SIAM J. SCI. COMPUT.
Vol. 36, No. 2, pp. A570–A587
© 2014 Society for Industrial and Applied Mathematics

CAN LOCAL SINGLE-PASS METHODS SOLVE ANY STATIONARY HAMILTON–JACOBI–BELLMAN EQUATION?*

SIMONE CACACE†, EMILIANO CRISTIANI‡, AND MAURIZIO FALCONE§

Abstract. The use of *local single-pass* methods (like, e.g., the fast marching method) has become popular in the solution of some Hamilton–Jacobi equations. The prototype of these equations is the *eikonal equation*, for which the methods can be applied saving CPU time and possibly memory allocation. Then some questions naturally arise: Can local single-pass methods solve any Hamilton–Jacobi equation? If not, where should the limit be set? This paper tries to answer these questions. In order to give a complete picture, we present an overview of some fast methods available in the literature and briefly analyze their main features. We also introduce some numerical tools and provide several numerical tests which are intended to exhibit the limitations of the methods. We show that the construction of a local single-pass method for general Hamilton–Jacobi equations is very hard, if not impossible. Nevertheless, some special classes of problems can actually be solved, making local single-pass methods very useful from a practical point of view.

Key words. fast marching methods, fast sweeping methods, eikonal equation, Hamilton–Jacobi equations, semi-Lagrangian schemes

AMS subject classifications. 65N12, 49L20, 49M99

DOI. 10.1137/130907707

1. Introduction.

The study of Hamilton–Jacobi (HJ) equations arises in several applied contexts, including classical mechanics, front propagation, control problems, and differential games, and has had a great impact in many areas, such as robotics, aeronautics, and electrical and aerospace engineering. In particular, for control/game problems, an approximation of the value function allows for the synthesis of optimal control laws in feedback form, and then for the computation of optimal trajectories. The value function for a control problem (resp., differential game) can be characterized as the unique viscosity solution of the corresponding Hamilton–Jacobi–Bellman (HJB) equation (resp., Hamilton–Jacobi–Isaacs (HJI) equation), and it is obtained by passing to the limit in Bellman's well-known dynamic programming (DP) principle. The DP approach can be rather expensive from a computational point of view, but in some situations it gives a real advantage over methods based on Pontryagin's maximum principle, because the latter approach allows one to compute only open-loop controls and locally optimal trajectories. Moreover, weak solutions to HJ equations are nowadays well understood in the framework of viscosity solutions, which offers the correct notion of solution for many applied problems.

The above remarks have motivated the research of efficient and accurate numerical methods. Indeed, an increasing number of techniques have been proposed for

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 29, 2013; accepted for publication (in revised form) December 26, 2013; published electronically March 27, 2014. This work was supported by AFOSR grant FA9550-10-1-0029, ITN–Marie Curie grant 264735-SADCO, and SAPIENZA 2009 "Analisi ed approssimazione di modelli differenziali nonlineari in fluidodinamica e scienza dei materiali."

http://www.siam.org/journals/sisc/36-2/90770.html

†Dipartimento di Matematica, SAPIENZA – Università di Roma, 00185 Rome, Italy (cacace@mat.uniroma1.it).

‡Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche, Rome, Italy (e.cristiani@iac.cnr.it).

§Corresponding author. Dipartimento di Matematica, SAPIENZA – Università di Roma, 00185 Rome, Italy (falcone@mat.uniroma1.it).

the approximation of viscosity solutions. They range from finite difference to finite volume, from discontinuous Galerkin to semi-Lagrangian schemes. In any case, for optimal control problems and games, the DP approach suffers from the so-called curse of dimensionality limitation; i.e., the size of nonlinear systems needed to solve high-dimensional problems becomes huge, making the numerical solution unfeasible in terms of both memory allocation and CPU time. The curse of dimensionality can sometimes be overcome by exploiting the peculiarities of the problem, if any (e.g., symmetry, periodicity, linearity), or by adopting a linearization based on the so-called "max-plus algebra" approach, which unfortunately presents other types of constraints; see, e.g., the book by McEneaney [23]. It is rather clear that the DP approach needs a big effort in the construction of numerical approximation schemes for two different reasons. The first, which is valid even in low dimensions, is due to the low regularity of viscosity solutions which are typically only Lipschitz continuous or even discontinuous, as in the case of constrained control problems and pursuit-evasion games. The second reason is related to the above-mentioned curse of dimensionality, which pushes towards methods with low memory allocation and, possibly, the definition of some rule to reduce the number of elementary operations and the CPU time.

Another motivation for efficient numerical methods is the approximation of front propagation problems via the level-set method. The motivation there is to reduce or eliminate the extra dimension which is added by the level-set method and obtain a fast and reliable algorithm. Starting in the 1980s, many efforts have been made to improve the efficiency of these numerical methods, a crucial step for the solution of real-world problems.

In this paper we deal with numerical methods for solving first-order nonlinear convex stationary HJB equations. In particular, we focus on the applicability of the fast marching method (FMM), introduced in the pioneering works by Tsitsiklis [29], Sethian [26], Helmsen et al. [18], and its generalizations; see, e.g., [1, 5, 6, 7, 8, 10, 11, 19, 21, 24, 27]. We analyze features and limitations of this kind of algorithm, aiming at understanding whether it is possible to construct local single-pass methods (see Definitions 1.1 and 1.2 below) to solve every HJB equation. Then we discuss whether or not the research on this topic should look for new future directions still based on the local single-pass idea and/or switch to other acceleration methods, such as fast sweeping methods (FSMs); see, e.g., [3, 20, 28, 30, 31] (see also the pioneering work [22, p. 168], where a sketch of the method is given, and [12], where a similar method in a discrete setting is proposed).

It is well known that FMM is an efficient numerical technique for solving the eikonal equation. This explains why we have decided to use as a guideline the following equations, which generalize the eikonal equation and are associated to some minimum time problems with target:

$$\sup_{a\in B(0,1)} \{-a\cdot\nabla T(x)\} = 1 \qquad \text{(homogeneous eikonal)}, \tag{1.1}$$

$$\sup_{a\in B(0,1)} \{-c_1(x)a\cdot\nabla T(x)\} = 1 \qquad \text{(nonhomogeneous eikonal)}, \tag{1.2}$$

$$\sup_{a\in B(0,1)} \{-c_2(a)a\cdot\nabla T(x)\} = 1 \qquad \text{(homogeneous anisotropic eikonal)}, \tag{1.3}$$

$$\sup_{a\in B(0,1)} \{-c_3(x,a)a\cdot\nabla T(x)\} = 1 \qquad \text{(nonhomogeneous anisotropic eikonal)}, \tag{1.4}$$

$$\sup_{a\in B(0,1)} \{-f(x,a)\cdot\nabla T(x)\} = 1 \qquad \text{(minimum time HJB)}, \tag{1.5}$$

where $x \in \mathbb{R}^d \backslash \mathcal{T}$, $\mathcal{T}$ is a closed nonempty target set in $\mathbb{R}^d$, c_1, c_2, c_3 are given strictly positive and Lipschitz continuous scalar functions, f is a given vector-valued Lipschitz continuous function, and $B(0,1)$ is the unit ball in $\mathbb{R}^d$, representing the set of the admissible controls. To simplify the presentation we will always consider the homogeneous Dirichlet condition $T = 0$ on $\mathcal{T}$, but also other boundary conditions can be applied, provided some compatibility conditions between the vectorfield f and $\partial\mathcal{T}$ hold true. Let us also note, for readers not familiar with control applications, that (1.1) and (1.2) can be written in a more standard form as

$$|\nabla T(x)| = 1 \qquad \text{and} \qquad c_1(x)|\nabla T(x)| = 1,$$

respectively. Moreover, the above relation shows the equivalence between the front propagation problem described by the level-set method and the minimum time problem, as one can find in [15]. To simplify the notation, we restrict the discussion to the case $d = 2$, but the results of the paper are valid in any dimension.

