Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 225 (2007) 1388–1408

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A semi-Lagrangian scheme for the curve shortening flow in codimension-2

E. Carlini [a], M. Falcone [a], R. Ferretti [b,*]

[a] *Dipartimento di Matematica, Università di Roma ''La Sapienza'', P.le Aldo Moro 2, I-00185 Roma, Italy*
[b] *Dipartimento di Matematica, Università di Roma Tre, Largo S. Leonardo Murialdo 1, I-00146 Roma, Italy*

Received 19 July 2006; received in revised form 24 January 2007; accepted 27 January 2007
Available online 6 February 2007

Abstract

We consider the model problem where a curve in $\mathbb{R}^3$ moves according to the mean curvature flow (the curve shortening flow). We construct a semi-Lagrangian scheme based on the Feynman–Kac representation formula of the solutions of the related level set geometric equation. The first step is to obtain an approximation of the associated codimension-1 problem formulated by Ambrosio and Soner, where the squared distance from the initial curve is used as initial condition. Since the ε-sublevel of this evolution contains the curve, the next step is to extract the curve itself by following an optimal trajectory inside each ε-sublevel. We show that this procedure is robust and accurate as long as the ''fattening'' phenomenon does not occur. Moreover, it can still single out the physically meaningful solution when it occurs.

AMS classification: 60H30; 65M25; 65M06; 74K10

Keywords: Mean curvature motion; Curve shortening; Semi-Lagrangian scheme

1. Introduction

We consider the problem of numerical approximation for the shortening flow of a curve in $\mathbb{R}^3$ by mean curvature motion (MCM). In particular, we develop a scheme in the framework of *level-set* models.

Geometrically the mean curvature vector points in the direction where the length of the curve decreases most. This results in the evolution which shortens the curve in the fastest way.

It is well known that the flow by mean curvature may develop singularities in finite time even if the initial manifold is smooth. This tends to rule out a parametric approach. In addition, one might be especially interested in the motion of curves that exhibit topological changes, i.e. merging and breaking. In the case of curves in $\mathbb{R}^3$, the behaviour should be to shorten the curve length, this meaning that when two segments of the curve

[*] Corresponding author. Tel.: +39 0654888218/3290571477; fax: +39 0654888080.
E-mail addresses: carlini@mat.uniroma1.it (E. Carlini), falcone@mat.uniroma1.it (M. Falcone), ferretti@mat.uniroma3.it (R. Ferretti).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.01.028

touch, the curve breaks and reconnects according to the acute angle direction (see the discussion of this point in [5]).

To set our contribution into perspective, let us recall that numerical approximations for MC motion of curves in $\mathbb{R}^3$ have been studied by several authors. Dziuk et al. [12–15] have obtained *a priori* convergence results for a finite element approximation which also suits the anisotropic case (see also [23] and the references therein). Although their schemes can handle singularities (like self-intersecting curves in $\mathbb{R}^2$), theoretical results require more regularity. More recently a codimension-2 version of the Merriman, Bence, Osher algorithm [22] has been proposed by Ruuth et al. [26]. The above approach has all the advantages of level set methods in capturing the curve motion, including topological changes, but is still impractical for generating highly accurate solutions since the local truncation error is $O(1/|\log(\Delta t)|)$. In a different method proposed by Burchard et al. [5], the curve is represented by the intersection of the zero level sets of two functions ϕ and ψ (more generally, a smooth manifold with codimension-k can be represented as the intersection of the zero level sets of k scalar functions). The motion of the curve is accomplished by evolving ϕ and ψ in $\mathbb{R}^3$ according to a suitable system of PDEs. Although the current state of the theory of viscosity solutions does not provide well-posedness results for this type of systems, such a representation has been successfully used for numerical simulations [5] (see also [11] for other generalizations). Even if we deal with isotropic motion, we should mention that the above approaches have also been applied to anisotropic curve motion, as well as to the motion of triple junctions.

The main goal of this paper is to give an approximation scheme for the curve shortening flow extending to codimension-2 the scheme for MCM in codimension-1 proposed by Falcone and Ferretti [18]. It is interesting to note that the scheme can be interpreted as a discretization of the Feynman–Kac representation formula for level set solutions of the geometric equation. The continuous representation formula was proved by Soner and Touzi [29] (the same representation formula has been obtained by Buckdahn et al. in [4] for the case of codimension-1 mean curvature flow). More recently, Kohn and Serfaty [21] have given another control-theoretic interpretation of motion by mean curvature showing that the solution of the degenerate parabolic problem associated to the level set formulation can be approximated by a family of discrete time, two-persons games (see the Introduction in [21] for a discussion of the relationship between the games and the stochastic control viewpoints).

In our approach to the codimension-2 problem, we take advantage of the formulation and results of Ambrosio and Soner [1], who have proved (see the Appendix for details) that an approximation of the curve shortening flow can be obtained by embedding the problem in a new codimension-1 evolutive problem using the squared distance from the original curve as initial condition. Taking the ε-sublevel set of the solution to this problem, the curve evolution is obtained in the limit for vanishing ε. This result is also useful for our approximation since we first compute the solution of the codimension-1 problem and then use this information to reconstruct the curve which is "inside" the ε-sublevel set. In particular, the reconstruction of the curve is based on the fact that the curve at time t is the 0-level set of the nonnegative solution $u(x,t)$, i.e. it is the set of the minimum points of u. Then, the algorithm for the curve reconstruction starts from a point on the curve and follows the minima by solving a suitable sequence of constrained minimization problems. In principle, this algorithm should not work properly if the "fattening" phenomenon occurs, as in the example of two linked circles analysed by Bellettini et al. [2]. However, we will present a couple of examples in which the algorithm selects the evolution corresponding to the fastest shortening.

The outline of the paper is as follows. In Section 2 we present the origin of our semi-Lagrangian scheme for the case of codimension 1. In Section 3 we describe the construction of the schemes in codimension 2 and in Section 4 their consistency analysis. In Section 5 some numerical tests are presented. Finally, we briefly review in the appendix the Ambrosio–Soner formulation of the MCM in arbitrary codimension.

2. Front propagation and SL- schemes in codimension 1

The level set method has been used for a long time in the approximation of front propagation problems (see the books [27,24]). In the case of a front propagating in the normal direction and driven by a known speed $c(x,t)$ it gives rise to the following first order Hamilton–Jacobi equation:

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

where u_0 is a continuous function having the initial configuration of the front Γ_0 as its zero level set (or a prescribed level set with a nonzero gradient on it). Although the discretization of such equation has been first obtained by finite differences, also semi-Lagrangian methods have been successfully applied, e.g. in [16,31,17]. The basic idea which is behind semi-Lagrangian approximations is to follow the characteristics of the problem by an ODE scheme and then compute the value at the foot of the characteristic by a numerical reconstruction based on grid values [30]. A simple way to make characteristics appear for problem (1) is to rewrite it as

$$\begin{cases} u_t + \max\limits_{a\in B(0,1)} \{c(x,t)a\cdot Du\} = 0 & \text{in } \mathbb{R}^n \times [0,\infty), \\ u(x,0) = u_0(x), \end{cases} \tag{2}$$

where $B(0,1) = \{x \in \mathbb{R}^n : |x| \leqslant 1\}$ is the unit ball in $\mathbb{R}^n$.

Naturally, the maximum is attained for the direction $a^*(x) = Du(x)/|Du(x)|$ which gives the normal direction to the front at every point. As in [17], this formulation can be applied to derive a scheme which at every point of the grid computes the solution following for one step all possible directions and reconstructing the values at the feet of characteristics $z(x,t,a,\Delta t) \equiv x - \Delta t c(x,t)a$ by interpolation. In $\mathbb{R}^2$, this corresponds to the scheme

$$\begin{cases} u_j^{n+1} = \min\limits_{a\in B(0,1)} \{I[u^n](z(x_j,t,a,\Delta t))\} & \text{in } \mathbb{R}^2 \times [0,\infty), \\ u(x,0) = u_0(x), \end{cases} \tag{3}$$

where $I[u^n](x)$ denotes a numerical reconstruction (by interpolation) performed at the point x on the discrete solution u^n, and we have adopted the standard notation $u_j^n = u(x_j,t_n)$, assuming for simplicity a structured grid $G = \{x_j : x_j = j\Delta x,\ j \in \mathbb{Z}^2\}$ and $t_n = n\Delta t$. It is interesting to note that this scheme represents a discrete version of the Lax–Hopf representation formula (see [17]). Whenever the speed c has constant sign this corresponds also to the Huygens principle and has a strong connection with the classical minimum time problem where a dynamical system $\dot{x} = -c(x,t)a$ controlled by the parameter $a \in B(0,1)$ has to be driven to a given target Ω_0 (in the front propagation problem, this is the internal region entoured by Γ_0, see [16,19]). The scheme (3) has a built-in up-wind correction which is obtained by computing a maximum at every point. This clearly results in an additional cost which can be reduced adopting efficient optimization methods as the one in [3]. However, the fact that the normal direction is computed more accurately with respect to finite difference schemes improves the global accuracy of the approximation.

When the evolution of the front is driven by its curvature the situation becomes more complex, but we can still follow the same approach. Formally, we can replace the given velocity in (2) by the curvature K setting $c(x,t) = K(x,t)$. Since the curvature depends on u the level set formulation leads to the second order Hamilton–Jacobi equation related to the MC flow:

$$\begin{cases} u_t - \operatorname{div}\left(\frac{Du}{|Du|}\right)|Du| = 0 & \text{in } \mathbb{R}^n \times [0,\infty), \\ u(x,0) = u_0(x), \end{cases} \tag{4}$$

where again the initial condition u_0 is a continuous function representing Γ_0 (the equation refers to the evolution in the inward normal direction as it is usually done in the experiments).

