Weighted Distance Maps Computation on Parametric Three-Dimensional Manifolds

Alexander M. Bronstein, Michael M. Bronstein, Ron Kimmel

Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel.

Abstract

We propose an efficient computational solver for the eikonal equations on parametric three-dimensional manifolds. Our approach is based on the fast marching method for solving the eikonal equation in $\mathcal{O}(n \log n)$ steps by numerically simulating wavefront propagation. The obtuse angle splitting problem is reformulated as a set of small integer linear programs, that can be solved in $\mathcal{O}(n)$. Numerical simulations demonstrate the accuracy of the proposed algorithm.

Key words: fast marching, fast sweeping, distance map, geodesic distance, parametric three-dimensional Riemannian manifolds, angle splitting

1 Introduction

Construction of distance maps is important in computational geometry. It arises in various applications including medical imaging, robot motion planning, navigation, and geophysics to name a few. The problem of computing distances on curved manifolds has been previously addressed primarily in the two-dimensional case. Sethian [Sethian(1996)] proposed an $\mathcal{O}(n \log n)$-efficient algorithm for computation of weighted distance maps on domains with weighted Euclidean metric, termed as *fast marching*. A similar algorithm was developed by Tsitsiklis [Tsitsiklis(1995)].

The main idea of the fast marching algorithm is to simulate a wave front advancing from a set of source points. The propagating front can be thought

Email addresses: alexbron@ieee.org (Alexander M. Bronstein), bronstein@ieee.org (Michael M. Bronstein), ron@cs.technion.ac.il (Ron Kimmel).

of as a "prairie fire" evolution with some velocity field F towards directions where the grid has not yet been "burnt out". At time $t = 0$, the fire starts at the source points, and the algorithm computes the time values T for each vertex at which the advancing fire front reaches it. Mathematically, the problem of weighted distance map computation can be formulated as the viscosity solution of the *eikonal equation*

$$\|\nabla T\| = F, \quad T(\mathrm{S}) = 0, \tag{1}$$

where S is the set of source points. Intuitively, the distance T grows from the source points with a gradient magnitude equal to F. In optics and acoustics, the eikonal equation governs the propagation of waves through a media with inhomogeneous (yet often isotropic) coefficients for the speed of light (sound). The solution of the eikonal equation demonstrates that light or acoustic waves traverse the path between two points, which takes the least time, a physics law known as *Fermat's principle*.

The fast marching method proposed by Sethian in [Sethian(1996)] can be easily generalized to any dimension, yet, it is limited to orthogonal grids. This limitation also applies to Tsitsiklis' method. In two dimensions, a variety of solutions has been proposed to overcome this limitation. Over the last decade, Sethian's algorithm was generalized to arbitrary triangulated two-dimensional manifolds [Kimmel and Sethian(1998)], unstructured meshes [Sethian and Vladimirsky(2000)], implicit unorganized surfaces [Mémoli and Sapiro(2001)], and parametric two-manifolds [Spira and Kimmel(2004)]. Unlike the well-studied two dimensional case, the situation differs dramatically for three-dimensional manifolds. Due to the problem of obtuse angles articulated later in this introduction, existing three-dimensional eikonal solvers can handle only weighted Euclidean (flat) manifolds, or non-flat manifolds with orthogonal parameterization. However, construction of such a grid is as complicated as computing the distance map itself.

This paper aims to fill this apparent gap by presenting an extension of the fast marching method to curved (non-Euclidean) three dimensional manifolds. Following [Spira and Kimmel(2004)], we focus our discussion on *parametric* manifolds, i.e. volumes consisting of a three-dimensional parameterization domain $\mathbf{U}$, which is mapped by $\mathbf{x} : \mathbf{U} \mapsto \mathbb{R}^m$ to some region in $\mathbb{R}^m$ where the manifold is immersed. The derivatives

$$\boldsymbol{\xi}_i = \frac{\partial \mathbf{x}}{\partial u^i}, \tag{2}$$

of $\mathbf{x}$ with respect to $\mathbf{u}$ constitute a local coordinate system on the parametric manifold, which is usually non-orthogonal. Geodesic distances in the volume are calculated according to the differential element

$$ds^2 = d\mathbf{u}^{\mathrm{T}}\mathbf{G}d\mathbf{u}, \tag{3}$$

where $d\mathbf{u} = (du^1, du^2, du^3)^{\mathrm{T}}$, and $\mathbf{G}$ is the *metric tensor*, whose elements are given by $g_{ij} = \boldsymbol{\xi}_i^{\mathrm{T}}\boldsymbol{\xi}_j$. The local coordinate system $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3\}$ is orthogonal if and only if $\mathbf{G}$ is diagonal.

Our goal is to find an approximate solution to the continuous eikonal equation

$$\|\nabla_{\mathbf{G}} T(\mathbf{u})\|^2 = (\nabla_{\mathbf{u}} T)^{\mathrm{T}}\mathbf{G}^{-1}(\mathbf{u})\nabla_{\mathbf{u}} T \;=\; F^2(\mathbf{u}) \tag{4}$$

on a discrete numerical grid, obtained by sampling the parameterization domain $\mathbf{U}$. The discrete solution is required to converge to the continuous one as the grid is refined. It is convenient to select a regular Cartesian grid with unit steps as the parametrization domain $\mathbf{U}$. Each grid point $\mathbf{u}_n$ is connected to its neighbors according to some connectivity pattern. Among the variety of possibilities, the simplest choice is to consider six-neighbor connectivity, which implies that each point $\mathbf{u}_n$ is connected to six neighboring points $\mathbf{u}_n + \mathbf{m}$, $\mathbf{m} = (\pm 1, 0, 0)^{\mathrm{T}}, (0, \pm 1, 0)^{\mathrm{T}}, (0, 0, \pm 1)^{\mathrm{T}}$. This pattern forms eight right-angle simplices in the parameterization domain with edges parallel to the axes u^1, u^2 and u^3 (Figure 1). Another possible grid connectivity is based on 18 neighbors: the six neighbors as before, plus additional 12 neighbors $\mathbf{m} = (0, \pm 1, \pm 1)^{\mathrm{T}}, (\pm 1, 0, \pm 1)^{\mathrm{T}}, (\pm 1, \pm 1, 0)^{\mathrm{T}}$. As shown in Figure 1(right), this scheme populates each octant with three simplices with the angles $45°, 45°, 60°$ at $\mathbf{u}_n$ and one $60°, 60°, 60°$ simplex (the angles are measured between the edges $\mathbf{x}(\mathbf{u}_n + \mathbf{m}_i) - \mathbf{x}(\mathbf{u}_n)$).

Let us examine a particular simplex formed by connecting some point $\mathbf{u}_0$ in the parameterization domain to three neighboring points $\mathbf{u}_i = \mathbf{u}_0 + \mathbf{m}_i$, where $\mathbf{m}_1, \mathbf{m}_2$, and $\mathbf{m}_3$ are integer displacement vectors selected according to the grid connectivity pattern. These points are mapped to the vertices $\mathbf{x}_i = \mathbf{x}(\mathbf{u}_i)$ in the volume. In local coordinates, we can write

$$\mathbf{x}_i = \mathbf{x}_0 + m_i^1\boldsymbol{\xi}_1 + \; m_i^2\boldsymbol{\xi}_2 + \; m_i^3\boldsymbol{\xi}_3 \;=\; \mathbf{x}_0 + \boldsymbol{\Xi}\mathbf{m}_i, \tag{5}$$

where $\boldsymbol{\Xi} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$. The angles formed at the vertex $\mathbf{x}_0$ of the simplex can be described by the scalar products $e_{ij} = (\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}(\mathbf{x}_j - \mathbf{x}_0) = \mathbf{m}_i^{\mathrm{T}}\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi}\mathbf{m}_j = \mathbf{m}_i^{\mathrm{T}}\mathbf{G}\mathbf{m}_j^{\mathrm{T}}$, or in matrix notation

