13]. Moreover, these schemes do not require an explicit and restrictive CFL condition for stability (see [16]). It is interesting to note that the first tentative steps in this direction can be found in [38] using a different approximation scheme. More recently, a semi-Lagrangian scheme has been proposed by Sethian and Vladimirsky in [34] in a more general framework which includes anisotropic front propagation on unstructured grids.

Our main contribution here is to introduce and analyze a new fast marching version of a semi-Lagrangian scheme, for which a priori error estimates are available in [12], and to prove an upper bound on its computational cost. We will also review the basic features of the FM-FD method and give a complete proof of its convergence under an explicit CFL condition which guarantees that the scheme is always meaningful and there are no complex solutions. To our knowledge this condition appears for the first time in the literature; our proof is presented in the appendix. Lastly we will recall the fast sweeping method studied by Zhao [39] (see also [37, 20, 21, 27, 28] for other sweeping methods and extensions) and also provide a sweeping version of our algorithm. Further extensions and new applications of the scheme presented in this paper can be found in [9].

The paper is organized as follows. In section 2 we recall the basic features of the FM-FD method introduced in [33] to solve (2) when $c(x)$ has a constant sign in its domain of definition. An example which shows that the FM-FD scheme can produce complex solutions is given in the same section, and the proof of convergence under a new CFL condition which always guarantees real solutions is presented in the appendix. Section 3 is devoted to the presentation of the fast marching semi-Lagrangian method (FM-SL) for (5). Section 4 contains some properties of the FM-SL scheme that will be useful in establishing its convergence, which will be proved in section 5. In the same section we analyze the computational complexity, showing that the FM-SL scheme has a complexity of order $O(N\ln(N_{nb}))$, where N is the total number of nodes of our computational grid and N_{nb} is the number of nodes in the *narrow band* (bounded by N). In section 6 we also present other fast algorithms and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

give the sweeping version of our method. Finally, section 7 is devoted to numerical tests and to comparisons between several FM schemes on a number of benchmarks.

2. The fast marching methods based on finite differences. The fast marching method has been introduced to reduce the computational effort needed to solve (2). The basic level set algorithm is based on a finite difference discretization and on an iterative procedure $T^{n+1} = F(T^n)$ which computes the approximate solution everywhere in $\mathbb{R}^n \setminus \Omega_0$ at every iteration. The FM-FD method instead follows the front concentrating the computational effort where it is needed, i.e., in a small neighborhood of the front, and it updates that neighborhood at every iteration to avoid useless computations. This is done by dividing the grid nodes into three subsets: *far* nodes, *accepted* nodes, and *narrow band* nodes. The narrow band nodes are the nodes where the computation actually takes place and their value can still change at the following iterations. The accepted nodes are those where the solution has been already computed and where the value cannot change in the following iterations. Finally, the far nodes are the remaining nodes where an approximate solution has never been computed. In physical terms, the far nodes are those in the space region which has never been touched by the front, the accepted nodes are those where the front has already passed through, and the narrow band nodes are, iteration by iteration, those lying in a neighborhood of the front.

The algorithm starts labeling as *accepted* only the nodes belonging to the initial front, i.e., belonging to $\Gamma_0 = \partial\Omega_0$, and ends only when all the nodes have been accepted. In this section, we will briefly sketch the FM-FD scheme for (2). In order to avoid cumbersome notation we will restrict the presentation to the case $n = 2$. In what follows, we will always consider the case of a positive normal velocity; i.e., we assume $c(x) > 0$ to guarantee a monotone (increasing) evolution of the front. The results in this section can be easily generalized to the n-dimensional case and to the case $c(x) < 0$.

We will take a square Q large enough to contain Ω_0; this is the domain where we want to compute T. Boundary conditions will be given on ∂Q and Γ_0 but, as a first step, we will consider the algorithm *without* boundary conditions on ∂Q. The implementation of boundary conditions in the scheme will be discussed in section 5.

We will assume that we are working on a structured grid of $M \times N$ nodes (x_i, y_j), $i = 1, \ldots, N$ and $j = 1, \ldots, M$. Δx and Δy will denote the (uniform) discretization steps, respectively, on the x and y axes. We will denote by $T_{i,j}$ and $c_{i,j}$, respectively, the values of T and c at (x_i, y_j).

Let us write (2) as

$$T_x^2 + T_y^2 = \frac{1}{c^2(x,y)}. \tag{9}$$

We replace the partial derivatives T_x and T_y by first order finite differences, and we choose for simplicity $M = N$ and $\Delta x = \Delta y$. It is well known that in order to obtain an approximation of the viscosity solution, an *up-wind* correction must be introduced. This leads to the equation

$$\begin{aligned} &\left(\max\left\{ \max\left\{ \frac{T_{i,j} - T_{i-1,j}}{\Delta x}, 0 \right\}, -\min\left\{ \frac{T_{i+1,j} - T_{i,j}}{\Delta x}, 0 \right\} \right\} \right)^2 \\ &+ \left(\max\left\{ \max\left\{ \frac{T_{i,j} - T_{i,j-1}}{\Delta x}, 0 \right\}, -\min\left\{ \frac{T_{i,j+1} - T_{i,j}}{\Delta x}, 0 \right\} \right\} \right)^2 = \frac{1}{c_{i,j}^2}. \end{aligned} \tag{10}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

2.1. The FM-FD algorithm. Let us briefly recall the main definitions and steps of the FM-FD method.

DEFINITION 2.1 (neighboring nodes for the finite difference scheme). *Let* $X = (x_i, y_j)$ *be a node. We define the set of neighboring nodes to* X *as*

$$N_{FD}(X) = \Big\{(x_{i+1}, y_j), (x_{i-1}, y_j), (x_i, y_{j+1}), (x_i, y_{j-1})\Big\}.$$

These are the nodes appearing in the stencil of the first order finite difference discretization. The definition can be easily extended to the n-dimensional case.

Sketch of the FM-FD algorithm.

Initialization.

1. The nodes belonging to the initial front Γ_0 are located and labeled as *accepted*. Their value is set to $T = 0$ (they form the set $\widetilde{\Gamma}_0$).
2. The initial narrow band is defined taking the nodes belonging to $N_{FD}(\widetilde{\Gamma}_0)$, external to Γ_0. These nodes are labeled as *narrow band*, setting the value to $T = \frac{\Delta x}{c}$.
3. The remaining nodes are labeled as *far*, and their value is set to $T = +\infty$ (in practice, the maximum floating point number).

Main cycle.

1. Among all the nodes in the narrow band we search for the minimum value of T. Let us denote this node by A.
2. A is labeled as *accepted* and is removed from the *narrow band*.
3. The nodes in $N_{FD}(A)$ which are not accepted are labeled as *active*. If among these nodes there are nodes labeled as *far*, they are transferred to the narrow band.
4. The value of T in the active nodes is computed (or recomputed), solving the second order equation (10) and taking the largest root.
5. If the narrow band is not empty, go back to 1.

Note that the narrow band is a reasonable approximation of the level set of $T(x, y)$.

The main interest in the FM-FD method is that its computational cost is bounded. In fact, every node cannot be accepted more than one time and every node has just four neighbors, so the bound on the maximum number of times a single node can be recomputed is four. This corresponds to a computational cost of $O(N)$, where N is the total number of nodes. We should add to that cost the search for the minimum value of T among the nodes in the narrow band, which costs $O(\ln(N_{nb}))$, where N_{nb} is the number of nodes in the narrow band. In conclusion, the algorithm has a global cost of $O(N \ln(N_{nb}))$ operations (see [38, 32, 33] for further details on the computational cost). This is not the case for the usual iterative/fixed point algorithm since in that case the approximate solution is obtained in the limit and, in practice, no one knows when the stopping criterion will apply; i.e., the maximum number of iterations is virtually unbounded.

Let us observe that it is necessary to introduce some conditions or to modify the scheme in order to avoid inconsistencies due to the appearance of imaginary solutions. In fact, let us consider the discretization (10) and suppose that

$$T_{i,j} < T_{i+1,j}, \qquad T_{i,j} < T_{i,j-1}, \qquad T_{i,j} > T_{i-1,j}, \qquad T_{i,j} > T_{i,j+1}.$$

It is easy to check that (10) corresponds to

$$\left(\frac{T_{i,j} - T_{i-1,j}}{\Delta x}\right)^2 + \left(\frac{T_{i,j+1} - T_{i,j}}{\Delta x}\right)^2 = \frac{1}{c_{i,j}^2},$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

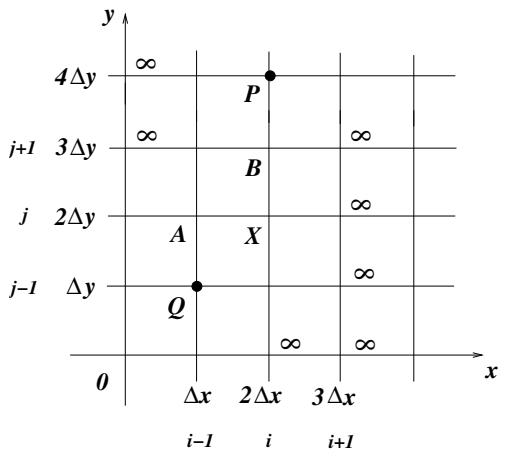

FIG. 1. *A configuration with complex roots.*

which gives

$$T_{i,j} = \frac{T_{i-1,j} + T_{i,j+1} \pm \sqrt{2\left(\frac{\Delta x}{c_{i,j}}\right)^2 - (T_{i-1,j} - T_{i,j+1})^2}}{2}. \tag{11}$$

We already noted that the term under the square root can be negative. Obviously this must be avoided since complex roots have no physical meaning. A situation where this occurs is the following example.

Consider the case where the initial front is the union of two points, i.e., $\Gamma_0 = P \cup Q$, $Q = (\Delta x, \Delta y)$ and $P = (2\Delta x, 4\Delta y)$ (see Figure 1). Let us consider the following velocity:

$$c(x,y) = \begin{cases} \varepsilon, & y \le \Delta y, \\ \varepsilon + \frac{1-\varepsilon}{\Delta y}(y - \Delta y), & \Delta y \le y \le 2\Delta y, \\ 1 - \frac{1-\varepsilon}{\Delta y}(y - 2\Delta y), & 2\Delta y \le y \le 3\Delta y, \\ \varepsilon, & y \ge 3\Delta y. \end{cases} \tag{12}$$

In this case the algorithm initializes the narrow band, computing a large value for B when ε is small and a small value for the node A which will be the first node accepted (after Γ_0). When the node X has to be computed, its value depends on $T(A)$ and $T(B)$. Since $c(X) = 1$ and $T(A) - T(B)$ is large (for ε small) the radicand in (11) will be negative (as numerical tests confirm).

This difficulty can be solved by either choosing the positive part of the radicand (as suggested in [22]) or changing discretization, as in [39]. However, both choices lead to a modification of the scheme, which can be difficult to handle when looking for theoretical results. We prefer to avoid changing the scheme, and we prove that under the CFL-like condition

$$\Delta x \le (\sqrt{2} - 1)\frac{c_{min}}{L_c} \tag{13}$$

the algorithm always computes *real* solutions at every node (here c_{min} is the minimum value of c, L_c is its Lipschitz constant, and again $\Delta x = \Delta y$). Condition (13) has a

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

clear meaning and allows us to give a proof of convergence to the viscosity solution. To our knowledge this is the first time this condition appears in the literature; a complete proof of the convergence result (Proposition 2.1) will be given in the appendix.

