

# A Low Complexity Algorithm for Non-Monotonically Evolving Fronts

Alexandra Tcheng<sup>1</sup> · Jean-Christophe Nave<sup>1</sup>

Received: 27 September 2015 / Revised: 15 May 2016 / Accepted: 25 May 2016  
© Springer Science+Business Media New York 2016

**Abstract** A new algorithm is proposed to describe the propagation of fronts advected in the normal direction with prescribed speed function  $F$ . The assumptions on  $F$  are that it does not depend on the front itself, but can depend on space and time. Moreover, it can vanish and change sign. The novelty of our method is that its overall computational complexity is predicted to be comparable to that of the Fast Marching Method (Sethian in Proceedings of the National Academy Sciences 93:1591–1595, 1996); (Vladimirsky in Interfaces Free Bound 8(3):281–300, 2006) in most instances. This latter algorithm is  $\mathcal{O}(N^n \log N^n)$  if the computational domain comprises  $N^n$  points. We use it in regions where the speed is bounded away from zero—and switch to a different formalism when  $F \approx 0$ . To this end, a collection of so-called *sideways* partial differential equations is introduced. Their solutions locally describe the evolving front and depend on both space and time. The well-posedness and geometric properties of those equations are addressed. We then propose a convergent and stable discretization of the PDEs. The resulting algorithm is presented together with a thorough discussion of its features. The accuracy of the scheme is tested when  $F$  depends on both space and time. Each example yields an  $\mathcal{O}(1/N)$  global truncation error. We conclude with a discussion of the advantages and limitations of our method.

**Keywords** Front propagation · Hamilton-Jacobi equations · Fast marching method · Level-set method · Viscosity solutions

---

The first author acknowledges the support of the Schulich Graduate Fellowship. This research was partly supported through the NSERC Discovery and Discovery Accelerator Supplement grants of the second author.

---

✉ Alexandra Tcheng  
alexandra.tcheng@mail.mcgill.ca

Jean-Christophe Nave  
jcnav@math.mcgill.ca

<sup>1</sup> Department of Mathematics and Statistics, McGill University, 805 Sherbrooke St. W, Montreal, QC H3A 0B9, Canada

## 1 Introduction

The design of robust numerical schemes describing front propagation has been a subject of active research for several decades. The need for such schemes is felt across many areas of applied sciences: geometric optics [32], optimal control [19, 50], lithography [2], shape recognition [24, 27], dendritic growth [21, 42], gas and fluid dynamics [28, 30, 49], combustion [56], etc. Depending on the problem at hand, various issues may arise. Consider the following two interface propagation phenomena: a fire propagating through a forest, and a large evolving population of bacteria in a Petri dish. In either case, space can be divided into distinct regions: burnt versus unburnt, and populated versus unpopulated. The boundaries between those regions form fronts that evolve in time. Those examples differ from one another in that a fire front can only propagate *monotonically*, whereas bacteria may advance or recede within their environment. This distinction led to different modelling approaches. Monotone propagation can be recast into a ‘static’ problem, as opposed to non-monotone evolution, which is intrinsically time-dependent. As a result, efficient single-pass algorithms for monotone propagation have been developed. In contrast, accurate algorithms for non-monotonically evolving fronts require a larger number of computations. In this paper, we propose a model that reconciles the advantages of previous methods—We accurately describe non-monotone front evolution with an algorithm that performs a low number of operations.

The Level-Set Method (LSM) developed by Osher and Sethian [34] accurately propagates fronts. This implicit approach embeds the front as the zero-level-set of an auxiliary function  $\phi$ . E.g.,  $\phi$  is negative in regions occupied by bacteria, and positive in other regions. Each contour of  $\phi$  is then evolved under the speed function  $F$  which may vanish and change sign. The first order discretization of this problem is both simple and robust. However, describing the evolution of an  $(n - 1)$ -dimensional front in  $\mathbb{R}^n$  requires solving for a function of  $n + 1$  variables, since  $\phi$  depends on space as well as time. Moreover, for accuracy reasons it is often desirable in applications to enforce the signed distance property  $|\nabla\phi| \approx 1$  in a neighborhood of the front. There exists a vast literature on lowering the computational complexity of the LSM [1, 33, 36, 41], and on maintaining the accuracy of the solution [9, 10, 36, 37, 48], but those features are incorporated at the expense of simplicity and efficiency.

The Fast Marching Method (FMM) developed by Sethian [39] and Tsitsiklis [52] requires the speed function to be independent of time, and bounded away from zero. The FMM builds the ‘first arrival time’ function  $\psi$  such that to every point  $\mathbf{x}$  in space is associated the value  $t = \psi(\mathbf{x})$  at which the front reaches  $\mathbf{x}$  [38–41] e.g.,  $\psi$  records the time at which the parcel of land burnt. The use of a Dijkstra-like data structure [17] renders this scheme very efficient: When the computational domain comprises  $N^n$  points, it runs in time  $\mathcal{O}(N^n \log N^n)$ . In [7], Falcone et al. proposed a Generalized FMM (GFMM) that accurately propagates fronts subject to a wide range of speed functions, including those that vanish and change sign. However, when  $F$  depends on time, the complexity of the GFMM may revert to that of the LSM.

The present work proposes an algorithm able to handle speed functions that change sign, while retaining the efficiency of the FMM. In general, a point  $\mathbf{x}$  in space may be reached by the front several times, and the arrival time cannot be described as a function of space. However, consider the set  $\mathcal{M} := \{(\mathbf{x}, t) : \mathbf{x} \text{ belongs to the front at time } t\}$ . The set  $\mathcal{M}$  consists of the surface traced out by the fronts as they evolve through space and time. If  $\mathcal{M}$  embeds as a  $C^k$ -manifold of dimension  $n$  in  $\mathbb{R}^n \times (0, T)$ , then each point  $(\mathbf{x}, t) \in \mathcal{M}$

belongs to a neighbourhood that is locally the image of a  $C^k$ -function of  $n$  variables. Under mild assumptions,  $\mathcal{M}$  is a compact subset of  $\mathbb{R}^n \times (0, T)$ , implying that we only need a finite number of functions to parametrize  $\mathcal{M}$ . Those local representations of the set  $\mathcal{M}$  are the images of functions that possibly depend on time as well as space. Our approach makes use of those other representations whenever the purely spatial one is not available—e.g., when  $n = 2$  and  $\mathcal{M}$  cannot be locally described by the standard first arrival time function  $\{t = \psi(x, y)\}$ , we may describe it as  $\{x = \tilde{\psi}(y, t)\}$  or  $\{y = \tilde{\psi}(x, t)\}$ . To this end, we introduce *sideways* PDEs solved by those  $C^k$ -functions. We illustrate how they relate to previous work, argue that they are well-posed, and show that their solution provides a local description of  $\mathcal{M}$ . Moreover, we describe a scheme to discretize them, prove that it converges to the correct viscosity solution, and show that it is stable.

In practice, the algorithm amounts to augmenting the FMM to describe  $\mathcal{M}$  near those points  $(\mathbf{x}, t)$  where  $F(\mathbf{x}, t) = 0$ . The different representations of  $\mathcal{M}$  that result are woven together along their overlapping parts using interpolation. To illustrate the method, examples are presented where an  $\mathcal{O}(1/N)$  global truncation error is achieved. Those tests all feature speed functions that vanish, and possibly depend on time. Since the algorithm always approximates a function of  $n$ -variables, the dimensionality of the problem is never raised, unlike what happens in the LSM. As a result, the computational complexity is expected to be comparable to that of the FMM. Tests reveal that our method has lower complexity than a local LSM.

*Outline of the article* We provide further motivation in Sect. 2. The case where  $|F| \geq \delta > 0$  depends on time is addressed in Sect. 3. The *sideways* PDEs used when  $F \approx 0$  are introduced in Sect. 4. A discussion of their properties is provided along with a convergent and stable scheme to discretize them. Pseudo-codes are given and discussed in Sect. 5. We predict and illustrate the complexity and accuracy of the method in Sects. 6 and 7. The advantages and weaknesses of our approach are discussed in Sect. 8, where an additional example addresses the limitations of the method.

## 2 Preliminaries

### 2.1 Problem Statement and Assumptions

Let a subset  $\mathcal{C}_0 \subset \mathbb{R}^n$  be closed with no boundary. Assume it is an orientable manifold of codimension one, with a well defined unique outer normal  $\hat{\mathbf{n}}_0(\mathbf{x})$ . Suppose  $\mathcal{C}_0$  is advected in time, and denote the resulting subset of  $\mathbb{R}^n$  at time  $t$  by  $\mathcal{C}_t$ . We want to describe  $\mathcal{C}_t$  for  $0 < t < T$  in the case where each point  $\mathbf{x} \in \mathcal{C}_t$  is advected under the velocity

$$\mathbf{v} = \mathbf{v}(\mathbf{x}, t) = F(\mathbf{x}, t)\hat{\mathbf{n}}(\mathbf{x}, t) \quad (1)$$

i.e., with the prescribed speed function  $F = F(\mathbf{x}, t)$ , in the direction of the outward normal to  $\mathcal{C}_t$ ,  $\hat{\mathbf{n}} = \hat{\mathbf{n}}(\mathbf{x}, t)$ .

We assume that the initial set  $\mathcal{C}_0$  is known exactly and is  $C^2$  in the sense that if it is given as the image of a map, e.g.,  $\gamma : S^{n-1} \longrightarrow \mathbb{R}^n$ , then  $\gamma \in C^2(S^{n-1})$ . The speed  $F = F(\mathbf{x}, t)$  is known exactly for all  $(\mathbf{x}, t)$  and does not depend on the curve itself, or any of its derivatives. For simplicity, we also make the following strong assumption: the map  $F : \mathbb{R}^n \times (0, T) \longrightarrow \mathbb{R}$  is analytic. In particular, this implies that the subset defined as  $\mathcal{F} := \{(\mathbf{x}, t) : F(\mathbf{x}, t) = 0\}$  is closed and has codimension one in  $\mathbb{R}^n \times [0, T]$ . Together, those assumptions guarantee that for any given  $t \in (0, T)$ , there exists a well defined normal  $\hat{\mathbf{n}} = \hat{\mathbf{n}}(\mathbf{x}, t)$  almost everywhere along  $\mathcal{C}_t$ .

## 2.2 Applications

An important application of the problem just described is robot-path planning. In [25] for example, the authors use the LSM to compute the time-optimal paths of ocean vehicles in dynamic continuous flows. In this context, the speed  $F$  is the sum of the speed of the vehicle and a time-dependent external velocity field modelling the ocean currents. An additional linear term may be considered to account for motions that are tangent to the external velocity field. See also [29].

The problem we are concerned with may also be regarded as a toy model. This is the approach taken by Carlini et al. in [6] which presents an early version of the GFMM. Their work is motivated by its applicability to dislocation dynamics where  $F$  is the convolution of a function of space with the level-set function. Going further along this line of thoughts, one may consider applying our toy model to a variety of problems, such as mean curvature flow or fluid dynamics. In [20], the authors apply the GFMM to image segmentation.

## 2.3 Previous Work

*The Level-Set Method* This approach embeds the curve  $\mathcal{C}_t$  as the zero-level-set of a function  $\phi : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}$ , i.e.,  $\mathcal{C}_t = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}$ . The level-set function  $\phi$  is shown to solve the following Initial Value Problem (IVP):

$$\begin{cases} \phi_t + F|\nabla\phi| = 0 & \text{on } \mathbb{R}^n \times (0, T) \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) & \text{on } \mathbb{R}^n \times \{0\} \end{cases} \quad (2)$$

where  $\phi_0(\mathbf{x})$  is such that  $\{\mathbf{x} : \phi_0(\mathbf{x}) = 0\} = \mathcal{C}_0$ . One of the most prominent feature of the LSM is that topological changes are accurately handled, and do not require special treatment.