It is interesting to note that the single-pass idea has also been applied to nonconvex Hamiltonians, e.g., the HJI equation corresponding to pursuit-evasion games [4, 9]. In pursuit-evasion games the structure of the solution is similar to the minimum time problem because the characteristic information propagates from a given target to the rest of the space. Clearly for games the structure of optimal trajectories (which coincide with the characteristics of the problem) is much more complicated due to the presence of two independent players. However, a complete theory for fast-marching-like methods in this framework is still missing and, as we will see later, it will be hard to develop it since these methods can fail even in the convex case (see the examples in section 5).

As mentioned above, in the last decades many numerical schemes and algorithms were proposed to solve the above equations. Some of these schemes are listed in the next section, together with their main properties. As is well known, one important feature held by fast-marching-like methods is that the solution to the HJ equation is computed in a finite number of steps. More precisely, these methods are *single-pass* in the sense of the following definition.

DEFINITION 1.1 (single-pass algorithm). *An algorithm is said to be* single-pass *if each mesh point is recomputed at most r times, where r depends only on the equation and the mesh structure, not on the number of mesh points.*

Single-pass algorithms usually divide the numerical grid into at least three subsets: *accepted* (*ACC*) region, *considered* (*CONS*) region, and *far* (*FAR*) region. Nodes in *ACC* are definitively computed, nodes in *CONS* are computed but their values are not yet final, and nodes in *FAR* are not yet computed.

We also introduce the following definition.

DEFINITION 1.2 (local single-pass algorithm). *A single-pass algorithm is said to be* local *if the computation at any mesh point involves only the values of first neighboring nodes, the region CONS is one-cell thick, and no information coming from the FAR region is used.*

The paper is organized as follows: In section 2 we summarize some of the existing methods to solve HJB equations and introduce two semi-Lagrangian numerical schemes. In section 3, we present new numerical tools which will be useful in investigating the applicability of local single-pass methods. In section 4, which is the core of the paper, we discuss the main features and limitations of the methods presented in sections 2 and 3, and we address the problem of extending local single-pass methods to general HJB equations. Finally, in section 5 we present several experiments and

numerical tests in order to compare the two schemes described in section 2 and to confirm the scenario depicted in section 4.

2. Background and general approximation schemes. Fast methods for HJB equations are usually designed to work with different local schemes, including finite difference and semi-Lagrangian (SL) schemes. Several results show that, in many cases on structured grids and at a reasonable cost, SL schemes provide better accuracy than other schemes (see, e.g., [11, 16]), due to their ability to follow directions which are oblique with respect to the coordinate axes. In this section we recall, for the reader's convenience, two SL schemes for HJB equations, which will be compared in section 5. Then we list and discuss some of the iterative and fast-marching-like methods available in the literature.

2.1. Two SL schemes. Let us introduce a structured grid G and denote its nodes by x_i, $i = 1, \dots, N$. The space step is assumed to be uniform and equal to $\Delta x > 0$. HJ equations can be discretized by means of the discrete version of the DP principle. In this way the relationship with the optimal control framework is never lost. Standard arguments [2] lead to the following discrete version of the HJB equation (1.5):

$$T(x_i) \approx \widehat{T}(x_i) = \min_{a \in B(0,1)} \left\{ \widehat{T}(\tilde{x}_{i,a}) + \frac{|x_i - \tilde{x}_{i,a}|}{|f(x_i, a)|} \right\}, \qquad x_i \in G, \tag{2.1}$$

where $\tilde{x}_{i,a}$ is a *nonmesh* point, obtained by integrating, until a certain final time τ, the ordinary differential equation

$$\begin{cases} \dot{y}(t) = f(y, a), & t \in [0, \tau], \\ y(0) = x_i \end{cases} \tag{2.2}$$

and then setting $\tilde{x}_{i,a} = y(\tau)$. To make the scheme fully discrete, the set of admissible controls $B(0, 1)$ is discretized in N_c points, and we denote by a^* the optimal control achieving the minimum in (2.1). Note that we can get different versions of the SL scheme (2.1) varying τ, the method used to solve (2.2), and the interpolation method used to compute $\widehat{T}(\tilde{x}_{i,a})$. Moreover, we remark that, in any single-pass method, the computation of $\widehat{T}(x_i)$ cannot involve the value $\widehat{T}(x_i)$ itself, because this self-dependency would make the method iterative.

A two-point SL scheme. This scheme is used, for example, in [27] and [29]. Equation (2.2) is solved by an explicit forward Euler scheme until the solution intercepts the line connecting two neighboring points $x_{i,1}$ and $x_{i,2}$ (see Figure 1a). The value $\widehat{T}(x_i)$ is computed by a one-dimensional linear interpolation of the values $\widehat{T}(x_{i,1})$ and $\widehat{T}(x_{i,2})$ with weights $\lambda_{i,1}$ and $\lambda_{i,2}$, respectively ($\lambda_{i,1} + \lambda_{i,2} = 1$).

A three-point SL scheme. This scheme is used, for example, in [11]. Equation (2.2) is solved by an explicit forward Euler scheme until the solution is at distance Δx from x_i, where it falls inside the triangle of vertices $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ (see Figure 1b). The value $\widehat{T}(x_i)$ is computed by a two-dimensional linear interpolation of the values $\widehat{T}(x_{i,1})$, $\widehat{T}(x_{i,2})$, and $\widehat{T}(x_{i,3})$ with weights $\lambda_{i,1}$, $\lambda_{i,2}$, and $\lambda_{i,3}$, respectively ($\lambda_{i,1} + \lambda_{i,2} + \lambda_{i,3} = 1$).

Remark 2.1. It is important to note that when the algorithm employs the single-pass technique (with *ACC-CONS-FAR* sets), the nodes in *CONS* can be either included in or excluded from the computation of $\widehat{T}(x_i)$. Indeed, we can decide to force

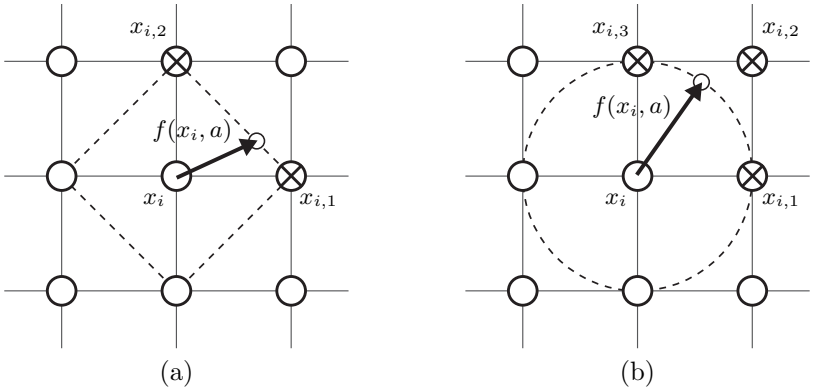

FIG. 1. (a) *Two-point SL scheme.* (b) *Three-point SL scheme.*

the scheme to employ only nodes in ACC, temporarily assuming that nodes in $CONS$ have very large values, so that they are automatically rejected by the minimum search in (2.1). Otherwise, we can employ nodes in ACC and $CONS$, although the values at nodes in $CONS$ are not in general correct since they can still vary in the following iterations.