Let us denote by A the node in the narrow band where the minimum value of T is attained. The algorithm labels A as *accepted* and starts to compute the neighboring nodes which are not accepted.

PROPOSITION 2.1. *Let $X = (x_i, y_j) \in N_{FD}(A)$ be the node where the FM-FD method computes a solution. Let us assume that*

$$c_{min} = \min_{Q\setminus\Omega_0} c(x) > 0 \tag{14}$$

and that the following CFL-like condition holds true:

$$\Delta x \leq (\sqrt{2}-1)\frac{c_{min}}{L_c}, \tag{15}$$

where L_c denotes the Lipschitz constant of c. Then we have

$$T(A) \leq T(X) \leq T(A) + f_X, \tag{16}$$

where $f_X := \Delta x/c(X)$.

The above result is crucial in order to obtain convergence in a finite number of steps. In fact, it shows that the minimum value of the nodes in the narrow band (which is actually the only value accepted at every iteration) is exact within the consistency error of the scheme. An approximate value is considered to be exact if the algorithm cannot replace it with a strictly lower value at any of the following iterations.

3. The fast marching method based on the semi-Lagrangian scheme. We will study a fast marching version of the semi-Lagrangian scheme studied in [12] under the assumptions (3) and (7) that we will keep here. It was proved in [3] that the numerical scheme stems from a discrete version of the dynamic programming principle applied to (6); this leads to the equation

$$\begin{cases} w(x) = \min\limits_{a\in B(0,1)} \{\beta w(x - hc(x)a)\} + 1 - \beta & \text{for } x \in \mathbb{R}^n\setminus\Omega_0, \\ w(x) = 0 & \text{for } x \in \partial\Omega_0, \end{cases} \tag{17}$$

where $\beta = e^{-h}$, h is the time step for the (hidden) dynamics, and w is an approximation of v. We will consider for simplicity a structured grid G, denoting its nodes by x_i, $i = 1, \ldots, N$, i.e., $G = \{x_i,\ i = 1, \ldots, N\}$. Note that the same scheme can be implemented on an unstructured grid as in [31]. We write (17) at every node, obtaining

$$\begin{cases} w(x_i) = \min\limits_{a\in B(0,1)} \{\beta w(x_i - hc(x_i)a)\} + 1 - \beta & \text{for } x_i \in G\setminus\Omega_0, \\ w(x_i) = 0 & \text{for } x_i \in G \cap \Omega_0, \end{cases} \tag{18}$$

where we defined $w = 0$ also in the internal nodes of Γ_0. It has been shown in [12] that under our assumptions (3) and (7), equation (18) has a unique solution w in the class of piecewise linear functions ($\mathbb{P}_1$ in the finite element notation) defined on the grid. Let us note that by applying the Kružkov transform (4) to the equation, one can also treat the case when $c = 0$ since in that case the minimum time function to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the target (i.e., the initial configuration of the front in the front propagation problem) will have infinite value at some points, whereas v will always stay bounded by 1. This allows us to run the computations also for $c = 0$ and to treat problems with state constraints (as we will see in the last section).

We will always approximate the v variable and use the fact that the Kružkov transform is monotone. In fact, since $T_1 > T_2$ if and only if $v_1 > v_2$ we can work on the v variable without changing the rules for the update of the narrow band because the crucial point is to label as *accepted* the node in the narrow band, where T (or v) attains its minimum. The above rule guarantees that we will process the nodes in an ordering which corresponds to increasing values of v.

The idea which is behind the FM-SL method is rather simple: we follow the initialization and all the steps of the classical FM-FD method except the step where the value at the node x_i is actually computed. That step would require us to iterate until convergence the scheme

$$w(x_i) = \min_{a \in B(0,1)} \{\beta w(x_i - hc(x_i)a)\} + 1 - \beta, \tag{19}$$

so that the typical fixed point iteration is applied "locally" at every single node following the order indicated by the FM-FD method. We will prove that for a semi-Lagrangian scheme based on a piecewise linear space reconstruction, just a single iteration is needed to compute the exact (within the accuracy of the scheme) value at every node so that the computational effort is very limited and of the same order as the FM-FD method.

3.1. Fast minimum search in $B(0,1)$. We will start improving the minimum search which is typical of the semi-Lagrangian schemes. The search for a minimum in the unit ball $B(0,1)$ will be solved algebraically for a linear interpolation, which allows us to compute the values $w(x_i - hc(x_i)a)$ using the known values at the nodes. Clearly, a new algebraic solution must be obtained (if possible) for other high order interpolations. Let us just recall that for the standard semi-Lagrangian scheme the search for the minimum is usually restricted to a discretization of the unit ball $B(0,1)$ which takes into account r points (or *controls* in the minimum time terminology) $a_1, a_2, \ldots, a_k, \ldots, a_r \in B(0,1)$.

For example, one can construct a uniform grid on $\partial B(0,1)$ with step $\Delta\theta = 2\pi/r$. To find the minimum, for every a_k the value $w(x_i - hc(x_i)a_k)$ is actually computed by interpolation. Although the choice of the type and order of the interpolation is completely free, the most popular choices are *linear*, using the three values at the nodes which are closer to $x_i - hc(x_i)a_k$, and *bilinear*, using the four values of w at the vertices of the cell containing $x_i - hc(x_i)a_k$.

Once all the values for a_k, $k = 1, \ldots, r$, are computed the minimum is obtained by comparison. It is worth noting that this algorithm is quite slow and requires a high computational cost; however, it can be applied to every high order interpolation. Moreover, it should be noted that this minimization problem is quite difficult since we expect to have nondifferentiable or even discontinuous solutions (if state constraints/obstacles are present in the domain) and that the comparison algorithm is very simple to implement and reasonably fast in low dimension especially when the search for the minimum can be restricted to the boundary of $B(0,1)$ (as will be the case in many examples). However, other algorithms for the minimization of nonsmooth functions can be applied, and the interested reader can find in [6] and [14] recent improvements on the solution of this problem. These algorithms converge to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

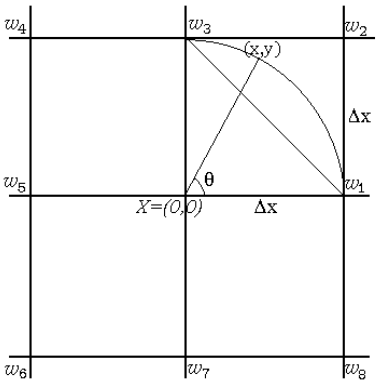

FIG. 2. *Search for optimal control.*

the minimum in the limit, so they cannot be applied here since we want to have an exact evaluation of the computational cost.

It is important to note that the time step h in (19) can vary at every node. We will denote by $h_i = h(x_i)$ the time step corresponding to the node x_i, by $c_i = c(x_i)$ the velocity at x_i, and by $\beta_i = e^{-h_i}$. When $c_i > 0$ it is always possible to choose

$$h_i = \frac{\Delta x}{c_i}. \tag{20}$$

In this way (19) can be written as

$$w(x_i) = \min_{a \in B(0,1)} \{\beta_i w(x_i - \Delta x\, a)\} + 1 - \beta_i. \tag{21}$$

In this situation, the nodes where $c_i = 0$ are actually treated apart from the other nodes: we just assign them the value $w = 1$ (which corresponds to $T = +\infty$) without any additional computation.

The method we propose here for the minimization problem has a low dimensional cost since for linear interpolation the search is restricted to the boundary of the unit ball. This is not a real restriction since, for our applications, the minimum in the unit ball is attained at the boundary. Later in this section we will show how this algorithm can be applied as a building block of our FM-SL scheme.

For simplicity, let us examine the situation in $\mathbb{R}^2$ considering a set of four cells each of side length Δx centered at the origin (see Figure 2). We want to compute the minimum of the function $w((0,0) - \Delta x\ a)$ for $a = (\cos\theta, \sin\theta)$ and $\theta \in [0, 2\pi)$. Let us introduce a vector $\mathbf{m} = (m_1, m_2, \ldots, m_8)$; the values of its components will be defined below. The minimum value for which we search will be given by $p = \min\{m_1, m_2, \ldots, m_8\}$.

Let us define the first four components of $\mathbf{m}$,

$$m_1 = w(\Delta x, 0)\,, \quad m_2 = w(0, \Delta x)\,, \quad m_3 = w(-\Delta x, 0)\,, \quad m_4 = w(0, -\Delta x),$$

and let us search for the minimum in every orthant.

Orthant I. Let w_1, w_2, and w_3 be the values of w corresponding, respectively, to the nodes $(\Delta x, 0)$, $(\Delta x, \Delta x)$, and $(0, \Delta x)$. The unique linear function $f(x, y)$ satisfying the conditions

$$f(\Delta x, 0) = w_1\,, \quad f(\Delta x, \Delta x) = w_2\,, \quad f(0, \Delta x) = w_3$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

is

$$f(x,y) = ax + by + c, \tag{22}$$

where

$$a = \left(\frac{w_2 - w_3}{\Delta x}\right), \qquad b = \left(\frac{w_2 - w_1}{\Delta x}\right), \qquad c = w_1 - w_2 + w_3.$$

Let us define the real function

$$F(\theta) = f(\Delta x \cos\theta, \Delta x \sin\theta) = a\Delta x \cos\theta + b\Delta x \sin\theta + c\,, \quad \theta \in [0, 2\pi) \tag{23}$$

and look for the minimum of $F(\theta)$ in the interval $(0, \pi/2)$. Note that the extreme values $\theta = 0$ and $\theta = \pi/2$ are not included since the values at the extrema of that interval have already been included in $\mathbf{m}$ (they are m_1 and m_2). By differentiating with respect to θ we obtain

$$F'(\theta) = 0 \Leftrightarrow \theta = \arctan(b/a).$$

The interesting case is when $w_2 < w_1$ and $w_2 < w_3$; otherwise the minimum is w_1 or w_3.

In this case, we get

$$a \neq 0\,, \quad b \neq 0\,, \quad b/a > 0\,, \quad \arctan(b/a) \in (0, \pi/2),$$

which means that the relative minimum is at $\theta_1^* = \arctan(b/a)$ and we set $m_5 = F(\theta_1^*)$.

If $w_2 \geq w_1$ or $w_2 \geq w_3$, we set $m_5 = +\infty$ (or the highest machine number).

Orthant II. Let w_3, w_4, and w_5 be the values of w, respectively, at the nodes $(0, \Delta x)$, $(-\Delta x, \Delta x)$, and $(-\Delta x, 0)$. The unique linear function $f(x,y)$ such that

$$f(0, \Delta x) = w_3\,, \quad f(-\Delta x, \Delta x) = w_4\,, \quad f(-\Delta x, 0) = w_5$$

is

$$f(x,y) = ax + by + c,$$

where

$$a = \left(\frac{w_3 - w_4}{\Delta x}\right), \qquad b = \left(\frac{w_4 - w_5}{\Delta x}\right), \qquad c = w_3 - w_4 + w_5.$$

Again we will consider the composite function $F(\theta)$ defined in (23), and we observe that it has a relative minimum in $(\pi/2, \pi)$ if and only if $w_4 < w_3$ and $w_4 < w_5$. In this case we have

$$a \neq 0\,, \quad b \neq 0\,, \quad b/a < 0\,, \quad \arctan(b/a) \in (-\pi/2, 0). \tag{24}$$

Since we are in the second orthant the value of θ where the minimum for F is attained is $\theta_2^* = \arctan(b/a) + \pi$. Proceeding as in the first orthant we set $m_6 = F(\theta_2^*)$.