*The Fast Marching Method* The FMM requires that  $F = F(\mathbf{x}) \geq \delta > 0$  on  $\mathbb{R}^n$ . Suppose the set  $\mathbb{R}^n \setminus \mathcal{C}_t$  consists of two connected components, and define  $\mathcal{A}_t$  to be the bounded one. The FMM solves the Eikonal equation

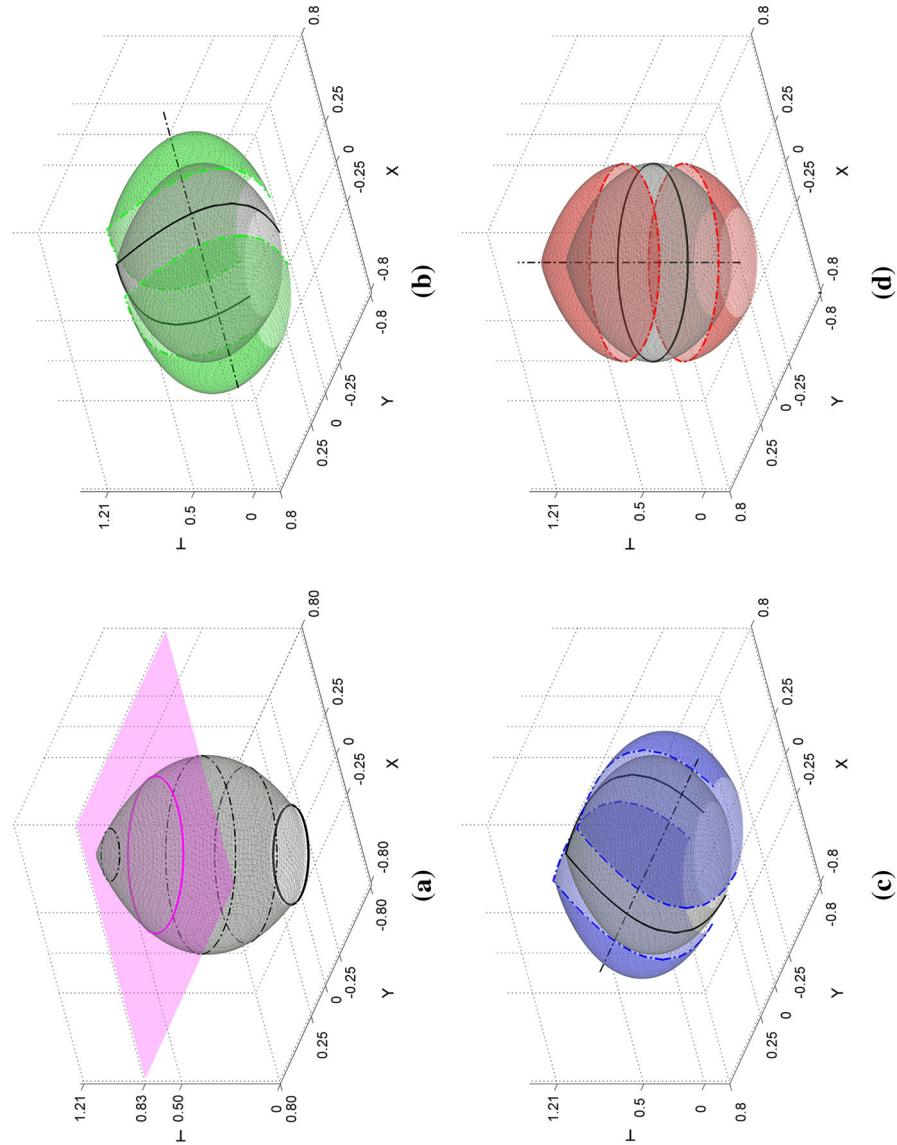
$$\begin{cases} |\nabla\psi| = \frac{1}{F} & \text{on } \mathcal{A}_0^c \setminus \mathcal{C}_0 \\ \psi(\mathbf{x}) = 0 & \text{on } \mathcal{C}_0 \end{cases} \quad (3)$$

whose unknown is the time  $\psi : \mathbb{R}^n \mapsto \mathbb{R}$  at which each point is reached by the curve. The FMM makes use of a Narrow Band to advance the front in a manner that enforces the characteristic structure of the PDE into the solution—See [19, 38–41]. Recent improvements include on the one hand the work of Zhao [54], who further lowered the complexity of the algorithm to develop the Fast Sweeping Method. On the other hand, Vladimirsky allowed the speed to be time-dependent. We discuss this latter method in Sect. 3.

## 2.4 Motivation

We present a simple example to motivate the need for an augmented FMM. Let the initial curve  $\mathcal{C}_0$  be a circle of radius  $r_0$  centered at the origin of  $\mathbb{R}^2$ , and the time-dependent speed be  $F(t) = 1 - ct$  where  $c > 0$ . The evolution of the curve occurs in two phases: (1) For  $t \in [0, \frac{1}{c}]$ , the circle expands until it reaches a maximal radius  $R$ . (2) For  $t \in (\frac{1}{c}, T]$ , the circle contracts until it collapses to the point  $(0, 0)$  at time  $T$ .

Consider the following atlas  $\mathcal{A}$  to describe the resulting  $C^0$ -manifold  $\mathcal{M}$  featured on Fig. 1. Define  $\mathcal{U} := \mathbb{R} \times [0, T]$ . Then  $\mathcal{A} = \cup_{i=1}^3 \{(\psi_{i,\pm}, \mathcal{W}_{i,\pm})\}$  where the real-valued functions  $\psi_{i,\pm}$  are defined as:



**Fig. 1** Chart decomposition of  $\mathcal{M}$  when  $c = 2$  and  $r_0 = 0.25$ . **a**  $\mathcal{M}$  and  $\mathcal{C}_{0.83}$ . **b**  $\mathcal{M}$ , along with  $\mathcal{V}_{1,-}$  and  $\mathcal{V}_{1,+}$  in green. **c**  $\mathcal{M}$ , along with  $\mathcal{V}_{2,-}$  and  $\mathcal{V}_{2,+}$  in blue. **d**  $\mathcal{M}$ , along with  $\mathcal{V}_{3,-}$  and  $\mathcal{V}_{3,+}$  in red (Color figure online)

$$\begin{aligned}\psi_{1,-} : \mathcal{U} &\longrightarrow [-R, 0] & \psi_{1,+} : \mathcal{U} &\longrightarrow [0, R] \\ \psi_{2,-} : \mathcal{U} &\longrightarrow [-R, 0] & \psi_{2,+} : \mathcal{U} &\longrightarrow [0, R] \\ \psi_{3,-} : \mathbb{R}^2 &\longrightarrow [0, \frac{1}{c}) & \psi_{3,+} : \mathbb{R}^2 &\longrightarrow (\frac{1}{c}, T]\end{aligned}\quad (4)$$

and

$$\psi_{1,\pm}(y, t) = \pm\sqrt{(r_0 - c t^2/2 + t)^2 - y^2} \quad (5)$$

$$\psi_{2,\pm}(x, t) = \pm\sqrt{(r_0 - c t^2/2 + t)^2 - x^2} \quad (6)$$

$$\psi_{3,\pm}(x, y) = \frac{1}{c} \left( 1 \pm \sqrt{1 - 2c \left( \sqrt{x^2 + y^2} - r_0 \right)} \right) \quad (7)$$

We also define the sets  $\mathcal{W}_{i,\pm}$  as the real part of the image of the functions  $\psi_{i,\pm}$ . The functions  $\psi_{3,\pm}$  can be verified to be the unique classical solutions to:

$$\begin{cases} |\nabla \psi_{3,-}(\mathbf{x})| = \frac{1}{F(\psi_{3,-}(\mathbf{x}))} & \text{on } \mathcal{U}_{3,-} \\ \psi_{3,-}(\mathbf{x}) = 0 & \text{on } \mathcal{C}_0 \end{cases} \quad (8)$$

$$\begin{cases} |\nabla \psi_{3,+}(\mathbf{x})| = -\frac{1}{F(\psi_{3,+}(\mathbf{x}))} & \text{on } \mathcal{U}_{3,+} \\ \psi_{3,+}(\mathbf{x}) = \frac{1}{c} & \text{on } \mathcal{C}_{1/c} \end{cases} \quad (9)$$

where  $\mathcal{U}_{3,-} = \{\mathbf{x} : r_0 < r(\mathbf{x}) < R\}$  and  $\mathcal{U}_{3,+} = \{\mathbf{x} : 0 \leq r(\mathbf{x}) < R\}$ . Together, the graphs of  $\psi_{3,-}$  and  $\psi_{3,+}$  describe all of  $\mathcal{M}$  but the circle of radius  $R$  reached at time  $t = \frac{1}{c}$ . On the other hand this circle lies in the union of the images of  $\psi_{1,\pm}$  and  $\psi_{2,\pm}$ . The functions  $\psi_{1,\pm}$  are the unique classical solutions to

$$\begin{cases} \mp(\psi_{1,\pm})_t + F(t)\sqrt{1 + (\psi_{1,\pm})_y^2} = 0 & \text{on } \mathbb{R} \times (0, T] \\ \psi_{1,\pm}(y, 0) = \pm\sqrt{r_0^2 - y^2} & \text{on } \mathbb{R} \times \{0\} \end{cases} \quad (10)$$

and similarly for  $\psi_{2,\pm}$ . This suggests the following procedure to build  $\mathcal{M}$ :

- (1) Solve for  $\psi_{3,-}$ .
- (Intermediate step) Solve for  $\psi_{1,\pm}$  and  $\psi_{2,\pm}$  restricted to  $[-R, R] \times [\frac{1}{c} - \epsilon, \frac{1}{c} + \epsilon]$  for some  $\epsilon > 0$ .

(2) Solve for  $\psi_{3,+}$ .

Some questions immediately come to mind. Criteria to decide when to move from (1) to the intermediate step must be chosen. Knowing which equation to solve within the intermediate step is also a concern. The practical aspects of how to reconcile the results of those steps need to be addressed carefully. We discuss all of these issues, and turn the above formal idea into an efficient algorithm that constructs  $\mathcal{M}$ .

## 2.5 Notation

To lighten the notation, we will now work in the setting where  $n = 2$ . All the results discussed extend to arbitrary  $n$ .

*Continuous setting* The letter  $\psi$  denotes functions whose image locally describes  $\mathcal{M}$ . Suppose  $\psi : \mathcal{U} \mapsto \mathbb{R}$  with  $\psi : (y, t) \mapsto \psi(y, t) = x$ . We define:

$$\Gamma_t := \{(x, y) \in \mathbb{R}^2 : \psi(y, t) = x, (y, t) \in \mathcal{U}\} \quad (11)$$

We distinguish between  $\hat{n}(\mathbf{x}, t)$  the two-dimensional outward normal to  $\mathcal{C}_t$  at  $\mathbf{x}$ ; and  $\hat{\mathbf{v}}(\mathbf{x}, t)$  the three-dimensional outward normal to  $\mathcal{M}$  at  $(\mathbf{x}, t)$ .

*Discrete setting* The spatial grids have fixed meshsize  $\Delta x = \Delta y =: h$ . We use

$$x_i = i \cdot h \quad y_j = j \cdot h \quad t^k = k \cdot \Delta t \quad (i, j, k) \in \mathbb{Z} \times \mathbb{Z} \times \{\mathbb{N} \cup \{0\}\} \quad (12)$$

We make no distinction between the continuous functions  $\psi$  and their discrete approximations, except in Sect. 4. Indices are used consistently, so that  $\psi_{ij}$  can be understood as  $\psi(x_i, y_j)$  and  $\psi_i^k$  as  $\psi(x_i, t^k)$ . Consequently, a point  $p \in \mathcal{M}$  may be described by one or more of the following three expressions:

$$p_j^k = (\psi_j^k, y_j, t^k) \quad p_i^k = (x_i, \psi_i^k, t^k) \quad p_{ij} = (x_i, y_j, \psi_{ij}) \quad (13)$$

### 3 A FMM for Time-Dependent Speeds: The $t$ -FMM

We first address the problem stated in Sect. 2.1 under the following restriction:

$$F = F(\mathbf{x}, t) \geq \delta > 0 \quad \forall (\mathbf{x}, t) \in \mathbb{R}^2 \times [0, T] \quad (14)$$

Allowing the speed to depend on time yields a non-autonomous control problem studied in [53]. In our context, the main result may be formulated as follows:  $\psi$  satisfies the following Hamilton-Jacobi-Bellman equation:

$$H(\nabla \psi, \psi, \mathbf{x}) := \|\nabla \psi(\mathbf{x})\| F(\mathbf{x}, \psi(\mathbf{x})) = 1 \quad (15)$$

The implementation of the resulting boundary-value problem:

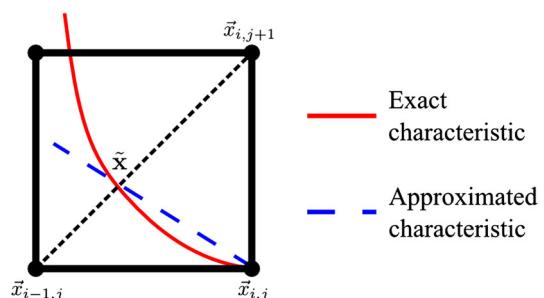
$$\begin{cases} \|\nabla \psi(\mathbf{x})\| = \frac{1}{F(\mathbf{x}, \psi(\mathbf{x}))} \leq \frac{1}{\delta} & \text{on } \mathcal{A}_0^c \setminus \mathcal{C}_0 \\ \psi(\mathbf{x}) = 0 & \text{on } \mathcal{C}_0 \end{cases} \quad (16)$$

is the same as the classical FMM except when a tentative value is assigned to a point in the Narrow Band. Following [53] this step is adjusted as follows. Let  $\mathbf{x}_{ij} = (x_i, y_j)$  and assume that  $\mathbf{x}_{i-1,j}$  and  $\mathbf{x}_{i,j+1}$  are Accepted neighbours of  $\mathbf{x}_{ij}$ —see Fig. 2. Then:  $\tilde{\mathbf{x}} = \xi \mathbf{x}_{i-1,j} + (1 - \xi) \mathbf{x}_{i,j+1}$  for some  $\xi \in [0, 1]$ . Letting  $\mathbf{v} = \mathbf{x}_{ij} - \tilde{\mathbf{x}}$ , we get  $|\mathbf{v}| = \sqrt{\xi^2 + (1 - \xi)^2} h$ . Associate the following value to Quadrant II:

$$\psi_{II} = \min_{\xi \in [0, 1]} \left\{ \psi(\tilde{\mathbf{x}}) + \sqrt{\xi^2 + (1 - \xi)^2} \frac{h}{F(\mathbf{x}_{ij}, \psi(\tilde{\mathbf{x}}))} \right\} \quad (17)$$

Proceeding similarly in the other quadrants yields the values  $\psi_I$ ,  $\psi_{III}$  and  $\psi_{IV}$ . The tentative value assigned to  $\psi_{ij}$  is then  $\psi_{ij} = \min\{\psi_I, \psi_{II}, \psi_{III}, \psi_{IV}\}$ . Note that in two dimensions the minimization problem (17) may be solved using a direct method; see “Appendix 1”.