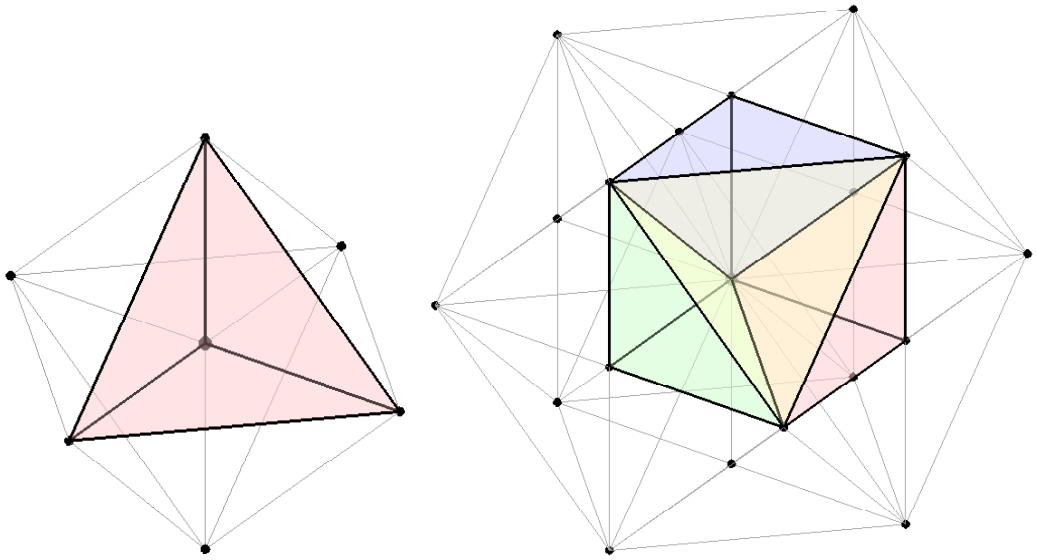

Fig. 1. Grid connectivity based on 6 (left) and 18 (right) neighbor points.

$$\mathbf{E} = \mathbf{M}^{\mathrm{T}}\mathbf{G}\mathbf{M}, \tag{6}$$

where $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$. The simplex is termed *acute* if the angles $\angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2$, $\angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_3$, and $\angle\mathbf{x}_2\mathbf{x}_0\mathbf{x}_3$ are non-obtuse, which holds if and only if all off-diagonal elements of $\mathbf{E}$ are non-negative. A simplex having at least one obtuse angle at the vertex $\mathbf{x}_0$ is termed *obtuse*. It is remarkable that the simplex geometry required for distance map computation is contained entirely in the matrix $\mathbf{E}$. This can be advantageous in applications where the manifold itself is unavailable, and only its metric is known.

Table 1 outlines the fast marching algorithm on three-dimensional parametric grids. Solution of the eikonal equation starts by setting zero distance to the set of source points and updating the neighboring points by simulating an advancing wavefront. By construction, the updated value cannot be smaller than the values of the supporting vertices. This monotonicity property ensures that the solution is always propagated outwards by fixing the vertex with the smallest T. The latter implies that the values of *fixed* vertices are never recomputed. Since the *update step* has constant complexity, the overall complexity of the fast marching algorithm is determined by the procedure that finds the smallest T in the *close* list. Heap sorting-based priority queue allows to implement this task in $\mathcal{O}(\log n)$, where n is the number of vertices. Since each vertex is fixed only once, the overall complexity is $\mathcal{O}(n \log n)$.

In the numerical core of the algorithm lies the update step, which given the time of arrival of the wavefront to three vertices of a simplex, computes the time of arrival to the fourth one. In the two-dimensional version of the fast

Input: Parameterization grid $\mathbf{U}$, weight function $F : \mathbf{U} \mapsto \mathbb{R}^+$, set of source points $S \subset \mathbf{U}$

Initialization:

1. Initialize the source points with $T(\mathrm{S}) = 0$ and assign them the *Fixed* attribute
2. Initialize the rest of the grid points with $T = \infty$ and assign them the *Unprocessed* attribute

Iteration:

1. Mark all *Unprocessed* vertices sharing a simplex with a *Fixed* vertex as *Close*
2. Update all *Close* vertices from three *Fixed* vertices in the simplex
3. Mark the *Close* vertex with minimum T as *Fixed*
4. If *Unprocessed* $\neq \emptyset$ return to step 1.

Output: The distance map $T : \Omega \mapsto \mathbb{R}^+$.

Table 1
Fast marching algorithm on parametric three-dimensional manifolds.

marching algorithm [Kimmel and Sethian(1998)], a vertex is updated by simulating a planar wavefront propagating inside the triangle; the values of the two supporting vertices allow to compute the front direction. In this paper, we extend this scheme to three dimensions and show that in the particular case of orthogonal grids, it corresponds to the first-order upwind discrete differential operator proposed by Sethian [Sethian(1996)].

Similarly to the two-dimensional case, we show that the discussed update procedure is suitable only for acute simplices. If this is not the case, it may occur that the supporting values are not yet available while updating the vertex, since the wavefront reaches some of the supporting vertices *after* it reaches the updated vertex. In the two dimensional case, Kimmel and Sethian [Kimmel and Sethian(1998)] proposed to "split" obtuse angles by adding virtual connections to other grid points. However, unlike the two-dimensional case where one split is always sufficient to solve the obtuse angle problem, in the three-dimensional case more virtual edges might be required. This fact prevents a simple extension of the two-dimensional splitting scheme to three dimensions and presents the main obstacle in developing an eikonal solver for general three-dimensional manifolds. Here, we introduce a simple procedure for splitting the obtuse angles, working entirely in the parametric domain, which extends the technique used in [Spira and Kimmel(2004)] for two-dimensional manifolds.

It is worthwhile noting that although our algorithm is based on fast marching, the order in which the grid points are updates does not necessarily have to be priority queue-based. In recent studies [Tsai et al.(2003)Tsai, Cheng, Osher,

and Zhao], [Zhao(2005)], [Qian et al.(2006b)Qian, Zhang, and Zhao], [Qian et al.(2006a)Qian, Zhang, and Zhao], and [Kao et al.(2002)Kao, Osher, and Tsai] as well as in classical works dating back to [Danielsson(1980)], an alternating raster scan update order is considered instead. The family of such fast sweeping algorithms is advantageous due to well-structured memory access and easy parallelization, which makes them especially attractive for implementation on single instruction multiple data (SIMD) architectures. The major disadvantage of the fast sweeping algorithms is that the number of iterations required for producing a consistent distance map is data-dependent. Our update scheme and splitting technique can be used within both families of algorithms.

The paper is organized as follows: In Section 2 we present an update scheme based on the planar wavefront model. Section 2.2 is dedicated to numerical stability analysis. In Section 3 we present a splitting procedure used to convert the numerical grid into one consisting of acute simplices only. In Section 4, numerical results assessing the proposed method are presented. Finally, Section 5 concludes the paper.

2 The Update step

Let $\mathbf{u}_0$ be a grid point being updated and let there be a simplex formed by $\mathbf{u}_0$ and other three grid points $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ with fixed times of wavefront arrival T_1, T_2 and T_3. Our goal is to simulate wavefront propagation from the supporting vertices along the simplex and compute the time of arrival T_0 to $\mathbf{u}_0$ [1].

Although the fast marching algorithm works on the orthogonal grid in the parameterization domain $\mathbf{U}$, the update must take into account the true geometry of the simplex. We henceforth assume, without loss of generality, that the simplex parameterized by $\mathbf{u}_i$, $i = 0, ..., 3$, lies in $\mathbb{R}^3$ in such a way that $\mathbf{x}_1 = 0$, $\mathbf{x}_2 = (x_2, 0, 0)^{\mathrm{T}}$, $\mathbf{x}_3 = (x_3, y_3, 0)^{\mathrm{T}}$, and $\mathbf{x}_0 = (x_0, y_0, z_0)^{\mathrm{T}}$, such that $x_2, y_3, z_0 > 0$. We refer to this coordinate system as to *canonical*. Note, that the values x_2, x_3, y_3, x_0, y_0 and z_0 can be expressed in terms of geometrical quantities of the simplex, and can be therefore precomputed for the given grid. We will further assume that the simplex is acute; treatment of obtuse simplices is addressed in Section 3.