If $w_4 \geq w_3$ or $w_4 \geq w_5$, we set $m_6 = +\infty$.

The analysis of the third and fourth orthants follows in the same way and will be skipped.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Once all the components of $\mathbf{m}$ have been set, we just compute $p = \min\{m_1, m_2, \ldots, m_8\}$ and substitute it in the expression

$$w(0,0) = \beta p + 1 - \beta. \tag{25}$$

This is done at every fixed point iteration until convergence. It is important to note that the above linear interpolation has a great advantage: the computation of the correct value of $w(0,0)$ does not require more than one iteration given the values at the neighboring nodes (along the axis directions and the diagonals) since $F(\theta)$ will not depend on $w(0,0)$. This property will *not* hold for other high-order interpolations, e.g., quadratic interpolation. Another advantage of linear interpolation with respect to the comparison of the values in a discrete unit ball is that it gives the exact value of the optimal direction at the cost corresponding to a discretization of $B(0,1)$ by just 8 directions.

3.2. The FM-SL scheme. This section is devoted to the presentation of the fast marching version of the SL-algorithm. For simplicity the presentation is given in $\mathbb{R}^2$, but the algorithm can be easily extended to $\mathbb{R}^n$. Let us start introducing the following definitions.

DEFINITION 3.1 (neighboring nodes for the SL scheme). *Let $X = (x_i, y_j)$ be a node of the grid. We define*

$$N_{FD}(X) = \Big\{(x_i, y_{j+1}), (x_i, y_{j-1}), (x_{i-1}, y_j), (x_{i+1}, y_j)\Big\},$$

$$D(X) = \Big\{(x_{i+1}, y_{j+1}), (x_{i+1}, y_{j-1}), (x_{i-1}, y_{j+1}), (x_{i-1}, y_{j-1})\Big\},$$

$$N_{SL}(X) = N_{FD}(X) \cup D(X).$$

The above definition is a natural extension of Definition 2.1 for the semi-Lagrangian scheme. According to the new definition, the nodes in the narrow band will also include the diagonal directions and not only the four directions N, S, E, W, as in the FM-FD method of section 2.

Sketch of the FM-SL algorithm.

Initialization (see Figure 3).

1. The nodes belonging to the initial front Γ_0 are located and labeled as *accepted*. Their value is set to $w = 0$. We will denote this set of nodes by $\widetilde{\Gamma}_0$.
2. The initial narrow band is defined according to the Definition 3.1, taking the nodes belonging to $N_{SL}(\widetilde{\Gamma}_0)$ external to Γ_0. These nodes are labeled as *narrow band*. Their value is set to $w = 1 - e^{-\frac{\Delta x}{c}}$ (which corresponds to $T = \Delta x/c$) if they belong to $N_{FD}(\widetilde{\Gamma}_0)$, or to $w = 1 - e^{-\frac{\sqrt{2}\Delta x}{c}}$ (which corresponds to $T = \sqrt{2}\Delta x/c$) if they belong to $D(\widetilde{\Gamma}_0)$.
3. We label as *far* all the remaining nodes of the grid; their value is set to $w = 1$ (which corresponds to the value $T = +\infty$).

Main cycle.

1. Among all the nodes in the narrow band we search for the minimum value of w. Let us denote this node by A.
2. The node A is labeled as *accepted* and is removed from the narrow band.
3. We label as *active* the nodes in $N_{SL}(A)$ which are not accepted. If there are far nodes, they are moved into the narrow band.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

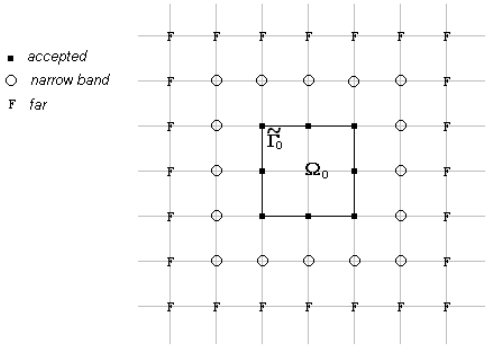

FIG. 3. *Initialization for FM-SL method, case* $c > 0$.

4. We compute (or recompute) the value w at the nodes belonging to $N_{FD}(A)$ which are active, iterating the fixed point operator

$$w(x_i) = \min_{a \in B(0,1)} \{\beta_i w(x_i - h_i c_i a)\} + 1 - \beta_i \,, \tag{26}$$

where $h_i c_i = \Delta x$. Note that just one iteration is needed, as we will see in the following sections. Then we compute by the same formula the value at the remaining active nodes in $N_{SL}(A) \setminus N_{FD}(A)$.
5. If the narrow band is empty, the algorithm stops; otherwise it goes back to step 1.

Although the algorithm advances the narrow band also in the diagonal directions, according to the new definition, it computes first the values at the neighboring nodes in the directions N, S, E, W (i.e., the finite difference directions) and then passes to the diagonal directions.

Some extensions: Obstacles, infinite velocity. We have seen that one can use our algorithm to deal with a front propagation with obstacles, i.e., regions where c vanishes. In [36, 18] the problem has been analyzed and several tests have been presented for a semi-Lagrangian method based on the linear interpolation, which treats the obstacle in a very simple way. The algorithm just assigns to the nodes belonging to an obstacle the value $w = 1$ in order to impose (indirectly and easily) a state constraints boundary conditions. In order to use the fast marching technique we just have to be careful and distinguish between nodes initialized to the value $w = 1$ because they are far and the ones to which was assigned the value $w = 1$ because they belong to an obstacle. In section 7 (Test 5) we will show a front propagating in the presence of obstacles.

Another interesting extension for applications to image processing is when the domain of computation contains points with infinite velocity. This is the case, for example, in the shape-from-shading problem when we have a point of maximal light intensity in the image (see, e.g., [29, 24]). Let us illustrate the idea which is behind our solution. Let x_{i_0} be a node such that

$$\lim_{x \to x_{i_0}} c(x) = +\infty.$$

Our equation $c(x)|\nabla T(x)| = 1$ can be written as

$$|\nabla T(x)| = g(x), \tag{27}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where $g(x) = 1/c(x)$. Clearly, (27) is a degenerate eikonal equation since g vanishes at x_{i_0}.

In order to compute $w(x_{i_0})$, we can set, according to (20), $h_{i_0} = 0$ and $\beta_{i_0} = 1$ and proceed as before, setting in (26)

$$h_{i_0} c_{i_0} = \Delta x. \tag{28}$$

Let us extend the function $h(x)$ outside the nodes in the domain $Q\backslash\Omega_0$. Our choice (28) can be justified by the fact that we would expect in our algorithm

$$\lim_{x\to x_{i_0}} c(x) = +\infty\,, \quad \lim_{x\to x_{i_0}} h(x) = 0, \quad \text{and} \quad \lim_{x\to x_{i_0}} c(x)h(x) = \Delta x.$$

Note that even if this argument is heuristic, it assigns to the node x_{i_0} the exact value for w. In fact, by (26), we get

$$w(x_{i_0}) = \min_{a\in B(0,1)} \{1w(x_{i_0} - \Delta x\ a)\} + 1 - 1 = w(x_{i_0} - \Delta x a^*),$$

where a^* is the *optimal control*. Since the front has an infinite velocity at x_{i_0} the minimum time of arrival on it coincides with the minimum time of arrival on the circle of radius Δx centered at x_{i_0}. In section 7 (Tests 6 and 7) we will show an application to a front propagation problem and to the shape-from-shading problem. It is interesting to note that theoretical results on discontinuous Hamiltonians can be found in [35] and [5].

4. Properties of the FM-SL scheme. We start with the following easy result on the semi-Lagrangian discretization.

PROPOSITION 4.1. *Let X be a node and assume that $w(X)$, defined by* (26), *is computed by interpolation using the three values $w^{(1)}$, $w^{(2)}$, $w^{(3)}$. Then*

$$w(X) \geq \min\left\{w^{(1)}, w^{(2)}, w^{(3)}\right\}. \tag{29}$$

Proof. Let $\beta = e^{-h}$, $h > 0$, and a^* be the optimal direction/control at X. The inequality

$$\beta w(X - h_i c_i a^*) + 1 - \beta \geq w(X - h_i c_i a^*)$$

is satisfied if and only if $w(X - h_i c_i a^*) \leq 1$. Since w is always less than or equal to 1 (due to the Kružkov transform) we have proved that

$$w(X) \geq w(X - h_i c_i a^*). \tag{30}$$

Since a simple property of linear interpolation guarantees that

$$\max\left\{w^{(1)}, w^{(2)}, w^{(3)}\right\} \geq w(X - h_i c_i a^*) \geq \min\left\{w^{(1)}, w^{(2)}, w^{(3)}\right\} \tag{31}$$

by (30) and (31) we end the proof. ∎

In order to prove that the fast marching version of our semi-Lagrangian scheme converges to the viscosity solution in a finite number of steps we have to prove first that the fast method for the minimum analyzed in section 3.1 matches the fast marching technique. This is necessary since the narrow band of the FM-SL method is larger than the narrow band of the FM-FD method as a consequence of the new definition of neighboring nodes. In particular we will show that the algorithm automatically

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

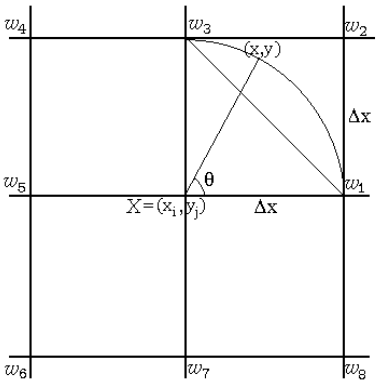

FIG. 4. *Analysis of the minimum in orthant I.*

rejects far nodes from the computation as in the standard up-wind finite difference discretization.

Let X be the node where we want to compute $w(X)$. Without loss of generality, we will assume that the optimal value is attained at a direction $\theta^* \in [0, \pi/2]$, i.e.,

$$a^* = (\cos\theta,\ \sin\theta)\,, \quad \theta \in [0, \pi/2]. \tag{32}$$

We will examine in detail all the possible configurations for this situation, which will be referred to in the following as the "minimum in orthant I" case (see Figure 4). For simplicity, let us assume $c > 0$ so that a node is labeled as *far* if and only if its value is $w = 1$.

PROPOSITION 4.2. *Let X be a node and let $w(X)$ be defined by* (26). *The value $w(X)$ will* not *be computed by interpolation using nodes labeled as* far.

Proof. Let us give the proof for the minimum in orthant I. The analysis for the other orthants is similar and can be easily obtained by symmetry arguments.