2.2. Some algorithms for HJB equations. Here we list and briefly describe some iterative and single-pass methods for solving HJB equations.

Iterative method (ITM). This method naturally exploits the fixed-point form of the discrete DP principle. It has been applied in [22] to stochastic control problems, where the deterministic case can be obtained vanishing the coefficient in front of the diffusion term. Later, a similar operator has been used in [17] to construct an approximation of the value function in a deterministic control problem. In the framework of viscosity solutions, the fully discrete DP scheme was first studied in [14]. The interested reader can also find a detailed description of the algorithm and some acceleration methods in [16]. A finite difference version for a specific application to the shape-from-shading problem can be found in [25].

Starting from some initial guess $\widehat{T}^{(0)}$ defined on the whole grid (compatible with the Dirichlet conditions imposed on $\mathcal{T}$), the nodes are visited in some unique and predefined order. At each visit, the numerical scheme is applied and a new value for the node is computed. This leads to a fixed-point algorithm of the form

$$\widehat{T}^{(n)} = \widehat{H}(\widehat{T}^{(n-1)}), \qquad n = 1, 2, 3, \dots,$$

where $\widehat{H}$ denotes a discrete Hamiltonian associated to the corresponding HJ equation. Gauss–Seidel-type or Jacobi–type iterations are possible. For a practical implementation, a criterion of the form

$$\max_{x_i \in G} |\widehat{T}^{(n)}(x_i) - \widehat{T}^{(n-1)}(x_i)| < tol \tag{2.3}$$

is needed in order to stop the computation at a desired precision tol. Clearly this method is not single-pass, since the number of iterations needed to reach convergence depends both on the grid size Δx and the dynamics underlying the equation. ITM was proved to be convergent, provided a suitable numerical scheme is employed.

Fast sweeping method (FSM) [3, 20, 28, 30, 31]. This method is similar to ITM, but the grid is visited in a multiple-direction predefined order. Usually, a rectangular grid is iteratively swept along four directions: $N \to S$, $E \to W$, $S \to N$, and $W \to E$, where N, S, E, W stand for North, South, East, and West, respectively. This method has been shown to be much faster than ITM, but, as ITM, it not single-pass. A well-known exception is given by the eikonal equation, for which it is proved that only one sweep (i.e., four visits of the whole grid) is enough to reach convergence (see [30] for details). FSM computes the same solution of ITM, provided the same scheme and the same stopping rule are employed.

Fast marching method (FMM) [18, 26, 29]. This method has been introduced as a fast solver for the eikonal equation. It differs from the previous ones, since the nodes are visited in a solution-dependent order, producing a single-pass method: the algorithm itself finds a correct order for processing the grid nodes. The order which is determined satisfies the *causality* principle; i.e., the computation of a node is declared completed only if its value cannot be affected by the future computation. As recalled in section 1, at each step the grid is divided into three regions: *ACC*, where computation is definitively done, *CONS*, where computation is going on, and *FAR*, where computation is not done yet. Then the node in *CONS* with the minimal value enters *ACC*, its first neighbors enter *CONS* (if not yet in) and are (re)computed.

Following [27], we remark that this *minimum-value rule* corresponds to computing the value function T step by step in ascending order (i.e., from the simplex containing $-\nabla T$). It follows that *CONS* expands under the gradient flow of the solution itself, which is exactly equivalent to saying that *CONS* is, at each step, an approximation of a level set of the value function. In the case of isotropic eikonal equation (1.2), the gradient of the solution coincides with the characteristic field, and hence FMM computes the correct solution. Moreover, FMM still works for problems with mild anisotropy, where gradient lines and characteristics define small angles and lie, at each point, in the same simplex of the underlying grid. On the other hand, when a strong anisotropy comes into play, as for a general anisotropic eikonal equation (1.3), FMM fails and there is no way to compute the viscosity solution following its level sets. Finally, we remark that FMM is also a local method, since each node is computed by means of first neighbors nodes only, and *CONS* is one cell thick. Moreover, FMM computes the same solution of ITM, provided the same scheme is employed.

Characteristic fast marching method (CFMM) [10]. This method is inspired by FMM, it is local and single-pass, and it can be used to solve some eikonal equations. It replaces the search for the minimum value in *CONS* with the search of the node where the characteristic line passes with maximal speed. The acceptance rule is also modified: a node x_i in *CONS* enters *ACC* if the point $x_i + f(x_i, a^*)$ falls in *ACC*. As the group marching method [21], more than one node can enter *ACC* at the same time, making the method in general faster than FMM. Note that CFMM does not always work if the solution of the equation is not differentiable.

Ordered upwind method (OUM) [27]. This method is inspired by FMM, but it is able to solve more general equations than the eikonal one, including nonhomogeneous anisotropic eikonal equations (1.4). This can be obtained by enlarging the stencil of the scheme, so that a value at a node x_i can be computed by using values at some nodes x_j that are far from the node x_i. This makes the method nonlocal. The maximal allowed distance $|x_i - x_j|$ depends on the degree of anisotropy of the equation. OUM is a single-pass method which computes the same solution of ITM (employing the same numerical scheme) only in the limit $\Delta x \to 0$.

A generalization of OUM was recently proposed in [1] to solve static convex HJ equations on highly nonuniform grids. The new method MAOUM (monotone acceptance OUM) computes the solution in a fast-marching fashion, but employs large stencils (even larger than OUM) that are precomputed for each grid node. This makes MAOUM two-pass and nonlocal.

Buffered fast marching method (BFMM) [8]. This method is inspired by FMM and can be used to solve any HJ equations modeling monotone front propagation. Although only first neighbors are involved in the computation, BFMM cannot be considered a local method, since *CONS* can increase its thickness. More precisely, the *CONS* region is extended by a *buffer* region, whose size depends on the dynamics of the equation and, in the worst case, it can cover the whole grid, thus making BFMM comparable to ITM. BFMM is not single-pass and computes the same solution of ITM, provided the same scheme is employed.

Progressive fast marching method (PFMM) [4]. This method can be considered a localization of BFMM. It is indeed a local method, but not single-pass. Some experimental results have shown that it can solve quite general problems, including pursuit-evasion games with state and control constraints. PFMM has been introduced for theoretical purposes only, since it is very slow (slower than ITM) and thus not usable in practice. It proposes a completely new rule for accepting nodes in *CONS*: in the *FAR* region, next to the *CONS* region, a layer of "tempting" values is placed and *progressively* increased, acting as an external boundary condition. For each tempting value, the solution is recomputed in *CONS*, recording the corresponding variations. The first node in *CONS* which is not affected by this external layer enters *ACC*. The "tempting" values can be considered as a guess on the outcome of the future computation, and the new rule of acceptance allows one to find the node in *CONS* that cannot be affected by it.