A direct discretization of (4) in the form suggested by the first order model has been proposed in [25]. A semi-Lagrangian version of this approach has been introduced by Strain in [31] and improved in a series of papers (see [32] and its references for the latest developments). In [31] the CIR method (from Courant, Isaacson and Rees) has been applied to level set models on uniform grids presenting several experiments where the method converges to the right solution. It is interesting to note that the SL-scheme implemented there has shown to be stable and accurate also in presence of changes of topology, faceting and curvature driven fronts. Moreover, the experiments show that choosing $\Delta t = \mathrm{O}(\Delta x)$ the method remains stable also for parabolic problems, a great improvement with respect to the traditional CFL condition $\Delta t = \mathrm{O}(\Delta x^2)$. In [32] the method has been improved introducing adaptive (quadtree) meshes and advanced contouring and redistancing techniques (the computation of the signed distance function on a collection of points is a crucial step in the level set method to reduce the error at the interface).

Another simple way to get a semi-Lagrangian scheme by extending the same argument used in the first order scheme (3), is to write first the equation as

$$\begin{cases} u_t - \max\limits_{a\in B(0,1)} \left\{ \operatorname{div}\left(\frac{Du}{|Du|}\right) a \cdot Du \right\} = 0 & \text{in } \mathbb{R}^n \times [0,\infty), \\ u(x,0) = u_0(x), \end{cases} \tag{5}$$

and let the normal direction be selected by the max operator. The evolution Γ_t of the interface is then tracked by taking the same level set of the solution $u(x,t)$ of (4). A drawback of this approach is that the (possibly unbounded) second order term appears in the velocity along characteristics. An alternative approach is to define characteristics as solution trajectories of a stochastic differential equation (see [18,6,7] for details). This leads to the following large time-step, *averaged scheme* for (4) which, following [18], can be written in $\mathbb{R}^2$ as

$$u_j^{n+1} = \frac{1}{2}\left(I[u^n](x_j + \sigma_j^n\sqrt{\Delta t}) + I[u^n](x_j - \sigma_j^n\sqrt{\Delta t}) \right), \tag{6}$$

where σ_j^n is defined by

$$\sigma_j^n = \frac{\sqrt{2}}{|D_j^n|} \begin{pmatrix} D_{2,j}^n \\ -D_{1,j}^n \end{pmatrix} \tag{7}$$

with $D_{1,j}^n$, $D_{2,j}^n$ and D_j^n suitable numerical approximations of respectively $u_{x_1}(x_j,t_n)$, $u_{x_2}(x_j,t_n)$ and $Du(x_j,t_n)$. Note that the numerical domain of dependence of u_j^{n+1} is given by the two regions around the points $x_j \pm \sigma_j^n\sqrt{\Delta t}$ which are about $2\sqrt{2\Delta t}$ apart. As expected, the discrete evolution takes into account the normal direction to the front with a speed which depends on the discrete curvature at the point x. An interesting variant of this scheme can be obtained by replacing the finite difference approximation of σ_j by a min–max operator as in Kohn–Serfaty [21]. In fact, this results in the following *min–max scheme*

$$u_j^{n+1} = \min_{\mu\in S^1}\left(\max(I[u^n](x_j + \sqrt{2\Delta t}\mu), I[u^n](x_j - \sqrt{2\Delta t}\mu)) \right), \tag{8}$$

where $S^1 = \{x \in \mathbb{R}^2 : |x| = 1\}$ is the unit sphere in $\mathbb{R}^2$. This scheme (which has been proposed in a very similar form by Catté et al. in [9]) has the advantage of detecting the gradient direction without requiring an approximation of the gradient itself, although it is more expensive due to the presence of the min–max operator. Construction and stochastic interpretation of the codimension-2 version of schemes (6) and (8) will be presented in Section 3. Section 4 is devoted to consistency analysis whereas in the last section we will compare the two schemes on some benchmarks. We point out that, in contrast with the SL techniques developed in [31,32], the approach pursued here focuses on the level set equation rather then the interface itself. In this respect, our SL techniques stem from a suitable characteristic-based representation formula for the viscosity solution $u(x,t)$. Moreover, we will not address here advanced issues such as contouring, redistancing and adaptivity, although for the latter some preliminary result for the codimension-1 case is shown in [8].

3. The approximation scheme for curves in R^3

Following [4,28,29], we will derive our scheme, as it has been done in [18], from the discretization of a stochastic representation formula. We will sketch in this section both the stochastic framework and the derivation of the scheme trying to keep the technical details to a minimum (a complete and rigorous treatment can be found in the references above).

To approximate the mean curvature flow of a curve Γ_0, let us start from the level set equation in $\mathbb{R}^3$:

$$\begin{cases} u_t = F(D^2u, Du) & \text{in } \mathbb{R}^3 \times [0,\infty), \\ u(x,0) = u_0(x), \end{cases} \tag{9}$$

where the initial condition is $u(x,0) = d(x,\Gamma_0)^2$, the squared distance from the curve Γ_0, or another continuous function vanishing on Γ_0 and positive on $\mathbb{R}^3 \setminus \Gamma_0$. For $p \in \mathbb{R}^3$, A a symmetric matrix in $\mathbb{R}^{3\times 3}$, the function F is given by

$$F(A,p) = \inf_{v\in\mathscr{N}(p)}\{\mathrm{trace}[AP_v]\},$$

$\mathscr{N}(p)$ being defined for $p \neq 0$ as

$$\mathscr{N}(p) = \{v \in U | P_v p = 0\},$$

where $U = \{v = (v_1, v_2) \in S^2 \times S^2 : v_1 \cdot v_2 = 0\}$ with $S^2 = \{x \in \mathbb{R}^3 : |x| = 1\}$ and P_v is a projection matrix. More precisely, we define

$$P_v := I_3 - v_1 v_1^{\mathrm{T}} - v_2 v_2^{\mathrm{T}}, \tag{10}$$

where a^{T} denotes the transpose of a vector a and I_3 is the identity matrix in $\mathbb{R}^3$. Since in this situation P_v is of rank 1, we can find a vector $\mu \in S^2$ such that

$$P_v = \mu\mu^{\mathrm{T}}$$

(μ is a normalized eigenvector associated to the nonzero eigenvalue of P_v). Then, $F(A,p)$ can be rewritten as

$$F(A,p) = \inf\{\mathrm{trace}[A\mu\mu^{\mathrm{T}}] : \mu \in S^2, \quad p \cdot \mu = 0\}. \tag{11}$$

Let now $(\Omega, \mathscr{F}, \mathscr{P}; \mathscr{F}_s, s \in [0,T])$ be a complete stochastic basis endowed with a three-dimensional Brownian motion $W = (W(s), s \in [0,T])$ and let $\mu(\cdot)$ be some S^2-valued and $(\mathscr{F}_s)$-progressively measurable process, then the generalized characteristics associated to (9) may be written as

$$\begin{cases} \mathrm{d}y_\mu(x,t;s) = \sqrt{2}\mu(s)\mu(s)^{\mathrm{T}}\,\mathrm{d}W(s), & s \in (0,t] \\ y_\mu(x,t;0) = x, \end{cases} \tag{12}$$

or equivalently as

$$\mathrm{d}y_\mu(x,t;s) = \sqrt{2}\mu(s)\mu^{\mathrm{T}}(s)\,\mathrm{d}W(s) = \sqrt{2}\mu(s)\,\mathrm{d}\widehat{W}(s),$$

where $\mathrm{d}\widehat{W}$ is the differential of a one-dimensional Brownian motion:

$$\mathrm{d}\widehat{W}(s) = \mu(s)^{\mathrm{T}}\,\mathrm{d}W(s) \in \mathbb{R}.$$

Note that the assumption that μ is measurable in s, guarantees the local existence of a unique strong solution (in the standard stochastic integral sense, see e.g. [20, Section 4.5]) for the Cauchy problem (12). In practice, $y_\mu(x,t;s)$ is a generalized characteristic going backward from (x,t) and having a diffusion coefficient $\mu(\cdot)$. The associated stochastic Cauchy problem is therefore:

$$\begin{cases} \mathrm{d}y_\mu(x,t;s) = \sqrt{2}\mu(s)\mathrm{d}\widehat{W}(s), & s \in (0,t] \\ y_\mu(x,t;0) = x. \end{cases} \tag{13}$$

Although more general representation formulae exist (see [29]), in the case u is a smooth solution of (9), by the Ito–Taylor expansion one can easily verify that

$$u(x,t) = \inf_{\mu(\cdot)\in\mathscr{U}(x,t)}\{u(y_\mu(x,t;t),0)\} \tag{14}$$

almost surely for any $y_\mu(x,t;s)$ solving (13), with

$$\mathscr{U}(x,t) = \{\mu(\cdot) : [0,T] \to S^2 \subset \mathbb{R}^3 : \mu(s)\cdot Du(y_\mu(x,t;s), t-s) = 0 \quad \text{a.e. for } s \in [0,t]\},$$

where $t \in [0,T]$ (note that this set is nonempty a.e. in (x,t) provided u is a.e. differentiable). Since (14) holds true almost surely for any trajectory $y_\mu(x,t;s)$ starting at (x,t) and satisfying (13), then it also holds for the probability expectation:

$$u(x,t) = \inf_{\mu(\cdot)\in\mathscr{U}(x,t)}\mathbb{E}\{u(y_\mu(x,t;t),0)\}. \tag{15}$$

The first step of our algorithm is to compute the solution of (9) (codimension-1 problem). Our aim is therefore to give a discrete version for (15) building a time-discrete approximation of the generalized characteristics over a time interval Δt and projecting on a uniform space grid of space step Δx. We will show that, although this construction is based on a stochastic representation formula, the result is purely deterministic, and, since (15) holds for any $t > 0$, it also allows for large time steps.