1. $w_1 = w_2 = w_3 = 1$: This configuration cannot occur. In fact, since the minimum is attained in orthant I we should have $w_4 = w_5 = w_6 = w_7 = w_8 = 1$. But this is not possible since we compute at X only when at least one of the nodes belonging to $N_{SL}(X)$ has been labeled as *accepted* in one of the previous iterations, and an accepted node must have a value lower than 1.
2. Among w_1, w_2, and w_3 there are two values equal to 1.
   (a) $w_1 = w_3 = 1$: this case cannot occur. In fact, since the minimum is attained in orthant I we must have $w_2 \leq w_1, w_3, w_4, \ldots, w_8$. The node that must be labeled as *accepted* is the one corresponding to the value w_2. This implies that the values w_1 and w_3 must be computed before X (see the sketch of the algorithm).
   (b) $w_1 = w_2 = 1$: the minimum value is w_3. A new iteration to compute $w(X)$ would not give a lower value, so the optimal value is obtained in just one iteration.
   (c) $w_2 = w_3 = 1$: the minimum value is w_1. Again, we will not get a lower value iterating, and the optimal value is obtained in just one iteration.
3. Among w_1, w_2, and w_3 only one value is equal to 1.
   (a) $w_2 = 1$: since f is linear the minimum will be attained by w_1 or w_3. The optimal value is obtained in just one iteration.
   (b) $w_1 = 1$, $w_3 \leq w_2$: the minimum is w_3.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

(c) $w_1 = 1$, $w_3 > w_2$: this is the most delicate case since $w_2 < w_1, w_3$. The minimum for F will be attained at some $\theta^* \in (0, \pi/2)$. The value $w(X)$, obtained by linear interpolation, will not be correct since it depends on $w_1 = 1$, which is a conventional value. Moreover, note that a new iteration of the fixed point map at X will not make $w(X)$ decrease since w_1 is frozen and so is $w(X)$. If this case could occur, we would not get convergence to the correct value even in the limit on the number of iterations. Note that this difficulty can occur neither for the global semi-Lagrangian scheme where *all* the nodes are computed at the same iteration nor for the FM-FD method where the values corresponding to far nodes are not used in the stencil. The following argument shows that this case also cannot occur for the FM-SL scheme. Since $w_1 = 1$, the corresponding node is labeled as *far* at the current iteration. This implies that the nodes labeled as *accepted* at the previous iteration do not belong to $N_{SL}(w_1)$. As a consequence, w_2 belongs to the narrow band. By Proposition 4.1 we have $w(X) > w_2$. This implies that X cannot be labeled as *accepted* before the nodes corresponding to w_2. Once w_2 becomes accepted the algorithm computes w_1 and w_3 before computing $w(X)$ so that the values at nodes labeled as *far* will not contribute.

(d) $w_3 = 1$, $w_1 \leq w_2$: the minimum is w_1. The optimal value is obtained in just one iteration.

(e) $w_3 = 1$, $w_1 > w_2$: analogous to case (3c). ∎

5. Convergence of the FM-SL scheme in a finite number of steps. As for the FM-FD method we have to prove that the minimal value of the nodes of the narrow band cannot decrease if we iterate the fixed point operator; i.e., it coincides with the value obtained by the discrete operator working on all the nodes. As we have seen, the values at the nodes belonging to the narrow band are not accepted all together. Only the minimal value is accepted at every iteration (this is a very pessimistic choice which simplifies the theoretical result). The following proposition shows the bounds on the number of times that one node can be recomputed, and it is a building block for the convergence of the scheme.

PROPOSITION 5.1. *Let X be a node in the narrow band such that $w(X) = w_{old}(X)$. Let us assume that at the current iteration the algorithm needs to compute a new value $w_{new}(X)$ for X. Moreover, let us assume that at the current iteration the following property holds true:*

(33) *If A belongs to the narrow band and B is* accepted, *then $w(A) \geq w(B)$.*

The following properties hold:

1. *If the value $w_{old}(X)$ was computed at an iteration in which a grid point $A_1 \in N_{FD}(X)$ was labeled as* accepted, *then it is impossible that $w_{new}(X) < w_{old}(X)$.*
2. *If the value $w_{old}(X)$ was computed at an iteration in which a grid point $A_2 \in D(X)$ was labeled as* accepted, *then to the node X a new value $w_{new}(X) < w_{old}(X)$ can be assigned but it will always satisfy the inequality $w_{new}(X) \geq w(A_2)$.*

Proof. Let us start with the first statement.

1. Let us assume that when the value w_{old} was assigned to X the node A_1 was the (unique) node belonging to $N_{FD}(X)$, which had been labeled as *accepted*.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

When the algorithm computed $w(X) = w_{old}(X)$ we certainly had

$$\min_{a \in \partial B(0,1)} w(X - \Delta x\ a) = w^* \leq w(A_1)$$

since there is a direction/control $\bar{a} \in \{(1,0),(0,1),(-1,0),(0,-1)\}$ such that $w(X - \Delta x\ \bar{a}) = w(A_1)$. The only possibility of having at X a value lower than $w_{old}(X)$ in the following iterations of the algorithm is that a value assigned to a node belonging to $N_{SL}(X)$ was lower than w^*. However, by Proposition 4.1 we know that this value cannot be computed using in the stencil the values at the nodes of the actual narrow band because they are all greater than $w(A_1) \geq w^*$, which has been accepted (as (33) assures). A lower value could be computed only using a stencil which contains nodes already accepted in one of the previous iterations since they all have values lower than $w(A_1)$. This is not possible since all the nodes which are neighbors of those accepted nodes have been computed already and they have a value greater than or equal to $w(A_1)$ since they have not been labeled as *accepted*.

2. Let us assume, for simplicity, that the node A_2 is the unique node belonging to $D(X)$ which has been labeled as *accepted* and let $w_{old}(X)$ be the value assigned at X at the same iteration. When a node $A_1 \in N_{FD}(X)$ has been labeled as *accepted* before A_2, the result holds true by the arguments of the above case 1.
   Let us assume that A_2 is the unique neighbor of X which has been labeled as *accepted*. Then we have

$$\min_{a \in \partial B(0,1)} w(X - \Delta x\ a) = w^* \geq w(A_2).$$

   It is always possible that using $w(A_2)$ one can obtain a new value $w_{new}(X)$ lower than $w_{old}(X)$. However, by (33) and Proposition 4.1 all the new values will be greater than or equal to $w(A_2)$; therefore $w_{new}(X) \geq w(A_2)$. □

Remark 5.1. Note that the previous proposition allows us to accelerate the algorithm. In fact, one can save CPU time by avoiding recomputing the values at the nodes corresponding to case 1. However, they cannot be labeled as *accepted* before their value is the minimum in the narrow band. An important consequence of Proposition 5.1 and the above observation is that every node can be computed at most 5 times; this is one of the reasons why the CPU time for FM-SL is slightly larger than that for the FM-FD method, where a node can be computed at most 4 times. We will see in the last section that the FM-SL method produces a more accurate approximation of the viscosity solution, which justifies a small increment in the CPU time.

The following result is an analogue of Proposition 2.1, and it is crucial to prove convergence in a finite number of steps.

PROPOSITION 5.2. *Let w be defined in* (26) *and let $w(X)$ be the value assigned at X at the same iteration when a node $Z \in N_{SL}(X)$ is labeled as* accepted. *Assume that*

$$c(x) \geq 0 \quad \textit{for any } \ x \in Q \backslash \Omega_0.$$

Then we have

$$w(X) \geq w(Z). \tag{34}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

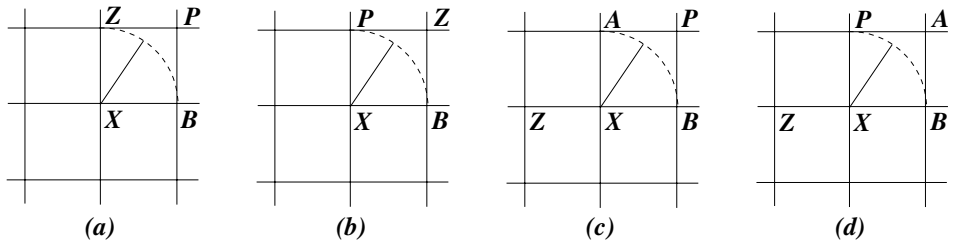

FIG. 5. *Four different configurations for Case* 2.

Proof. We examine all the cases corresponding to a minimum in orthant I (see Figure 4). The proof will be obtained by induction on the number of iterations of the algorithm.

At the first step the result holds true by our initialization.

Let us consider the nth step of the algorithm. The induction hypothesis implies that at the current iteration the values of nodes in the narrow band are greater than values of nodes labeled as accepted. Therefore (33) holds true, so we can apply Proposition 5.1. Our proof will be divided into three parts.

Case 1. $w_1, \ldots, w_8$ are narrow band or far (before Z is labeled as *accepted*).

If Z belongs to orthant I, we have seen by Proposition 4.1 that

$$w(X) \geq \min\{w_1, w_2, w_3\} = w(Z).$$

If Z does not belong to orthant I, we have

$$w(X) \geq \min\{w_1, w_2, w_3\} \geq w(Z)$$

since Z as been labeled as *accepted*.

Case 2. One node $w_1, \ldots, w_8$ is accepted (before Z is labeled as *accepted*).

Let us denote by P this node. When P was accepted the value at X was $w_{old}(X)$. Now the value at X has to be recomputed. We can only have one of the following situations:

1. P belongs to orthant I.
   (a) Z belongs to orthant I
      i. See Figure 5(a). By Proposition 5.1, Z and B cannot be assigned to a lower value after P became accepted, so $w_{new}(X) = w_{old}(X)$ and $w_{old}(X) \geq w(Z)$ since Z is the node chosen to be labeled as *accepted*.
      ii. See Figure 5(b). When Z is accepted the minimum is attained at P, and this implies again $w_{new}(X) = w_{old}(X)$.
   (b) Z does not belong to orthant I
      i. See Figure 5(c). In the iterations between the acceptance of P and that of Z the values $w(A)$ and $w(B)$ cannot be changed. Moreover, the minimum is attained in orthant I so we have $w_{new}(X) = w_{old}(X)$.
      ii. See Figure 5(d). We know that the value $w(A)$ has not been replaced, $w(B)$ cannot be lower than $w(P)$, and the minimum is attained in orthant I. Then the minimum is attained at P and $w_{new}(X) = w_{old}(X)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

2. P does not belong to orthant I.
   Since the minimum is attained in orthant I this means that P has no effect on the computation at X and we are back to Case 1.

Case 3. More than one value among $w_1, \dots, w_8$ has been labeled as *accepted* (before Z is labeled as *accepted*).

This case can be solved by the arguments in Case 2. □

As for the FM-FD method (see [33]) we can now conclude that the value of the node which is labeled as *accepted* at every iteration cannot be decreased if we iterate the fixed point operator. In fact, let us denote this value w_{min}. Since all the nodes in the narrow band have values greater than w_{min}, the previous result implies that using those nodes we cannot assign to a node a value lower than w_{min}. In conclusion, the up-winding is respected and the value w_{min} can be considered exact since it cannot be improved on the same grid (of course it can be improved if we reduce the discretization steps).

Remark 5.2. The FM-SL scheme does not require a stability CFL-like condition, as required by the FM-FD scheme.

5.1. Convergence to the viscosity solution and conclusions. The semi-Lagrangian scheme is consistent, as has been proved, e.g., in [16]. Moreover, choosing $\Delta x = \Delta y$, we get that the local truncation error is $O(\Delta x)$.