3. New tools and verification methods. In this section we consider four additional fast-marching-like methods. The first two are acceleration techniques which are expected to provide the same solution of ITM whenever they are applicable. The last two, labeled *dumb*, are not new methods for solving HJ equations; rather they are *verification tools*. They will be used to analyze features and limitations of the methods already presented, with the aim of giving a comprehensive classification of the equations that can be solved by local single-pass algorithms. Our ultimate goal is to discuss the possibility that local single-pass methods for solving general HJ equations may not exist. We give two preliminary definitions.

DEFINITION 3.1 (safe node). *Let $x_i \in CONS$ and let $x^*_{i,1}, \dots, x^*_{i,p}$ be the neighboring interpolation points of x_i achieving the minimum in* (2.1) *($p = 2$ or $p = 3$, depending on the employed SL scheme). Denote by $\lambda^*_{i,1}, \dots, \lambda^*_{i,p}$ the corresponding interpolation weights, and define, for $j = 1, \dots, p$,*

$$b_{i,j} = \begin{cases} 1 & \text{if } x^*_{i,j} \in ACC, \\ 0 & \text{otherwise.} \end{cases}$$

The node x_i is said to be safe *if $\sum_{j=1}^{p} \lambda^*_{i,j} b_{i,j} = 1$.*

The previous definition (see Figure 2 in the case $p = 3$), means that the computation at x_i involves values of nodes in *ACC* only. By Remark 2.1, it is clear that the notion of *safeness* makes sense only if the scheme in use can employ nodes in *CONS*. Otherwise, all the nodes are safe by construction. Then we always allow the scheme to use nodes in *CONS*. From a practical point of view, we remark that round-off errors

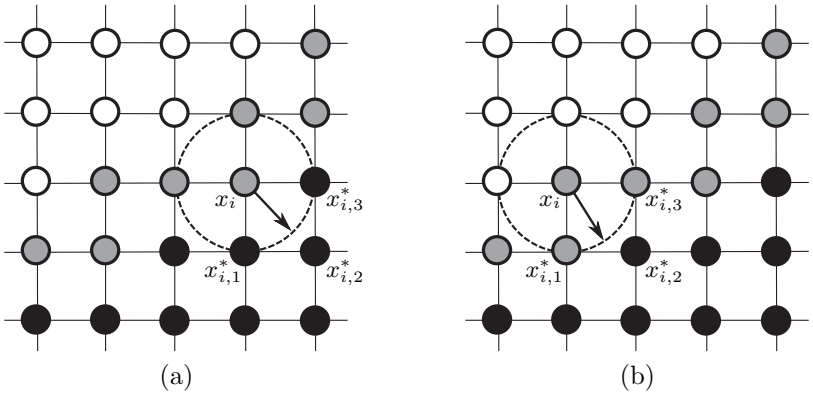

FIG. 2. *Black nodes belong to ACC, gray nodes belong to CONS, and white nodes belong to FAR. The arrow denotes the optimal velocity field $f(x_i, a^*)$, and $x^*_{i,1}, x^*_{i,2}, x^*_{i,3}$ are the corresponding neighboring interpolation points involved in the computation of $T(x_i)$.* (a) *x_i is safe: its value depends on nodes in ACC only.* (b) *x_i is not safe: its value depends on nodes in ACC and CONS.*

can prevent the safeness to be satisfied. Then we can relax this condition by requiring that $\sum_{j=1}^{p} \lambda_{i,j} b_{i,j} \geq 1 - \sigma$ for some tolerance $\sigma > 0$.

DEFINITION 3.2 (exact node). *Let T^{exact} be the solution computed by ITM, setting tol equal to machine precision in* (2.3)*. A node $x_i \in CONS$ is said to be* exact *if its value $T(x_i)$ coincides with $T^{exact}(x_i)$ up to machine precision.*

The two new fast-marching-like tools are as follows.

Safe fast marching method (SFMM). This method is identical to FMM but for the rule of acceptance: at each step, the safe node with minimal value in *CONS* enters *ACC*. This method is local and single-pass.

Safe method (SM). This method is identical to FMM but for the rule of acceptance: at each step, whichever safe node in *CONS* enters *ACC*. This method is local and single-pass.

Note that the existence of safe nodes in *CONS* is not in general guaranteed; then SFMM and SM could stop prematurely before *ACC* covers the whole domain. In the next section we will discuss when these method can be successfully employed.

The two verification methods are as follows.

Safe dumb method (SDM). This method is identical to FMM but for the rule of acceptance: at each step, whichever safe and exact node in *CONS* enters *ACC*.

This method is not usable in practice, since it assumes that one already knows the solution of ITM (or any other equivalent method). It clearly computes the same solution of ITM and it is nonlocal, because it employs the information contained in the final solution of ITM, defined everywhere. SDM is introduced for theoretical purposes since it represents a limit for the applicability of any local single-pass method. Indeed, if there are no nodes in *CONS* which are both safe and exact, we can conclude that the numerical domain of dependence of every exact node in *CONS* includes nodes in *CONS*. Since one cannot say if values at nodes in *CONS* are exact or not, we face a loop dependency that cannot be resolved keeping the method local and single-pass. As a consequence, if SDM fails, then any local single-pass method will fail.

Dumb method (DM). This method is identical to FMM but for the rule of acceptance: at each step, whichever exact node in *CONS* enters *ACC*. Similar considerations made for SDM apply.

This method is also introduced for theoretical purposes. Unexpectedly, DM does not always work. For some pathological dynamics and choice of the mesh, it can happen that *CONS* does not contain any exact node, and the algorithm stops (see section 5 for an example). In such cases it seems that enlarging *CONS* or breaking the single-pass property is the only way to make the algorithm process all the nodes.

Remark 3.1. Every fast-marching-like method requires a selection rule to move nodes from *CONS* to *ACC*. It is possible that more than one node in *CONS* satisfies that rule at the same step of the algorithm. In this case we can either move one node at random among the correct ones or move all the nodes at once. In the latter case we often get an additional speed-up of the algorithm. For example, in FMM one can find two or more nodes in *CONS* with the minimum value, while in SM one can find two or more safe nodes. Investigating the difference of the two implementations is beyond the scope of the paper, since we are mainly interested in the applicability of the methods rather than in their performance. Then we always move in ACC one node at a time.

4. Applicability of local single-pass methods. In this section we address the problem of extending the range of applicability of local single-pass methods to general HJ equations. To this end, we focus on three algorithms discussed in the previous sections which are local and single-pass, namely, FMM, SFMM, and SM. In order to point out their features and limitations, we will also employ the two verification methods SDM and DM.

From the numerical point of view, it is meaningful to divide HJB equations into two classes. Given a mesh, we have the following:

(ISO) Equations whose characteristic lines coincide or lie in the same simplex of the gradient lines of their solutions. The prototype equation is the eikonal isotropic (1.2).

(¬ISO) Equations for which there exists at least a grid node where the characteristic line and the gradient of the solution do not lie in the same simplex.

FMM works for equations of type ISO and fails for equations of type ¬ISO (see [27] for further details and explanations). Let us introduce two other classes for HJB equations of type (1.5):

(REG) Equations with noncrossing (regular) characteristic lines. Characteristics spread from the target $\mathcal{T}$ to the rest of the domain without intersecting.

(¬REG) Equations with crossing characteristic lines. Characteristics start from the target $\mathcal{T}$ and then meet in finite time, creating shocks. As a result, the solution T is not differentiable at shocks.