We focus on the case where the wave propagation velocity is $F \equiv 1$ in the entire volume. An extension to weighted distance maps can be achieved by

[1] Since the point $\mathbf{u}_0$ might be part of more than one simplex, contributions from all the simplices sharing $\mathbf{u}_0$ should be considered. We choose the one that yields the smallest updated value for T.

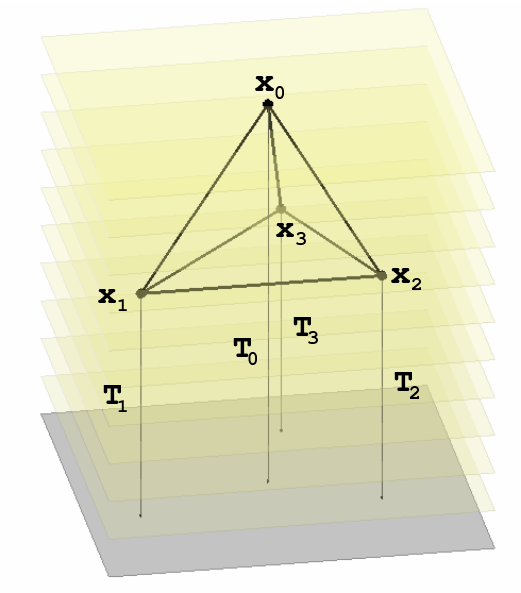

Fig. 2. Planar wavefront approximation scheme.

casting the original eikonal equation $\|\nabla_{\mathbf{G}} T\| = F$ to $\|\nabla_{\tilde{\mathbf{G}}} T\| = 1$, where $\tilde{\mathbf{G}} = \mathbf{G}/F$ represents a modified metric.

We adopt the planar model, assuming the wavefront to be a plane propagating from a (virtual) source described by the equation $\mathbf{n}^{\mathrm{T}}\mathbf{x} + p = 0$, where $\mathbf{n}$ defines the propagation direction. We demand that the supporting vertices $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lie at distances T_1, T_2 and T_3, respectively, from the latter plane (Figure 2, left). Formally, this can be expressed as the following set of coupled equations:

$$\begin{aligned} \mathbf{n}^{\mathrm{T}}\mathbf{x}_1 + p &= T_1 \\ \mathbf{n}^{\mathrm{T}}\mathbf{x}_2 + p &= T_2 \\ \mathbf{n}^{\mathrm{T}}\mathbf{x}_3 + p &= T_3, \end{aligned} \tag{7}$$

which, in the canonical coordinate system, yields $p = T_1$ and

$$\begin{aligned} n_x &= \frac{T_2 - T_1}{x_2} \\ n_y &= \frac{T_3 - T_1}{y_3} - \frac{(T_2 - T_1)x_3}{x_2 y_3} \\ n_z &= \pm\sqrt{1 - n_x^2 - n_y^2}. \end{aligned} \tag{8}$$

The negative solution for n_z corresponds to wavefront propagating downwards and thus reaching $\mathbf{x}_0$ before $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. In order to enforce consistency of the update, the negative solution is discarded [Kimmel and Sethian(1998)]. The value of T_0 is updated according to the distance of the vertex $\mathbf{x}_0$ from the planar source, namely,

$$T_0 = \mathbf{n}^{\mathrm{T}}\mathbf{x}_0 + p \;=\; x_0 n_x + y_0 n_y + z_0 n_z + T_1. \tag{9}$$

In the special case where the grid is Cartesian, one has $\mathbf{x}_1 = (h,0,0)^{\mathrm{T}}$, $\mathbf{x}_2 = (0,h,0)^{\mathrm{T}}$, $\mathbf{x}_3 = (0,0,h)^{\mathrm{T}}$ and $\mathbf{x}_0 = (0,0,0)^{\mathrm{T}}$, or in canonical coordinates, $\mathbf{x}_1 = (0,0,0)^{\mathrm{T}}$, $\mathbf{x}_2 = \left(\sqrt{2}h, 0, 0\right)^{\mathrm{T}}$, $\mathbf{x}_3 = \left(\sqrt{\frac{1}{2}}h, \sqrt{\frac{3}{2}}h, 0\right)^{\mathrm{T}}$ and $\mathbf{x}_0 = \left(\sqrt{\frac{1}{2}}h, \sqrt{\frac{1}{6}}h, \sqrt{\frac{1}{3}}h\right)^{\mathrm{T}}$, where h denotes the grid step. Equation (9) reduces to

$$\begin{aligned} T_0 = &\pm\frac{1}{3}\sqrt{(T_1+T_2+T_3)^2 - 3(T_1^2+T_2^2+T_3^2-h^2)} \\ &+\frac{T_1+T_2+T_3}{3}. \end{aligned} \tag{10}$$

Reformulating the latter result, T_0 can be considered as a solution of

$$\left(\frac{T_0-T_1}{h}\right)^2 + \left(\frac{T_0-T_2}{h}\right)^2 + \left(\frac{T_0-T_3}{h}\right)^2 = 1, \tag{11}$$

which is exactly the three-dimensional extension of the first-order Cartesian fast marching scheme

$$\left(\overline{D}_x T\right)^2 + \left(\overline{D}_y T\right)^2 + \left(\overline{D}_z T\right)^2 = 1 \tag{12}$$

proposed by Sethian [Sethian(1996)], where

$$\overline{D}_x T = \max\left\{D_{-x}T, -D_{+x}T, 0\right\} \tag{13}$$

is the upwind first-order discrete derivative along the x axis, and D_{-x}, D_x are the standard backward and forward discrete derivatives along the x axis. $\overline{D}_y$ and $\overline{D}_z$ are defined in a similar way.

As its two-dimensional analog, the planar update scheme produces a first-order approximation to the continuous distance map. This map is exact when the source is indeed a plane and is particularly accurate for extended surface-shaped sources whose curvature radius is significantly larger than the grid step.

2.1 Monotonicity and consistency

The update of $\mathbf{x}_0$ has to be monotone, i.e. an increase of T_1, T_2 or T_3 should increase T_0. In other words, we require that

$$\frac{\partial T_0}{\partial T_i} > 0 \tag{14}$$

for $i = 1, 2, 3$, or in vector notation,

$$\nabla_T T_0 = (\nabla_T \mathbf{n}^{\mathrm{T}})\mathbf{x}_0 + \nabla_T p > 0. \tag{15}$$

Differentiating equations (7) and $\mathbf{n}^{\mathrm{T}}\mathbf{n} = 1$ with respect to T_i, one obtains $\nabla_T p = (1, 0, 0)^{\mathrm{T}}$ and

$$\nabla_T \mathbf{n}^{\mathrm{T}} = \begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} (\mathbf{x}_2, \mathbf{x}_3, \alpha\mathbf{n})^{-1}, \tag{16}$$

where $\alpha \neq 0$ is some proportion coefficient. Hence,

$$\begin{aligned} \nabla_T T_0 &= \begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_2 & x_3 & \alpha n_x \\ 0 & y_3 & \alpha n_y \\ 0 & 0 & \alpha n_z \end{pmatrix}^{-1} \mathbf{x}_0 + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &= \frac{1}{x_2 y_3 n_z} \begin{pmatrix} -y_3 n_z & n_z(x_3 - x_2) & -x_3 n_y + y_3 n_x + x_2 n_y \\ y_3 n_z & -x_3 n_z & x_3 n_y - y_3 n_x \\ 0 & x_2 n_z & -x_2 n_y \end{pmatrix} \mathbf{x}_0 + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{17}$$