We will prove that the solution computed by the FM-SL method is identical to the solution computed by the standard semi-Lagrangian scheme where the computation is repeated on every node of the grid until convergence. Naturally, if the two schemes compute the same values, convergence of the FM-SL method to the viscosity solution is just a consequence of that of the standard semi-Lagrangian scheme.

THEOREM 1. *Let $(V_i)_{i=1,\dots,N}$ be the matrix containing the final values on the n-dimensional grid and let*

$$V_i = F(V_{i-k}, \dots, V_{i+l}) \tag{35}$$

be the iteration corresponding to the numerical scheme. Let $\widehat{V}$ be the matrix of the approximate solution corresponding to the fixed point iteration (35) *and let $\overline{V}$ be the matrix containing the final values of the approximate solution corresponding to the fast marching technique applied to the* same *scheme (i.e., the result obtained when the narrow band is empty). Then $\overline{V} = \widehat{V}$.*

Proof. The two matrices coincide if and only if

$$\overline{V}_i = F(\overline{V}_{i-k}, \dots, \overline{V}_{i+l}) \quad \text{for any} \quad i = 1, \dots, N. \tag{36}$$

Assume the narrow band is empty and take $\overline{V}$ as initial guess for the fixed point technique; this will not change the solution since the value is computed by the same scheme. When all the nodes are accepted the equality (36) must hold for every i. In fact, if the equality is not true at one node, then its value can still be improved, implying that the list of narrow band or far nodes is not empty, which gives us a contradiction. □

The above results allow us to draw some conclusions about the order of complexity of the FM-SL scheme. The values $w(X)$ computed by (26) are an approximation of $v(X)$, which has been computed at most 5 times for every nodes. This means that the computational cost can be estimated as in the FM-FD scheme. One component is given by the cost of the heap-sort method to select the minimum value in the narrow band, and the other component is given by the computational cost at every node.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

This globally gives a cost $O(N\log(N_{nb}))$, where N is the total number of nodes and N_{nb} the number of nodes in the narrow band (see [33]).

Since the values which have been labeled as *accepted* at every iteration cannot be improved by the global fixed point iteration, i.e., they coincide with the same values obtained by the global fixed point operator, the a priori error estimates in [12] are still valid for the solution obtained by the FM-SL method. In the last section we will present several tests which confirm these theoretical results.

Boundary conditions on ∂Q. We define outside Q a strip of ghost nodes where we set $w = 1$. If they enter the narrow band, at the end of the iteration, their value is set back to $w = 1$ to avoid their contributing to the computation of other internal nodes. When the minimal value on the nodes of the narrow band is 1, the ghost nodes will be the only nonaccepted nodes and we can stop the computation. In general, any constant larger than the maximum of the solution in Q can be used to assign the value at the ghost nodes (a typical choice is to set the solution to $+\infty$ if there is no a priori estimate on the solution).

Note that in our case, the normal velocity has always the same (positive) sign, so in the case of a constant velocity the front propagation starting from $\Gamma_0 \subset Q$ will hit the boundary of Q and both T and w are increasing approaching ∂Q. The values computed by the algorithm on the nodes of the boundary will always be lower than 1, and the choice of the above boundary condition is then well adapted to this situation. However, when c is variable or when there are obstacles in the domain we can also have a different situation: the front propagates more rapidly in some directions, and this could require enlarging the domain to get a correct solution to our problem. Finally, let us observe that the use of homogeneous Neumann boundary conditions is less appealing because it strongly affects the fronts near the boundary because all the level curves must be orthogonal to the boundary to satisfy $\nabla v(x)\cdot\eta(x) = 0$ for any $x \in \partial Q$ (here $\eta(x)$ denotes the exterior normal to Q).

6. Other fast schemes. As some authors have remarked, it is possible to improve the finite difference method. In the paper by Tsitsiklis [38] one can find an algorithm which can be parallelized directly with a complexity $O(N)$. There are at least two ways to accelerate convergence and/or reduce the CPU time:

1. Reduce the computational effort for the minimum search by accepting more than one node in the narrow band at every iteration (group marching method).
2. Avoid searching for the minimum value in the narrow band (fast sweeping method), obtaining convergence in more than one iteration.

We will briefly illustrate these two techniques.

Group marching. The group marching (GM) method has been introduced by Kim [22] to solve the eikonal equation on a structured grid by a discretization as that of FM-FD. Although we do not compare this algorithm with the others studied in the previous section, we will give a brief presentation of its main features for completeness. Let us denote by Γ the set of nodes belonging to the narrow band, and let us choose $\Delta x = \Delta y$. Define

$$T_{\Gamma,min} = \min\{T_{i,j} \mid (x_i, y_j) \in \Gamma\} \quad \text{and} \quad c_{\Gamma,\max} = \max\{c_{i,j} \mid (x_i, y_j) \in \Gamma\}.$$

The GM method labels as *accepted*, *all at once*, the nodes belonging to the set G defined by

$$G := \left\{ (x_i, y_j) \in \Gamma \ : \ T_{i,j} \le T_{\Gamma,min} + \frac{\Delta x}{\sqrt{2}}\ \frac{1}{c_{\Gamma,\max}} \right\}. \tag{37}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

At every iteration the update of the narrow band is obtained as in the FM-FD method, including the four neighbors of every node that have been labeled as *accepted*. It is clear that if the set G is large, the GM method can be much faster than the FM-FD method because more than one node at a time in accepted. On the other hand, it is rather difficult to give an estimate of the acceleration parameter since the cardinality of G depends on the values $\{T_{i,j} : (x_i, y_j) \in \Gamma\}$ and on the velocity of propagation. It could be that $G = \{T_{\Gamma,min}\}$, and this would imply a computational cost of $O(N \ln(N_{nb}))$ instead of getting $O(N)$, as one would expect by some tests in [22].

Fast sweeping. The fast sweeping (FS) method is based on an idea first introduced in [11] and was extensively analyzed in [39] and [37]. The crucial idea is that the algorithm sweeps the whole (two-dimensional) domain with four alternating orderings repeatedly,

$$(38) \qquad (1)\ i = 1, \dots, N\,,\ j = 1, \dots, M; \qquad (2)\ i = N, \dots, 1\,,\ j = 1, \dots, M;$$

$$(39) \qquad (3)\ i = N, \dots, 1\,,\ j = M, \dots, 1; \qquad (4)\ i = 1, \dots, N\,,\ j = M, \dots, 1$$

(where N and M are the number of nodes in each dimension), and it updates the value at a grid point only if the new value is smaller than the current one. This idea can be easily extended to n-dimensional domains.

Computing the values in this special ordering, the algorithm is able to follow simultaneously a family of characteristics in a certain direction. As proved in [39], the FS method converges in 2^n iterations, where n is the dimension of the problem if the initial front Γ_0 is just a point on the grid and the function c is constant. If those assumptions do not hold, the FS method has been shown to be of complexity $O(N)$ and to converge in a finite number of iterations although the bound for the number of iterations is not explicitly written out. See [27] for an extension on triangular meshes and an upper bound to the number of iterations needed by the FS method to reach convergence.

Let us note that the discretization used in [39] is the same as that used in the FM-FD method described in section 2, and that in any case the numerical evidence shows that the convergence is more rapid with respect to the classical iterative method.

The FS method has an easy extension to the semi-Lagrangian case. In fact, we can easily substitute the finite difference discretization by the semi-Lagrangian discretization maintaining the ordering in which nodes are visited. Obviously, we expect that at least in the case $c(x) \equiv$ const. the FS semi-Lagrangian scheme (FS-SL) can compute in four iterations exactly the same solution as FM-SL.

In the next section we run this algorithm in the case $c(x) \equiv 1$ with two different initial fronts and see that this intuition is actually true.

7. Numerical experiments. In this section we present some numerical experiments performed with MATLAB 7 on a PC equipped with a Pentium IV 2.80 GHz processor, 512 MB RAM.

The main goal is to compare the FM-FD method and the FM-SL method described in previous sections. We also compare these methods with the semi-Lagrangian iterative method and FS method based on a semi-Lagrangian discretization described in section 6. First, two tests are devoted to approximate the solution of model problems where we know the exact solution, so we can compute the L^∞ error and L^1 error. Other tests are devoted to solving more complicated problems and applications

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

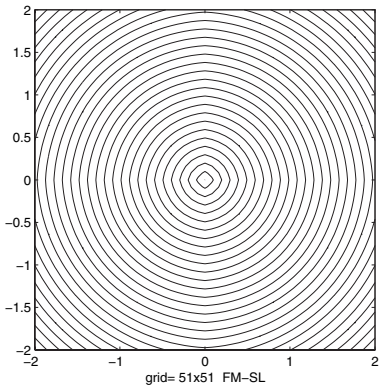

FIG. 6. *Level sets of $T(x)$ computed by the FM-SL method,* 51×51 *grid.*

in which the velocity function $c(x)$ does not satisfy standard assumptions such as Lipschitz continuity and boundedness.

If not specified otherwise, we choose $Q = [-2, 2]^2$ as our computational domain.

7.1. Tests on model problems. In the following tests we compare the exact solution T with the solution $\widehat{T}$ computed by the FM-FD method and the FM-SL method described above. Note that in the implementation of the FM-SL algorithm we have used the observation in Remark 5.1 to speed up the computation.

We compute

$$E_{\infty,\Delta x} = \max_{i,j} |T_{i,j} - \widehat{T}_{i,j}|\,, \qquad E_{1,\Delta x} = (\Delta x)^2 \sum_{i,j} |T_{i,j} - \widehat{T}_{i,j}| \tag{40}$$

and the rate of convergence r in some model problems in $\mathbb{R}^2$. We consider 51×51, 101×101, and 201×201 grids[1] corresponding, respectively, to $\Delta x = 0.08$, $\Delta x = 0.04$, and $\Delta x = 0.02$.

Since we know that there is a constant C such that

$$E_{p,\Delta x} \le C \Delta x^r \quad \text{and} \quad E_{p,\Delta x/2} \le C \left(\frac{\Delta x}{2}\right)^r, \qquad p = 1, \infty,$$

we obtain that the numerical rate of convergence is

$$r = \log_2 \left(\frac{E_{p,\Delta x}}{E_{p,\Delta x/2}}\right), \qquad p = 1, \infty.$$

Moreover, we compare these algorithms with the classical iterative semi-Lagrangian method in which we choose $\max_{i,j} |w_{i,j}^{(k)} - w_{i,j}^{(k-1)}| < \varepsilon$, $\varepsilon = 10^{-7}$, as the stopping criterion and with the FS-SL method performing just four iterations in different order.

Let us finally remark that in all cases condition (15) holds.

Test 1. $\Gamma_0 = (0, 0)$, $c(x, y) \equiv 1$. Exact solution: $T(x, y) = \sqrt{(x^2 + y^2)}$.

Results are summarized in Figure 6 and Table 1. As expected, in all cases errors reduce as Δx decreases. The numerical rate of convergence (Table 2) is in the interval $[0.5, 1]$ for both methods.

[1] In these grids there is a node corresponding to the point $(0, 0)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 1
Errors for Test 1.