Let us comment on the applicability of the local single-pass methods by making use of the classifications introduced above.

(1) SM solves REG. SM can be applied in the case REG, provided SDM works. Let us denote by x_i one safe node in *CONS* (x_i exists because we assume SDM can be applied). By definition of safeness, the value at x_i depends only on values at nodes in *ACC*, which can be assumed to be exact by induction. Then the exactness of the value at x_i is guaranteed by the property REG, which implies that no characteristics will reach x_i in the future from another direction, possibly changing its value. In other words, the information passes through x_i one and only one time. Then once x_i is reached by the region *ACC*, it is ready to enter *ACC*.

(2) Is the minimum-value rule really needed? Having in mind the FMM (and its ancestor, Dijkstra's algorithm [13]), one can be convinced that giving priority to the smallest value among nodes in *CONS* is an essential request to making the method work. On the contrary, by the above comment (1), we know that a method like SM, which makes no distinction among nodes with respect to their values, works in the case REG (both ISO and ¬ISO), provided SDM works. The choice of the minimum value becomes essential only in the ¬REG case, where characteristics reach some point from two or more different directions. We discuss this point in the next comment.

(3) Handling shocks in the ¬REG case. Let us consider the ¬REG case, and let x be a point belonging to a shock, i.e., where the solution is not differentiable. By definition, the value $T(x)$ is carried by two or more characteristic lines reaching x at the same time. Similarly, let x_i be a grid node Δx-close to the shock. In order to mimic the continuous case, x_i has to be approached by the *ACC* region approximately at the same time from the directions corresponding to the characteristic lines. In this case, the value $T(x_i)$ is correct (no matter which upwind direction is chosen) and, more important, the characteristic information stops at x_i and it is no longer propagated, getting stuck by the *ACC* region. As a consequence, the shock is localized properly.

We remark that FMM is able to handle shocks when applied to ISO & ¬REG equations. Indeed, thanks to the minimum-value rule, the evolving region *CONS* is, every time, a good approximation of the level sets of the final solution, and shocks are reached by the *ACC* region approximately at the same step of the algorithm.

(4) ¬ISO case requires *CONS* not to be an approximation of the level sets of the solution. In order to solve correctly ¬ISO equations, *CONS* cannot be at any time an approximation of the level sets of the solution. This is due to the fact that the anisotropy shifts the characteristic directions, so that they no longer coincide with the gradient directions. *CONS*, to be correctly enlarged, should not follow the gradient direction, and then it no longer coincides with the level sets. In section 5 we show an example for (1.3). See also [27] for a more detailed explanation.

(5) Can local single-pass methods solve general HJB equations? Let us consider the ¬ISO & ¬REG case. By comments (3)–(4), in order to solve ¬ISO equations, the *CONS* region cannot be an approximation of a level set of the solution. But, doing so, a node x_i close to a shock can be reached by *ACC* at different times. When *ACC* reaches x_i for the first time, it is impossible to detect the presence of the shock by using only local information. Indeed, only a global view of the solution allows one to know that another characteristic line will reach x_i at a later time. As a consequence, the algorithm continues the enlargement of *CONS* and *ACC*, thus making an error that cannot be redressed in the future. Test 4 in section 5 shows an example in which a shock crosses a region with strong anisotropy. In this situation, it seems impossible to get the correct solution without the addition of nonlocal information regarding the location of the shock, or going back to nodes in *ACC* at a later time.

Table 1 summarizes the comments above. Note that the word "no" in the table should be read as "not in general," since some exceptions are possible. It is plain that SFMM is the most versatile of all the methods (whenever it can be applied), since it joins advantages of both SM and FMM.

5. Numerical tests. The first aim of this section is comparing the two SL schemes described in section 2.1, in order to understand which one has to be preferred for practical implementations of the fast methods described above. In the following

TABLE 1
A bird's-eye view on the applicability of local single-pass methods.

	ISO & REG	ISO & ¬REG	¬ISO & REG	¬ISO & ¬REG
FMM	yes	yes	no	no
SM	yes	no	yes (if SDM works)	no
SFMM	yes	yes	yes (if SDM works)	no

we will denote the two schemes by SL-2p and SL-3p, respectively. The second aim is confirming the theoretical observations in section 4 and investigating the theoretical bounds given by SDM and DM in order to understand how close SFMM is to that limit. This will give an idea about how much room is still present for further improvements in the field of local single-pass methods. In all the tests the solution computed by ITM on an 801^2 grid (with $tol = 10^{-16}$) will be referred to as the "exact" solution and will be denoted by T^{exact}.

In Table 2 we list five reference HJB equations, together with the class they belong to. In all the cases we set $d = 2$, $a = (a_1, a_2) \in B(0,1)$, and $\mathcal{T} = \{(0,0)\}$. Moreover, λ, μ, and ε denote generic positive parameters. Finally we define $m_{\lambda,\mu}(a) = (1 + (\lambda\, a_1 + \mu\, a_2)^2)^{-\frac{1}{2}}$ and denote by χ_S the characteristic function of a set S.

TABLE 2
Equations considered for numerical tests and the class they belong to.

Equation	Dynamics	Class
HJB-A	$f(x,y,a) = a$	ISO & REG
HJB-B	$f(x,y,a) = (1 + \chi_{\{x>1\}})\, a$	ISO & ¬REG
HJB-C	$f(x,y,a) = m_{\lambda,\mu}(a)\, a$	¬ISO & REG
HJB-D	$f(x,y,a) = (m_{\lambda,\mu}(a) + \varepsilon(x-1)\chi_{\{x>1\}})\, a$	¬ISO & ¬REG
HJB-E	$f(x,y,a) = (1 + \lvert x+y\rvert) m_{\lambda,\mu}(a)\, a$	¬ISO & ¬REG

Test 0 (SL-2p vs. SL-3p). In this test we compare the schemes described in section 2.1 by means of FSM in terms of accuracy and number of iterations. We consider equations HJB-A and HJB-D (for $\varepsilon = 0.02$). Relative errors in norms L^1 and L^∞ with respect to the "exact" solution T^{exact} are defined as

$$E_1 := \frac{1}{N}\sum_{i=1}^{N} \frac{|T^{exact}(x_i) - \widehat{T}(x_i)|}{|T^{exact}(x_i)|} \qquad \text{and} \qquad E_\infty := \max_{i=1,\dots,N} \frac{|T^{exact}(x_i) - \widehat{T}(x_i)|}{|T^{exact}(x_i)|}\,.$$

By "sweep" we mean *four* iterations executed in four different directions. When reporting the number of sweeps of FSM, we include the final "stopping" sweep, needed to realize that convergence is reached, namely, the stopping rule (2.3) is satisfied. We choose $tol = 10^{-16}$ (machine precision). Results are reported in Table 3.

We recall that, as discussed in section 2.2, the convergence of FSM is ensured in 1 sweep for equation HJB-A; see [30]. Nevertheless, real algorithms involving double precision computations can require 2 sweeps to reach machine precision. The third sweep reported in Table 3 is the "stopping" sweep.

It is rather clear that SL-3p overcomes SL-2p in terms of both accuracy and number of sweeps. This is likely due to the fact that SL-3p can propagate the characteristic information of the HJB equation along diagonal directions easier than SL-2p.