**Fig. 2** If the characteristic comes from Quadrant II



This method converges to the correct viscosity solution, and is globally first order accurate [43, 44, 53]. Its complexity is  $\mathcal{O}(N^n \log N^n)$ . We will refer to this modified FMM as the ‘ $t$ -FMM’. The results presented in this section yield Algorithm 2 given in Sect. 5. In the general case  $|F| \geq \delta > 0$ , the PDE becomes  $||\nabla \psi(\mathbf{x})|| |F(\mathbf{x}, \psi(\mathbf{x}))| = 1$ .

## 4 The Sideways Representation

An option to study the evolution of propagating curves or surfaces is to represent the front as a function that depends on time, e.g.,  $y = Y(x, t)$  [40]. Although this approach fails to capture the global properties of the front, we use it near regions where  $F$  vanishes.

### 4.1 Smooth Setting

Consider the solution  $\phi$  to the IVP (2). Suppose  $\phi \in C^1(\mathbf{x}_0, t_0)$  and  $\phi(\mathbf{x}_0, t_0) = 0$ . Assume furthermore that  $\phi_x(\mathbf{x}_0, t_0) \neq 0$ , so that the mapping is locally invertible. From the Implicit Function Theorem there exist open neighbourhoods  $(\mathbf{x}_0, t_0) \in \mathcal{V}$  and  $\mathcal{U} \subset \mathbb{R} \times [0, T]$ , as well as a function  $\psi \in C^1(\mathcal{U})$  satisfying

$$\psi : \mathcal{U} \longrightarrow \mathbb{R}, \quad \psi : (y, t) \mapsto x = \psi(y, t), \quad (\psi(y, t), y, t) \in \mathcal{V} \quad (18)$$

and  $\phi(\psi(y, t), y, t) = 0 \forall (y, t) \in \mathcal{U}$ . Taking full derivatives of  $\phi$  with respect to  $y$  and  $t$ , and using the fact that in  $\mathcal{V}$ ,  $\phi$  satisfies the LSE pointwise gives:

$$(-\phi_x \psi_t) + F \sqrt{\phi_x^2 + (-\phi_x \psi_y)^2} = 0 \iff -\psi_t \pm F \sqrt{1 + \psi_y^2} = 0 \quad (19)$$

where  $\phi_x$  and  $F$  are evaluated at  $(x, y, t) = (\psi(y, t), y, t)$ . The sign used in the last equation depends on  $\phi_x = \pm \sqrt{\phi_x^2}$ . We let  $a := -\text{sign}(\phi_x(\mathbf{x}_0, t_0))$ .

Now, let  $\psi$  satisfy the following IVP:

$$\begin{cases} \psi_t + aF(\psi, y, t)\sqrt{1 + \psi_y^2} = 0 & \text{on } \mathcal{U} \cap (\mathbb{R} \times (t_0, T)) \\ \psi(y, t_0) = \psi_0(y) & \text{on } \mathcal{U} \cap (\mathbb{R} \times \{t_0\}) \end{cases} \quad (20)$$

where  $\psi_0$  is chosen such that  $\phi(\psi_0(y), y, t_0) = 0$ . Then for all  $t \in (t_0, T)$  the set  $\Gamma_t$  locally describes the curve at time  $t$ , i.e.,  $\Gamma_t = \mathcal{C}_t \cap \mathcal{V}$ . We now investigate the case where  $\mathcal{M}$  is merely  $C^0$ . For simplicity, we work with  $t_0 = 0$ .

### 4.2 Vanishing Viscosity Setting

Equation (20) is a Cauchy problem of the form

$$\begin{cases} \psi_t + H(y, t, \psi, \psi_y) = 0 & \text{on } \mathcal{U} \cap (\mathbb{R} \times (0, T)) \\ \psi(y, 0) = \psi_0(y) & \text{on } \mathcal{U} \cap (\mathbb{R} \times \{0\}) \end{cases} \quad (21)$$

where the Hamiltonian  $H : \mathbb{R}^1 \times (0, T) \times \mathbb{R} \times \mathbb{R}^1 \rightarrow \mathbb{R}$  is defined as  $H(y, t, \psi, \psi_y) = aF(\psi, y, t)\sqrt{1 + \psi_y^2}$ . The function  $\psi_0$  is defined such that for all  $y \in \mathcal{U} \cap (\mathbb{R} \times \{0\})$  we have  $(\psi_0(y), y) \in \mathcal{C}_0$ . We resort to the rich theory of viscosity solutions of Hamilton-Jacobi equations to study various properties of this problem [4, 5, 12–14, 18, 23, 46, 47]. We first address the well-posedness of (21).

**Theorem 1** (Existence and Uniqueness) *There exists a unique viscosity solution  $\psi$  to problem (21).*

*Proof* The assumptions on  $H$  required to apply Theorem 1.1 in [45] can be verified to hold in our context, with the exception of (H3) in [45]. However, it may be modified to get  $\gamma_{R,P} \in \mathbb{R}$  if  $p \in B_N(0, P)$  for some  $P > 0$ .  $\square$

Next, we verify that (21) enjoys the geometric property advertised in Sect. 4.1.

**Theorem 2** ( $\Gamma_t$  locally describes  $\mathcal{C}_t$ ) *The set  $\Gamma_t$  satisfies  $\Gamma_t = \mathcal{C}_t \cap \mathcal{V}$ .*

*Proof* Consider the IVP (2) again:

$$\begin{cases} \phi_t + F|\nabla\phi| = 0 & \text{on } \mathbb{R}^2 \times (0, T) \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) & \text{on } \mathbb{R}^2 \times \{0\} \end{cases} \quad (22)$$

Since it is known that  $\mathbf{x} \in \mathcal{C}_t \cap \mathcal{V}$  if and only if  $\phi(\mathbf{x}, t) = 0$ , we may prove the theorem by showing that:  $\mathbf{x} \in \Gamma_t$  if and only if  $\phi(\mathbf{x}, t) = 0$ .

→ We argue by contradiction. Suppose the set  $\mathcal{T} = \{T > t > 0 : \exists \mathbf{x} \in \Gamma_t \text{ s.t. } \phi(\mathbf{x}, t) \neq 0\}$  is not empty and define  $t^* = \inf \mathcal{T}$ . Since  $\phi$  is continuous,  $\mathcal{T}$  is open and  $t^* \notin \mathcal{T}$ . Therefore, for all  $\mathbf{x}^* \in \Gamma_{t^*}$ ,  $\phi(\mathbf{x}^*, t^*) = 0$ , but for any  $\epsilon > 0$  sufficiently small, there exists  $\mathbf{x}_\epsilon \in \Gamma_{t^*+\epsilon}$  such that  $\phi(\mathbf{x}_\epsilon, t^* + \epsilon) \neq 0$ . If  $\mathcal{M}$  is differentiable at  $(\mathbf{x}^*, t^*)$ , this contradicts the argument presented in Sect. 4.1: The Implicit Function Theorem guarantees that the set  $\mathcal{V}$  is open. If  $\mathcal{M}$  is not differentiable at  $(\mathbf{x}^*, t^*)$ , then fix  $\epsilon$  and for  $\delta > 0$  consider  $\mathbf{x}^0 \in \Gamma_{t^*+\epsilon}$  such that  $\|\mathbf{x}_\epsilon - \mathbf{x}^0\| \leq \delta$  and  $\mathcal{M}$  is differentiable at  $(\mathbf{x}^0, t^* + \epsilon)$ . For any  $\delta$ , such a point can be found since for any  $T > t^* + \epsilon > 0$  the singularities of  $\Gamma_{t^*+\epsilon}$  are subsets of measure 0.<sup>1</sup> Again, the Implicit Function Theorem guarantees that there is a neighbourhood  $\tilde{\mathcal{V}}$  of  $(\mathbf{x}^0, t^* + \epsilon)$  where  $\phi(\mathbf{x}, t) = 0$  for any  $\mathbf{x} \in \Gamma_t \cap \tilde{\mathcal{V}}$ . Considering the sequence  $\delta_n = \{\frac{1}{n} : n \in \mathbb{N}\}$  and the corresponding sequence  $\{\mathbf{x}^n\}_{n=1}^\infty$ , we arrive at the conclusion that  $\phi(\mathbf{x}_\epsilon, t^* + \epsilon) \neq 0$  contradicts the continuity of  $\phi$ .

← Assume that there exists  $(\mathbf{x}, t) \in \mathcal{V}$  such that  $\phi(\mathbf{x}, t) = 0$ , but there is no  $y$  such that  $\mathbf{x} = (\psi(y), y) \in \Gamma_t$ . We re-use the arguments given in the proof of →: If  $\mathcal{M}$  is differentiable at  $(\mathbf{x}, t)$  then this contradicts the argument in Sect. 4.1. If  $\mathcal{M}$  is not differentiable at  $(\mathbf{x}, t)$ , then we can find a sequence  $\mathbf{x}^n \in \Gamma_t$  converging to  $\mathbf{x}$  such that  $\phi(\mathbf{x}^n, t) = 0$ , and obtain the contradiction that  $\psi$  is not continuous.  $\square$

Letting  $\phi_y(\mathbf{x}_0, t_0) \neq 0$  in Sect. 4.1 and following a similar reasoning, we get that  $\psi : (x, t) \mapsto y = \psi(x, t)$  with  $(x, \psi(x, t), t) \in \mathcal{V}$  satisfies:

$$\begin{cases} \psi_t + aF(x, \psi, t)\sqrt{\psi_x^2 + 1} = 0 & \text{on } \mathcal{U} \cap (\mathbb{R} \times (0, T)) \\ \psi(x, 0) = \psi_0(x) & \text{on } \mathcal{U} \cap (\mathbb{R} \times \{0\}) \end{cases} \quad (23)$$

in the viscosity sense. In subsequent sections, we will refer to Problems (20) and (23) as the  $yt$ - and  $xt$ -representations of  $\mathcal{M}$ . Those problems provide *sideways* representations of the evolving front.

### 4.3 Discretization

Finite-differences schemes for problems such as (21) have been discussed [15, 16, 45]. Based on these works, we propose the following discretization for Equation (20). In this subsection only, we will distinguish between the continuous function  $\psi$ , and its discrete approximation  $\chi$ . The spatial derivative  $\chi_y$  must be computed in an upwind fashion. We define

<sup>1</sup> This follows directly from the fact that (21) is a first order Hamilton-Jacobi equation.

$$D_l^+ \chi^r := \frac{\chi_{l+1}^r - \chi_l^r}{h} \quad D_l^- \chi^r := \frac{\chi_l^r - \chi_{l-1}^r}{h} \quad (24)$$

and suggest:

$$\chi_l^{r+1} = \chi_l^r - a \cdot \Delta t \cdot F(\chi_l^r, y_l, t^r) \cdot \sqrt{1 + \text{upw}(\chi^r, l, r, \alpha)} \quad (25)$$

where

$$\begin{aligned} \text{upw}(\chi^r, l, n, \alpha) := & \max\{\alpha, 0\} \left( \min\{D_l^+ \chi^r, 0\}^2 + \max\{D_l^- \chi^r, 0\}^2 \right) \\ & - \min\{0, \alpha\} \left( \max\{D_l^+ \chi^r, 0\}^2 + \min\{D_l^- \chi^r, 0\}^2 \right) \end{aligned} \quad (26)$$

The constant  $\alpha$  acts as a switch and is defined as  $\alpha = \text{sign}(aF(\chi_l^r, y_l, t^r))$ .