Let us start from the approximation of (13). We consider a partition $t_n = n\Delta t, n = 0, 1, \ldots, [\frac{T}{\Delta t}]$ (where $[\cdot]$ denotes the integer part) of the time interval $[0, T]$. Let y_j denote the approximation of the stochastic trajectory y at time t_j. We are interested in approximation scheme for the trajectories converging in the standard weak sense (see [20]), which means that

$$|\mathbb{E}\{g(y_j)\} - \mathbb{E}\{g(y(j\Delta t))\}| \to 0, \tag{16}$$

for every smooth function g. In particular, we are interested in the situation in which $g = u$, and y is a solution of (13) for a fixed diffusion coefficient $\mu(\cdot)$. The easiest way to obtain (16) is to use the weak Euler method (see [20]). When this method is applied to (13), taking into account that for increasing s we integrate characteristics backwards from the point (x, t_{n+1}), we obtain:

$$\begin{cases} y_{j+1} = y_j + \sqrt{2}\mu(t_j)\Delta\widehat{W}_j \\ y_0 = x. \end{cases}$$

Even if $\Delta\widehat{W}_j$ should represent a Gaussian variable with mean 0 and variance Δt, first order weak convergence can be achieved just using a variable with 2-points discrete probability density:

$$P(\Delta\widehat{W}_j = \pm\sqrt{\Delta t}) = \frac{1}{2}, \tag{17}$$

for the one-dimensional Brownian process (see [20] for this and more general approximation schemes for stochastic differential equations). Then, the representation formula (15) is written on a single time step $[t_n, t_{n+1}]$ so that

$$u(x, t_{n+1}) = \inf_{\mu(\cdot)\in\mathscr{U}(x,t_{n+1})} \mathbb{E}\{u(y_\mu(x, t_{n+1}; \Delta t), t_n)\}.$$

Denoting now by $u_{\Delta t}(x, t)$ a time-discrete approximation, and replacing the function $\mu(\cdot)$ with a constant value μ on $(t_n, t_{n+1}]$, we can write a first time discretization of (15) as:

$$u_{\Delta t}(x, t_{n+1}) = \min_{\mu\in U^{n+1}_{\Delta t}(x)} \mathbb{E}\{u_{\Delta t}(x + \sqrt{2}\mu\Delta\widehat{W}_{n+1}, t_n)\},$$

where we introduce the set $U^j_{\Delta t}(x) = \{\mu \in S^2 \subset \mathbb{R}^3 : \mu \cdot Du_{\Delta t}(x, t_j) = 0\}$. Note that $\mu \in U^{n+1}_{\Delta t}(x)$ to match the definition of the stochastic Euler method. This would result in an implicit scheme. To avoid this, we rather consider the following explicit scheme:

$$u_{\Delta t}(x, t_{n+1}) = \min_{\mu\in U^n_{\Delta t}(x)} \mathbb{E}\{u_{\Delta t}(x + \sqrt{2}\mu\Delta\widehat{W}_n, t_n)\}.$$

Then, using (17), we obtain a third discrete-time approximation:

$$u_{\Delta t}(x, t_{n+1}) = \min_{\mu\in U^n_{\Delta t}(x)} \left\{\frac{1}{2}u_{\Delta t}(x + \sqrt{2\Delta t}\mu, t_n) + \frac{1}{2}u_{\Delta t}(x - \sqrt{2\Delta t}\mu, t_n)\right\}.$$

In the algorithm, we treat the constraint $\mu \cdot Du = 0$ by penalization and compute an infimum over S^2:

$$u_{\Delta t}(x, t_{n+1}) = \min_{\mu\in S^2} \left\{\frac{1}{2}u_{\Delta t}(x + \sqrt{2\Delta t}\mu, t_n) + \frac{1}{2}u_{\Delta t}(x - \sqrt{2\Delta t}\mu, t_n) + \frac{1}{\alpha}|Du_{\Delta t}(x, t_n) \cdot \mu|^2\right\} \tag{18}$$

with $0 < \alpha$.

Next, we discretize (18) with respect to the space variable. Setting up a space grid of step Δx, we denote by x_j the nodes of a lattice where $j = (j_1, j_2, j_3) \in \mathbb{Z}^3$ and $x_j = (j_1\Delta x, j_2\Delta x, j_3\Delta x) \in \mathbb{R}^3$, by u^n_j the fully discrete approximation of the continuous solution $u(x_j, t_n)$ and by $u^n = (u^n_j)_{j\in\mathbb{Z}^3}$. On this grid, the solution at the stochastic up-wind points $x_j \pm \sqrt{2\Delta t}\mu$ is evaluated by a numerical interpolation $I[u^n](\cdot)$ and the gradient is replaced by a centered differences approximation, so that defining

$$D_j[u^n] = \frac{1}{2\Delta x}\begin{pmatrix} u^n_{j_1+1,j_2,j_3} - u^n_{j_1-1,j_2,j_3} \\ u^n_{j_1,j_2+1,j_3} - u^n_{j_1,j_2-1,j_3} \\ u^n_{j_1,j_2,j_3+1} - u^n_{j_1,j_2,j_3-1} \end{pmatrix},$$

the fully discrete, "averaged" scheme is given by

$$u^{n+1}_j = S_j(u^n), \tag{19}$$

where

$$S_j(u^n) \equiv \min_{\mu\in S^2}\left\{\frac{1}{2}I[u^n](x_j+\sqrt{2\Delta t}\mu) + \frac{1}{2}I[u^n](x_j-\sqrt{2\Delta t}\mu) + \frac{1}{\alpha}(D_j[u^n]\cdot\mu)^2\right\}.$$

The scheme is deterministic and explicit. The numerical domain of dependence has a radius of $\sqrt{2\Delta t}$ and this allows to circumvent the classical parabolic CFL condition of explicit schemes without stability losses. However, the two reconstruction stencils of $I[u^n](x_j \pm \sqrt{2\Delta t}\mu)$ are $2\sqrt{2\Delta t}$ apart, and in order to resolve smaller structures (that is, for *accuracy* instead of *stability* reasons) it might be necessary to reduce the time step. We point out that, in the codimension-1 case, no minimization is required if $Du \neq 0$, and the scheme reduces to the one proposed in [18].

The second (min–max) approach requires to replace (15) with a more general representation formula. Following [28], we can write the solution as:

$$u(x,t) = \inf_{\mu}(\operatorname{ess\,sup}_{\Omega} u(y_\nu(x,t,t),0)), \tag{20}$$

where, roughly speaking, μ maps the interval $[0,+\infty[$ into a suitable compact subset of $\mathbb{R}^n$; for all the technical assumptions and definitions needed for this formula we refer to [28]. Using the same discretization strategy as sketched before, we obtain the codimension-2 version of the min–max scheme (8)

$$u^{n+1}_j = \min_{\mu\in S^2}\Big(\max(I[u^n](x_j+\sqrt{2\Delta t}\mu), I[u^n](x_j-\sqrt{2\Delta t}\mu))\Big), \tag{21}$$

where the only difference with the codimension-1 version is that the minimization is accomplished over the three-dimensional sphere S^2. Note that a time-discrete version of (21) is outlined in [21].

As we said, our final aim is to compute the evolution of the curve Γ_t, for $t = t_n$. Therefore, the second phase of the algorithm consists in extracting the curve contained in the ε-sublevel set of $u(x,t_n)$. The problem of determining the location of the curve inside this 'tube' is solved by following an 'optimal trajectory'. The idea is to exploit the property that the curve is made by points achieving the minimum value. Once a starting point of the curve is located, the algorithm moves (with a fixed step $\Delta\xi$) towards a new point achieving a minimum. Of course we expect to find two such points, and in order to keep the same direction on the curve Γ_{t_n}, the search for a new increment is restricted to a half space once the first choice has been made.

We sketch below the main steps of this post–processing phase.

3.1. Optimal trajectory algorithm

Step 1 Find a point ξ_0, such that the numerical solution u^n achieves a minimum value. Then, ξ_0 will be the starting point for our curve reconstruction.

Step 2 Evaluate the corresponding optimal direction:

$$\hat{\eta}^n_0 = \operatorname*{argmin}_{\eta\in S^2}\{u^n(\xi_0+\Delta\xi\eta)\},$$

where $\Delta\xi \in \mathbb{R}$ is the curvilinear abscissa step. Set $j = 1$ and

$$\xi_1 = \xi_0 + \Delta\xi\hat{\eta}^n_0$$

Step 3 For $j \geqslant 1$, compute

$$\hat{\eta}^n_{j+1} = \operatorname*{argmin}_{\substack{\eta\cdot\hat{\eta}^n_j>0\\ \eta\in S^2}}\{u^n(\xi_j + \Delta\xi\eta)\},$$

$$\xi_{j+1} = \xi_j + \Delta\xi\hat{\eta}^n_{j+1}$$

(note that the condition $\eta\cdot\hat{\eta}^n_j > 0$ ensures that the algorithm keeps the same direction along the curve)
Step 4 If $|\xi_0 - \xi_{j+1}| \leqslant \varepsilon_{\text{tol}}$ then **Stop**, otherwise increment j and go to **Step 3**.

The stop condition of step 4, with ε_{tol} a fixed tolerance depending on $\Delta\xi$, works fine for closed or periodic curve (such as the ones presented in the section on numerical tests). However, this procedure may fail in detecting the proper endpoint for open or non-periodic curves, in which case suitable initial or final conditions are needed.

4. Consistency analysis

Our goal here is to prove consistency for our SL-schemes. Regardless of the stochastic arguments which have been used in the construction of the scheme, the consistency analysis will be carried out here with merely deterministic techniques.

We point out that the following consistency estimates refer to the approximation of the solution u and should be intended as rough measures of accuracy for the scheme. The reconstruction of the evolving curve Γ_t results from a further post-processing step whose precision is not discussed here. The section is split in two subsections, depending on the version of the scheme considered.