Dealing instead with fast-marching-like methods, the two schemes show a difference in the order of acceptance of the nodes in *CONS*. In particular, we note that

TABLE 3
Test 0: *SL-2p and SL-3p schemes comparison.*

Equation	Grid	Scheme	E_∞	E_1	# sweeps
HJB-A	101^2	SL-2p	0.130	0.016	3
HJB-A	101^2	SL-3p	0.079	0.009	3
HJB-A	201^2	SL-2p	0.094	0.008	3
HJB-A	201^2	SL-3p	0.058	0.004	3
HJB-A	401^2	SL-2p	0.050	0.003	3
HJB-A	401^2	SL-3p	0.030	0.002	3
HJB-D	101^2	SL-2p	0.888	0.053	8
HJB-D	101^2	SL-3p	0.635	0.029	4
HJB-D	201^2	SL-2p	0.535	0.027	7
HJB-D	201^2	SL-3p	0.405	0.014	4
HJB-D	401^2	SL-2p	0.245	0.010	7
HJB-D	401^2	SL-3p	0.189	0.005	3

TABLE 4
Test 1: *ISO & REG.*

Grid	Method	E_∞	E_1
101^2	FSM	0.079	0.009
101^2	FMM	0.079	0.009
101^2	SM	0.079	0.009
201^2	FSM	0.057	0.004
201^2	FMM	0.057	0.004
201^2	SM	0.057	0.004
401^2	FSM	0.029	0.001
401^2	FMM	0.029	0.001
401^2	SM	0.029	0.001

algorithms based on SL-3p provide a larger number of safe nodes in *CONS*, thus extending the applicability of SM, SFMM, and SDM.

From now on, only the scheme SL-3p will be employed for all the following tests.

Test 1 (ISO & REG). In this test we compare FSM, FMM, and SM against HJB-A. Errors with respect to the "exact" solution T^{exact} are reported in Table 4. The three methods lead to the same error because they compute exactly the same solution. A fortiori, SFMM does as well. This confirms that SM can be applied in ISO & REG cases and that picking the minimum value in *CONS*, as the acceptance rule, is not strictly needed here to compute the correct solution.

Test 2 (ISO & ¬REG). In this test we compare FSM, FMM, SFMM, and SM against HJB-B. Errors with respect to the "exact" solution T^{exact} are reported in Table 5. FSM, FMM, and SFMM lead to the same error because they compute exactly the same solution. Conversely, SM cannot be used here, since it is not able to properly locate the shocks (see Figure 3).

Test 3 (¬ISO & REG). In this test we compare FSM, FMM, and SM against HJB-C. Errors with respect to the "exact" solution T^{exact} are reported in Table 6. FSM and SM lead to the same error because they compute exactly the same solution. A fortiori, SFMM does as well. Conversely, FMM fails (although it is quite robust),

TABLE 5
Test 2: *ISO & ¬REG.*

Grid	Method	E_∞	E_1
101^2	FSM	0.079	0.011
101^2	FMM	0.079	0.011
101^2	SFMM	0.079	0.011
101^2	SM	0.583	0.019
201^2	FSM	0.057	0.006
201^2	FMM	0.057	0.006
201^2	SFMM	0.057	0.006
201^2	SM	0.606	0.014
401^2	FSM	0.029	0.002
401^2	FMM	0.029	0.002
401^2	SFMM	0.029	0.002
401^2	SM	0.603	0.011

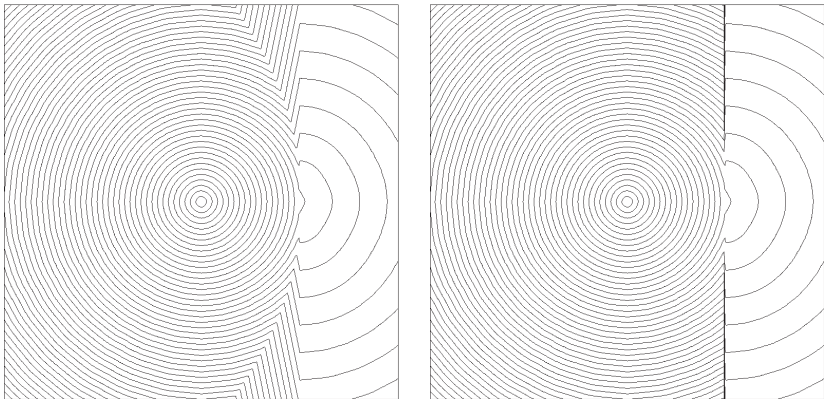

FIG. 3. *Test* 2: *level sets of the solutions computed by FSM and SM.*

TABLE 6
Test 3: *¬ISO & REG.*

Grid	Method	E_∞	E_1
101^2	FSM	0.635	0.029
101^2	FMM	0.635	0.058
101^2	SM	0.635	0.029
201^2	FSM	0.404	0.014
201^2	FMM	0.408	0.049
201^2	SM	0.404	0.014
401^2	FSM	0.189	0.005
401^2	FMM	0.290	0.044
401^2	SM	0.189	0.005

since it does not compute the same solution as FSM. This comes from the fact that FMM is not able to deal with substantial anisotropies, as discussed in section 2.2 (see also [27] for more details).

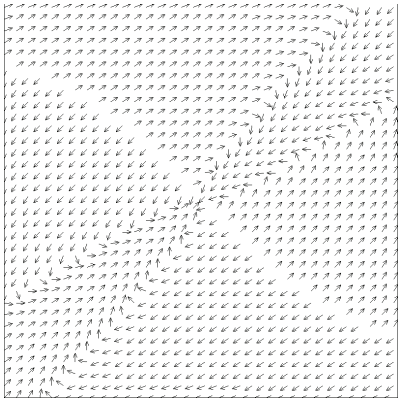

FIG. 4. *Test* 4: *optimal vector field* f^*_{FSM} *computed by FSM (zoom around the origin).*

TABLE 7
Test 4: $\neg ISO$ *&* $\neg REG$.

Grid/method	FSM	FMM	SFMM	SM	DM	SDM
101^2	0.114	0.170	0.115	0.124	0.114	0.114
201^2	0.061	0.132	0.062	0.072	0.061	0.061
401^2	0.024	0.109	0.025	0.036	0.024	0.024

Test 4 (¬ISO & ¬REG). In this test we compare FSM, FMM, SFMM, SM, DM, and SDM against HJB-E (for $\lambda = 6$ and $\mu = 5$). Figure 4 shows some optimal directions (characteristic lines) computed by means of FSM. Both the strong inhomogeneity (characteristic lines hardly bend in the I and III quadrants) and the shock (the cubic-like curve in the II and IV quadrants) are visible. Table 7 reports the error E_1 with respect to the "exact" solution T^{exact}. In this case only "dumb" methods (DM and SDM) are able to compute the same solution of FSM, although SFMM is very close to FSM. The differences among the methods are much more evident looking at the level sets of the corresponding solutions, as reported in Figure 5. FSM is able to respect the anisotropy; indeed the level sets of its solution around the origin are ellipses, as expected. Moreover, it properly catches the shock. FMM tries catching the shock, but fails to respect the anisotropy. SM tries respecting the anisotropy, but fails to catch the shock. Finally, SFMM is a kind of mix between FMM and SM.