**Proposition 1** (Convergence.) *Let  $M$  be defined as the local bound on  $F$ , i.e.,  $M_l^r = \sup_{(x, y, t) \in B(p_l^r, 2h)} \{|F(x, y, t)|\}$ , where  $p_l^r = (\chi_l^r, y_l, t^r)$ . Assume that  $\max\{|D_l^+ \chi^r|, |D_l^- \chi^r|\} \leq P$  for all  $l \in L$  and  $0 \leq r \leq R$ . Suppose  $\Delta t$  satisfies*

$$M_l^r \cdot \Delta t \leq \frac{h}{2P} \quad (27)$$

*Then the above scheme is such that  $\chi \rightarrow \psi$  as  $h$  and  $\Delta t \rightarrow 0$ , with rate*

$$\|\chi - \psi\|_\infty \leq c\sqrt{\Delta t} \quad (28)$$

*for all  $l$ , where the constant  $c$  depends on  $\|\psi_0\|$ ,  $\|D\psi_0\|$ , the numerical Hamiltonian  $g$ , and  $R\Delta t$  where  $0 \leq r \leq R$ .*

*Proof* We proceed by showing that the scheme is *monotone* and *consistent* in the sense of [45]. The results then follow from Theorem 3.1 of that same paper. The scheme can be rewritten as

$$\chi_l^{r+1} = \chi_l^r - \Delta t \cdot g(y_l, t^r, \chi_l^r, D_l^+ \chi^r, D_l^- \chi^r) \quad (29)$$

where the numerical Hamiltonian  $g$  is verified to be consistent, i.e.,

$$g(y, t, s, \delta, \delta) = H(y, t, s, \delta) \quad \forall (y, t) \in \mathcal{U}, s \in \mathbb{R}, |\delta| < P \quad (30)$$

except near extrema of  $\chi$ , where we may have  $g(y, t, s, \delta, \delta) = H(y, t, s, 2\delta)$ . However, as argued by Sethian (cf. [34, 39, 40]), the upwind discretization of the gradient still guarantees convergence. We verify monotonicity by showing

$$G(\chi_{l-1}^r, \chi_l^r, \chi_{l+1}^r) = \chi_l^r - a \cdot \Delta t \cdot F(u, y_l, t^r) \cdot \sqrt{1 + \text{upw}(\chi^r, l, r, \alpha)} \quad (31)$$

is a non-decreasing function of each of its argument, for fixed  $u$ ,  $y_l$  and  $t^r$ . We only treat the case  $\alpha > 0$ , since the other case is symmetric. Writing  $F = F(u, y_l, t^r)$  for short gives

$$G(b, c, d) = \begin{cases} c - a\Delta t F \sqrt{1 + \left(\frac{d-c}{h}\right)^2} & \text{if } d - c < 0, c - b < 0 \\ c - a\Delta t F \sqrt{1 + \left(\frac{c-b}{h}\right)^2} & \text{if } d - c > 0, c - b > 0 \\ c - a\Delta t F \sqrt{1 + \left(\frac{d-c}{h}\right)^2 + \left(\frac{c-b}{h}\right)^2} & \text{if } d - c < 0, c - b > 0 \\ c - a\Delta t F & \text{if } d - c > 0, c - b < 0 \end{cases} \quad (32)$$

For the first case:  $G_b, G_d \geq 0$  are trivial to check while  $G_c \geq 0$  only if

$$1 \geq \left( F^2 \left( \frac{\Delta t}{h} \right)^2 - 1 \right) \left( -\frac{d-c}{h} \right)^2 \iff \frac{\sqrt{1+P^2}}{P} \geq M_l^r \frac{\Delta t}{h} \quad (33)$$

Case 2 yields the same condition, whereas Case 3 gives the more restrictive one present in the assumption of the claim. Case 4 is trivial.  $\square$

**Proposition 2** (Stability.) *The above scheme is stable, provided that*

$$\Delta t < \min \left\{ \frac{h}{2PM_l^r}, \frac{P-2}{KP\sqrt{1+2P^2}}, \frac{2}{P\delta} \right\} \quad (34)$$

for some  $\delta > 0$ . The constant  $P$  is such that  $\max \{|D_l^+ \chi^r|, |D_l^- \chi^r|\} \leq P$  for all  $l \in L$  and  $0 \leq r \leq R$ .  $K$  is the Lipschitz constant of  $F$ .

*Proof* Applying Theorem 7 of [31] to our scheme, it is possible to show that for  $h$  small enough, the explicit Euler map defined as

$$S_{\Delta t}^l(\chi) = \chi_l - a\Delta t \cdot F(\chi_l, y_l, t)\sqrt{1 + \text{upw}(\chi^r, l, r, \alpha)} \quad (35)$$

is a strict contraction in  $\ell_\infty$ . Bounding  $S_{\Delta t}^l(\chi) - S_{\Delta t}^l(\tau)$  from below (resp. above) yields the 2nd (resp. 3rd) bound in (34).  $\square$

When defining ‘upw’, we implicitly assumed that both  $\chi_{l+1}^r$  and  $\chi_{l-1}^r$  were known. In the instance where one of those values is not known, we set  $\chi_l^{r+1}$  to  $+\infty$ . Indeed, no value can be assigned to  $\chi_l^{r+1}$  since it is not possible to infer where the characteristic going through the point  $p_l^r = (\chi_l^r, y_l, t^r)$  comes from.

*Remark 1* Assuming  $P = \mathcal{O}(1/h)$ , we revisit the bounds on  $\Delta t$  given in (34). The first one is not very restrictive since  $F \approx 0$  implies that  $M_l^r$  is be small. The others are  $\mathcal{O}(h)$  as is usual for the CFL condition of an advection problem.

## 5 Algorithms and Discussion

The algorithms make use of four lists. The ‘accepted’ list  $\mathcal{A}$  and ‘narrow band’ list  $\mathcal{N}$  are lists of triplets, e.g.,  $p_{ij} = (x_i, y_j, \psi_{ij})$ . The ‘pile’ list  $\mathcal{P}$  and ‘far away’ list  $\mathcal{Fa}$  are lists of coordinates, e.g.,  $(x_i, y_j)$ .

### 5.1 Algorithm 1, Main Loop

The main loop is such that if  $F = F(x, y) \geq \delta > 0, \forall (x, y) \in \mathbb{R}^2$ , it reduces to the classical FMM. Indeed, the sideways formulations are only used when  $F \approx 0$ . Let the point accepted during the first steps of the main loop be labelled as  $p_{\alpha\beta} = (x_\alpha, y_\beta, \psi_{\alpha\beta})$ .

*Update Pile:* Lines 7–17 This is only performed if  $\psi_{\alpha\beta}$  is below a certain predefined time  $T$ , which is in contrast with the standard FMM, where the size of the computational domain determines  $T$ . In order to decide whether a nearest neighbour  $(x_a, y_b)$  of  $p_{\alpha\beta}$  should be put in  $\mathcal{P}$ , three criteria are used: position, status and orientation. To clarify what is meant by line 9, consider the following situation: If  $F(p_{\alpha\beta}) > 0$  (which means the curve is locally expanding) and  $(x_a, y_b)$  lies *inside* the curve  $C_{\psi_{\alpha\beta}}$ , then  $(x_a, y_b)$  is not added to  $\mathcal{P}$  since it is not likely to

**Algorithm 1** Main Loop

---

```

1: while  $\mathcal{N}$  do  $\neq \emptyset$ 

2:   Find the point  $p_{\alpha\beta}$  in  $\mathcal{N}$  with the smallest time value.
3:   Remove  $[ p_{\alpha\beta} ; \hat{v}(p_{\alpha\beta}) ]$  from  $\mathcal{N}$  and add it to  $\mathcal{A}$ .
4:   if  $(x_\alpha, y_\beta) \in \mathcal{F}_a$  then
5:     remove  $(x_\alpha, y_\beta)$  from  $\mathcal{F}_a$ .
6:   end if

7:   if  $\psi_{\alpha\beta} < T$  then
8:     for all neighbours  $(x_a, y_b)$  of  $(x_\alpha, y_\beta)$  do
9:       if  $(x_a, y_b)$  could be traversed by the curve then
10:        if  $(x_a, y_b) \in \mathcal{F}_a$  then
11:          add  $(x_a, y_b)$  to  $\mathcal{P}$ 
12:        else
13:          depending on the orientation of the points in  $\mathcal{A}$  with spatial coord.  $(x_a, y_b)$ , add  $(x_a, y_b)$ 
              to  $\mathcal{P}$ . (See discussion.)
14:        end if
15:      end if
16:    end for
17:  end if

18:  for all  $(x_i, y_j) \in \mathcal{P}$  do
19:    Compute  $\psi_{ij}$  and  $\hat{v}_{ij}$  using Algorithm 2 and let  $p_{ij} \leftarrow (x_i, y_j, \psi_{ij})$ .
20:    Remove  $(x_i, y_j)$  from  $\mathcal{P}$ .
21:    if  $p_{ij}$  fails the Sign Test then
22:      proceed to Algorithm 3.
23:    end if
24:    if  $\psi_{ij} < +\infty$  then
25:      if  $\exists q_{ij} \in \mathcal{N}$  with the same spatial coord. as  $p_{ij}$  then
26:        remove  $[ q_{ij} ; \hat{v}(q_{ij}) ]$  from  $\mathcal{N}$ .
27:      end if
28:      add  $[ p_{ij} ; \hat{v}(p_{ij}) ]$  to  $\mathcal{N}$ 
29:    end if
30:  end for

31: end while

```

---

be traversed by the curve within a short time. Next, if the pair  $(x_a, y_b)$  was traversed by the curve in the past, Line 13 picks, out of all the points in  $\mathcal{A}$  with spatial coordinates  $(x_a, y_b)$ , the point with the *largest* time value (i.e., the one that was most recently traversed by the front), and calls it  $p_{ab}$ . Suppose that  $\hat{v}(p_{ab}) > 0$ , which means  $F(p_{ab}) < 0$  and the front was contracting at  $p_{ab}$ . If  $F(p_{ab}) > 0$ , then  $(x_a, y_b)$  is added to  $\mathcal{P}$ .

*Update the Narrow Band:* Lines 18–30 This procedure assigns tentative values to the points in  $\mathcal{P}$  using either the standard or the  $t$ -FMM. Since those algorithms are only valid in regions where  $|F| \geq \delta > 0$ , they only involve points that lie in one such region. In Lines 4–5 of Algorithm 2: an accepted neighbour is ‘compatible’ if its  $\hat{v}_3$  component has the same sign as  $\hat{v}_3(p_{\alpha\beta})$ . Lines 21–23 represent the main modification to the standard FMM algorithm. The Sign Test (described below) is performed to check if the value returned by Algorithm 2 is valid. If the value is not valid, Algorithm 3 attempts to return a new tentative point using a sideways representation. If Algorithm 3 fails, no new point is added to  $\mathcal{N}$ . Else, the updating procedure of  $\mathcal{N}$  is identical to the standard FMM (Lines 25–28).

*The Sign Test* When  $(x_i, y_j)$  is assigned a value  $\psi_{ij}$  using the ( $t$ -)FMM, the Sign Test is performed as follows. Suppose the point  $p_{i-1,j} = (x_{i-1}, y_j, \psi_{i-1,j})$  was used in the

**Algorithm 2** Solve  $|\nabla\psi| = 1/F$ 


---

```

1: if  $F = F(\mathbf{x})$  then
2:   use the standard FMM procedure.
3: else if  $F = F(t)$  or  $F = F(\mathbf{x}, t)$  then
4:    $u_{\pm} \leftarrow \psi_{i \pm 1, j}$  if defined and compatible, or  $+\infty$  otherwise.
5:    $v_{\pm} \leftarrow \psi_{i, j \pm 1}$  if defined and compatible, or  $+\infty$  otherwise.
6:    $\Theta \leftarrow [0, 0, 0, 0]$ 
7:   for Quadrant = 1...4 do
8:     if Quadrant == 1 then
9:        $\psi_v \leftarrow v_+, \psi_u \leftarrow u_+, \tau_v \leftarrow \frac{h}{|F(x_i, y_{j+1}, \psi_v)|}, \tau_u \leftarrow \frac{h}{|F(x_{i+1}, y_j, \psi_u)|},$ 
10:      end if
11:      (and similarly for other quadrants)
12:      if ( $\psi_v = +\infty$ ) and ( $\psi_u = +\infty$ ) then
13:         $\theta \leftarrow +\infty,$ 
14:      else
15:         $\theta \leftarrow \min_{\xi \in [0, 1]} \{\xi \psi_v + (1 - \xi) \psi_u + \sqrt{\xi^2 + (1 - \xi)^2} (\xi \tau_v + (1 - \xi) \tau_u)\}$ 
16:        (see “Appendix 1” for details)
17:      end if
18:       $\Theta(\text{Quadrant}) \leftarrow \theta,$ 
19:    end for
20:     $\psi_{ij} \leftarrow \min(\Theta)$ 
21:  end if

```

---

computation of  $\psi_{ij}$ . Considering the line in  $xyt$ -space from  $p_{i-1,j}$  to  $p_{ij}$ , we check the number of times  $d$  the speed changes sign along this line. If  $d = 0$ , the algorithm can keep running the  $(t)$ -FMM. If  $d = 1$ ,  $p_{ij}$  fails the Sign Test. If  $d > 1$ , the grid has to be refined. Remark that the parameter  $\delta$  is not actually used to switch representation.

## 5.2 Algorithm 3, Sideways Representation

This algorithm is called by the main loop when the speed  $F$  is close to 0.

In order to work locally, the first step defines a square of side length at most  $2sh$  as the new computational grid. The parameter  $s \in \mathbb{N}$  is chosen as large as possible, but subject to the restriction that the inner product between any two normals  $\hat{n}$  at the accepted points of the subgrid is positive. The representation is chosen based on the length of the components of the normal at  $p_{\alpha\beta}$ , e.g., if  $|\hat{v}_1| > |\hat{v}_2|$  then the  $yt$ -representation is used.

Then data are converted according to the procedure illustrated on Fig. 3, which is now detailed. The algorithm must switch from the  $xy$ - to the  $yt$ -representation; i.e.,  $\mathcal{M}$  is locally sampled by points of the form  $p_{lm} = (x_l, y_m, t)$  (as on Fig. 3a), and the  $yt$ -representation requires points of the form  $p_m^r = (x, y_m, t^r)$ . Points that do not have an orientation compatible with the current representation are discarded, e.g., if  $\hat{v}_1(p_{\alpha\beta}) > 0$ , then all the points with  $\hat{v}_1 \leq 0$  are discarded; see Fig. 3b. Note that  $p_{\alpha\beta}$  is represented *exactly* on both the  $xy$ - and the  $yt$ -grid. One-dimensional interpolation is then used line by line to populate the sideways grid; see Fig. 3c.