4.1. Averaged scheme

In this first case, we consider a penalized approximation of the function F, that is

$$F_\beta(A,p) = \inf_{\mu\in S^2}\left\{\text{trace}[A\mu\mu^{\text{T}}] + \frac{1}{\beta}|p\cdot\mu|^2\right\}.$$

Convergence of the method of penalization for constrained minimization problems ensures that

$$\lim_{\beta\to 0}F_\beta(A,p) = F(A,p).$$

Since we use a penalization technique to treat the constraints which appear in the definition of F in (11), we develop the consistency analysis for a penalized continuous problem where the function F is replaced by

$$F_\beta(A,p) = \min_{\mu\in S^2}\{\text{trace}[A\mu\mu^{\text{T}}] + \frac{1}{\beta}|p\cdot\mu|^2\} = \text{trace}[A\bar{\mu}\bar{\mu}^{\text{T}}] + \frac{1}{\beta}(p\cdot\bar{\mu})^2 == \bar{\mu}^{\text{T}}A\bar{\mu} + \frac{1}{\beta}(p\cdot\bar{\mu})^2. \tag{22}$$

In (22), we have written the matrix P_v as $P_v = \mu\mu^{\text{T}}$ and explicitly referred to the vector $\bar{\mu}$ which attains the minimum. We consider a smooth solution $u(x,t)$ whose samples are contained in a vector w and we assume that the order of convergence of the space reconstruction is r, then we get

$$\begin{aligned}
S_j(w) &= \min_{\mu\in S^2}\left\{\frac{1}{2}I[w](x_j + \sqrt{2\Delta t}\mu) + \frac{1}{2}I[w](x_j - \sqrt{2\Delta t}\mu) + \frac{1}{\alpha}(D_j[w]\cdot\mu)^2\right\}\\
&= \min_{\mu\in S^2}\left\{\frac{1}{2}u(x_j + \sqrt{2\Delta t}\mu,t) + \frac{1}{2}u(x_j - \sqrt{2\Delta t}\mu,t) + \text{O}(\Delta x^r)\right.\\
&\qquad\left. + \frac{1}{\alpha}(Du(x_j,t)\cdot\mu)^2 + \frac{1}{\alpha}\left[(D_j[w]\cdot\mu)^2 - (Du(x_j,t)\cdot\mu)^2\right]\right\}\\
&= \min_{\mu\in S^2}\left\{\frac{1}{2}u(x_j + \sqrt{2\Delta t}\mu,t) + \frac{1}{2}u(x_j - \sqrt{2\Delta t}\mu,t) + \text{O}(\Delta x^r) + \frac{1}{\alpha}(Du(x_j,t)\cdot\mu)^2 + \frac{\text{O}(\Delta x^2)}{\alpha}\right\},
\end{aligned}$$

where we have used the second-order convergence rate of the centered difference approximation for the gradient. The relationship between the two penalization parameters α and β will be clarified below. Expressing the points $x_j \pm \sqrt{2\Delta t}\mu$ by a third-order Taylor expansion we obtain

$$\begin{aligned}S_j(w) &= u(x_j,t) + \min_{\mu\in S^2}\left\{\frac{\sqrt{2\Delta t}}{2}Du(x_j,t)\cdot\mu + \frac{\Delta t}{2}\mu^{\mathrm{T}}D^2u(x_j,t)\mu + \frac{1}{12}D^3u(x_j,t,\sqrt{2\Delta t}\mu) - \frac{\sqrt{2\Delta t}}{2}Du(x_j,t)\cdot\mu\right.\\ &\quad\left. + \frac{\Delta t}{2}\mu^{\mathrm{T}}D^2u(x_j,t)\mu - \frac{1}{12}D^3u(x_j,t,\sqrt{2\Delta t}\mu) + \mathrm{O}(\Delta t^2) + \mathrm{O}(\Delta x^r) + \frac{1}{\alpha}(Du(x_j,t)\cdot\mu)^2 + \frac{\mathrm{O}(\Delta x^2)}{\alpha}\right\}\\ &= u(x_j,t) + \min_{\mu\in S^2}\left\{\Delta t\mu^{\mathrm{T}}D^2u(x_j,t)\mu + \frac{1}{\alpha}(Du(x_j,t)\cdot\mu)^2 + + \mathrm{O}(\Delta t^2) + \mathrm{O}(\Delta x^r) + \frac{\mathrm{O}(\Delta x^2)}{\alpha}\right\}.\end{aligned} \tag{23}$$

In (23), $D^3u(x_j,t,\sqrt{2\Delta t}\mu)$ denotes the third-order differential of u computed at (x_j,t) with increment $\sqrt{2\Delta t}\mu$ (note that the odd terms of the expansion disappear because of the opposite signs). Now, using $\bar{\mu}$ defined in (22) instead of the true minimizer in (23), and setting $\beta = \alpha\Delta t$ gives the inequality

$$S_j(w) \leqslant u(x_j,t) + \Delta t F_\beta(D^2u(x_j,t), Du(x_j,t)) + \mathrm{O}(\Delta t^2) + \mathrm{O}(\Delta x^r) + \frac{\mathrm{O}(\Delta x^2)}{\alpha}.$$

Using in turn the minimizer of (23) instead of $\bar{\mu}$ in (22) gives a reversed inequality, and at last the consistency error

$$L_{\Delta x,\Delta t}(x_j,t) = \mathrm{O}(\Delta t) + \mathrm{O}\left(\frac{\Delta x^r}{\Delta t}\right) + \mathrm{O}\left(\frac{\Delta x^2}{\alpha\Delta t}\right). \tag{24}$$

Note that the scheme is conditionally consistent, and the relationship between Δt and Δx could be optimized so as to achieve the highest consistency rate. In the numerical tests we will rather choose $\Delta t \sim \Delta x$ and use a cubic reconstruction ($r = 4$) to reduce numerical viscosity (this fact is crucial for a correct detection of the minima of $u(x,t)$). The choice of α is not critical: in fact, the scheme is consistent with a continuous problem where the penalization parameter is $\beta = \alpha\Delta t$, which vanishes even for constant α. Moreover, the curve Γ_t lies on the minima of $u(x,t)$ where the constraint $Du\cdot\mu = 0$ is identically satisfied, so that penalization needs not to be overly accurate.

4.2. Min–max scheme

First, note that we can rewrite the function F as

$$F(A,p) = \min_{\mu\in S^2,\mu\cdot p=0}\{\mathrm{trace}[A\mu\mu^{\mathrm{T}}]\} = \min_{\mu\in S^2}\max\{\mathrm{trace}[A\mu\mu^{\mathrm{T}}] - \mu\cdot p, \mathrm{trace}[A\mu\mu^{\mathrm{T}}] + \mu\cdot p\} = \mathrm{trace}[A\tilde{\mu}\tilde{\mu}^{\mathrm{T}}] = \tilde{\mu}^{\mathrm{T}}A\tilde{\mu},$$

where we have denoted by $\tilde{\mu}$ the minimizer (note that since $\mu\in S^2$, the choice of the sign is irrelevant). Here and in the sequel, we make use of the fact that if two functions $f_1(\mu)$ and $f_2(\mu)$ depend continuously on μ and span the same set, their max is minimized when they attain the same value.

Following the notation and technique introduced above, we have for this version of the scheme:

$$S_j(w) = \min_{\mu\in S^2}\left\{\max\left(I[w](x_j + \sqrt{2\Delta t}\mu), I[w](x_j - \sqrt{2\Delta t}\mu)\right)\right\} \tag{25}$$

and therefore, denoting by $\bar{\mu}_j$ the minimizer in (25),

$$S_j(w) = I[w](x_j + \sqrt{2\Delta t}\bar{\mu}_j). \tag{26}$$

The values within the max can be expressed as

$$\begin{aligned}I[w](x_j \pm \sqrt{2\Delta t}\mu) &= u(x_j \pm \sqrt{2\Delta t}\mu, t) + \mathrm{O}(\Delta x^r)\\ &= u(x_j,t) \pm \sqrt{2\Delta t}Du(x_j,t)\cdot\mu + \Delta t\mu^{\mathrm{T}}D^2u(x_j,t)\mu + \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r).\end{aligned} \tag{27}$$

Now, the max of $I[w](x_j \pm \sqrt{2\Delta t}\mu)$ is minimized when the two values coincide, so that

$$\begin{aligned} &u(x_j,t) + \sqrt{2\Delta t}Du(x_j,t)\cdot\bar{\mu}_j + \Delta t\bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j + \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r) \\ &\quad = u(x_j,t) - \sqrt{2\Delta t}Du(x_j,t)\cdot\bar{\mu}_j + \Delta t\bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j + \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r) \end{aligned}$$

and therefore

$$\sqrt{2\Delta t}Du(x_j,t)\cdot\bar{\mu}_j = \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r) \tag{28}$$

(note that (28) shows that the minmax operation basically selects again the direction orthogonal to the gradient). Using now (26)–(28) in (25), we obtain

$$S_j(w) = u(x_j,t) + \Delta t\bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j + \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r).$$

Taking now into account that

$$\begin{aligned} F(D^2u(x_j,t),Du(x_j,t)) &= \tilde{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\tilde{\mu}_j \\ &\leqslant \max\{\bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j - \bar{\mu}_j\cdot Du(x_j,t), \bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j + \bar{\mu}_j\cdot Du(x_j,t)\} \\ &\leqslant \bar{\mu}_j^{\mathrm{T}}D^2u(x_j,t)\bar{\mu}_j + \mathrm{O}(\Delta t) + \mathrm{O}\left(\frac{\Delta x^r}{\Delta t^{1/2}}\right) \end{aligned}$$

(note that we have used again (28) and that the terms $\mathrm{O}(\Delta t)$ and $\mathrm{O}(\Delta x^r/\Delta t^{1/2})$ are not the leading consistency terms and may therefore be omitted) we get

$$S_j(w) \geqslant u(x_j,t) + \Delta tF(D^2u(x_j,t),Du(x_j,t)) + \mathrm{O}(\Delta t^{3/2}) + \mathrm{O}(\Delta x^r).$$

By interchanging again the roles of the minimizers $\tilde{\mu}$ and $\bar{\mu}$ we can obtain the reverse inequality, so that at last

$$L_{\Delta x,\Delta t}(x_j,t) = \mathrm{O}(\Delta t^{1/2}) + \mathrm{O}\left(\frac{\Delta x^r}{\Delta t}\right).$$

This latter consistency estimate would in principle be optimized under smaller time steps then the former. In practice we will compare the two schemes with the same steps, obtaining however a somewhat lower performance of the min–max version.