Method	Δx	L^∞ error	L^1 error	CPU time (sec)
FM-FD	0.08	0.0875	0.7807	0.5
FM-SL	0.08	0.0329	0.3757	0.7
SL (46 it)	0.08	0.0329	0.3757	8.4
FS-SL	0.08	0.0329	0.3757	0.8
FM-FD	0.04	0.0526	0.4762	2.1
FM-SL	0.04	0.0204	0.2340	3.1
SL (86 it)	0.04	0.0204	0.2340	60
FS-SL	0.04	0.0204	0.2340	3.2
FM-FD	0.02	0.0309	0.2834	9.4
FM-SL	0.02	0.0122	0.1406	14
SL (162 it)	0.02	0.0122	0.1406	443.7
FS-SL	0.02	0.0122	0.1406	12.5

TABLE 2
Rate of convergence in L^∞ *and* L^1 *norms computed by errors in Table* 1.

Method	L^∞ $(0.08 \to 0.04)$	L^∞ $(0.04 \to 0.02)$	L^1 $(0.08 \to 0.04)$	L^1 $(0.04 \to 0.02)$
FM-FD	0.7342	0.7675	0.7132	0.7487
FM-SL	0.6895	0.7417	0.6831	0.7349

The FM-SL and semi-Lagrangian methods give exactly the same errors in accordance with Theorem 1, and they are also equal to the errors of FS-SL, as expected, since FS-SL converges in four iterations in the case c is constant. These errors number about half that of the FM-FD method, although both are first order methods. This is due to the fact that semi-Lagrangian discretization is able to follow every direction of the characteristic flow.

Both methods based on the fast marching technique are dramatically faster than the iterative semi-Lagrangian method. Nevertheless we want to note that only one iteration of the iterative scheme is less expensive with respect to the single iteration needed by fast marching–based algorithms. This is due to the fact that the narrow band technique requires that we (1) compute a minimum over nodes in the narrow band and (2) access the data in an almost random manner rather than in a systematic way along the loop indices (see [22]). Finally we note that the CPU time needed by the FM-SL method is slightly larger than the CPU time needed by the FM-FD method. This due to the fact that (1) the narrow band is bigger in the first method; and therefore the search for the minimum in the narrow band is more expensive; and (2) in the FM-SL method we need to compute the minimum over the unit ball $B(0,1)$.

Test 2. Γ_0 = unit square centered in $(-1,1)$ and rotated by $11.25^\circ \cup$ circle with radius $R = 0.5$ centered in $(0,-1) \cup$ square with side 0.4 centered in $(1.4, 1.4)$, $c(x,y) \equiv 1$. Exact solution: $T(x,y) =$ minimum between the distance function of the square rotated, the circle, and the square.

Results are summarized in Figure 7 and Table 3. In this test the shape of the initial front is much more complicated, but errors have the same behavior as in the previous simple Test 1, although the difference between errors is smaller.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

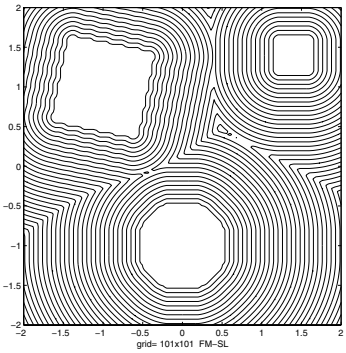

FIG. 7. *Level sets of $T(x)$ computed by the FM-SL method,* 101×101 *grid.*

TABLE 3
Errors for Test 2.

Method	Δx	L^∞ error	L^1 error	CPU time (sec)
FM-FD	0.08	0.0625	0.2154	0.5
FM-SL	0.08	0.0440	0.1849	0.7
SL (30 it)	0.08	0.0440	0.1849	4.9
FS-SL	0.08	0.0440	0.1849	0.7
FM-FD	0.04	0.0393	0.1120	2.2
FM-SL	0.04	0.0215	0.1044	3.1
SL (55 it)	0.04	0.0215	0.1044	34.1
FS-SL	0.04	0.0215	0.1044	2.9
FM-FD	0.02	0.0248	0.0669	10.2
FM-SL	0.02	0.0135	0.0633	14.5
SL (102 it)	0.02	0.0135	0.0633	246.6
FS-SL	0.02	0.0135	0.0633	11.4

TABLE 4
Rate of convergence in L^∞ and L^1 norms computed by errors in Table 3.

Method	L^∞ $(0.08 \to 0.04)$	L^∞ $(0.04 \to 0.02)$	L^1 $(0.08 \to 0.04)$	L^1 $(0.04 \to 0.02)$
FM-FD	0.6693	0.6642	0.9435	0.7434
FM-SL	1.0332	0.6714	0.8246	0.7218

FS-SL seems to be the best method. It has the smallest error and the CPU time is slightly larger than that of FM-FD. This is probably due to the fact that the structure of the narrow band is very complicated and is very large in terms of nodes.

Also in this case the rate of convergence (Table 4) is greater than 0.5.

7.2. Applications. In the following we try to use the FM-SL method in some classical applications of the eikonal equation such as the minimum time problem and shape-from-shading. We consider some cases not covered by the theory in which $c(x, y)$ is discontinuous, $c(x, y)$ vanishes in some regions (state constraints), and $c(x, y)$ has infinite values. We also consider the *anisotropic* case in which the velocity field c depends on (x, y) and on the control a. The results we obtained are very satisfactory even in these cases.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

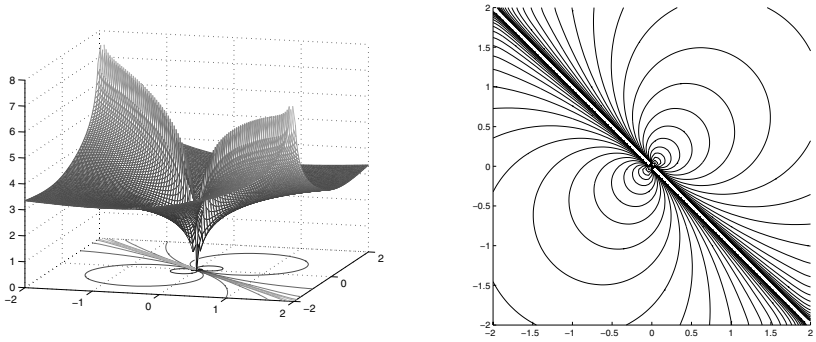

FIG. 8. *Value function T (left) and level sets of T (right).*

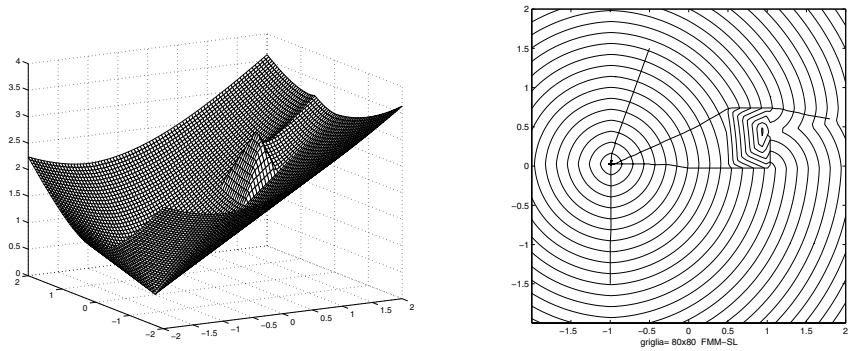

FIG. 9. *Value function T (left) and level sets of T with some optimal trajectories (right).*

Test 3: Nonconstant velocity. $\Gamma_0 = \partial B(0, \frac{\Delta x}{2})$, $c(x, y) = |x + y|$. In this case the velocity field is nonconstant. Figure 8 shows the value function $T(x, y)$ and level sets of T. On the line $x = -y$ the solution T is not defined since its correct value is $T = +\infty$. The FS-SL method needs 12 iterations to reach convergence and is more than three times slower than the FM-SL method on a 101×101 grid.

Test 4: Discontinuous vector field. $\Gamma_0 = (-1, 0)$.

$$c(x, y) = \begin{cases} 0.4, & (x, y) \in [0.5, 1] \times [0, 0.5], \\ 1 & \text{elsewhere.} \end{cases}$$

In this case the velocity field is discontinuous. Figure 9 shows the value function $T(x, y)$ and level sets of T. Figure 9 (right) also shows some optimal trajectories which start from four different points and reach the target Γ_0 in the minimum time with speed $c(x, y)$. The FS-SL method converges in 8 iterations.

Test 5: State constraint problem. $\Gamma_0 = (-1, -1)$.

$$c(x, y) = \begin{cases} 0, & (x, y) \in ([0, 0.5] \times [-2, 1.5]) \cup ([1, 1.5] \times [-1.5, 2]), \\ 1 & \text{elsewhere.} \end{cases}$$

In this test the velocity field vanishes in two different regions (the obstacles). Figure 10 shows the computational domain, the value function $T(x, y)$, and level sets of

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

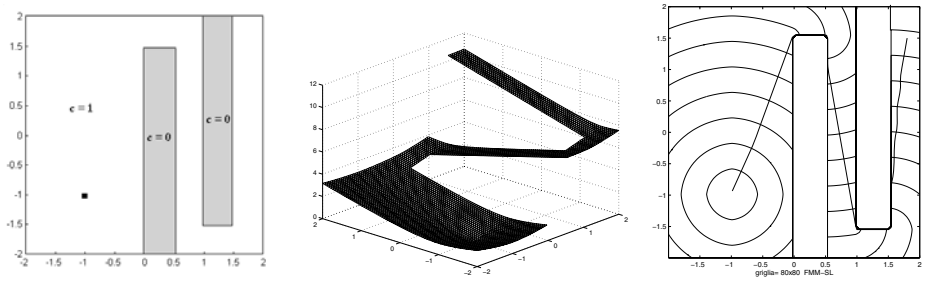

FIG. 10. *Domain of the equation (left), value function T (center), and level sets of T with one optimal trajectory (right).*

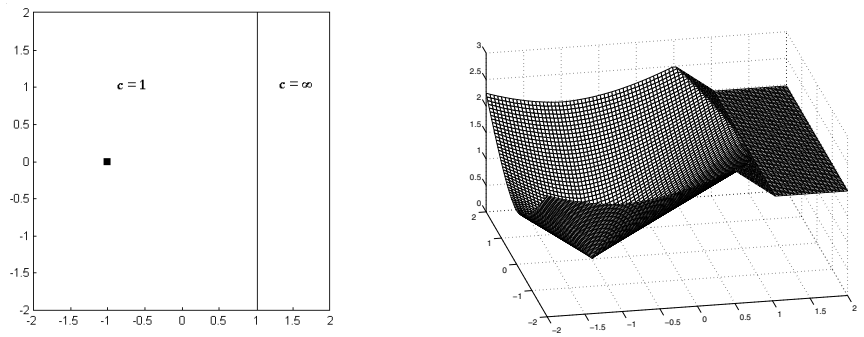

FIG. 11. *Domain of the equation (left) and value function T (right).*

T. Figure 10 (right) also shows one optimal trajectory which starts from the point $(1.8, 1.5)$ and reaches Γ_0 in the minimum time avoiding obstacles. We remark that since we use the Kružkov transform and compute v, we do not need to modify the numerical scheme to deal with state constraints. Also in this case the FS-SL method converges in 8 iterations.

Test 6: Infinite velocity. $\Gamma_0, = (-1, 0)$.

$$c(x,y) = \begin{cases} +\infty, & x \geq 1, \\ 1 & \text{elsewhere.} \end{cases}$$

In this case the front can propagate instantaneously in the region $R = \{x \geq 1\}$. It corresponds to the case of the following degenerate eikonal equation (see [30]):

$$|\nabla T(x,y)| = f(x,y) \quad \text{with } f = 0 \text{ in } R.$$

Figure 11 shows the computational domain and value function $T(x, y)$. In this test we used the technique described in section 3.2 in order to deal with this kind of vector field.