Substituting $\nabla_T \mathbf{n}^{\mathrm{T}}$ and $\nabla_T p$ into (15) and using the fact that $x_2, y_3, n_z > 0$, after some algebraic manipulations, the monotonicity conditions assume the form

$$\mathbf{q}_i^{\mathrm{T}} \mathbf{n} < 0, \tag{18}$$

where

$$\begin{aligned} \mathbf{q}_1 &= \mathbf{x}_3 \times \mathbf{x}_0 \;\; = \;\; -(\mathbf{x}_0 - \mathbf{x}_3) \times (\mathbf{x}_0 - \mathbf{x}_1) \\ \mathbf{q}_2 &= \mathbf{x}_0 \times \mathbf{x}_2 \;\; = \;\; -(\mathbf{x}_0 - \mathbf{x}_1) \times (\mathbf{x}_0 - \mathbf{x}_2) \\ \mathbf{q}_3 &= \mathbf{x}_0 \times \mathbf{x}_3 + \mathbf{x}_2 \times \mathbf{x}_0 - \mathbf{x}_2 \times \mathbf{x}_3 \\ &= -(\mathbf{x}_0 - \mathbf{x}_2) \times (\mathbf{x}_0 - \mathbf{x}_3). \end{aligned} \tag{19}$$

Recognizing in $\mathbf{q}_i$ the normals to the faces of the simplex, the monotonicity conditions have a very simple geometric interpretation: the wavefront propagation direction $\mathbf{n}$ must lie inside the semi-infinite cone created by the simplex.

Yet another requirement is the consistency condition $T_0 > T_i$. Substituting $\mathbf{n}^{\mathrm{T}}\mathbf{x}_0 + T_1 > \mathbf{n}^{\mathrm{T}}\mathbf{x}_i + T_1$ yields

$$\mathbf{n}^{\mathrm{T}}(\mathbf{x}_0, \mathbf{x}_0 - \mathbf{x}_2, \mathbf{x}_0 - \mathbf{x}_3) > 0, \tag{20}$$

meaning that the wavefront propagation direction $\mathbf{n}$ has to form an acute angle with the simplex edges. Since this has to hold for every $\mathbf{n}$ coming from within the simplex, in order to satisfy the consistency condition, the simplex must be acute. Here we address the update of an acute simplex only and defer the issue of obtuse angle splitting to the next session.

Conditions (18) and (20) should guarantee that update is performed only from a simplex that contains the characteristic direction, which makes the update scheme *upwind* [Sethian and Vladimirsky(2000)]. However, since $\mathbf{n}$ is only an approximation of the characteristic direction, it may happen that the conditions are not satisfied although the true characteristic does lie inside the simplex. For a sufficiently small simplex this can happen only if any of the three inner products $\mathbf{n}^{\mathrm{T}}\mathbf{x}_0$, $\mathbf{n}^{\mathrm{T}}(\mathbf{x}_0 - \mathbf{x}_2)$, and $\mathbf{n}^{\mathrm{T}}(\mathbf{x}_0 - \mathbf{x}_3)$ is sufficiently close to zero. This corresponds to the situation in which T_0 can be computed from one of the triangles (two-dimensional simplices) $\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2$, $\mathbf{x}_1\mathbf{x}_0\mathbf{x}_3$, $\mathbf{x}_2\mathbf{x}_0\mathbf{x}_3$, or one of the edges (one-dimensional simplices) $\mathbf{x}_1\mathbf{x}_0$, $\mathbf{x}_2\mathbf{x}_0$, $\mathbf{x}_3\mathbf{x}_0$. In this case, a two-dimensional or a one-dimensional update is performed.

To derive the two-dimensional update scheme, let us assume without loss of generality that T_0 is being updated from the triangle $\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_2$, whose vertices in the canonical coordinates are $\mathbf{x}'_1 = \mathbf{x}_1$, $\mathbf{x}'_2 = \mathbf{x}_2$, and $\mathbf{x}'_0 = (x_0, \sqrt{y_0^2 + z_0^2})$. As in the three-dimensional case, we adopt the planar wavefront model and solve the coupled system of equations

$$\begin{aligned} \mathbf{n}^{\mathrm{T}}\mathbf{x}'_1 + p &= T_1 \\ \mathbf{n}^{\mathrm{T}}\mathbf{x}'_2 + p &= T_2 \\ \mathbf{n}^{\mathrm{T}}\mathbf{n} &= 1, \end{aligned} \tag{21}$$

which, in the canonical coordinate system, yields

$$T_0 = \mathbf{n}^{\mathrm{T}}\mathbf{x}'_0 + p \;=\; T_1 + x_0\frac{T_2 - T_1}{x_2} + y'_0\sqrt{1 - \frac{(T_2 - T_1)^2}{x_2^2}}, \tag{22}$$

where $y_0' = \sqrt{y_0^2 + z_0^2}$. The unit vector $\mathbf{n} \in \mathbb{R}^2$ expresses the wavefront propagation direction in the triangle plane. Derivation of the monotonicity condition closely follows that of the three-dimensional update scheme. Differentiating equations (21) with respect to T_i, one obtains $\nabla_T p = (1, 0)^{\mathrm{T}}$ and

$$\nabla_T \mathbf{n}^{\mathrm{T}} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} (\mathbf{x}_2, \alpha\mathbf{n})^{-1}, \tag{23}$$

where $\alpha \neq 0$ is some proportion coefficient. Hence,

$$\nabla_T T_0 = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_2 & \alpha n_x \\ 0 & \alpha n_y \end{pmatrix}^{-1} \mathbf{x}_0 + \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{x_2 n_y} \begin{pmatrix} x_2 n_y + y_0' n_x - x_0 n_y \\ x_0 n_y - y_0' n_x \end{pmatrix}. \tag{24}$$

Demanding $\nabla_T T_0 > 0$, we get

$$\begin{aligned} (-y_0', x_0)\mathbf{n} &> 0 \\ (y_0', x_2 - x_0)\mathbf{n} &> 0, \end{aligned} \tag{25}$$

which can be interpreted geometrically as requiring $\mathbf{n}$ to come from within the triangle $\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_2$. The consistency conditions for the two-dimensional update are

$$T_0 - T_i = \mathbf{n}^{\mathrm{T}}(\mathbf{x}_0', \mathbf{x}_0' - \mathbf{x}_1) > 0. \tag{26}$$

Like in the three-dimensional case, requiring (26) to be satisfied by every $\mathbf{n}$ coming within the triangle $\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_2$ requires the angle $\angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2$ to be acute. The latter demand is satisfied if the simplex is acute.

If conditions (25),(26) are satisfied in none of the three triangles, the path is restricted to the simplex edges, and the trivial Dijkstra-type update

$$T_0 = \min\left\{T_1 + \|\mathbf{x}_1 - \mathbf{x}_0\|, T_2 + \|\mathbf{x}_2 - \mathbf{x}_0\|, T_3 + \|\mathbf{x}_3 - \mathbf{x}_0\|\right\}$$

is performed. The complete update step is summarized in Table 2.

Input: The simplex vertices coordinates $\mathtt{x}_1$, $\mathtt{x}_2$, $\mathtt{x}_3$ and $\mathtt{x}_0$; the arrival times T_1, T_2, and T_3 of the supporting vertices, and the current value T_0.