The sideways PDE can now be solved. As mentioned in Sect. 4.3, if either  $\psi_{l-1}^{r-1}$  or  $\psi_{l+1}^{r-1}$  are set to  $+\infty$ , then Algorithm 4 sets  $\psi_l^r$  to  $+\infty$ . As depicted on Fig. 3d, this has the effect of shrinking the size of the set where the PDE is solved: At most  $s$  time steps can be taken before all the boundary information available has been used up. The time step is restricted by Proposition 2.

Using one-dimensional interpolation again, a value is assigned to the point  $(x_i, y_j)$  in  $\mathcal{P}$ . This is the situation illustrated on Fig. 3e, f. If  $(x_i, y_j)$  is not traversed, then a value is assigned to  $(x_\alpha, y_\beta)$ —recall that the point that has just been accepted is  $p_{\alpha\beta} = (x_\alpha, y_\beta, \psi_{\alpha\beta})$ . If this

cannot be done either, then this representation failed. The algorithm then attempts using the other representation; e.g., the  $xt$ -representation. If both representations fail, then Algorithm 3 fails entirely. Note that this is expected to happen if  $(x_i, y_j)$  and  $(x_\alpha, y_\beta) \notin \mathcal{C}_t$  for any  $t$  in the interval  $(p_{\alpha\beta}, T)$ . See Example 2 in Sect. 7. In such cases, no point is added to  $\mathcal{N}$ .

*Remark 2* It may not be necessary to solve for  $\psi_l^r$  until all the information has been used up. As soon as a value can be assigned to either  $(x_i, y_j)$  or  $(x_\alpha, y_\beta)$ , the scheme can stop calling Algorithm 4 and switch from sideways back to  $xy$ -representation.

---

**Algorithm 3** Sideways representation

---

- 1: Consider the square of  $2s \times 2s$  points centered at  $(x_i, y_j)$ .
  - 2: Based on  $\hat{v}(p_{\alpha\beta})$  pick a representation. Suppose it is the  $yt$ -representation.
  - 3:  $a \leftarrow -\text{sign}(\hat{v}_1(p_{\alpha\beta}))$
  - 4: Using the procedure illustrated on Fig. 3a–c, populate the sideways grid with boundary data. Those points that cannot be assigned any value get  $+\infty$ .
  - 5: Compute  $\psi_l^r = \psi(x, y_l, t^r)$  using Algorithm 4 and moving forward in time. (Fig. 3d)
  - 6: Using one-dimensional interpolation, verify whether  $(x_i, y_j)$  or  $(x_\alpha, y_\beta)$  has been traversed by the curve.
  - 7: **if**  $(x_i, y_j)$  or  $(x_\alpha, y_\beta)$  is traversed by the curve **then**
  - 8:     the corresponding point and normal  $[p_{ij}; \hat{v}(p_{ij})]$  or  $[p_{\alpha\beta}; \hat{v}(p_{\alpha\beta})]$  is returned.
  - 9: **else**
  - 10:     this sideways representation failed. Go back to line 2 and try  $xt$ -rep.
  - 11: **end if**
  - 12: **if** both sideways representations failed **then**
  - 13:     Algorithm 3 has failed. Return  $[(\infty, \infty, \infty); (\infty, \infty, \infty)]$ .
  - 14: **end if**
- 

---

**Algorithm 4** Solve  $\psi_t + aF(\psi, y, t)\sqrt{1 + \psi_y^2} = 0$ 


---

```

if ( $\psi_{l-1}^{r-1} < +\infty$ ) & ( $\psi_{l+1}^{r-1} < +\infty$ ) then
     $\alpha \leftarrow \text{sign}(aF(\psi_l^{r-1}, y_l, t^{r-1}))$ 
     $\psi_l^r \leftarrow \psi_l^{r-1} - a \cdot \Delta t \cdot F(\psi_l^{r-1}, y_l, t^{r-1}) \cdot \sqrt{1 + \text{upw}(\psi^{r-1}, l, r, \alpha)}$ 
else
     $\psi_l^r \leftarrow +\infty$ 
end if

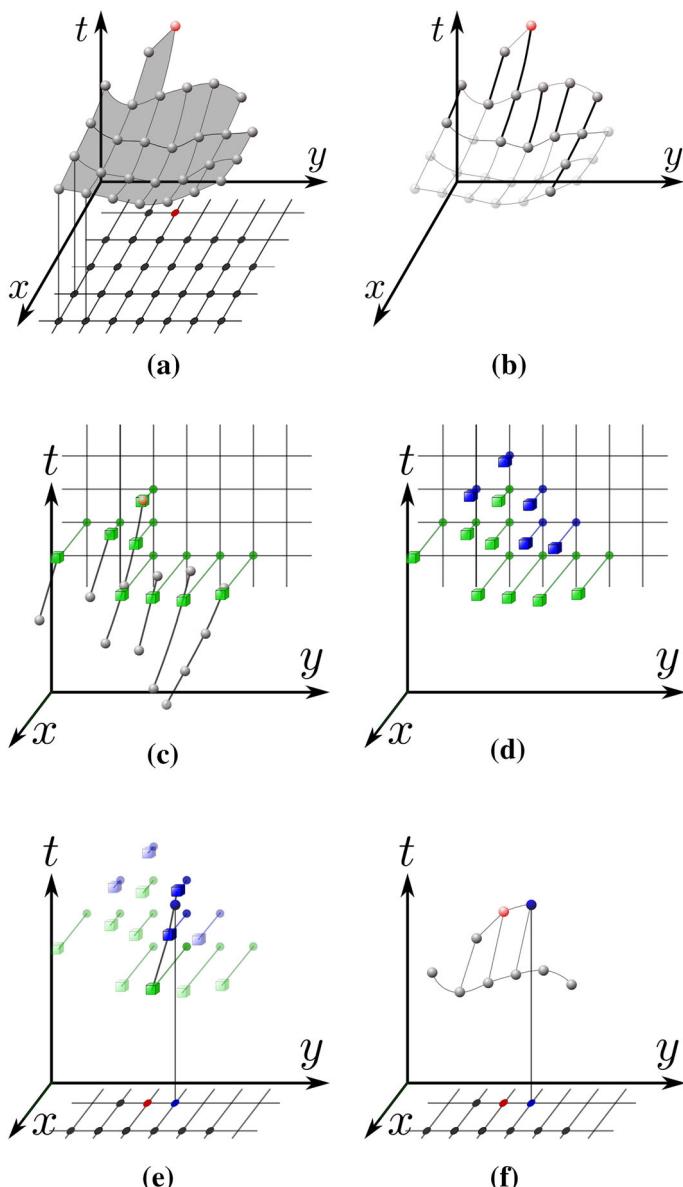
```

---

### 5.3 General Remarks

When the code ends,  $\mathcal{N}$  is empty whereas  $\mathcal{Fa}$  may still contain points. The set  $\mathcal{A}$  provides a discrete sampling of  $\mathcal{M}$ . It may contain multiple triplets sharing the same spatial coordinates. Using this point cloud, and possibly  $\hat{v}$ , a continuous representation of  $\mathcal{M}$  can be obtained—See [3, 8, 22, 26, 35, 55]. Given a time  $t \in (0, T)$ , a contouring algorithm can then be used to find  $\mathcal{C}_t$  (see [26]). By construction,  $\mathcal{M}$  is expected to be undersampled in regions where  $F \approx 0$ , unless the points computed in the sideways representations are also recorded.

*In summary* The Main Loop (Algorithm 1) is similar to the main loop of the classical FMM—and can be verified to be identical to it when  $|F| \geq \delta > 0$ . Depending on the domain of



**Fig. 3** Converting the data using interpolation. Note that the domain shrinks by two points every time step. **a** Some data in the \$xy\$-representation. The point \$p\_{\alpha\beta}\$ appears in red. **b** Only keep those points with a compatible orientation to perform interpolation. **c** One-dimensional interpolation yields the green square data in the \$yt\$-representation. **d** Solving the sideways PDE yields the dark blue squares. **e** One-dimensional interpolation is used to assign a value to \$(x\_i, y\_j)\$. **f** A new point of the form \$(x\_i, y\_j, \psi\_{ij})\$ has been computed (Color figure online)

$F$ , the Eikonal equation  $|\nabla\psi| = 1/F$  is either solved using the classical FMM solver or the  $t$ -FMM (Algorithm 2). In those regions of space-time where the  $xy$ -representation fails, the algorithm switches to a sideways representation (Algorithm 3). The corresponding sideways equation is solved (Algorithm 4) either until a point in the  $xy$ -plane is traversed by the curve or until the code can no longer proceed. The method then reverts back to  $xy$ -representation.

## 6 Complexity of the Method

We derive some estimates for the computational time of the method. Consider a spatial grid of  $N^2$  points with meshsize  $h$ . Let  $\Delta t \sim h$ , and define  $N^*$  to be the number of gridpoints traversed by  $\mathcal{C}_t$  when  $0 < t < T$ . (i.e., if a given gridpoint  $(x_i, y_j)$  is traversed at times  $t_1$  and  $t_2$  where  $0 < t_1 < t_2 < T$ , then this contributes +2 to  $N^*$ .) By construction, the computational time depends on the size of the set  $\mathcal{F}_M := \mathcal{F} \cap M$ . Indeed, Algorithm 3 is only called when Algorithm 1 fails, which occurs whenever an accepted point computed by Algorithm 1 is within a spatial distance  $h$  of  $\mathcal{F}_M$ . Let the number of points computed by Algorithm 3 be  $\tilde{N}$ . Since the complexity of Algorithm 1 is well-known [39], we focus on the complexity of a single call to Algorithm 3.

On the square of side  $2s$ , the narrow band forms a one-dimensional subset. Using interpolation therefore takes  $\mathcal{O}(s)$  operations. Algorithm 4 performs at most  $s^2$  operations, at most twice (one for each attempted representation). However, by Remark 2, and the fact that as  $N$  increases, the number of attempts taken by Algorithm 3 tends to one for almost all points, the complexity tends to  $\mathcal{O}(s)$  for large  $N$ .

Given the assumption that  $F$  is analytic, we expect  $N^* - \tilde{N} = \mathcal{O}(N^2)$  and  $\tilde{N} = \mathcal{O}(N)$ . In practice, the number of points in the local grid  $s$  can be chosen as  $kN$  for  $k \ll 1$ . The overall complexity can therefore be estimated as:

$$\mathcal{O}(N^2 \log(N^2)) + \mathcal{O}(N) \times \mathcal{O}(kN) = \underbrace{\mathcal{O}(N^2 \log(N^2))}_{(t)-\text{FMM}} + \underbrace{\mathcal{O}(kN^2)}_{\text{augmented part}} \quad (36)$$

Note that if  $\mathcal{F} = \emptyset$ , we recover the usual complexity of the FMM.

## 7 Numerical Tests

In this section, we illustrate how the method works with a variety of examples. We first discuss the methodology used to assess the convergence of the algorithms, and briefly summarize which features and results are expected. We then present the examples. More details are provided in “Appendix 2”.

### 7.1 Error Measurement

The error associated to each point  $p_{ij}$  is computed using Method 1 for all examples, except Example 4 when  $F < 0$  where Method 2 is used.

*Method 1:  $E_{ij}$*  We define  $E_{ij} = |\phi(p_{ij})|$ , where  $\phi(x, y, t)$  is the distance function to  $\mathcal{C}_t$  for all  $t$ , and  $p_{ij} = (x_i, y_j, \psi_{ij})$ .



**Fig. 4** Legend for the figures featuring  $\mathcal{A}$

**Method 2:  $G_{ij}$**  The IVP (2) is solved on a very fine grid using second order stencils in space, and Runge-Kutta 2 in time. At each time step, the zero-contour of  $\phi$  is found and sampled. The resulting list of points  $\mathcal{B}$  provides a very accurate discrete approximation of  $\mathcal{M}$ . The error associated to  $p_{ij}$  is defined as the smallest three-dimensional distance to this exact cloud of points, i.e.,  $G_{ij} = \min_{q \in \mathcal{B}} \{|p_{ij} - q|\}$ .

## 7.2 Tests Performed

*Accuracy of Algorithm 4* To verify the accuracy of the sideways method, we pick a domain  $\mathcal{U}$ , initialize say  $x = \psi(y_m, t^0)$  with exact data for some initial time  $t^0$ , and run Algorithm 4 for different gridsizes. The result is a subset of  $\mathcal{M}$ , encoded as a list of points of the form  $p_m^r = (\psi_m^r, y_m, t^r)$ . An error is associated to each point  $p_m^r$  such that  $\psi_m^r < \infty$  using either Method 1 or 2, i.e.,  $E_m^r = |\psi_{\text{exact}}(y_m, t^r) - \psi_m^r|$  or  $G_m^r = \min_{q \in \mathcal{B}} \{|p_m^r - q|\}$ . A two-dimensional  $L_1$  norm is then used to report the results in Fig. 5, e.g.,  $L_1 = h^2 \cdot \sum_{m \in M} \sum_{r \in R} E_m^r$ .