This technique allows us to reconstruct a perfect *flat* surface on R as the theory and the physical sense require. This technique can be very useful in shape-from-shading problems.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

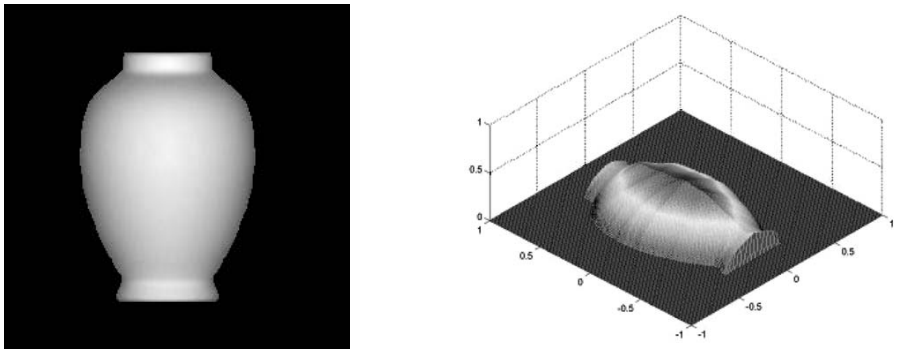

FIG. 12. *Initial image (left) and reconstructed surface (right).*

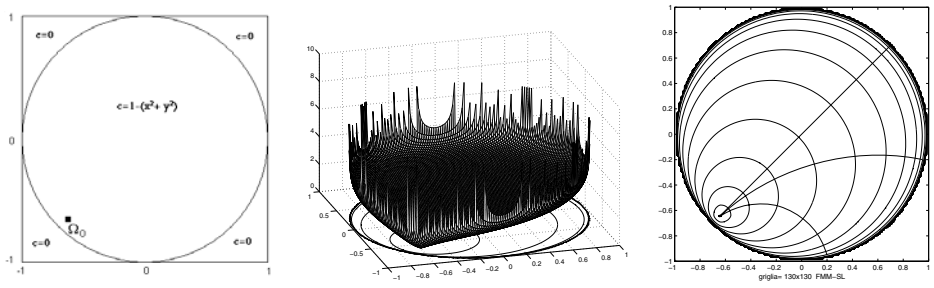

FIG. 13. *Domain of the equation (left), value function T (center), and level sets of T with some optimal trajectories (right).*

Test 7: Shape-from-shading. $Q = [-1,1]^2$, $\Gamma_0 =$ silhouette of a vase.

$$c(x,y) = \left(\sqrt{\frac{1}{I(x,y)^2} - 1}\right)^{-1}, \quad I(x,y) = \text{intensity light function.}$$

In this test we solve the shape-from-shading problem in the simple case of a vase. Figure 12 (left) shows the initial image and Figure 12 (right) shows the reconstructed surface. By the symmetry of the problem we guess that all characteristic curves start from the right and left sides of the image, so we can impose Dirichlet boundary condition just on the right and left sides of the domain and state constraints elsewhere as in [10] (see also [29, 24], where different boundary conditions are applied).

Test 8: Poincaré model. $Q = [-1,1]^2$, $\Gamma_0 = (-0.65, -0.65)$.

$$c(x,y) = \begin{cases} 1 - (x^2+y^2), & x^2+y^2 < 1, \\ 0 & \text{elsewhere.} \end{cases}$$

This example is an interesting application of the eikonal equation to the Poincaré model of the hyperbolic geometry. Figure 13 shows the computational domain, the value function $T(x,y)$, and level sets of T. The FS-SL method converges in 8 iterations.

As result of the particular choice of the velocity field (see [25]), the optimal trajectories of the associated minimum time problem correspond to the hyperbolic

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

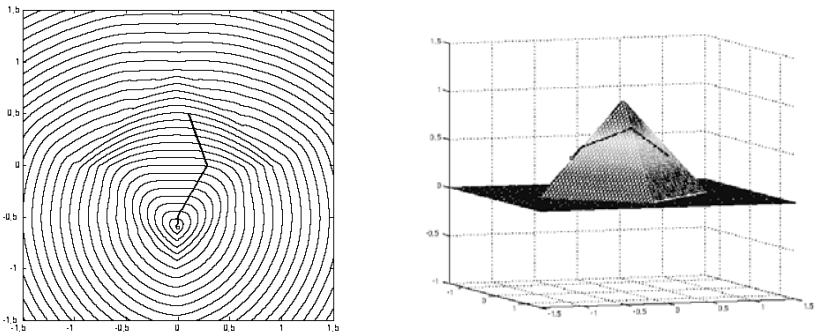

FIG. 14. *Level sets of T (left) and an optimal trajectory on the surface z (right).*

straight lines. Moreover, the level sets of T are hyperbolic circles with center Γ_0 (i.e., the sets of points which have the same hyperbolic distance from Γ_0).

Test 9: Geodesics on a nonsmooth surface. $Q = [-1.5, 1.5]^2, \Gamma_0 = (0, -0.6)$.

$$\text{Surface} : z(x,y) = \begin{cases} 1 - (|x| + |y|), & |x| + |y| < 1, \\ 0 & \text{elsewhere.} \end{cases}$$

In this case we want to solve a minimum time problem on a surface $z = z(x, y)$. The three-dimensional problem can be easily reduced to a two-dimensional problem modifying the velocity field according to the function z. In fact, if the intrinsic velocity on the surface in equal to 1, it can be shown (see [32, 23]) that the velocity of the corresponding two-dimensional problem becomes

$$c(x, y, a) = \frac{1}{\sqrt{1 + (\nabla z \cdot a)^2}}.$$

Figure 14 shows the level sets of T and the surface with an optimal trajectory on it. The starting point is $(0, 0.5)$.

We remark that the dependence of c on a changes the properties of the solution of the equation. In fact the equation for anisotropic front propagation is

$$\begin{cases} v(x) + \max\limits_{a \in B(0,1)} \{c(x,a)a \cdot \nabla v(x)\} = 1, & x \in \mathbb{R}^n \backslash \Omega_0, \\ v(x) = 0, & x \in \partial\Omega_0. \end{cases} \tag{41}$$

In this case the fast marching technique is no longer directly applicable (there is no guarantee that convergence is reached in just one iteration; see [34]). This is true for the FM-SL method too, but we stress that scheme (26) requires tiny modifications to deal with this kind of velocity field. Moreover, if we use the function w computed by the FM-SL method as a starting point of the iterative semi-Lagrangian scheme, we can reach convergence in very few iterations.

Appendix. Convergence of the FM-FD method in a finite number of steps.

Proof of Proposition 2.1. We will assume that B, C, and D are the neighbors of X which can have a label *accepted*, *narrow band,* or *far* (see Figure 15). We will

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

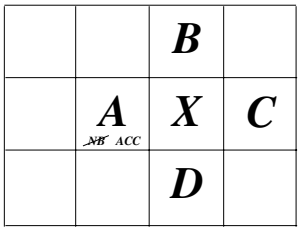

FIG. 15. *The neighboring nodes of* X.

prove the result by induction on the number of iterations of the algorithm. We will always assume

$$T(B) \leq T(D), \tag{42}$$

which is not restrictive since we can always switch the B and D.

In the first iteration we simply have $T(X) = 0 + f_X$ and (16) is satisfied. Let us consider the nth step of the algorithm. The induction hypothesis implies that at each iteration the values of nodes in the narrow band are greater than values of nodes labeled as *accepted* at the same iteration. Therefore, by construction we have that, given two nodes Y and Z,

$$\text{if } Y \text{ has become accepted before } Z, \text{ then } T(Y) \leq T(Z).$$

The proof will be divided into four cases.

Case 1. B is far. C and D are narrow band or far.

By assumption $T(B) = +\infty$, and since $T(B) \leq T(D)$ this implies that D must be far. Moreover, we have

$$T(C) \geq T(A)$$

since A has been chosen among all the nodes of the narrow band to become accepted. Also X must be far, since it has never been computed. Then by (10) we get

$$T(X) = T(A) + f_X. \tag{43}$$

Since $f_X > 0$, (43) implies

$$T(A) \leq T(X) \leq T(A) + f_X.$$

Case 2. B is narrow band. C and D are narrow band or far.

Also in this case X is far. We have

$$T(A) \leq T(B)\,, \qquad T(A) \leq T(C)$$

since A is the minimal node in narrow band. Moreover, the assumption (42) implies that $T(X)$ will be computed by the values at A and B. From (10) we get

$$T(X) = \frac{T(A) + T(B) + \sqrt{2f_X^2 - (T(A) - T(B))^2}}{2} \tag{44}$$

and then

$$T(X) \geq \frac{T(A) + T(B)}{2} \geq \frac{T(A) + T(A)}{2} = T(A). \tag{45}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Since $T(X)$ solves

$$(T(X) - T(A))^2 + (T(X) - T(B))^2 = f_X^2$$

we have

$$(T(X) - T(A))^2 \leq f_X^2.$$

Since all the terms in the above equation are positive we conclude that

$$T(X) - T(A) \leq f_X. \tag{46}$$

Case 3. B is accepted. C and D are narrow band or far.

This situation occurs when X has been already computed once (when B has been labeled as *accepted*). Let us denote its value by $T_{old}(X)$. The node X is then in the narrow band and has to be recomputed because A has just been labeled as *accepted*. Let us note that in the previous computation $T_{old}(X)$ has been computed according to the rules examined in Case 1 or 2. Then we have

$$T(B) \leq T_{old}(X) \leq T(B) + f_X.$$

Moreover $T(A) \leq T_{old}(X)$ because A just became accepted and $T(B) \leq T(A)$ since B became accepted before A (induction).

These inequalities imply

$$T(B) \leq T(A) \leq T_{old}(X) \leq T(B) + f_X \tag{47}$$

and

$$0 \leq T(A) - T(B) \leq f_X. \tag{48}$$

The value at X, which will be denoted by $T_{new}(X)$, will depend on $T(A)$ and $T(B)$.

By (48) and (47) we derive

$$\begin{aligned} T_{new}(X) &= \frac{T(A) + T(B) + \sqrt{2f_X^2 - (T(A) - T(B))^2}}{2} \\ &\geq \frac{T(A) + (T(B) + f_X)}{2} \geq \frac{T(A) + T(A)}{2} = T(A) \end{aligned} \tag{49}$$

and

$$T_{new}(X) \leq \frac{T(A) + T(B) + \sqrt{2}f_X}{2} \leq T(A) + \frac{\sqrt{2}}{2}f_X \leq T(A) + f_X.$$

Case 4. B is narrow band or far. C is accepted. D is narrow band or far.