1. If conditions (18),(20) hold, compute T_0' according to (9).
   else
   1.1. If conditions (25),(26) are satisfied for the triangle $\mathtt{x}_0\mathtt{x}_1\mathtt{x}_2$, compute T_0' according to (22).
   1.2. Else if conditions (25),(26) are satisfied for the triangle $\mathtt{x}_0\mathtt{x}_1\mathtt{x}_3$, compute T_0' according to (22).
   1.3. Else if conditions (25),(26) are satisfied for the triangle $\mathtt{x}_0\mathtt{x}_2\mathtt{x}_3$, compute T_0' according to (22).
   1.4. Else, set

$$T_0' = \min\left\{T_1 + \|\mathbf{x}_1 - \mathbf{x}_0\|, T_2 + \|\mathbf{x}_2 - \mathbf{x}_0\|, T_3 + \|\mathbf{x}_3 - \mathbf{x}_0\|\right\}$$

2. Output the time of arrival $\min\{T_0, T_0'\}$.

Table 2
The three-dimensional acute simplex update scheme.

2.2 Numerical stability

So far, we have shown that our update scheme yields a consistent solution to the eikonal equation assuming infinite-precision arithmetics. However, round-off and truncation errors typical to finite-precision arithmetics may compromise the *numerical* stability of the algorithm.

Let us assume that T_i is affected by a small numerical error ϵ, which, in turn, influences the computed time of arrival T_0. Using first-order Taylor expansion [2] in the three-dimensional update scheme,

$$\tilde{T}_0 \approx T_0 + \frac{\partial T_0}{\partial T_i}\epsilon \;\le\; T_0 + \left(\left|\frac{\partial T_0}{\partial T_1}\right| + \left|\frac{\partial T_0}{\partial T_2}\right| + \left|\frac{\partial T_0}{\partial T_3}\right|\right)\epsilon. \tag{27}$$

Observe that from (20) it follows that $\frac{\partial T_0}{\partial T_1} + \frac{\partial T_0}{\partial T_2} + \frac{\partial T_0}{\partial T_3} = 1$. When monotonicity conditions (18) are satisfied, one has $\frac{\partial T_0}{\partial T_i} \ge 0$, and consequently $\tilde{T}_0 \le T_0 + \epsilon$. Since the error in T_0 does not grow with the distances T_i, the three-dimensional update is numerically stable.

[2] Arrival time T_0 may not be differentiable. In such a case, the maximum over the left- and right-sided derivatives of T_0 with respect to T_i must be used instead of $\frac{\partial T_0}{\partial T_i}$ and the presented analysis is still valid. We are grateful to one of our referees for noting this issue.

Let us now examine the two-dimensional update, assuming without loss of generality that T_0 is being updated from the triangle $\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_2$. It is easy to observe from (24) that $\frac{\partial T_0}{\partial T_1} + \frac{\partial T_0}{\partial T_2} = 1$, and we obtain again that $\tilde{T}_0 \leq T_0 + \epsilon$. Since the one-dimensional update $T_0 = T_i + \|\mathbf{x}_i - \mathbf{x}_0\|$ does not amplify numerical errors, we conclude that the entire update scheme is numerically stable.

3 Splitting obtuse angles

For three-dimensional manifolds with non-trivial metric, we are liable to encounter obtuse simplices, which cause inconsistent update. In fact, observe that when 6-neighborhood connectivity is used, four of the eight simplices formed at $\mathbf{x}_0$ will have at least one non-acute angle at $\mathbf{x}_0$. The situation is somehow ameliorated when higher-order connectivity is used, but obtuse simplices are usually inevitable. This section describes a preprocessing algorithm, which creates an acute numerical grid by adding virtual connections to non-neighboring grid points.

Unlike the two-dimensional case, a three-dimensional simplex can have from zero to three obtuse angles at the updated vertex. The splitting procedure has to connect the updated grid point $\mathbf{u}_0$ to, at least, one grid point, so that

- C1. the obtuse simplex $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ is replaced by at least two acute simplices formed at $\mathbf{x}_0$; and
- C2. the new simplices span a spatial angle containing that spanned by the original simplex.

The first condition guarantees that the produced simplices are acute; the second condition guarantees that the splitting will not limit the set of directions, from where the vertex $\mathbf{x}_0$ can be updated.

3.1 Splitting simplices with one obtuse angle

We start with the case where the simplex has only one obtuse angle at $\mathbf{x}_0$ (without loss of generality, $\angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2 > 90°$, or, equivalently, $e_{12} < 0$). In this case, a single virtual connection of $\mathbf{u}_0$ to some $\mathbf{u}_4 = \mathbf{u}_0 + \mathbf{m}_4$, $\mathbf{m}_4 \in \mathbb{Z}^3$, is sufficient to perform the splitting. By condition (C1), the virtual edge $\mathbf{x}_0\mathbf{x}_4$ has to form acute angles with $\mathbf{x}_0\mathbf{x}_1$, $\mathbf{x}_0\mathbf{x}_2$ and $\mathbf{x}_0\mathbf{x}_3$, which holds if and only if

$$\mathbf{m}_i^{\mathrm{T}}\mathbf{G}\mathbf{m}_4 \geq 0 \tag{28}$$

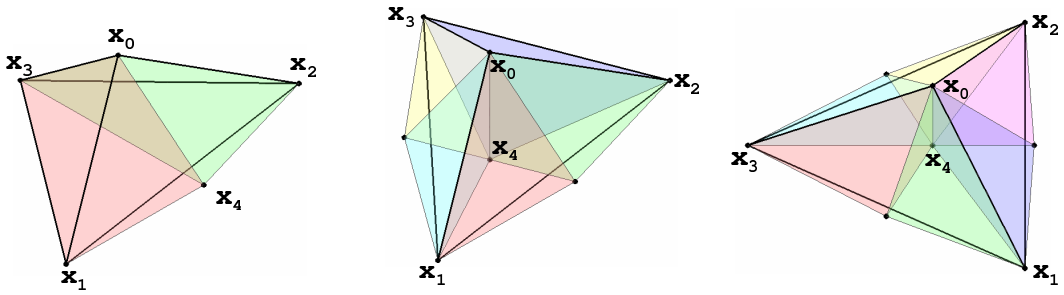

Fig. 3. Splitting a vertex $\mathbf{x}_0$ with one (left), two (middle), and three (right) obtuse angles.

for $i = 1, 2, 3$. Condition (C2) is satisfied by the requirement that the vector $\mathbf{x}_4$ lies outside the face $\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2$ (Figure 3, left). Let us denote by $\mathbf{n} = \mathbf{x}_1 \times \mathbf{x}_2$ the normal to the face $\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2$ pointing outside the simplex; $\mathbf{x}_4$ has to form an acute angles with $\mathbf{n}$, namely

$$\mathbf{n}^{\mathrm{T}}\mathbf{m}_4 \geq 0. \tag{29}$$

Our goal is to satisfy constrains (28), (29) with the shortest possible connection. This gives rise to the minimization problem

$$\min_{\mathbf{m}_4 \in \mathbb{Z}^3} \|\mathbf{m}_4\|_1 \text{ s.t. } \mathbf{A}\mathbf{m}_4 \geq \epsilon, \tag{30}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{M}^{\mathrm{T}}\mathbf{G} \\ \mathbf{n}^{\mathrm{T}} \end{pmatrix}, \tag{31}$$

and $\epsilon > 0$ is a small constant used to disallow the trivial solution $\mathbf{m}_4 = 0$. The choice of the ℓ_1 norm is especially convenient, since it allows to reformulate the problem as an *integer linear program* (ILP). Indeed, let us define $m^+, m^- \geq 0$, so that $m = m^+ - m^-$ and $|m| = m^+ + m^-$. Then, the problem can be reformulated as

$$\min_{\mathbf{m}_4^+, \mathbf{m}_4^- \in \mathbb{Z}^3} \mathbf{1}^{\mathrm{T}}\mathbf{m}_4^+ + \mathbf{1}^{\mathrm{T}}\mathbf{m}_4^- \text{ s.t. } \begin{cases} \mathbf{A}\left(\mathbf{m}_4^+ - \mathbf{m}_4^-\right) \geq \epsilon \\ \mathbf{m}_4^+, \mathbf{m}_4^- \geq 0. \end{cases} \tag{32}$$

where $\mathbf{1} = (1, 1, 1)^{\mathrm{T}}$. Once the vertex $\mathbf{x}_4$ is found, the simplex $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ can be split into $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_3\mathbf{x}_4$ and $\mathbf{x}_0\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4$, both of which are acute. We postpone the discussion on how to practically solve (32) to Section 3.3.