Test 5 (¬ISO & ¬REG: easy case). In this test we compare FSM and SFMM against HJB-E (for $\lambda = 5$ and $\mu = 5$). Due to the fact that $\lambda = \mu$, the shock has a particular symmetry with respect to the axes. This symmetry makes SFMM work, since *CONS* "luckily" reaches the shock at the same time from both sides (see Figure 6). This example shows that local single-pass schemes can solve ¬ISO & ¬REG equations in some special cases.

Test 6 (¬ISO & ¬REG: hard case). In this test we show that even SDM and DM can fail to compute the correct solution; i.e., it can happen that either there is no safe node in *CONS* and/or there is no exact node in *CONS*. Therefore, the methods stop abruptly before *ACC* covers the whole domain. We consider again the equation HJB-E (for $\lambda = 10$ and $\mu = 5$). This case is even more pathological than that depicted in Figure 4: characteristic lines bend too much compared to the mesh size; i.e., they can significantly change direction within a single cell. We caught the precise moment

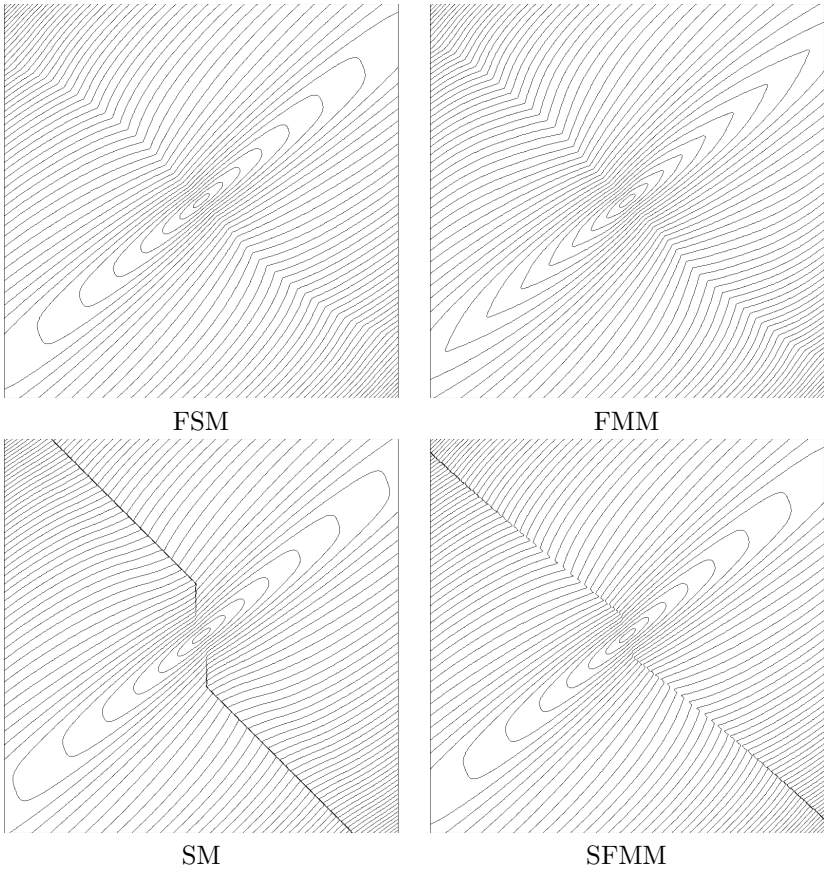

FIG. 5. *Test* 4: *level sets of the solutions computed by FSM, FMM, SM, and SFMM.*

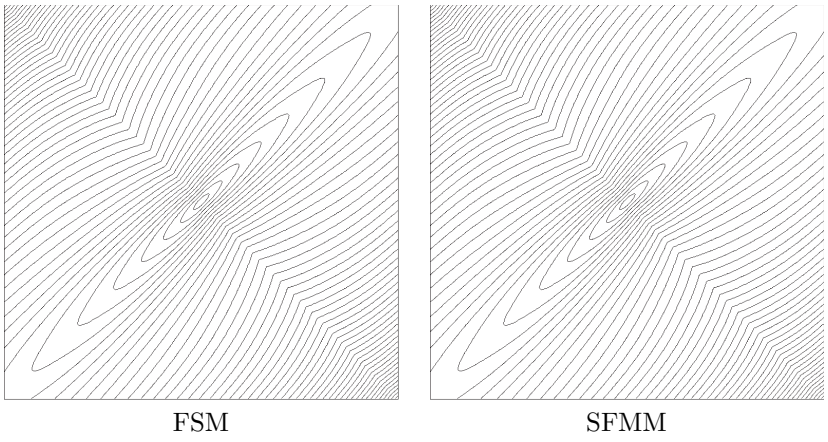

FIG. 6. *Test* 5: *level sets of the solutions computed by FSM and SFMM.*

in which both SDM and DM stop working, due to the lack of safe and exact nodes in *CONS*. In Figure 7 the black central node is the target, gray nodes represent the *ACC* region, while white nodes are in *CONS*. For each node in *CONS* we plot the

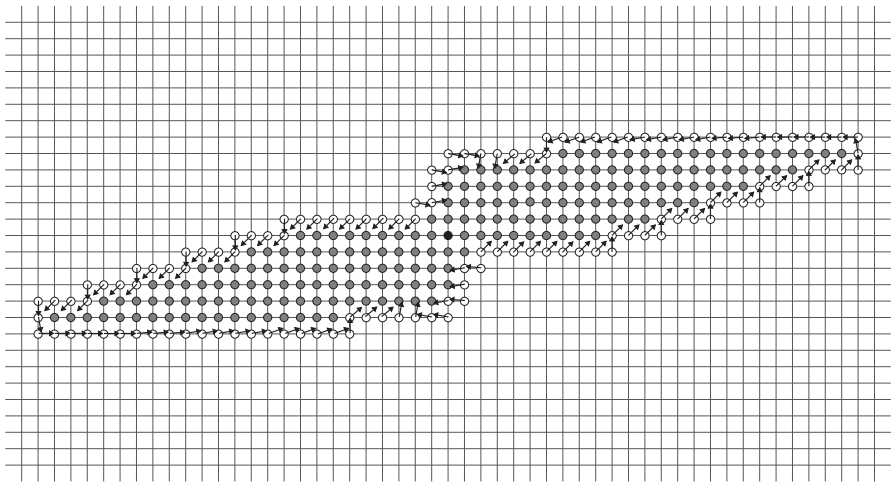

FIG. 7. *Test* 6: *SDM fails because no safe nodes are found in CONS. Note the loop dependency following the optimal vector field* f^*_{SDM}.

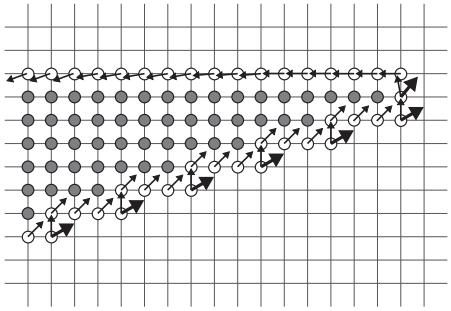

FIG. 8. *Test* 6: *DM fails because no exact nodes are found in CONS. Exact information flows back from the FAR region, breaking locality and the single-pass property.*

optimal vector field f^*_{SDM} computed by means of the current solution of SDM. It is evident that every node in *CONS* depends on other nodes in *CONS*, so that a loop is created and no safe node is present.