5. Numerical tests

In all the tests we started our approximations by following the evolution of an ε-sublevel set, as described in Section 3. Although the parabolic behaviour of the evolution equation results in a thickening of the ε-sublevel, the optimal trajectory algorithm does not suffer from this effect. It has been implemented with a $\Delta\xi \simeq \Delta x$, so as to avoid the detection of local minima due to the reconstruction, and with a tolerance of $\varepsilon_{\mathrm{tol}} = \Delta\xi$ for determining the endpoint of closed curves.

The Optimal Trajectory Algorithm combines a cubic interpolation and a minimization. The minimization has been performed via Brent's routine `PRAXIS` from NETLIB (see [3] for details) which works on the principle of finding conjugate directions by means of interpolated line searches. This routine has proved to be robust enough to carry out the minimization of an unsmooth function, while retaining superlinear convergence whenever the function is smooth. Once minimization is performed with a high accuracy, the prevailing error of the Optimal Trajectory Algorithm is due to interpolation.

The convergence rate of the reconstructed curve results from both the semi-Lagrangian scheme (19) or (21), which should be at most first-order accurate, and the Optimal Trajectory Algorithm. In spite of this, numerical errors are remarkably small even for the coarsest grids and measured convergence rates usually go beyond the value 1.

5.1. Simple curves

The first set of tests presents the evolution of simple curves. Following [5,26], in the first two tests we have computed errors between numerical and exact evolution, along with convergence rates, for both the averaged and the min–max scheme. For Test 1, CPU times are also shown. Error tables report the maximum E_H over all iterations of the Hausdorff distance between exact and computed curves, each successive row representing a reduction of the discretization steps by a factor of about $\sqrt{2}$. In the first two tests, the initial condition is smooth and very large time steps may be used, whereas in the third the initial curve presents corners, and the time step has been reduced in order to track more precisely the evolution of singularities (this situation could also be handled with a time-adaptive scheme, see [8]). Note that in tests 1 and 2 we have indicated an *approximate* $\Delta x/\Delta t$ ratio. With a fixed final time, such large time steps do not allow to keep a precise ratio, and therefore each refinement has been performed so as to be as close as possible to the relationship chosen. This partly explains some irregularity in the measured convergence rates.

Test 1: Evolution of an oblique circle in $\mathbb{R}^3$

We approximate (9) in $[-1.5, 1.5]^3 \times [0, 0.45]$, with

$$u_0(x_1,x_2,x_3) = \left(\frac{\sqrt{3}}{2}x_2 + \frac{1}{2}x_3\right)^2 + \left(\sqrt{x_1^2 + \left(\frac{1}{2}x_2 - \frac{\sqrt{3}}{2}x_3\right)^2} - 1\right)^2,$$

that is, a function vanishing on a circle centered at the origin, with radius 1 and contained in an oblique plane, so as to avoid an alignment with the grid. We present in Table 1 the errors evaluated at time $T = 0.45$ (note that the circle should collapse at the theoretical time $T^* = 0.5$). The ratio $\Delta x/\Delta t \approx 1.25$ has been used to determine the time step for each refinement of the space grid. The CPU time refers to the seconds needed to run each test in a CLUSTER IBM power 5 (eight processors at 1.9 GhZ). For a comparison, with the same level of code optimization such a platform performs the product of two 5000×5000 matrices in 96 s. Note that the min–max scheme has a slightly lower overall convergence rate and, due to the fact that it minimizes a non-smooth function instead of a smooth one, it also has higher execution times. In both cases, however, the increase in CPU times is approximately of order Δx^{-4}, that is, of the same order as the total number of nodes in the space–time grid.

Fig. 1 refers to iterations $n = 0, 2, 4, 6, 8, 10$ for the collapsing circle with time step $\Delta t = 0.0375$ and $\Delta x = 0.046875$.

Test 2: Evolution of a helical curve in $\mathbb{R}^3$

We take (9) in $[0, 1]^3 \times [0, 0.05]$, for

$$u_0(x_1,x_2,x_3) = (x_1 - 0.5 - 0.3\cos(2\pi x_3))^2 + (x_2 - 0.5 - 0.3\sin(2\pi x_3))^2,$$

i.e. a function vanishing on a helix centered on the axis $(0.5, 0.5, x_3)$ and with radius 0.3.

The pictures in Fig. 2 refer to a simulation with $\Delta x = 0.015626$, $\Delta t = 0.00625$, cubic interpolation. Periodic conditions are used at the boundary. We plot the ε sublevel set at time iterations n = 1–8, together with the helical curve inside each sublevel set.

In Fig. 3 we plot only the helical curve evolution. As the figure shows, the curve remains helical but with a shrinking radius, as obtained for the same test by Burchard, Cheng, Merriman and Osher in [5]. The comparison with [5] shows a similar order of magnitude for errors, with a lower convergence rate but with a lower number of time steps in our case. Tests are run with the ratio $\Delta x/\Delta t \approx 2.5$ and the corresponding results are presented in Table 2.

Table 1
Numerical errors, convergence rates and CPU times for Test 1

Grid		Averaged scheme			Min–max scheme		
Nodes	Δt	E_H	Rate	CPU (s)	E_H	Rate	CPU (s)
16^3	0.15	9.47×10^{-3}		28	2.01×10^{-2}		54
23^3	0.1125	2.98×10^{-3}	3.33	120	8.08×10^{-3}	2.63	206
32^3	0.09	9.77×10^{-4}	3.21	378	1.99×10^{-3}	4.03	748
45^3	0.05	3.48×10^{-4}	2.97	2212	1.29×10^{-3}	1.25	3523
64^3	0.0375	2.05×10^{-4}	1.54	9047	6.80×10^{-4}	1.85	13,161
90^3	0.02647	8.79×10^{-5}	2.44	35,739	3.75×10^{-4}	1.71	55,411

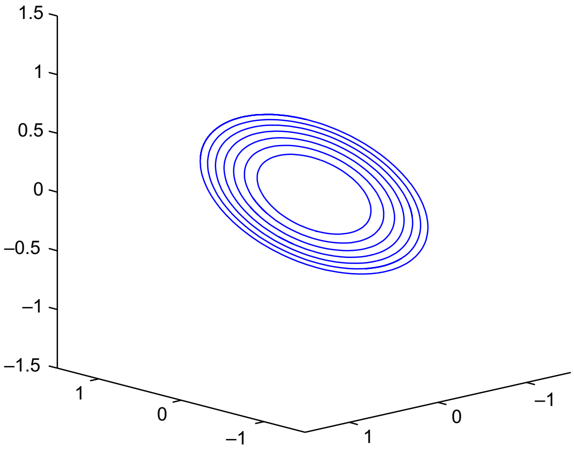

Fig. 1. Evolution of the oblique circle, post-processed curve.

Table 2
Numerical errors and convergence rates for Test 2

Grid		Averaged scheme		Min–max scheme	
Nodes	Δt	E_{H}	Rate	E_{H}	Rate
16^3	2.5×10^{-2}	8.09×10^{-3}		9.51×10^{-3}	
23^3	1.6×10^{-2}	4.96×10^{-3}	1.41	8.87×10^{-3}	0.2
32^3	$1.25\ \times 10^{-2}$	3.58×10^{-3}	0.94	4.83×10^{-3}	1.75
45^3	$8.34\ \times 10^{-3}$	2.23×10^{-3}	1.36	2.83×10^{-3}	1.54
64^3	6.25×10^{-3}	1.61×10^{-3}	0.94	2.28×10^{-3}	0.62
90^3	4.55×10^{-3}	1.09×10^{-3}	1.12	1.87×10^{-3}	0.57

Test 3: evolution of a nonsmooth manifold

We consider (9) in $[-1.5, 1.5]^3 \times [0, 0.5]$, for u_0 describing an initial curve which joins with segments the points $(-1, 0, 0)$, $(0, 1, 1)$, $(1, 0, 0)$ and $(0, -1, 1)$.

The test is performed with the averaged scheme, $\Delta x = 0.06$, $\Delta t = 0.005$ and cubic interpolation. Fig. 4 shows the evolution of the curve every 5 iterations. In particular, the figure confirms both the regularization of the curve and its asymptotic alignment with the plane $x_3 = 1/2$.

5.2. Intersecting curves and fattening

In the second set of tests (which have been carried out with the averaged scheme), we present the evolution of curves which eventually generate a double point, and therefore a region of fattening for the ε-sublevel set. In this situation, we single out the physically meaningful evolution as the one which shortens the curve in the fastest way, in particular resolving double points by keeping acute angles connected. In the tests performed (and definitely in the two tests presented), the scheme has shown some tendency to follow this rule of breaking of the curve, provided the solution u is sufficiently accurate and the Courant number is relatively low. However, we have no theoretical justification for this, and in fact we also experienced some situation in which this rule was not respected.

Test 4: evolution of two linked circles in $\mathbb{R}^3$

We consider (9) in $[-1.5, 1.5]^2 \times [-1.5, 2.5] \times [0, 0.5]$, for

$$u_0(x_1, x_2, x_3) = \min(\mathscr{C}_1, \mathscr{C}_2),$$

where $\mathscr{C}_1$ and $\mathscr{C}_2$ are the squared distances in $\mathbb{R}^3$ from two linked oblique circles with radius 1.