3.2 Splitting simplices with two or three obtuse angles

Let us now assume that the simplex has two acute angles at $\mathbf{x}_0$, without loss of generality, $\angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_2, \angle\mathbf{x}_1\mathbf{x}_0\mathbf{x}_3 > 90^\circ$ (or, equivalently, $e_{12}, e_{13} < 0$). Obviously, one virtual edge is insufficient to satisfy condition (C1) without violating (C2). Sacrificing optimality, we first construct a virtual connection to some point $\mathbf{x}_4$, satisfying (C1) and lying inside the basis $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$, thus, violating condition (C2). Such a connection can be found by solving

$$\min_{\mathbf{m}_4 \in \mathbb{Z}^3} \|\mathbf{m}_4\|_1 \,\text{s.t.}\, \mathbf{M}^{\mathrm{T}}\mathbf{G}\mathbf{m}_4 \geq \epsilon, \tag{33}$$

which, as before, can be reformulated as an integer linear program. Once the vertex $\mathbf{x}_4$ is found, the simplex is split into three simplices, $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4$, $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_3\mathbf{x}_4$ and $\mathbf{x}_0\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4$. The latter one is acute, since $e_{23} \geq 0$, whereas the two former ones are obtuse, since $e_{12}, e_{13} < 0$, yet have only *one* obtuse angle. Therefore, the procedure for splitting a simplex with one obtuse angle described before can be applied to each of the two obtuse simplices, resulting in a total of five simplices (Figure 3, middle). In the same manner, one can split a simplex with three obtuse angles, obtaining a total of six simplices (Figure 3, right). We note that the described procedure might generate more splits than required theoretically. However, the overhead does not appear very significant.

3.3 Heuristic simplex splitting algorithm

Though solution of ILPs is believed to be an NP-hard problem, here, the number of optimization variables is constant (six). Therefore, splitting a single simplex by solving (32) or (33) can be considered to have constant complexity for any practical purpose. However, the particular geometry of the problem allows to obtain a suboptimal solution using a simple heuristic algorithm instead of resorting to a cumbersome general ILP solver. Here, we briefly outline an algorithm for an appoximate solution of problem (33), where the objective is to minimize $\|\mathbf{m}\|_1$ over integer triples $\mathbf{m} = (m, n, p)^{\mathrm{T}}$ lying inside the semi-unbounded cone $\mathbf{B}\mathbf{m} \geq 0$, $\mathbf{B} = \mathbf{M}^{\mathrm{T}}\mathbf{G}$. The cone is given by the intersection of three planes $\mathbf{b}_i^{\mathrm{T}}\mathbf{m} = 0$, where $\mathbf{b}_i^{\mathrm{T}}$ are the rows of $\mathbf{B}$, or, equivalently, by the set of three rays $\mathbf{r}_1 = \mathbf{b}_3 \times \mathbf{b}_1, \mathbf{r}_2 = \mathbf{b}_1 \times \mathbf{b}_2$, and $\mathbf{r}_3 = \mathbf{b}_2 \times \mathbf{b}_3$ emanating from the origin. We define

$$\Delta_i = \text{sign } r_1^i \;+\; \text{sign } r_2^i \;+\; \text{sign } r_3^i. \tag{34}$$

Input: Vectors $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$.

1. For $p = 1, 2, ...$
   2. Project $\mathbf{r}_i$ onto the plane $u^3 = p$ according to (35). Order the resulting vectors so that $v_1^1 \leq v_2^1 \leq v_3^1$.
   3. Compute
$$P = \max\left\{\frac{\mathbf{v}_2^2 - \mathbf{v}_1^2}{\mathbf{v}_2^1 - \mathbf{v}_1^1}, \frac{\mathbf{v}_3^2 - \mathbf{v}_1^2}{\mathbf{v}_3^1 - \mathbf{v}_1^1}\right\}$$
$$Q = \min\left\{\frac{\mathbf{v}_2^2 - \mathbf{v}_1^2}{\mathbf{v}_2^1 - \mathbf{v}_1^1}, \frac{\mathbf{v}_3^2 - \mathbf{v}_1^2}{\mathbf{v}_3^1 - \mathbf{v}_1^1}\right\}$$
$$R = \frac{\mathbf{v}_3^2 - \mathbf{v}_2^2}{\mathbf{v}_3^1 - \mathbf{v}_2^1}$$
$$D = v_1^2 + Q(v_2^1 - v_1^1) - v_2^2$$
   4. For $m = \lceil v_1^1 \rceil, ..., \lfloor v_3^1 \rfloor$
      5. Compute
$$\alpha = v_1^2 + P(m - v_1^1)$$
$$\beta = v_1^2 + Q(m - v_1^1)$$
$$\gamma = v_2^2 + R(m - v_2^1)$$
      6. If ($D < 0$ And $\min\{\lfloor\alpha\rfloor, \lfloor\gamma\rfloor\} \geq \lceil\beta\rceil$) Or ($D > 0$ And $\lfloor\alpha\rfloor \geq \max\{\lfloor\gamma\rfloor, \lfloor\beta\rfloor\}$), Output: $(m, n = \lfloor\alpha\rfloor, p)$ and stop

Table 3
Heuristic algorithm for approximate solution of ILP (33) under the assumption $\Delta_3 = 3$.

Existence of an i such that $|\Delta_i| = 3$ is manifest of the fact that the i-th elements of the vectors $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are aligned in the same direction ($\pm\mathbf{u}^i$ depending on the sign of Δ_i).

Let us first explore the case where, without loss of generality, $\Delta_3 = +3$. The cone formed by $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ intersects the plane $u^3 = p > 0$ at a triangle, whose vertices are given by

$$\mathbf{v}_i = \left(\frac{pr_i^1}{r_i^3}, \frac{pr_i^2}{r_i^3}, p\right)^{\mathrm{T}} \tag{35}$$

(Figure 4, left). We increase p in unit steps starting from $p = 1$, until the triangular intersection region contains at least one integer point. Table 3 outlines an algorithm to perform this task.

Obviously, the increase of p enlarges the intersection region. A pessimistic bound on p can be obtained by demanding the radius of the incircle of $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ to be larger than $\frac{\sqrt{2}}{2}$. This yields

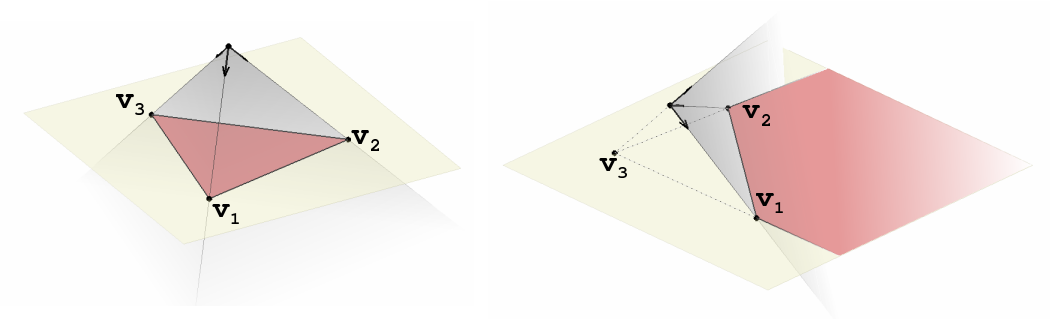

Fig. 4. The semi-infinite cone $\mathbf{Bm} \geq 0$ (grayed) can intersect a plane parallel to one of the axes $\mathbf{u}_i$ at a triangle (left) or a semi-infinite region (right).

$$p \leq \left\lceil \frac{P_1}{2\sqrt{2}S_1} \right\rceil, \tag{36}$$

where S_1 and P_1 are the area and the perimeter of triangle $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ for $p = 1$, respectively.