In Figure 8 we show a detail of Figure 7 and plot the optimal vector fields f^*_{DM} (small arrows) and f^*_{ITM} (large arrows), computed by means of the solution of DM and ITM, respectively (if the two optimal vector fields coincide, only one is plotted).

It can be seen that f^*_{ITM} points either toward the *FAR* region or toward nodes in *CONS*, whose f^*_{ITM} also points toward the *FAR* region (recursively). This means that the values at nodes in *ACC* and *CONS* are not enough to compute exact values in *CONS*, even if we perform an additional stabilization by iterating the scheme on *CONS* up to convergence. We infer that we are facing a large loop in the numerical domain of dependence of the nodes in *CONS*, which includes also nodes currently in *FAR*. This particular behavior of the characteristic flow is also confirmed by the fact that ITM requires in this case a huge number of iterations to reach convergence compared to that of the previous tests. The thickness of *CONS* must be increased (as in BFMM) in order to resolve the dependency.

6. Conclusions. The above tests and considerations allow us to sketch some final comments and suggest new directions for a future analysis of acceleration methods. First, we want to stress that all the considerations debated in the paper have a theoretical value. Indeed, from the practical point of view, it is not always possible to know in advance if an equation falls in the class ISO or REG, and thus it is not evident how to choose a method which is able to solve it. Only ITM and FSM can be safely used if no a priori knowledge of the solution is available.

1. SM is one of the simplest methods one can imagine; nevertheless it is able to solve a large class of equations, including the homogeneous anisotropic eikonal equation (1.3). Therefore, methods such as OUM and PFMM are in some sense "more complicated" than necessary. In our opinion, the reason the minimum-value rule has been given a crucial role so far is simply that the Dijkstra method uses it. Nevertheless, on graphs, the distinction between REG and ¬REG is not visible, nor is the condition of safeness. The importance of the latter condition was missed because the different possibilities in which *CONS* nodes can be used have been completely underestimated (see Remark 2.1). In this respect, we point out that SM is very similar to CFMM since the acceptance rule used in CFMM actually coincides with that of SM (see Definition 3.1). In [10] it was already noted that, running CFMM, *CONS* does not coincide with the level set of the solution, but this fact was not as fully exploited as in this paper.

2. The reliability of the SFMM to solve ¬ISO & ¬REG equations cannot be known in advance. Even if the method computes a solution, i.e., ACC covers the whole grid, we do not know if that solution is correct or not. On the contrary, if the method stops, due to the lack of safe nodes in *CONS*, the user can eventually conclude that this method cannot be used for the equation under consideration.

3. Our experience suggests that there is not much room between SFMM and SDM, meaning that it is quite difficult to define precisely a class of equations that can be solved by SDM and not by SFMM. Since we have seen that SDM is a sort of limit of applicability for local single-pass schemes, we conclude that it is relatively fruitless to investigate new local single-pass schemes. In order to solve more general equations one should look for new methods based on larger and dynamic stencils that will likely produce more complicated implementations.

Acknowledgments. The authors are grateful to Alexander Vladimirsky and Michael Breuß for the interesting and motivating discussions they had with them during the preparation of this paper.

REFERENCES

[1] K. Alton and I. M. Mitchell, *An ordered upwind method with precomputed stencil and monotone node acceptance for solving static convex Hamilton-Jacobi equations*, J. Sci. Comput., 51 (2012), pp. 313–348.

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

[3] M. Boué and P. Dupuis, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, SIAM J. Numer. Anal., 36 (1999), pp. 667–695.

[4] S. Cacace, E. Cristiani, and M. Falcone, *A local ordered upwind method for Hamilton-Jacobi and Isaacs equations*, in Proceedings of the 18th IFAC World Congress, 2011, pp. 6800–6805.

[5] E. Carlini, M. Falcone, N. Forcadel, and R. Monneau, *Convergence of a generalized fast-marching method for an eikonal equation with a velocity-changing sign*, SIAM J. Numer. Anal., 46 (2008), pp. 2920–2952.

[6] A. Chacon and A. Vladimirsky, *Fast two-scale methods for eikonal equations*, SIAM J. Sci. Comput., 34 (2012), pp. A547–A578.
[7] D. L. Chopp, *Some improvements of the fast marching method*, SIAM J. Sci. Comput., 23 (2001), pp. 230–244.
[8] E. Cristiani, *A fast marching method for Hamilton-Jacobi equations modeling monotone front propagations*, J. Sci. Comput., 39 (2009), pp. 189–205.
[9] E. Cristiani and M. Falcone, *A fast marching method for pursuit-evasion games*, in Communications to SIMAI Congress, Vol. 1, L. Puccio, ed., 2006.
[10] E. Cristiani and M. Falcone, *A characteristics driven fast marching method for the eikonal equation*, in Numerical Mathematics and Advanced Applications (ENUMATH 2007, Graz, Austria, 2007), K. Kunisch, G. Of, and O. Steinbach, eds., Springer, Berlin, Heidelberg, 2008, pp. 695–702.
[11] E. Cristiani and M. Falcone, *Fast semi-Lagrangian schemes for the Eikonal equation and applications*, SIAM J. Numer. Anal., 45 (2007), pp. 1979–2011.
[12] P. Danielsson, *Euclidean distance mapping*, Computer Graphics Image Process., 14 (1980), pp. 227–248.
[13] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math., 1 (1959), pp. 269–271.
[14] M. Falcone, *A numerical approach to the infinite horizon problem of deterministic control theory*, Appl. Math. Optim., 15 (1987), pp. 1–13; corrigenda: Appl. Math. Optim., 23 (1991), pp. 213–214.
[15] 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 Verlag, Berlin, 1994.
[16] 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.
[17] R. Gonzalez and E. Rofman, *On deterministic control problems: An approximation procedure for the optimal cost* I. *The stationary problem*, SIAM J. Control Optim., 23 (1985), pp. 242–266.
[18] J. J. Helmsen, E. G. Puckett, P. Colella, and M. Dorr, *Two new methods for simulating photolithography development in* 3*D*, in Optical Microlithography IX, Proc. SPIE 2726, 1996, pp. 253–261.
[19] W.-K. Jeong and R. T. Whitaker, *A fast iterative method for eikonal equations*, SIAM J. Sci. Comput., 30 (2008), pp. 2512–2534.
[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] S. Kim, *An* $\mathcal{O}(N)$ *level set method for eikonal equations*, SIAM J. Sci. Comput., 22 (2001), pp. 2178–2193.
[22] H. J. Kushner, *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations*, Academic Press, New York, 1977.
[23] W. M. McEneaney, *Max-Plus Methods for Nonlinear Control and Estimation*, Birkhäuser Systems and Control Series, 2006.
[24] E. Prados and S. Soatto, *Fast marching method for generic shape from shading*, in Variational, Geometric, and Level Set Methods in Computer Vision (VLSM 2005), N. Paragios et al., eds., Lecture Notes in Comput. Sci. 3752, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 320–331.
[25] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM J. Numer. Anal., 29 (1992), pp. 867–884.
[26] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595.
[27] 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.
[28] 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.
[29] J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528–1538.
[30] H.-K. Zhao, *A fast sweeping method for eikonal equations*, Math. Comp., 74 (2005), pp. 603–627.
[31] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang, *Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method*, Comput. Vis. Image Und., 80 (2000), pp. 295–314.