*Accuracy of the full scheme* When testing the accuracy of the full scheme, we distinguish between different regions of the resulting set  $\mathcal{A}$ . In regions computed by the (*t*-)FMM, a two-dimensional  $L_1$  norm is used:  $L_1 = h^2 \cdot \sum_{i \in I} \sum_{j \in J} E_{ij}$ . The sideways representations form one-dimensional sets of  $\mathbb{R}^2 \times [0, T]$ . Consequently, a one-dimensional  $L_1$  norm is used to study those points:  $L_1 = h \cdot \sum_{i \in I} \sum_{j \in J} E_{ij}$ . The global error (computed using all the points in  $\mathcal{A}$ ) is a two-dimensional  $L_1$  norm.

## 7.3 Expectations

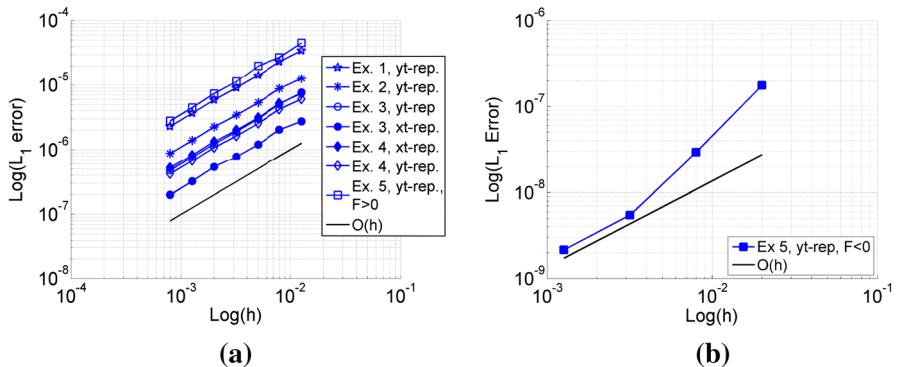
As  $h \rightarrow 0$ , the first order (*t*-)FMM scheme is used almost everywhere. The global error should therefore be first order. We expect the call to Algorithm 3 to increase the constant of convergence. The behaviour of the scheme in the presence of singularities is investigated in Example 4, as well as in Sect. 8.

In all examples but the fourth one, the initial curve  $\mathcal{C}_0$  is a circle of radius  $r_0 = 1/4$  centred at the origin. All tests are initialized with exact values. The legend used for the figures featuring the set  $\mathcal{A}$  is presented on Fig. 4. To reconstruct the curves  $\mathcal{C}_t$ , Delaunay triangulations are used.<sup>2</sup>

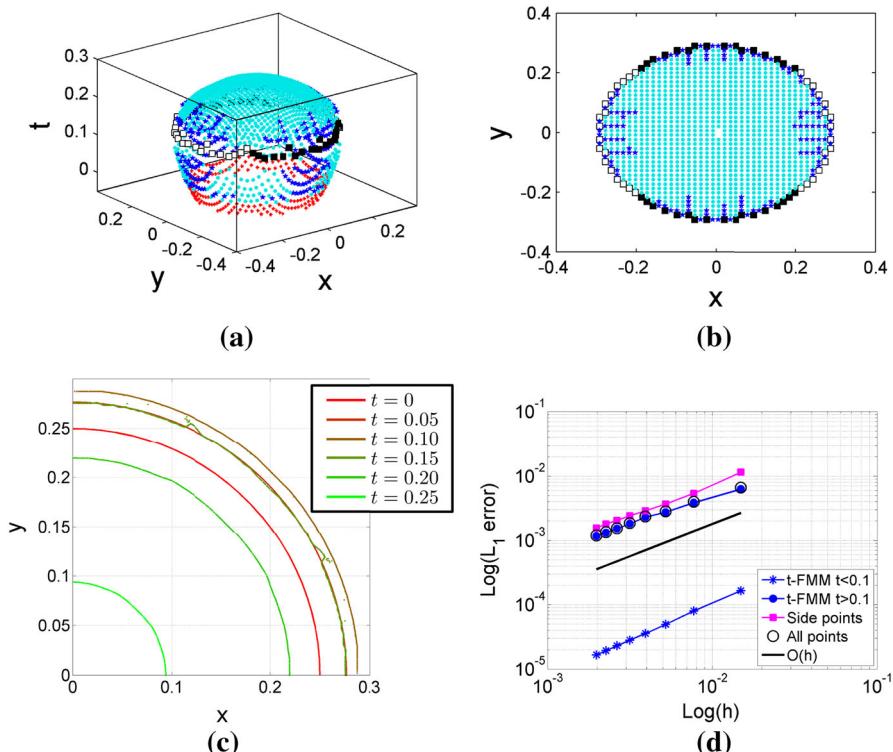
## 7.4 Example 1: $F = F(t) = 1 - e^{10t-1}$

The example illustrates the basic ideas of the method. The speed is such that the circle expands up to time  $t = 0.1$  and then contracts until it collapses to the origin of  $\mathbb{R}^2$ . The results reported on Fig. 5 clearly indicate that the accuracy of the sideways representation is  $\mathcal{O}(h)$ . This is higher than the  $\mathcal{O}(h^{1/2})$  rate predicted in Sect. 4.3. When the entire code is run, the set  $\mathcal{A}$  is presented on Fig. 6a, b. One-dimensional optimization (i.e., The solution of Eq. (17) occurs when  $\xi \in \{0, 1\}$ ) is used for those points traversed by a characteristic that aligns with a spatial axis. We note that the sideways points are computed in the *yt*- (resp. *xt*-)representation when  $\hat{n}$  aligns better with the *x*- (resp. *y*)-axis. As expected, the sampling of

<sup>2</sup> TriScatteredInterp MatLab function

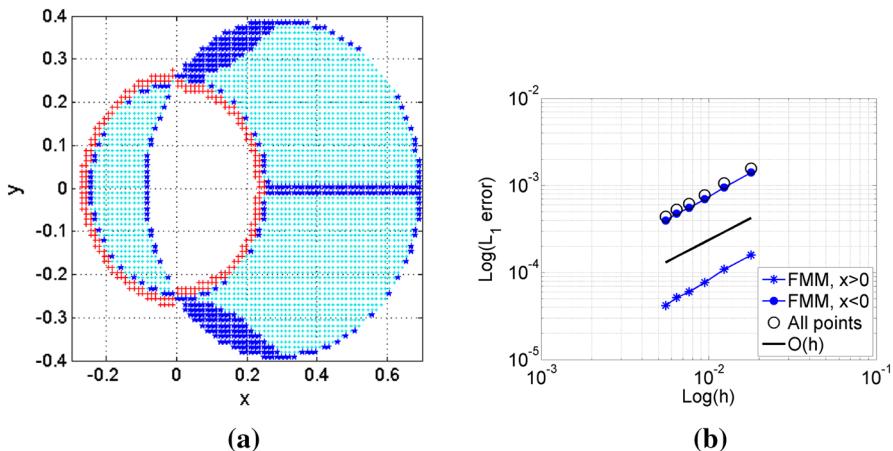


**Fig. 5** Convergence results for the sideways scheme. **a** Method 1,  $E_{ij}$ . **b** Method 2,  $G_{ij}$



**Fig. 6** Example 1. **a** The set  $\mathcal{A}$ . **b** The set  $\mathcal{A}$ . **c** The reconstructed curves  $C_t$  at various times, first quadrant. **d** Convergence results

the surface is sparser near the plane  $t = 0.1$ . The global convergence results are presented in Fig. 6d. We distinguish between the bottom part of the surface, the top part, and those points computed using the sideways representation. The results pertaining to the bottom and the top parts show that the  $t$ -FMM has  $\mathcal{O}(h)$  accuracy, as predicted in Sect. 3. However, the results



**Fig. 7** Example 2. **a** The set  $\mathcal{A}$ . **b** Convergence results

for the top part converge with a larger constant. We conclude that changing representation does deteriorate the accuracy of the sampling but only to a mild extent.

Further investigations reveal that those points computed using one-dimensional optimization in the  $t$ -FMM, just after  $t = 0.1$  bear the largest errors. This slightly affects the accuracy of the reconstructed curve, as can be seen from Fig. 6c, however a simple remedy would be to get rid of those outliers via some post-processing of the data. Two reasons can explain these larger errors: One-dimensional optimization is less accurate than two-dimensional optimization, and the constant of convergence of the  $t$ -FMM seems to depend on  $\delta$  where  $|F| \geq \delta > 0$ .

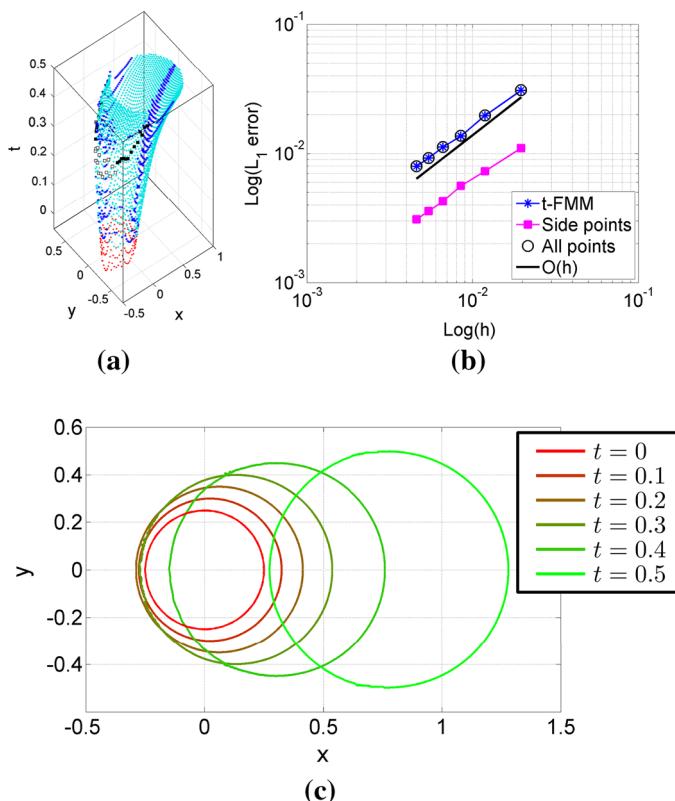
### 7.5 Example 2: $F = F(x) = x$

The given speed is such that the curve remains a circle whose radius grows while its center shifts to the right. Our method adequately handles this case as a single problem, although the speed changes sign across the  $y$ -axis. As expected, Algorithm 3 fails near the points  $(0, 0.25)$  and  $(0, -0.25)$ , as shown on Fig. 7a. The sideways scheme is first order both in the  $xt$ - and the  $yt$ -charts (Fig. 5). Note however that it is not called by the main loop. The results for the full scheme show that it converges with  $\mathcal{O}(h)$  accuracy everywhere (Fig. 7b). Remark that a bi-directional FMM was proposed in [11] to solve a related problem.

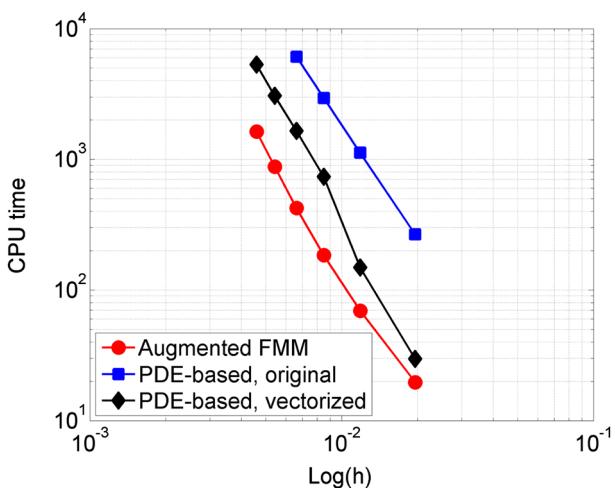
### 7.6 Example 3: $F = F(x, y, t)$

This example differs from the previous ones in that the set  $\mathcal{F}$  is not confined to a single temporal plane, or specific spatial locations (see “Appendix 2”). Thus it illustrates the method in its full generality. The exact solution  $C_t$  is a circle that only grows at first, and then starts moving in the positive  $x$ -direction, as shown on Fig. 8c. Our method is observed to perform very well; We present the resulting surface and the first order convergence results on Figs. 5 and 8.