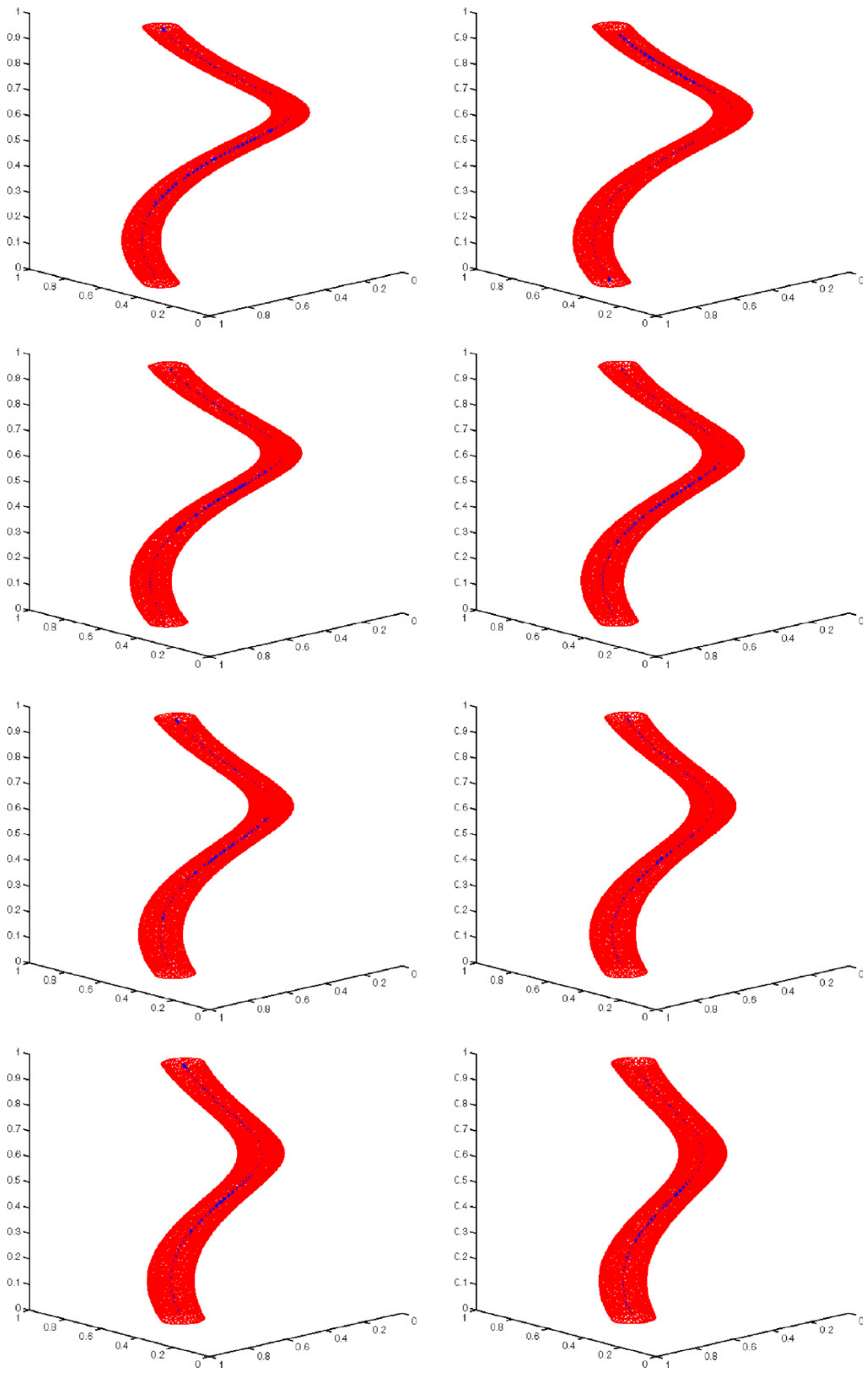

Fig. 2. Evolution of the ε-sublevel for the helix, along with the post-processed curve.

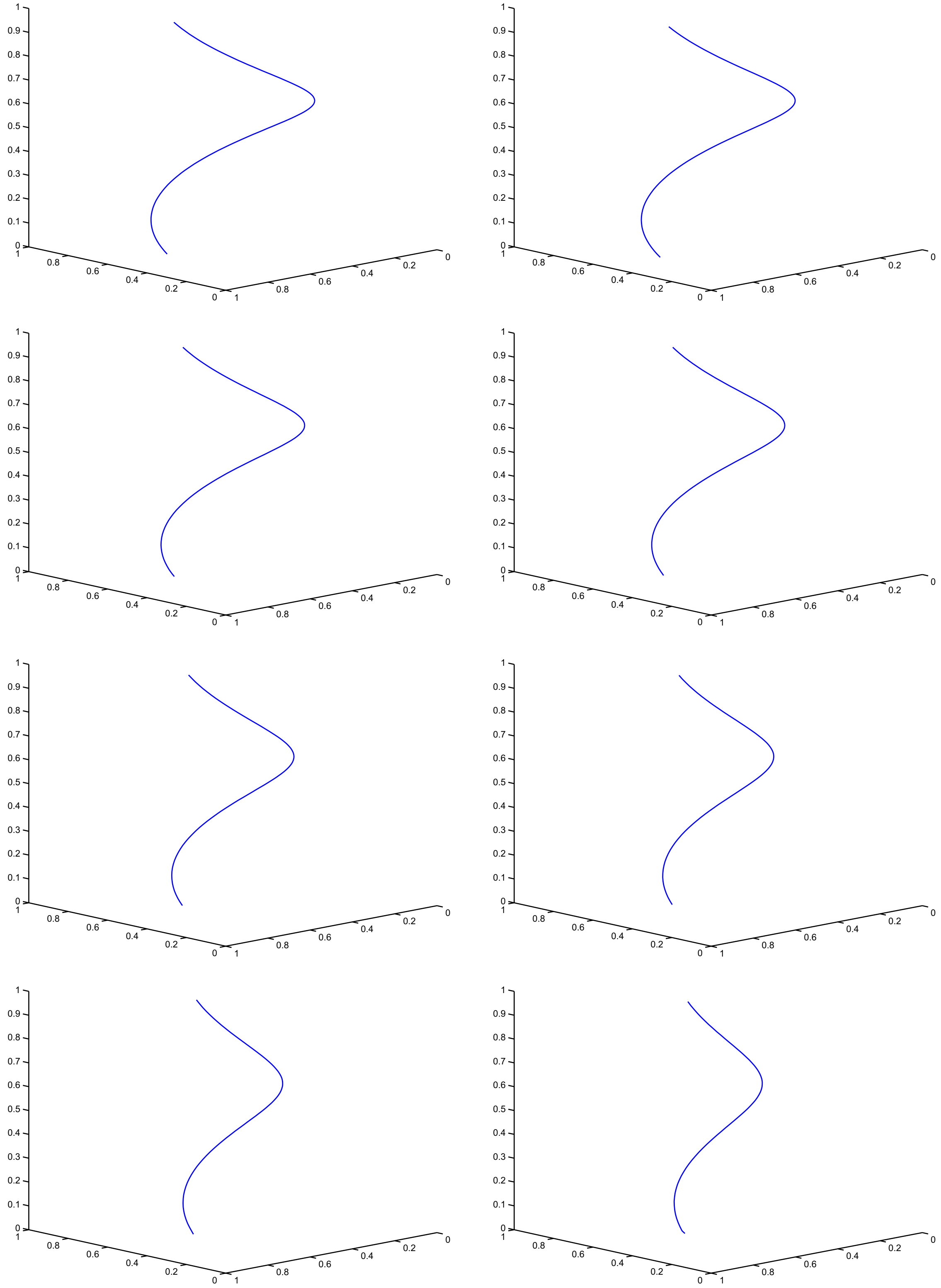

Fig. 3. Evolution of the helix, post-processed curve.

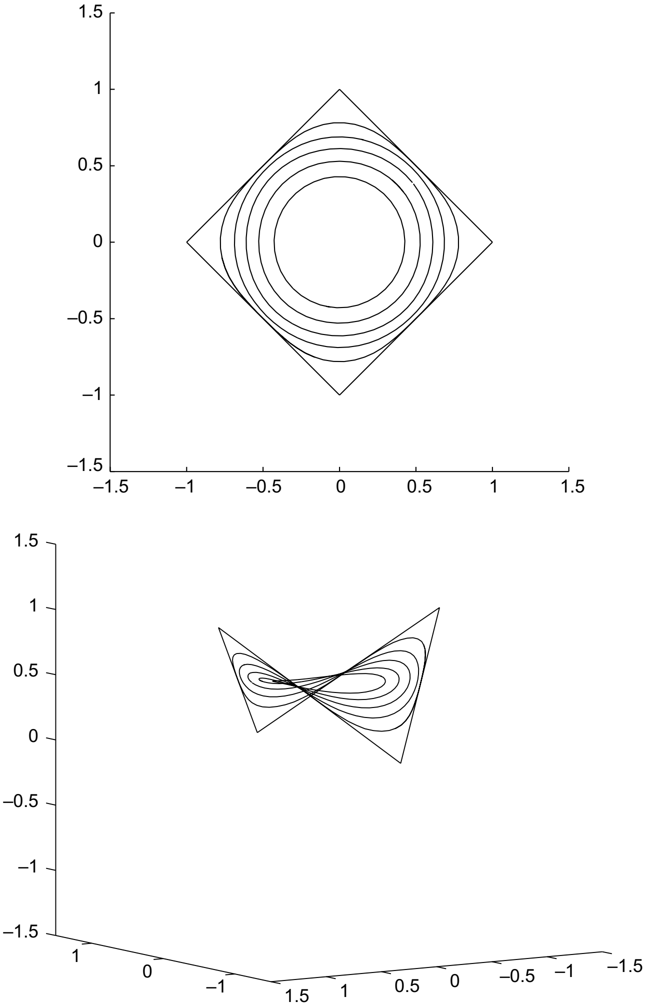

Fig. 4. Evolution of the nonsmooth manifold, post-processed curve ($x_1 - x_2$ projection and perspective).

The results in Fig. 5 are obtained with $\Delta x = 0.04$, $\Delta t = 0.005$ and cubic interpolation. Pictures refer to iterations $n = 20, 30, 40, 50, 60, 70$. The two curves shrink independently until they touch and a three dimensional fattening appears, as shown by the level surfaces, and theoretically analysed by Bellettini, Novaga, Paolini in [2]. Fig. 6 shows a snapshot of the post-processing and Fig. 7 the complete post-processed curve. Note that the optimal trajectory algorithm reconstructs the curve with some oscillations at the breaking of the double point. This effect is not due to instability of the scheme (and in fact, it does not propagate to subsequent iterations). On the contrary, it is related to some inherent difficulty of the post-processing algorithm to follow the minima of u near the onset time of fattening.