Let us now address the more exotic case when $|\Delta_i| < 3$ for $i = 1, 2, 3$, i.e. the cone does not intersect any of the planes normal to one of the axes $\mathbf{u}_i$ at more than two points. Let us assume without loss of generality that $\mathbf{r}_2$ and $\mathbf{r}_3$ intersect the plane $u^3 = 1$ at two points $\mathbf{v}_2$, $\mathbf{v}_3$ given, as before, by (35). Although the ray $\mathbf{r}_1$ does not intersect the plane, $-\mathbf{r}_1$ does at $\mathbf{v}_1$ (defined as before). In this case, the intersection of the cone with the plane is the semi-infinite region formed by the rays $\mathbf{v}_1\mathbf{v}_2$, $\mathbf{v}_1\mathbf{v}_3$ emanating from $\mathbf{v}_1$ and excluding the interior of triangle $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3$ (Figure 4, right). Fixing $p = 1$ and running steps 3-6 of the algorithm described in Table 3 while reversing the sign of D, handles this special case as well.

Last, we address the suboptimal solution of ILP (32). This problem is very similar to (33), except that it has four linear constraints instead of three. The latter can be interpreted as finding the shortest integer vector $\mathbf{m}$ in a semi-infinite cone with quadrilateral section. The cone can be split into two cones with triangular section, which makes the previously described algorithm applicable.

4 Numerical results

Three numerical experiments were performed to assess the accuracy of the proposed algorithm. In the first experiment, a distance map was computed from a point source on an orthogonal $31 \times 31 \times 31$ grid with step $h = 1/31$, 6-neighbor connectivity, and non-uniform weight varying according to

$$F(\mathbf{x}) = \begin{cases} 1.8 & : 0.1225\sin(4\pi x^1) + 1.2x^2 + 0.05 < 0.33 \\ 0.9 & : 0.33 \leq 0.1225\sin(4\pi x^1) + 1.2x^2 + 0.05 < 0.66 \\ 1.08 & : 0.66 \leq 0.1225\sin(4\pi x^1) + 1.2x^2 + 0.05 < 1 \\ 4 & : 0.1225\sin(4\pi x^1) + 1.2x^2 + 0.05 \geq 1 \end{cases} \tag{37}$$

The level sets of the computed distance function are depicted in Figure 5.

In the second experiment, a non-weighted distance map was measured from a point source on a non-Euclidean volume

$$x^4 = 0.9\sin(2\pi(x^1 - 0.5))\sin(2\pi(x^2 - 0.5))\sin(2\pi(x^3 - 0.5)) \tag{38}$$

for $x^1, x^2, x^3 \in [0, 1]$ discretized on a $31 \times 31 \times 31$ grid. 18-neighbor grid connectivity was used and obtuse angle splitting was performed. The computed distance map is presented in Figure 6. Similar results were obtained using the proposed technique with a raster scan grid update order.

In the third experiment, the fast marching algorithm was executed in a Euclidean hyperplane with obtuse tessellation discretized on grids of varying sizes ranging from 5^3 to 51^3. Analytic geodesic distances were used to compute the numerical errors. From Figure 7 it is evident that the algorithm has first-order convergence when the splitting technique is employed, and no convergence when the obtuse tessellation is used as is. Figure 8 shows the level sets of the distance function computed on an obtuse 31^3 tessellation with two angles of 120° with and without obtuse angle splitting.

5 Conclusions

We presented an extension of the fast marching method for efficient computation of distance maps on parametric three-dimensional curved manifolds. For that goal, we extended the technique proposed in [Spira and Kimmel(2004)] as an alternative to unfolding triangles in cases of obtuse angles in two-dimensional manifolds. We presented an extension of the update step to three dimensions based on the planar wavefront propagation model. Numerical simulations show that the proposed method achieves plausible results in solving the eikonal equation on both Euclidean and non-Euclidean three-dimensional manifolds. Our method can be used both in the priority queue (fast marching type) and raster scan-based (fast sweeping type) families of algorithms. In sequel works, we intend to use our three-dimensional eikonal solver to track white matter fiber bundles in diffusion tensor MRI medical data.

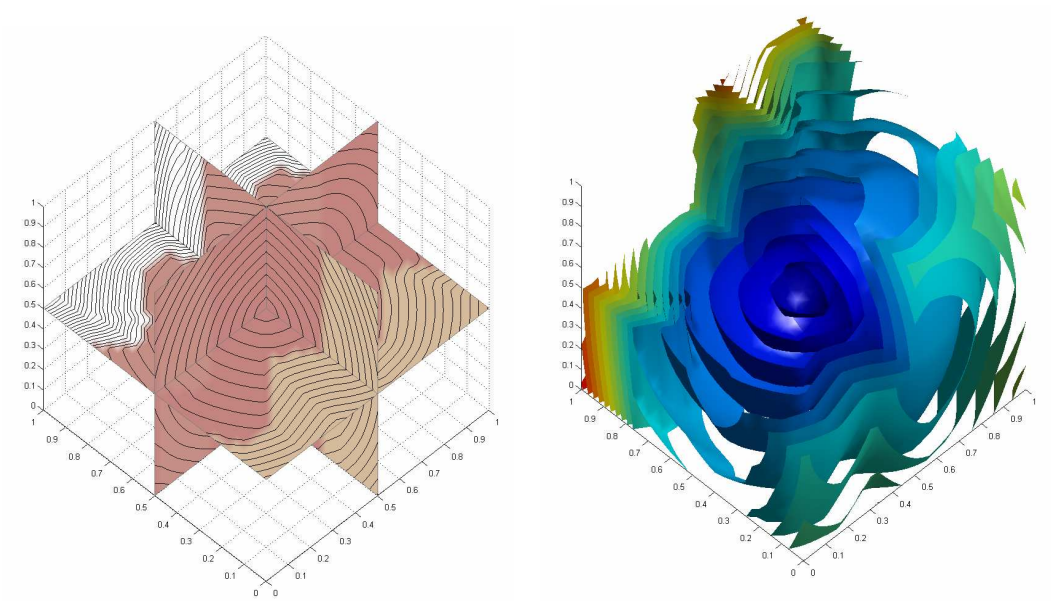

Fig. 5. Distance map from a point source in a Euclidean domain with non-uniform weight F, computed using the fast marching algorithm on a $31 \times 31 \times 31$ grid. Left: slices through the volume with equi-distance contours with step 0.05; color shading represent the configuration of the weight. Right: equi-distant surfaces (distance function level sets) with step 0.1.

References

[Danielsson(1980)] Danielsson, P., 1980. Euclidean distance mapping. Computer Graphics and Image Processing 14, 227–248.

[Kao et al.(2002)Kao, Osher, and Tsai] Kao, C., Osher, S., Tsai, Y., 2002. Fast sweeping methods for static Hamilton-Jacobi equations. Tech. rep., Department of Mathematics, University of California, Los Angeles.

[Kimmel and Sethian(1998)] Kimmel, R., Sethian, J. A., 1998. Computing geodesic paths on manifolds. Proc. of National Academy of Sciences 95 (15), 8431–8435.

[Mémoli and Sapiro(2001)] Mémoli, F., Sapiro, G., 2001. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. Journal of Computational Physics 173 (1), 764–795.

[Qian et al.(2006a)Qian, Zhang, and Zhao] Qian, J., Zhang, Y., Zhao, H., 2006a. A fast sweeping method for static convex Hamilton-Jacobi equations. SIAM Journal on Numerical AnalysisTo appear in SINUM.

[Qian et al.(2006b)Qian, Zhang, and Zhao] Qian, J., Zhang, Y., Zhao, H., 2006b. Fast sweeping methods for eikonal equations on triangulated meshes. SIAM Journal on Numerical AnalysisTo appear.