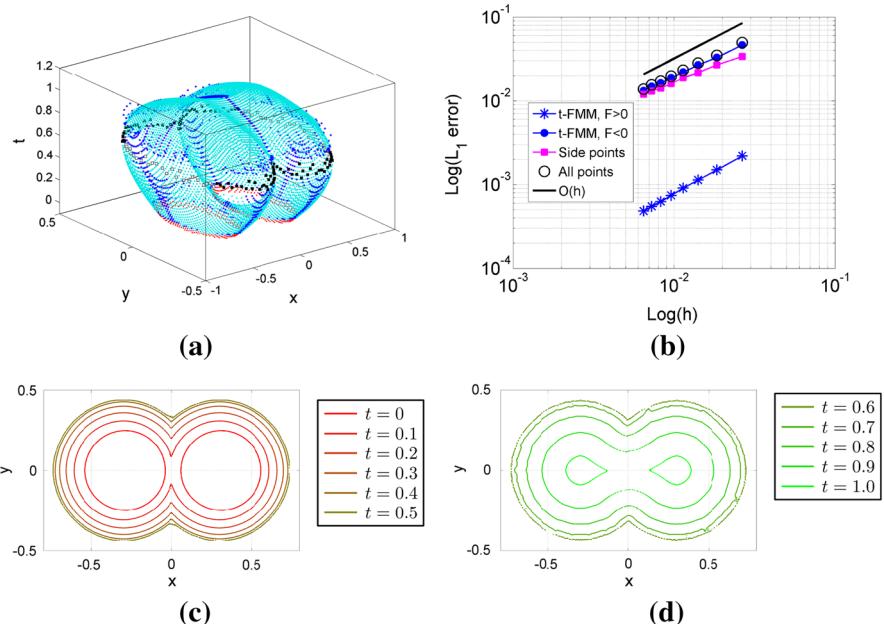
*Complexity* Figure 9 compares our method against the PDE-based fast local LSM presented in [36]. Both the original method (as described in the paper) and a vectorized version are run. It is clear from the results that although the trend is the same, our method is faster than the local LSM: It differs from the original version by one order of magnitude.



**Fig. 8** Example 3. **a** The set  $\mathcal{A}$ . **b** Convergence results. **c** The reconstructed curves  $\mathcal{C}_t$  at various times



**Fig. 9** CPU times of our method and the PDE-based local level-set method [36], when running Example 3



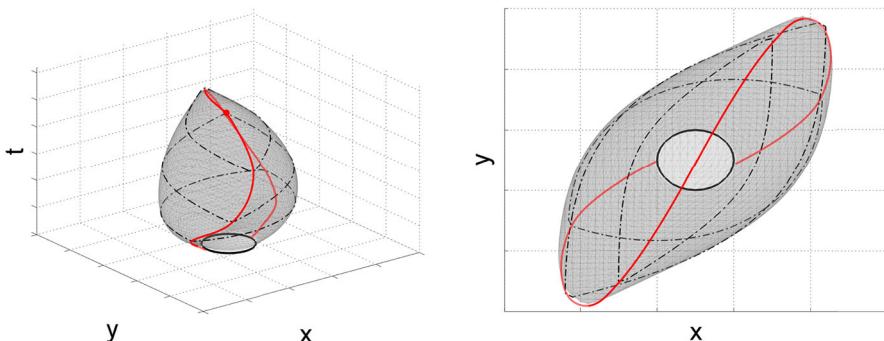
**Fig. 10** Example 4. **a** The set  $\mathcal{A}$ . **b** Convergence results. **c** The reconstructed curves  $\mathcal{C}_t$  at various times  $t < 0.5$ . **d** The reconstructed curves  $\mathcal{C}_t$  at various times  $t > 0.5$

## 7.7 Example 4: Two Merging Circles

This example tests the ability of the scheme to capture topological changes. The initial codimension-one manifold consists of two disjoint circles of radius  $r_0 = 1/4$ , with centres at  $(-.3, 0)$  and  $(.3, 0)$ . The speed is such that the circles first expand, until they touch and merge. Then the speed changes sign, which makes the curve shrink until it pinches off and splits into two distinct curves. The set  $\mathcal{A}$  is presented in Fig. 10a. The accuracy of the sideways scheme is investigated on a domain that comprises the shock when  $F > 0$ , and the rarefaction when  $F < 0$ . First order convergence is obtained in each case (Fig. 5). The full scheme also shows first order convergence (Fig. 10b). The convergence of the sideways points and the top part is a little shy of first order, but this can be attributed to the measurement method. Those results demonstrate how robust the overall scheme is. Note that a similar example was tackled in [7], with a speed  $F$  that depended linearly on time. The reconstructed curves can be seen on Fig. 10c, d.

## 8 Discussion

We illustrate a limitation of the scheme with an ultimate example. The speed is such that the initial circle immediately develops a kink along the  $x$ -axis at time  $t = 0$ .  $\mathcal{C}_t$  is shaped like an almond, turning in the counterclockwise direction while expanding. Then, the sign of the speed changes, forcing the curve to contract while retaining its slanted shape. See “Appendix 2” for details and Fig. 11 for an illustration. As far as the authors know, this is the first example in the literature of a curve evolution featuring a singularity whose location changes with time, for which an exact solution is known. Some results are presented on Fig. 12. The singularity



**Fig. 11**  $\mathcal{M}$  for the almond example. The shock appears as a *red plain line* (Color figure online)

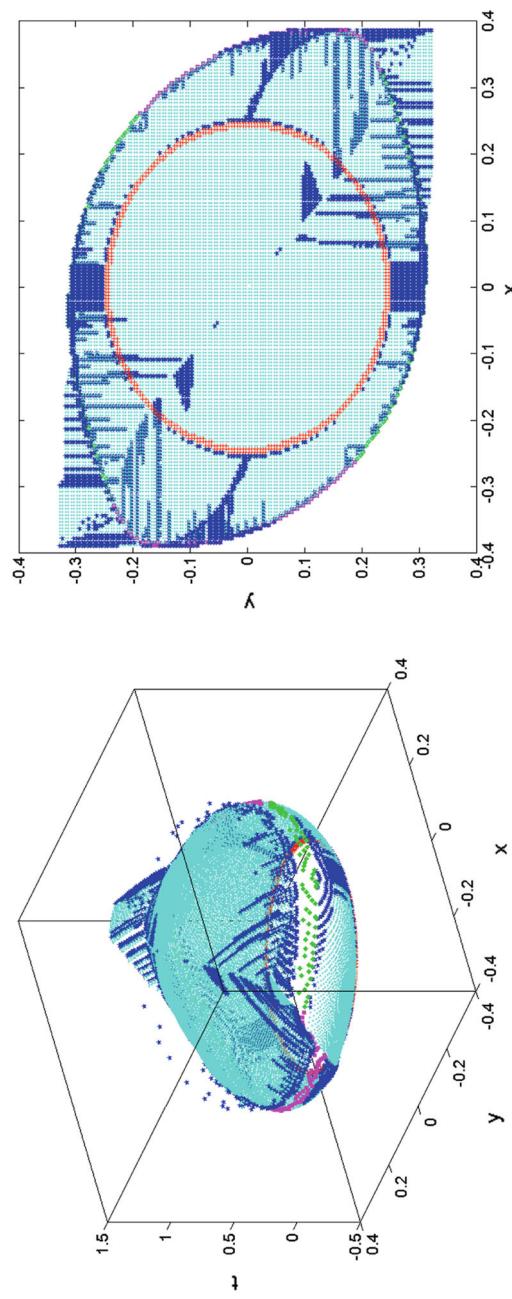
is clearly visible, and has the expected figure-eight shape. Nevertheless some points ‘escape’ through the shock when the speed changes sign, and start out two new fronts that keep on expanding. This results from our usage of the outward normal to distinguish between the outside and the inside of the curve: Discontinuities in  $\hat{v}$  result in mistagging of points. Note however that the speed  $F$  does not satisfy the assumptions of this paper, since it is a  $C^0$  function of  $\mathbb{R}^2 \times [0, T]$ .

A weakness of the method is that it is fairly sensitive to the accuracy of the normal, which is used to ‘glue’ representations. Moreover, as it stands, Algorithm 3 may fail even though  $(x_i, y_j)$  or  $(x_\alpha, y_\beta)$  belongs to  $\mathcal{C}_t$  for some  $t \in (0, T)$ . Two situations make such a scenario possible: (1) the time steps taken are too small, or (2) too little information obtained from interpolation is available. These problems could be resolved by making the parameters (e.g.,  $h$  and  $s$ ) depend on the Lipschitz constants of  $F$  and its derivatives, as well as the local curvatures of  $\Gamma_t$ . Doing so properly still requires investigation.

An obvious benefit of building the method on the standard FMM is that the sideways representations need only be used to compute a small number of points. As a result, the numerical complexity of the method competes with a local LSM.

*Conclusion* Our aim was to devise an algorithm with low complexity able to describe the non-linear evolution of codimension one manifolds subject to a space- and time-dependent speed function that changes sign. We illustrated how pre-existing methods can be combined to achieve this goal. The fact that we always dealt with explicit representations of the manifold implied that the dimensionality of the problem was never raised. The resulting algorithm was found to have a global truncation error of  $\mathcal{O}(h)$ . We tested it against a number of examples, some of which do not appear in the current literature.

Overall, the present work thoroughly introduces a new algorithm, along with proofs of convergence and stability, as well as sturdy numerical results. We believe that the main idea on which it relies—i.e., to change representation based on the speed function  $F$ —may be extended and improved in many ways that shall be explored.



**Fig. 12** Almond example: The set  $\mathcal{A}$

**Acknowledgments** The authors wish to thank Prof. A.Oberman for helpful discussions. The second author would like to thank the organizers of the 2011 BIRS workshop “Advancing numerical methods for viscosity solutions and applications”, Profs. Falcone, Ferretti, Mitchell, and Zhao for stimulating discussions which eventually lead to this work.

## Appendix 1: Direct Method to Compute $\psi_{\text{II}}$ in the $t$ -FMM, in 2D

We provide a direct method for solving the minimization problem appearing in Equation (17), in two dimensions. Introducing  $\tau(y) = h/|F(\mathbf{x}_{ij}, \psi(y))|$ , we first use linear interpolation to simplify the quantity we wish to minimize:

$$\begin{aligned} \psi(\tilde{\mathbf{x}}) + \sqrt{\xi^2 + (1 - \xi)^2} \frac{h}{|F(\mathbf{x}_{ij}, \psi(\tilde{\mathbf{x}}))|} &= \psi(\tilde{\mathbf{x}}) + \sqrt{\xi^2 + (1 - \xi)^2} \tau(\tilde{\mathbf{x}}) \\ &\approx \xi \psi(\mathbf{x}_{i-1,j}) + (1 - \xi) \psi(\mathbf{x}_{i,j+1}) \\ &\quad + \sqrt{\xi^2 + (1 - \xi)^2} \\ &\quad \times (\xi \tau(\mathbf{x}_{i-1,j}) + (1 - \xi) \tau(\mathbf{x}_{i,j+1})) \\ &=: f(\xi) \end{aligned} \quad (37)$$

Minimizing  $f$  over  $\xi \in (0, 1)$  amounts to finding the roots of  $0 = c_4\lambda^4 + c_3\lambda^3 + c_2\lambda^2 + c_1\lambda + c_0$  where  $\lambda \in (0, 1)$  is such that  $f'(\lambda) = 0$ . This quartic can be solved either directly with closed formulas, or with Newton’s method—we use the latter. For each root  $r_i \in (0, 1)$  the corresponding value of  $\psi$  is computed as  $\psi_{\text{II},r_i} = f(r_i)$ . If  $\psi_{\text{II},r_i} < \psi(\mathbf{x}_{i-1,j})$  or  $\psi_{\text{II},r_i} < \psi(\mathbf{x}_{i,j+1})$ , then  $\psi_{\text{II},r_i}$  is discarded. Values arising from minimization in one dimension are also computed as  $\psi_{\text{II},0} = \psi(\mathbf{x}_{i,j+1}) + \tau(\mathbf{x}_{i,j+1})$  and  $\psi_{\text{II},1} = \psi(\mathbf{x}_{i-1,j}) + \tau(\mathbf{x}_{i-1,j})$ . The global minimum is found by comparing all those values.

## Appendix 2: Implementation Details for the Examples

We give some details about the examples presented in Sect. 7. All tests were performed using MATLAB® [51]. In particular, finding the minimum value in  $\mathcal{N}$  is done using `min`.

*Computing the outward normal* In regions where the level-set function  $\phi$  is  $C^1$ , we have  $\hat{\mathbf{v}} = (\phi_x, \phi_y, \phi_t)/|(\phi_x, \phi_y, \phi_t)|$ . We use the Implicit Function Theorem: e.g., If  $\psi(y, t) = x$  satisfies  $\phi(\psi(y, t), y, t) = 0$ , then  $\phi_y = -\phi_x \psi_y$  and  $\phi_t = -\phi_x \psi_t$ . Since  $\phi_x \neq 0$ , we set  $\mathbf{v} = (+\text{sign}(\phi_x), \psi_y, \psi_t)$  and  $\hat{\mathbf{v}} = \mathbf{v}/|\mathbf{v}|$ . In the  $(t)$ -FMM as well as in the sideways representations, the points used to approximate the derivatives using finite-differences are the ones involved in the computation of the new point. E.g., In the  $t$ -FMM, if two-dimensional optimization was used in Quadrant III to obtain  $\psi_{ij}$ , then  $\hat{\mathbf{v}}(p_{ij}) = \mathbf{v}/|\mathbf{v}|$  where

$$\mathbf{v} = ((\psi_{ij} - \psi_{i-1,j})/h, (\psi_{ij} - \psi_{i,j-1})/h, -\text{sign}(\hat{\psi}_3(p_{ij-1}))) \quad (38)$$

*Choice of parameters* In all examples, the number of points in each dimension is  $N + 1$ , and the spatial grid spacings are even:  $h = dx = dy$ . The size of the local grid in Algorithm 3 is  $s = \lfloor \frac{N}{3} \rfloor$ . When  $F$  depends on time, we use adaptive time-stepping. Before the time where  $F = 0$ , we set  $\Delta t = r_1 h$ . Passed that time, we let  $\Delta t = r_2 h$ . To assess the convergence of the sideways methods, a  $yt$ -grid with spacings  $h$  and  $\Delta t = h/2$  is built. The exact normal  $\hat{\mathbf{v}}$  is assigned to the points as they are accepted in all the examples, except Example 1 where it is computed as explained in the above paragraph.