Test 5: evolution of two linked helices in $\mathbb{R}^3$

The second example for fattening is obtained by considering two linked helices, as in [5]. We take (9) in $[-0.75, 0.75] \times [-0.5, 0.5] \times [0, 2] \times [0, 0.02]$, for

$$u_0(x_1, x_2, x_3) = \min(\mathscr{C}_1, \mathscr{C}_2),$$

where $\mathscr{C}_1$ and $\mathscr{C}_2$ are the squared distances in $\mathbb{R}^3$ from two helices with centers at $(x_1, x_2) = (\pm 0.25, 0)$ and radius 0.3. Here, the space grid has $65 \times 65 \times 112$ nodes and the time step is $\Delta t = 0.001$. Fig. 8 shows the post-processed curves at iterations 0, 10, 20. Again, the post-processing phase has some loss of accuracy around the

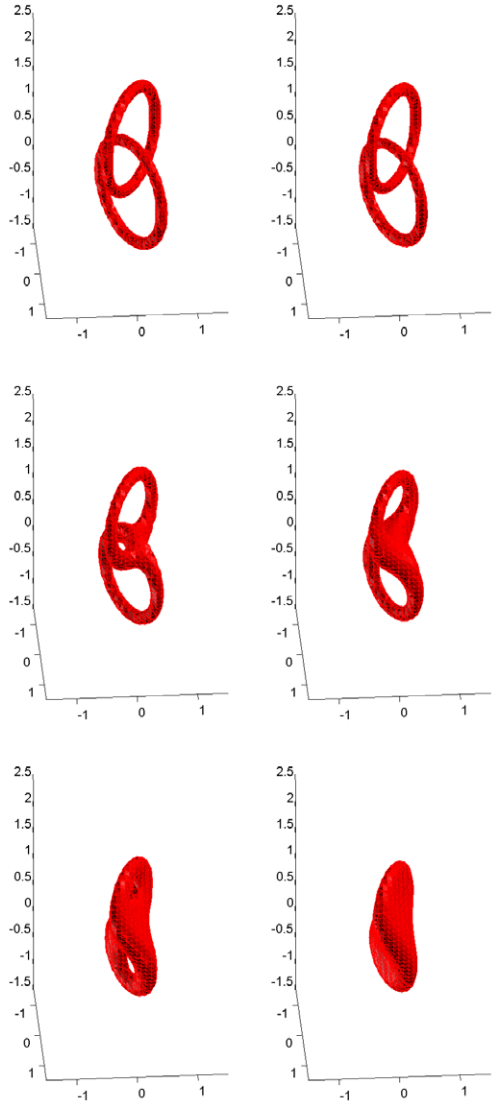

Fig. 5. Evolution of the ε-sublevel for the linked circles, $\varepsilon = 0.01$.

breaking of the double point, but according to the parameters used the topology of the reconstructed curve is correct.

Appendix A. The level set approach to mean curvature flow in arbitrary codimension (Ambrosio–Soner, [1])

As we said in the introduction, our work is inspired by the approach introduced by Ambrosio and Soner [1]. We will briefly recall here, for reader's convenience, the main features of such approach to curve evolution in codimension-k. As far as we know this is the only analytical characterization which has been shown to converge to the curve.

Let $\Gamma \subseteq \mathbb{R}^{\mathbb{N}}$ be a smooth surface with codimension $k \geqslant 1$, $u : \mathbb{R}^N \mapsto [0, \infty)$ be an auxiliary function such that

$$\Gamma = \{x \in \mathbb{R}^N : u(x) = 0\}. \tag{29}$$

Let us assume that u is smooth near Γ and its gradient does not vanish outside Γ. The key step is to express the curvature properties of Γ in terms of the derivatives of this auxiliary function u. To this end, consider the

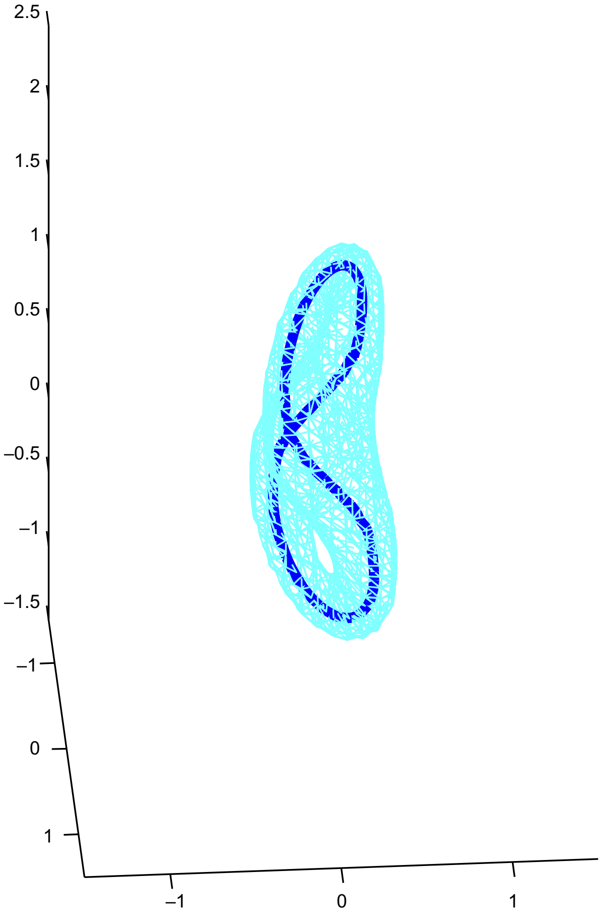

Fig. 6. An ε-sublevel for the linked circles, along with the post-processed curve.

ε-level set Γ^ϵ of u for small $\varepsilon > 0$. Let us first consider the case $k = 1$ and for a nonzero (column) vector $p \in \mathbb{R}^N$ define the projection matrix

$$P_p = I_N - \frac{pp^{\mathrm{T}}}{|p|^2}, \tag{30}$$

where a^{T} denotes the transpose of a and I_N is the identity matrix in $\mathbb{R}^N$. For any $x \notin \Gamma$ but in a small neighbourhood of Γ let us define the symmetric, $N \times N$ matrix

$$M(x) = \frac{1}{|Du(x)|} P_{Du(x)} D^2u(x) P_{Du(x)},$$

where $Du(x)$ and $D^2u(x)$ denote respectively gradient and hessian matrix of u. Moreover, let

$$\lambda_1(M) \leqslant \lambda_2(M) \leqslant \cdots \leqslant \lambda_{N-1}(M)$$

be the eigenvalues of $M(x)$ corresponding to eigenvectors orthogonal to $Du(x)$ (note that $Du(x)$ is an eigenvector associated to the zero eigenvalue since $M(x)Du(x) = 0$).

The above eigenvalues correspond to the principal curvatures of the codimension-1 surface Γ^ϵ, oriented by Du. Since Γ has codimension-k, for ε small enough, we expect Γ^ϵ to have very large $k - 1$ principal curvatures and the remaining N-k principal curvatures to be related to the geometry of Γ.

Ambrosio and Soner have given the following *level set definition for the codimension-k mean curvature flow*: for a symmetric matrix $A \in \mathbb{R}^{N \times N}$ and $p \in \mathbb{R}^N$ with $p \neq 0$, set

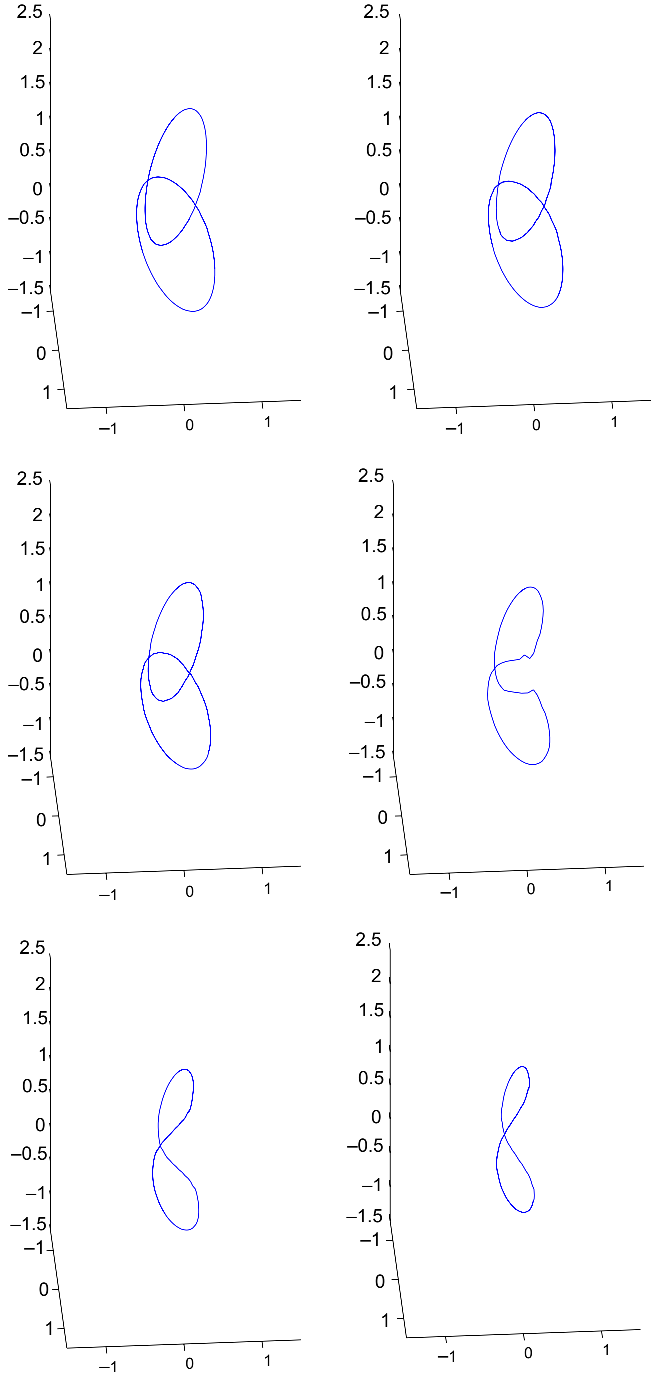

Fig. 7. Evolution of the linked circles, post-processed curve.