[Sethian(1996)] Sethian, J. A., 1996. A fast marching level set method for monotonically advancing fronts. Proc. of National Academy of Sciences 93 (4), 1591–1595.

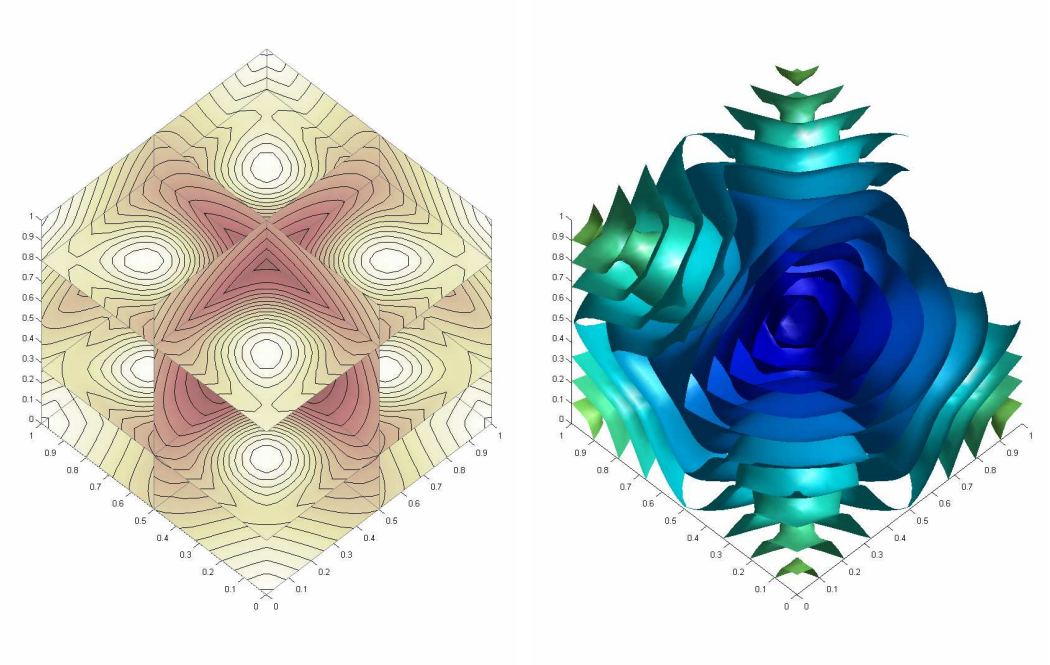

Fig. 6. Distance map from a point source in a non-Euclidean domain given by $x^4 = 0.9\sin(2\pi(x^1 - 0.5))\sin(2\pi(x^2 - 0.5))\sin(2\pi(x^3 - 0.5))$, computed using the fast marching algorithm on a $31 \times 31 \times 31$ grid. Left: slices through the volume with equi-distance contours with step 0.05; color shading represent the configuration of the slowness field. Right: equi-distant surfaces (distance function level sets) with step 0.1.

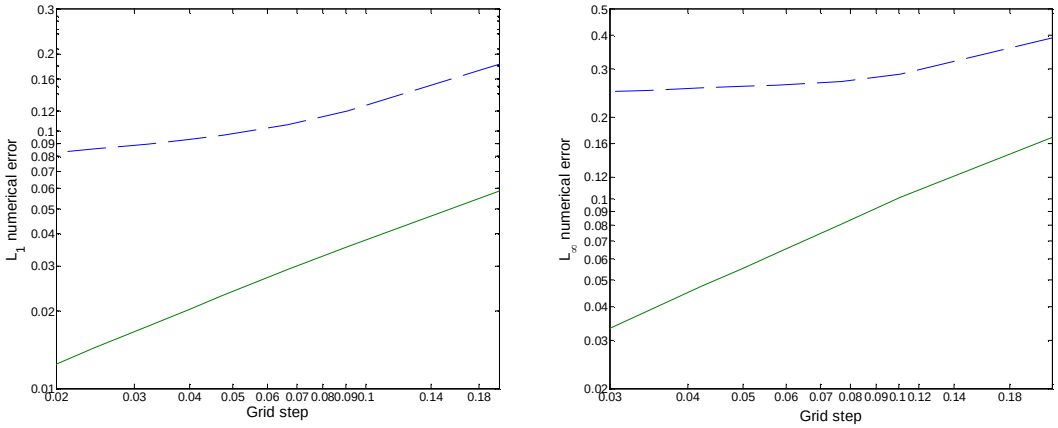

Fig. 7. Numerical error (left: L_1; right: L_∞) as function of the grid step for the fast marching algorithm on an obtuse tessellation with obtuse angle splitting (solid) and without obtuse angle splitting (dashed).

[Sethian and Vladimirsky(2000)] Sethian, J. A., Vladimirsky, A., 2000. Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes. PNAS 97 (11), 5699–5703.

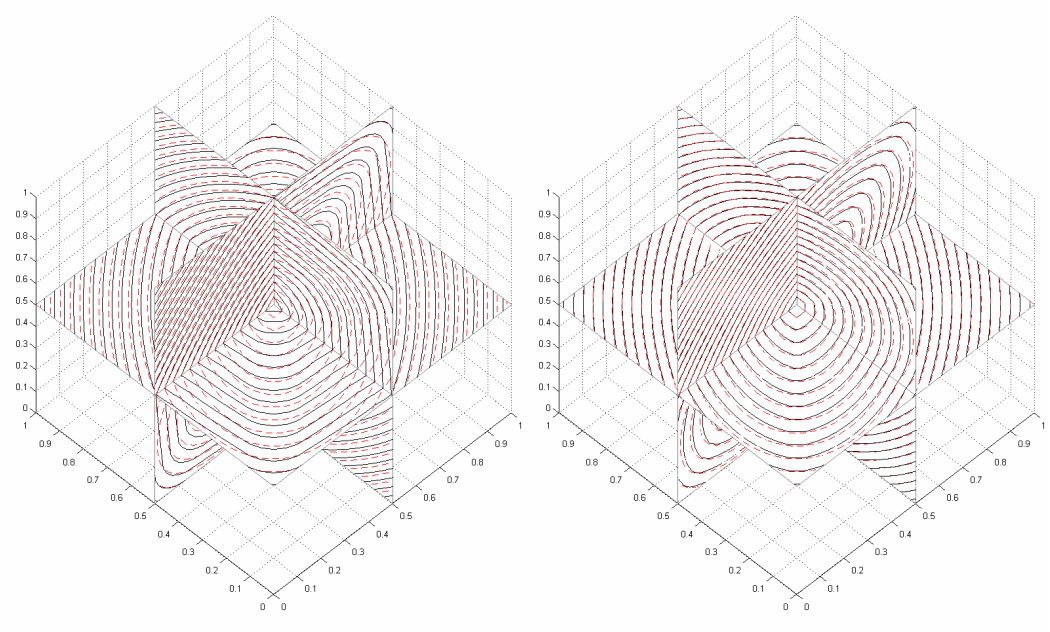

Fig. 8. Equi-distant contours computed by the fast marching algorithm on an obtuse tessellation without obtuse angle splitting (left) and with obtuse angle splitting (right). The true level sets of the distance function are shown in dashed red.

[Spira and Kimmel(2004)] Spira, A., Kimmel, R., 2004. An efficient solution to the eikonal equation on parametric manifolds. Interfaces and Free Boundaries 6 (4), 315–327.

[Tsai et al.(2003)Tsai, Cheng, Osher, and Zhao] Tsai, Y., Cheng, L., Osher, S., Zhao, H., 2003. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. SIAM Journal on Numerical Analysis 41 (2), 673–694.

[Tsitsiklis(1995)] Tsitsiklis, J. N., 1995. Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 40 (9), 1528–1538.

[Zhao(2005)] Zhao, H. K., 2005. Fast sweeping method for eikonal equations. Mathematics of Computation 74, 603–627.