*Example 1* The exact solution to the Level-Set Equation is  $\phi(x, y, t) = \sqrt{x^2 + y^2} - R(t)$  where  $R(t) = \left(r_0 - \frac{e^{10t}-1}{10e} + t\right)$ . Domain:  $[-.321, .319]^2$ .  $T_F = 0.3$ .  $xt$ - and  $yt$ -rep.:  $r_1 = 1/3, r_2 = 2$ . Skewed rep.:  $r_1 = r_2 = 1$ . Domain for convergence of Algorithm 4:  $(y, t) \in [-0.25, 0.25] \times [0, 0.3]$ .

*Example 2* The signed distance function to  $\mathcal{C}_t$  is given as  $\phi(x, y, t) = \sqrt{(x - x_c(t))^2 + y^2} - r(t)$  where  $x_c(t) = r_0 \sinh t$  and  $r(t) = r_0 \cosh t$ . Note that  $\phi$  does not solve the Level-Set Equation. Domain:  $[-1.01, 0.99]^2$ .  $T_F = 1$ .  $xt$ - and  $yt$ -rep.:  $r_1 = 1/3, r_2 = 2$ . Skewed rep.:  $r_1 = 1/3, r_2 = 5$ . Domain for convergence of Algorithm 4:  $(y, t) \in [-0.25, 0.25] \times [0, 1]$  and  $(x, t) \in [-0.25, 0.25] \times [0, 1]$ .

*Example 3* The exact solution to the Level-Set Equation is  $\phi(x, y, t) = \sqrt{(x - gt)^2 + y^2} - (r_0 + ct)$  where  $b = 10, c = 1/2$  and  $g(t) = \arctan(b(t - 0.5)) + \frac{\pi}{2}$ . The speed is

$$F = \frac{(x - gt)(g't + g)}{\sqrt{(x - gt)^2 + y^2}} + c \implies F \approx \begin{cases} c & \text{for } t \text{ small} \\ \frac{(x - \pi t)\pi}{\sqrt{(x - \pi t)^2 + y^2}} + c & \text{for } t \text{ large} \end{cases} \quad (39)$$

We expect the circle to first expand (when  $t$  is small), and then expand while moving to the right with speed  $\pi$  (when  $t$  is large). Domain:  $[-1.51, +1.49]^2$ .  $T_F = 0.5$ .  $xt$ - and  $yt$ -rep.:  $r_1 = 1/3, r_2 = 2$ . Skewed rep.:  $r_1 = 1/3, r_2 = 5$ . Domain for convergence of Algorithm 4:  $(y, t) \in [-0.25, 0.25] \times [0, 0.5]$ .

*Example 4* The set  $\mathcal{C}_0$  consists of two disjoint circles of radius  $r_0 = 0.25$ , with centres at  $(-0.3, 0)$  and  $(0.3, 0)$ . The speed is  $F = 1 - e^{2t-1}$ . The circles touch along the  $y$ -axis when  $t \approx 0.08$ . When  $t < 0.5$  the exact solution to the Level-Set Equation is  $\phi(x, y, t) = \min \left\{ \sqrt{(x + 0.3)^2 + y^2} - R(t), \sqrt{(x - 0.3)^2 + y^2} - R(t) \right\}$  where  $R(t) = r_0 - \frac{e^{2t}-1}{2e} + t$ . Domain:  $[-1.5+0.01e, +1.5+0.01e]^2$ .  $T_F = 1.2$ .  $xt$ - and  $yt$ -rep.:  $r_1 = 1/3, r_2 = 2$ . Skewed rep.:  $r_1 = 1/3, r_2 = 5$ . Domain for convergence of Algorithm 4:  $(y, t) \in [-0.5, 0.5] \times [0.2, 0.5]$  and  $(y, t) \in [-0.5, 0.5] \times [0.5, .52]$ .

*The Almond example* The exact solution to the Level-Set Equation is

$$\phi(x, y, t) = \left( \sqrt{x^2 + y^2} - r_0 + \frac{e^{ct} - 1}{ce} - t(1 + C) \right) + \frac{t|xt - y|}{\sqrt{1 + t^2}} \quad (40)$$

The constants are set to be:  $r_0 = 1/4$ ,  $c = 1$ , and  $C = .65$ . The function  $\phi$  is made up of two parts: The first one in brackets is qualitatively the same as in Example 1. Domain:  $[-0.5, 0.5]^2$ .  $T_F = 1.9$ .  $xt$ - and  $yt$ -rep.:  $r_1 = 1/3, r_2 = 2$ . Skewed rep.:  $r_1 = 1/2, r_2 = 6$ .

## References

1. Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. *J. Comput. Phys.* **118**(2), 269–277 (1995)
2. Adalsteinsson, D., Sethian, J.A.: A level set approach to a unified model for etching, deposition, and lithography. I. Algorithms and two-dimensional simulations. *J. Comput. Phys.* **120**(1), 128–144 (1995)
3. Amenta, N., Bern, M., Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98, pp. 415–421. ACM, New York, NY, USA (1998)
4. Bardi, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Modern Birkhäuser Classics. Birkhäuser, Boston (2008)
5. Barles, G.: Existence results for first order Hamilton Jacobi equations. *Ann. Inst. H. Poincaré Anal. Non Linéaire* **1**(5), 325–340 (1984)

6. Carlini, E., Cristiani, E., Forcadel, N.: A non-monotone fast marching scheme for a Hamilton-Jacobi equation modeling dislocation dynamics. ENUMATH 2005, Santiago de Compostela (Spain) (2007)
7. Carlini, E., Falcone, M., Forcadel, N., Monneau, R.: Convergence of a generalized fast-marching method for an eikonal equation with a velocity-changing sign. SIAM J. Numer. Anal. **46**(6), 2920–2952 (2008)
8. Chazal, F., Cohen-Steiner, D., Mérigot, Q.: Geometric inference for probability measures. Found. Comput. Math. **11**(6), 733–751 (2011)
9. Cheng, L.T., Tsai, Y.H.: Redistancing by flow of time dependent eikonal equation. J. Comput. Phys. **227**(8), 4002–4017 (2008)
10. Chopp, D.L.: Some improvements of the fast marching method. SIAM J. Sci. Comput. **23**(1), 230–244 (2001)
11. Chopp, D.L.: Another look at velocity extensions in the level set method. SIAM J. Sci. Comput. **31**(5), 3255–3273 (2009)
12. Crandall, M.G., Evans, L.C., Lions, P.L.: Some properties of viscosity solutions of Hamilton-Jacobi equations. Trans. Amer. Math. Soc. **282**(2), 487–502 (1984)
13. Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Amer. Math. Soc. (N.S.) **27**(1), 1–67 (1992)
14. Crandall, M.G., Lions, P.L.: Viscosity solutions of Hamilton-Jacobi equations. Trans. Am. Math. Soc. **277**(1), 1–42 (1983)
15. Crandall, M.G., Lions, P.L.: Two approximations of solutions of Hamilton-Jacobi equations. Math. Comput. **43**(167), 1–19 (1984)
16. Crandall, M.G., Tartar, L.: Some relations between nonexpansive and order preserving mappings. Proc. Am. Math. Soc. **78**(3), 385–390 (1980)
17. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1), 269–271 (1959)
18. Evans, L.: Partial Differential Equations. Graduate studies in Mathematics. American Mathematical Society, Providence (2010)
19. Falcone, M.: The minimum time problem and its applications to front propagation. In: Visintin, A., Buttazzo, G. (eds.) Motion by Mean Curvature and Related Topics, pp. 70–88. De Gruyter verlag, Berlin (1994)
20. Forcadel, N., Le Guyader, C., Gout, C.: Generalized fast marching method: applications to image segmentation. Numer. Algorithms **48**(1–3), 189–211 (2008)
21. Gibou, F., Fedkiw, R., Caflisch, R., Osher, S.: A level set approach for the numerical simulation of dendritic growth. J. Sci. Comput. **19**(1–3), 183–199 (2003). Special issue in honor of the sixtieth birthday of Stanley Osher
22. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. SIGGRAPH Comput. Graph. **26**(2), 71–78 (1992)
23. Koike, S.: A Beginner's Guide to the Theory of Viscosity Solutions. MSJ Memoirs. Mathematical Society of Japan, Tokyo (2004)
24. Lions, P.L., Rouy, E., Tourin, A.: Shape-from-shading, viscosity solutions and edges. Numerische Mathematik **64**(1), 323–353 (1993)
25. Lolla, T., Ueckermann, M., Yigit, K., Haley, P., Lermusiaux, P.: Path planning in time dependent flow fields using level set methods. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 166–173. River Centre, Saint Paul, Minnesota, USA (2012)
26. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. **21**(4), 163–169 (1987)
27. Malladi, R., Sethian, J.A., Vemuri, B.C.: A fast level set based algorithm for topology-independent shape modeling. J. Math. Imaging Vision **6**(2–3), 269–289 (1996)
28. Merriman, B., Bence, J.K., Osher, S.J.: Motion of multiple functions: a level set approach. J. Comput. Phys. **112**(2), 334–363 (1994)
29. Mitchell, I.: Dynamic programming algorithms for planning and robotics in continuous domains and the Hamilton-Jacobi equation (Slides presented for IROS meeting in France). <http://www.cs.ubc.ca/mitchell/Talks/mitchellIROS.pdf> (2008)
30. Mulder, W., Osher, S., Sethian, J.A.: Computing interface motion in compressible gas dynamics. J. Comput. Phys. **100**(2), 209–228 (1992)
31. Oberman, A.M.: Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-Jacobi equations and free boundary problems. SIAM J. Numer. Anal. **44**(2), 879–895 (2006). (electronic)
32. Osher, S., Cheng, L.T., Kang, M., Shim, H., Tsai, Y.H.: Geometric optics in a phase-space-based level set and eulerian framework. J. Comput. Phys. **179**(2), 622–648 (2002)

33. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces, Applied Mathematical Sciences, vol. 153. Springer, New York (2003)
34. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phys.* **79**, 12–49 (1988)
35. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: Proceedings of the Conference on Visualization '02. VIS '02, pp. 163–170. IEEE Computer Society, Washington, DC, USA (2002)
36. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-based fast local level set method. *J. Comput. Phys.* **155**(2), 410–438 (1999)
37. Russo, G., Smereka, P.: A remark on computing distance functions. *J. Comput. Phys.* **163**(1), 51–67 (2000)
38. Sethian, J.: Numerical methods for propagating fronts. In: Concus, P., Finn, R. (eds.) *Variational Methods for Free Surface Interfaces*, pp. 155–164. Springer, New York (1987)
39. Sethian, J.: A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.* **93**, 1591–1595 (1996)
40. Sethian, J.: Fast marching methods. *SIAM Rev.* **41**(2), 199–235 (1999)
41. Sethian, J.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (1999)
42. Sethian, J.A., Strain, J.: Crystal growth and dendritic solidification. *J. Comput. Phys.* **98**(2), 231–253 (1992)
43. Sethian, J.A., Vladimirsky, A.: Ordered upwind methods for static Hamilton-Jacobi equations. *Proc. Natl. Acad. Sci. USA* **98**(20), 11069–11074 (2001)
44. Sethian, J.A., Vladimirsky, A.: Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms. *SIAM J. Numer. Anal.* **41**(1), 325–363 (2003)
45. Souganidis, P.E.: Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. *J. Differ. Equ.* **59**(1), 1–43 (1985)
46. Souganidis, P.E.: Existence of viscosity solutions of Hamilton-Jacobi equations. *J. Differ. Equ.* **56**(3), 345–390 (1985)
47. Subbotin, A.: *Generalized Solutions of First Order PDEs: The Dynamical Optimization Perspective*. Systems & Control. Birkhäuser, Boston (1994)
48. Sussman, M., Fatemi, E.: An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.* **20**(4), 1165–1191 (1999). (electronic)
49. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**(1), 146–159 (1994)
50. Takei, R., Tsai, R.: Optimal trajectories of curvature constrained motion in hamilton-jacobi formulation (to appear). *J. Sci. Comput.* **54**(2), 622–644 (2013)
51. The MathWorks Inc., N.M.U.S.: MatLab and Statistics Toolbox Release 2010b, Version 7.11.0.584. The Mathworks, Inc., Natick, Massachusetts, United States. (2010)
52. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Trans. Automat. Control* **40**(9), 1528–1538 (1995)
53. Vladimirsky, A.: Static PDEs for time-dependent control problems. *Interfaces Free Bound* **8**(3), 281–300 (2006)
54. Zhao, H.: A fast sweeping method for Eikonal equations. *Math. Comput.* **74**(250), 603–627 (2005)
55. Zhao, H.K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2001. pp. 194–201 (2001)
56. Zhu, J., Ronney, P.: Simulation of front propagation at large non-dimensional flow disturbance intensities. *Combust. Sci. Technol.* **100**(1–6), 183–201 (1994)