$$X = P_p A P_p \tag{31}$$

and let

$$\lambda_1(X) \leqslant \lambda_2(X) \leqslant \cdots \leqslant \lambda_{N-1}(X)$$

be the eigenvalues of X corresponding to eigenvectors orthogonal to p (observe again that 0 is an eigenvalue of X corresponding to p) and define

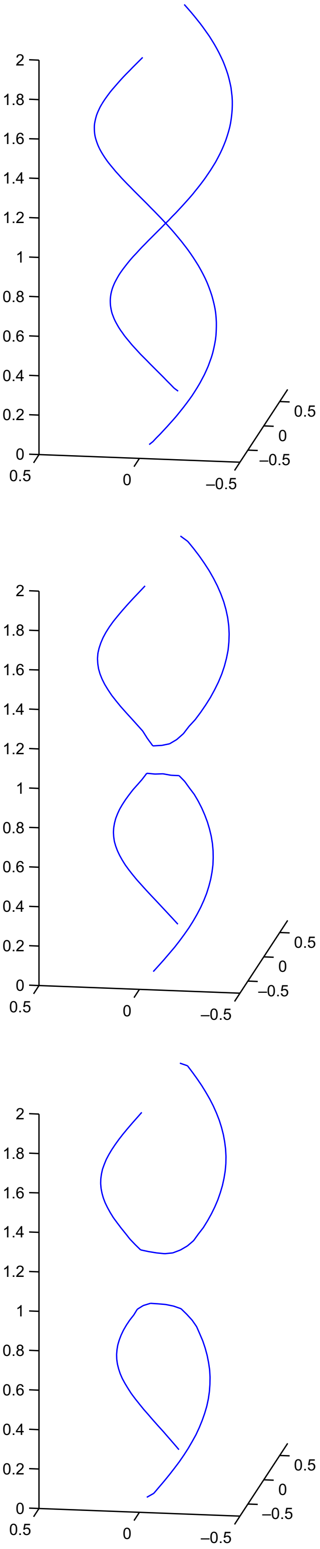

Fig. 8. Evolution of the linked helices, post-processed curves.

$$\widehat{F}_k(A,p) = \sum_{i=1}^{N-k} \lambda_i(X).$$

Given an initial data Γ_0, and a nonnegative function u_0 vanishing on Γ_0, the codimension-k mean curvature flow of Γ_0 is given by $\Gamma_t = \{x \in \mathbb{R}^N : u(x,t) = 0\}$ with $u(x,t)$ solution of

$$\begin{cases} u_t = \widehat{F}_k(D^2u, Du) & \text{in } \mathbb{R}^3 \times [0,\infty) \\ u(x,0) = u_0(x). \end{cases} \tag{32}$$

Ambrosio and Soner [1] have proved existence and uniqueness of a viscosity solution to the Cauchy problem (32) provided $u_0 : \mathbb{R}^N \mapsto [0,\infty)$ satisfies (29) and is uniformly continuous (the result is an extension of the classical result by Chen, Giga, Goto [10]). Moreover, they prove that Γ_t depends only on Γ_0 but not on u_0. Hence Γ_t is a well-defined evolution of Γ_0.

Let us now recall a representation of the differential operator corresponding to the codimension-k case. This representation uses a generalized form of the projection matrix (30). Let us define as usual $S^{N-1} = \{z \in \mathbb{R}^N : |z| = 1\}$, and the set

$$U = \{v = (v_1,\ldots,v_k) \in (S^{N-1})^k : v_i \cdot v_j = 0 \quad \text{for all } 1 \leqslant i \neq j \leqslant k\}. \tag{33}$$

Then, for any matrix (with orthonormal columns) $v \in U$, let P_v be the projection onto a vector space orthogonal to v, i.e.

$$P_v := I_N - \sum_{j=1}^{k} v_j v_j^{\mathrm{T}} \tag{34}$$

and for $p \neq 0$, let

$$\mathcal{N}(p) = \{v \in U | P_v p = 0\} \tag{35}$$

(note that $\mathcal{N}(0) = U$). With the above notations, (32) can be written as

$$u_t = F_k(D^2u, Du) \quad \text{in } \mathbb{R}^N \times [0,T), \tag{36}$$

where

$$F_k(A,p) = \inf_{v \in \mathcal{N}(p)} \{\mathrm{trace}[AP_v]\}.$$

Finally, Soner and Touzi have proved the equivalence between the two definition (32) and (36):

Proposition A.1 [28]. For any $p \in \mathbb{R}^N$, $p \neq 0$, and symmetric matrix $A \in \mathbb{R}^{N\times N}$, the two operators F and $\widehat{F}$ coincide, i.e.

$$F_k(A,p) = \widehat{F}_k(A,p).$$

We just recall that (36) corresponds to a dynamic programming equation for a stochastic control problem as it has been proved in [28].

References

[1] L. Ambrosio, M. Soner, Level set approach to mean curvature flow in arbitrary codimension, J. Differ. Geom. 43 (1996) 693–737.

[2] G. Bellettini, M. Novaga, M. Paolini, An example of three dimensional fattening for linked space curves evolving by curvature, Commun. Part. Differ. Eq. 23 (1998) 1475–1492.

[3] R. Brent, Algorithms for Minimization Without Derivatives, Prentice-Hall, Englewood Cliffs, 1973.

[4] R. Buckdahn, P. Cardaliaguet, M. Quincampoix, A representation formula for the mean curvature motion, SIAM J. Math. Anal. 33 (2001) 827–846.

[5] P. Burchard, L.T. Cheng, B. Merriman, S. Osher, Motion of curves in three spatial dimensions using a level set approach, J. Comput. Phys. 170 (2001) 720–741.

[6] E. Carlini, Semi-Lagrangian schemes for first and second order Hamilton–Jacobi equations, Ph.D. Thesis, Università di Roma "La Sapienza", Roma, 2004.

[7] E. Carlini, M. Falcone, R. Ferretti, Convergence of a large time-step scheme for mean curvature motion, in preparation.

[8] E. Carlini, M. Falcone, R. Ferretti, A time-adaptive semi-Lagrangian approximation to mean curvature motion, Numerical Mathematics and Advanced Applications – ENUMATH 2005, Springer, 2006, pp. 732–739.

[9] F. Catté, F. Dibos, G. Koepfler, A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets, SIAM J. Numer. Anal. 32 (1995) 1895–1909.

[10] Y.G. Chen, Y. Giga, S. Goto, Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations, J. Differ. Geom. 33 (1991) 749–786.

[11] L.T. Cheng, P. Burchard, B. Merriman, S. Osher, Motion of curves constrained on surfaces using a level-set approach, J. Comput. Phys. 175 (2002) 604–644.

[12] K. Deckelnick, G. Dziuk, On the approximation of the curve shortening flow, Pitman Res. Notes Math. Ser. 326 (1995) 100–108.

[13] G. Dziuk, Convergence of a semi-discrete scheme for the curve shortening flow, Math. Models Methods Appl. Sci. 4 (1994) 589–606.

[14] G. Dziuk, Discrete anisotropic curve shortening flow, SIAM J. Numer. Anal. 36 (1999) 1808–1830.

[15] G. Dziuk, E. Kuwert, R. Schätzle, Evolution of elastic curves in $\mathbb{R}^n$: existence and computation, SIAM J. Math. Anal. 33 (2002) 1228–1245.

[16] M. Falcone, The minimum time problem and its applications to front propagation, in: G. Buttazzo, A. Visintin (Eds.), Motion by mean curvature and related topic, (Trento, 1992), de Gruyter, Berlin, 1994, pp. 70–88.

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

[18] M. Falcone, R. Ferretti, Consistency of a large time-step scheme for mean curvature motion, in: F. Brezzi, A. Buffa, S. Corsaro, A. Murli (Eds.), Numerical Mathematics and Advanced Applications – ENUMATH 2001, Springer Verlag, 2003, pp. 495–502.

[19] M. Falcone, T. Giorgi, P. Loreti, Level sets of viscosity solutions and applications, SIAM J. Appl. Math. 54 (1994) 1335–1354.

[20] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, Berlin, 1992.

[21] R.V. Kohn, S. Serfaty, A deterministic-control-based approach to motion by curvature, Courant Institute, Commun. Pure Appl. Math. 59 (2006) 344–407.

[22] B. Merriman, J. Bence, S. Osher, Diffusion-generated motion by mean curvature, in: J. Taylor (Eds.), AMS Selected Lectures in Mathematics, The Computational Crystal Grower's Workshop, AMS Providence, 1993, pp. 73–83.

[23] K. Mikula, Solution of nonlinearly curvature driven evolution of plane curves, Appl. Numer. Math. 31 (1999) 191–207.

[24] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.

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

[26] S.J. Ruuth, B. Merriman, J. Xin, S. Osher, Diffusion-generated motion by mean curvature for filaments, J. Nonlinear Sci. 11 (2001) 473–493.

[27] J.A. Sethian, Level set method. Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science, Cambridge Monographs on Applied and Computational Mathematics, vol. 3, Cambridge University Press, Cambridge, 1996.

[28] H.M. Soner, N. Touzi, A stochastic representation for the level set equations, Commun. Part. Differ. Eq. 27 (9&10) (2002) 2031–2053.

[29] H.M. Soner, N. Touzi, A stochastic representation for mean curvature type geometric flows, Ann. Probab. 31 (2003) 1145–1165.

[30] A.N. Staniforth, J. Côtè, Semi-Lagrangian integration schemes for atmospheric models – a review, Monsoon Weather Rev. 119 (1991) 2206–2223.

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

[32] J. Strain, A fast semi-Lagrangian contouring method for moving interfaces, J. Comput. Phys. 169 (2001) 1–22.