In this case X has already been computed because it is a neighbor of C. It belongs to the narrow band and has a value $T_{old}(X)$. Besides

$$T(A) \leq T_{old}(X) \tag{50}$$

since on the contrary X would have been chosen instead of A as the node to be accepted and

$$T(A) \leq T(B) \tag{51}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

for the same reason. Moreover we have $T(C) \leq T(A)$ by induction and $T(B) \leq T(D)$ by assumption. In conclusion, the nodes contributing to the computation of $T(X)$ are C and B or only C. The fact that A has been labeled as *accepted* has no effect on the computation so we are again in Case 1 or 2. This implies,

$$T(C) \leq T_{new}(X) \leq T(C) + f_X \leq T(A) + f_X.$$

Now we prove that $T_{new} \geq T(A)$. When $T_{old}(X)$ was computed the algorithm was in the Case 1 or 2, so

$$T(C) \leq T_{old}(X) \leq T(C) + f_X. \tag{52}$$

Moreover we have

$$T(C) \leq T(B) \tag{53}$$

by induction.

If $T(B) > T_{old}(X)$, then the node contributing to the computation of $T(X)$ is only C, so we have

$$T_{new}(X) = T(C) + f_X \geq T_{old}(X) \geq T(A).$$

Otherwise, if $T(B) \leq T_{old}(X)$, the nodes contributing to the computation of $T(X)$ are C and B.

Using this last assumption, (52), and (53) we have

$$\begin{aligned} T(C) \leq T(B) \leq T_{old}(X) \leq T(C) + f_X &\Rightarrow 0 \leq T(B) - T(C) \leq f_X \\ &\Rightarrow (T(B) - T(C))^2 \leq f_X^2. \end{aligned} \tag{54}$$

Moreover, by (50) and (52) we have

$$T(C) + f_X \geq T(A). \tag{55}$$

Computation of X leads to

$$\begin{aligned} T_{new}(X) &= \frac{T(C) + T(B) + \sqrt{2f_X^2 - (T(C) - T(B))^2}}{2} \\ &= \frac{(T(C) + f_X) - f_X + T(B) + \sqrt{2f_X^2 - (T(C) - T(B))^2}}{2}. \end{aligned} \tag{56}$$

Using (55), (54), and (51) we obtain

$$\begin{aligned} T_{new}(X) &\geq \frac{T(A) - f_X + T(B) + \sqrt{2f_X^2 - (T(C) - T(B))^2}}{2} \\ &\geq \frac{T(A) - f_X + T(B) + \sqrt{f_X^2}}{2} \geq \frac{T(A) + T(A)}{2} = T(A). \end{aligned} \tag{57}$$

Finally, let us remark that the cases when two or more nodes among B, C, and D are accepted can be treated as in the previous cases. Note that if D is accepted, then B must also be accepted since $T(B) \leq T(D)$.

To complete the proof, it is necessary to show that the expression appearing under the square root in the computation of $T(X)$ expressed as a function of its two

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

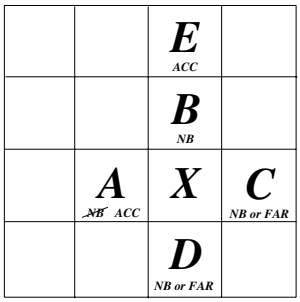

FIG. 16. *Proof that radicand is positive under the CFL-like condition* (15).

neighbors is nonnegative. Let us start by proving that the hypothesis (15) guarantees that

$$\frac{c(Z)}{c(Z')} \leq \sqrt{2} \tag{58}$$

for any couple of nodes Z and Z' such that

$$|Z - Z'| = \Delta x.$$

In fact, by assumption we have

$$|c(Z) - c(Z')| \leq L_c |Z - Z'|.$$

If $|Z - Z'| = \Delta x$, we have that

$$|c(Z) - c(Z')| \leq L_c \Delta x \leq (\sqrt{2} - 1) c_{min} \leq (\sqrt{2} - 1) c(Z'),$$

which implies

$$c(Z) - c(Z') \leq (\sqrt{2} - 1) c(Z')$$

and then

$$c(Z) \leq \sqrt{2}\, c(Z').$$

Let us examine the three cases where we need to show that the radicand is nonnegative.

Case 2. Since B is in the narrow band, there must be at least one neighbor belonging to accepted. Let E be this node (see Figures 16 and 1). Moreover, $T(A) \leq T(B)$ since A has been chosen to be labeled as *accepted* and $T(E) \leq T(A)$ because E became accepted before A. By the previous results, we get

$$T(E) \leq T(B) \leq T(E) + f_B,$$

which implies

$$T(A) \leq T(B) \leq T(E) + f_B \leq T(A) + f_B$$

and

$$0 \leq T(B) - T(A) \leq f_B. \tag{59}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Choosing $Z = X$ and $Z' = B$ in (58), we get

$$\frac{c(X)}{c(B)} \leq \sqrt{2}$$

and then

$$\sqrt{2} f_X \geq f_B. \tag{60}$$

Finally (59) and (60) imply

$$\sqrt{2} f_X \geq T(B) - T(A) \geq 0,$$

so we can conclude that

$$2 f_X^2 - (T(B) - T(A))^2 \geq 0.$$

Case 3 and 4. In these cases, (48) and (54) guarantee, respectively, that the expression appearing under the radicand is always positive. □

Let us show now that the value at the node which is labeled as *accepted* at every iteration is exact. Let us denote this value by T_{min}. Since all the nodes in the narrow band have values greater than T_{min}, the previous result implies that using those nodes we cannot assign to a node a value lower than T_{min}. In conclusion (see [33]), the up-winding is respected and the value T_{min} can be considered exact since it cannot be improved on the same grid (of course it can be improved if we reduce the discretization steps).

Note that Theorem 1 is valid also for the FM-FD method.

Acknowledgment. The authors wish to thank M. Sagona for helpful discussions on the FM-FD and FM-SL methods.

REFERENCES

[1] S. Augoula and R. Abgrall, *High order numerical discretization for Hamilton-Jacobi equations on triangular meshes*, J. Sci. Comput., 15 (2000), pp. 197–229.
[2] M. Bardi and I. Capuzzo Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Boston, 1997.
[3] M. Bardi and M. Falcone, *An approximation scheme for the minimum time function*, SIAM J. Control Optim., 28 (1990), pp. 950–965.
[4] G. Barles, *Solutions de viscosité des equations de Hamilton-Jacobi*, Springer, Berlin, 1994.
[5] F. Camilli and A. Siconolfi, *Hamilton-Jacobi equations with measurable dependence on the state variable*, Adv. Differential Equations, 8 (2003), pp. 733–768.
[6] E. Carlini, M. Falcone, and R. Ferretti, *An efficient algorithm for Hamilton-Jacobi equations in high dimension*, Comput. Vis. Sci., 7 (2004), pp. 15–29.
[7] D. L. Chopp, *Some improvements of the fast marching method*, SIAM J. Sci. Comput., 23 (2001), pp. 230–244.
[8] M. G. Crandall and P.-L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19.
[9] E. Cristiani, *Fast Marching and Semi-Lagrangian Methods for Hamilton-Jacobi Equations with Applications*, Ph.D. thesis, Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, SAPIENZA - Università di Roma, Rome, Italy, 2007.
[10] E. Cristiani, M. Falcone, and A. Seghini, *Numerical solution of the perspective shape from shading problem*, in Proceedings of Control Systems: Theory, Numerics and Applications PoS (CSTNA2005) 008, http://pos.sissa.it/.
[11] P. Danielsson, *Euclidean distance mapping*, Comput. Graphics Image Process., 14 (1980), pp. 227–248.
[12] M. Falcone, *The minimum time problem and its applications to front propagation*, in Motion by Mean Curvature and Related Topics, A. Visintin and G. Buttazzo, eds., de Gruyter, Berlin, 1994, pp. 70–88.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[13] M. Falcone, *Numerical solution of dynamic programming equations*, in Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations, M. Bardi and I. Capuzzo Dolcetta, eds., Birkhäuser, Boston, 1997, Appendix A, pp. 471–504.

[14] M. Falcone, *Some remarks on the synthesis of feedback controls via numerical methods*, in Optimal Control and Partial Differential Equations, J. L. Menaldi, E. Rofman, and A. Sulem, eds., IOS Press, The Netherlands, 2001, pp. 456–465.

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

[16] 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.

[17] M. Falcone, T. Giorgi, and P. Loreti, *Level sets of viscosity solutions: Some applications to fronts and rendez-vous problems*, SIAM J. Appl. Math., 54 (1994), pp. 1335–1354.

[18] M. Falcone and C. Truini, *A Level-Set Algorithm for Front Propagation in the Presence of Obstacles*, Technical Report, Dipartimento di Matematica, Università di Roma "La Sapienza," Rome, Italy, 2003.

[19] R. Fedkiw and S. Osher, *Level set methods and dynamic implicit surfaces*, Appl. Math. Sci. 153, Springer, New York, 2003.

[20] C.-Y. Kao, S. Osher, and J. Qian, *Lax-Friedrichs sweeping scheme for static Hamilton–Jacobi equations*, J. Comput. Phys., 196 (2004), pp. 367–391.

[21] C.-Y. Kao, S. Osher, and Y.-H. Tsai, *Fast sweeping methods for static Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 42 (2005), pp. 2612–2632.

[22] S. Kim, *An $O(N)$ level set method for eikonal equations*, SIAM J. Sci. Comput., 22 (2001), pp. 2178–2193.

[23] R. Kimmel and J. A. Sethian, *Computing geodesic paths on manifold*, Proc. Natl. Acad. Sci. USA, 95 (1998), pp. 8431–8435.

[24] R. Kimmel and J. A. Sethian, *Optimal algorithm for shape from shading and path planning*, J. Math. Imaging Vision, 14 (2001), pp. 237–244.

[25] B. O'Neill, *Elementary Differential Geometry*, Academic Press, New York, London, 1966.

[26] S. Osher, *A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations*, SIAM J. Math. Anal., 24 (1993), pp. 1145–1152.

[27] J. Qian, Y.-T. Zhang, and H. K. Zhao, *Fast sweeping methods for eikonal equations on triangular meshes*, SIAM J. Numer. Anal., 45 (2007),pp. 83–107.

[28] J. Qian, Y.-T. Zhang, and H. Zhao, *A fast sweeping method for static convex Hamilton-Jacobi equations*, J. Sci. Comput., 31 (2007), pp. 237–271.

[29] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM J. Numer. Anal., 29 (1992), pp. 867–884.

[30] M. Sagona, *Numerical Methods for Degenerate Eikonal Type Equation and Applications*, Ph.D. thesis, Dipartimento di Matematica, Università di Napoli "Federico II," Naples, Italy, 2001.

[31] M. Sagona and A. Seghini, *An adaptive scheme for the shape from shading problem*, in Numerical Methods for Viscosity Solutions and Applications, M. Falcone and Ch. Makridakis, eds., World Scientific, Singapore, 2001, pp. 197–219.

[32] J. A. Sethian, *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK, 1999.

[33] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595.

[34] J. A. Sethian and A. Vladimirsky, *Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325–363.

[35] P. Soravia, *Boundary value problems for Hamilton-Jacobi equations with discontinuous Lagrangian*, Indiana Univ. Math. J., 51 (2002), pp. 451–477.

[36] C. Truini, *Approssimazione numerica del problema controllistico di tempo minimo e applicazione all'evoluzione dei fronti*, Master Thesis, Dipartimento di Matematica, Università di Roma "La Sapienza," Rome, Italy, 1995.

[37] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao, *Fast sweeping algorithms for a class of Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 41 (2003), pp. 673–694.

[38] J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528–1538.

[39] H. Zhao, *A fast sweeping method for eikonal equations*, Math. Comp., 74 (2005), pp. 603–627.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.