sciendo

TATRA MOUNTAINS
Mathematical Publications
DOI: 10.2478/tmmp-2023-0031
Tatra Mt. Math. Publ. **87** (2024), 145–170

MACROPHAGES TRAJECTORIES SMOOTHING BY EVOLVING CURVES

Giulia Lupi — Karol Mikula — Seol Ah Park

Department of Mathematics and Descriptive Geometry,
Slovak University of Technology in Bratislava, SLOVAKIA

ABSTRACT. When analyzing cell trajectories, we often have to deal with noisy data due to the random motion of the cells and possible imperfections in cell center detection. To smooth these trajectories, we present a mathematical model and numerical method based on evolving open-plane curve approach in the Lagrangian formulation. The model contains two terms: the first is the smoothing term given by the influence of local curvature, while the other attracts the curve to the original trajectory. We use the flowing finite volume method to discretize the advection-diffusion partial differential equation. The PDE includes the asymptotically uniform tangential redistribution of curve grid points. We present results for macrophage trajectory smoothing and define a method to compute the cell velocity for the discrete points on the smoothed curve.

1. Mathematical model

In this paper, we present a mathematical model and numerical method for macrophage trajectories smoothing based on evolving open plane curve approach in the Lagrangian formulation, namely, we solve the equation

$$\frac{\partial \mathbf{x}}{\partial t} = -\delta k \mathbf{N} + \lambda[(\mathbf{x}_0 - \mathbf{x}) \cdot \mathbf{N}]\mathbf{N} + \alpha \mathbf{T}, \tag{1}$$

where $\mathbf{x}$ represents the evolving curve and $\mathbf{x}_0$ the initial curve. The terms evolving the curve in the normal direction $\mathbf{N}$ and the tangential direction $\mathbf{T}$ are given in details below.

The paper is structured as follows: in this section, we discuss the mathematical model and its possible applications. In Section 2, we show the numerical

© 2023 Mathematical Institute, Slovak Academy of Sciences.
2020 Mathematics Subject Classification: 35Q80, 92C55, 74S10, 65D10.
Keywords: smoothing, macrophage, velocity, finite volume method.
Supported by the Marie Skłodowska-Curie grant agreement no. 955576.
Licensed under the Creative Commons BY-NC-ND 4.0 International Public License.

discretization of the suggested model. In Section 3, we expose the numerical experiments made on cell trajectories. Moreover, the smoothed curves allow us to compute the velocity of the cell. To do that, we propose a method to find the new velocity for the discrete points on the evolving curve based on the evolution of the length of the segments composing the piecewise linear initial condition.

Let Γ be an open plane curve with fixed endpoints

$$\begin{aligned} \Gamma : [0,1] &\to \mathbb{R}^2, \\ u &\mapsto \mathbf{x}(u), \end{aligned} \tag{2}$$

where $\mathbf{x}(u) = (x_1(u), x_2(u))$ is a point of the curve Γ. We consider the time evolution of the curve Γ, $\Gamma_t = \{\mathbf{x}(t,u), u \in [0,1]\}$ where t is the time and $\mathbf{x}(t,u)$ represents the position of the point $\mathbf{x}(u) \in \Gamma_t$ at time t. From now on, we will write $\mathbf{x}$ instead of $\mathbf{x}(t,u)$ for clarity. The evolution of a point $\mathbf{x} \in \Gamma$ is driven by the following general equation

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{V}(\mathbf{x}, t), \tag{3}$$

where $\frac{\partial \mathbf{x}}{\partial t}$ is the time derivative representing the velocity of the evolving curve. The velocity vector field $\mathbf{V}$ that drives the curve motion is given by

$$\mathbf{V}(\mathbf{x}, t) = -\delta k(\mathbf{x}, t)\mathbf{N}(\mathbf{x}, t) + \lambda[(\mathbf{x}_0 - \mathbf{x}) \cdot \mathbf{N}(\mathbf{x}, t)]\mathbf{N}(\mathbf{x}, t), \tag{4}$$

where $\mathbf{N}$ is the unit normal vector to the evolving curve, k is its curvature, $\mathbf{x}_0$ is the initial condition, " $\cdot$ " is the scalar product and δ, λ are given positive parameters. From now on, we will indicate by

$$w(\mathbf{x}, t) = (\mathbf{x}_0 - \mathbf{x}) \cdot \mathbf{N}(\mathbf{x}, t). \tag{5}$$

The curvature term $-k\mathbf{N}$, weighted by the parameter δ, regularizes the curve, while w, weighted by the parameter λ, attracts the evolving curve next to the initial condition. The vector function $(\mathbf{x}_0 - \mathbf{x})$ realizes the minimum distance between the curve $\mathbf{x}$ and the initial curve $\mathbf{x}_0$. We define it in the following way: we fix a point $\mathbf{x} = \mathbf{x}(u) \in \Gamma_t$ on the evolving curve Γ_t, parametrized by $u \in [0,1]$. Then we define

$$(\mathbf{x}_0 - \mathbf{x})(u) = \underset{\mathbf{v} \in \chi_u}{\arg\min} |\mathbf{v}|, \tag{6}$$

where

$$\chi_u = \{\mathbf{v} \in \mathbb{R}^2 : \mathbf{v} = \mathbf{x}_0(q) - \mathbf{x}(u), q \in [0,1]\}.$$

For every point on the evolving curve, parametrized by $u \in [0,1]$, $(\mathbf{x}_0 - \mathbf{x})$ selects the point on the original curve, parametrized by $q \in [0,1]$, which realizes the shortest distance and considers the vector $\mathbf{v} = \mathbf{x}_0(q) - \mathbf{x}(u)$. Then, since the tangential velocity does not influence the shape of the evolving curve, we consider only $w(\mathbf{x}, t)$, the component of the vector in the normal direction.

We rewrite equation (3) as

$$\frac{\partial \mathbf{x}}{\partial t} = \beta \mathbf{N}, \tag{7}$$

where the normal velocity β is given by

$$\beta = -\delta k + \lambda w. \tag{8}$$

By construction of the model, the values of the curvature regularization term $-\delta k$ and of the attracting term λw always have opposite sign, therefore for some nonzero values of δ and λ the normal velocity β will be zero. Then, since the tangential velocity α only influences the distribution of the grid points, the evolving curve will reach an equilibrium state.

On the other hand, from the numerical point of view, the tangential velocity redistributes the points on the curve, see [3, 5, 8, 9], this improves both the stability and the robustness of numerical computations. To redistribute the points asymptotically uniformly, we enrich the model as follows

$$\frac{\partial \mathbf{x}}{\partial t} = \beta \mathbf{N} + \alpha \mathbf{T}, \tag{9}$$

where β is defined in (8) and α will be defined in Section 2.1.

Equation (1) can be helpful in different applications where one needs to smooth a curve and remove the noise while keeping the curve similar to the original. In [2, 11, 12], models have been proposed for segmentation and tracking of immune system cells. Once these trajectories are obtained, one may be interested in finding the actual velocity of the cells (see for example [4]). We applied the proposed model for smoothing macrophage trajectories. Our goal was to derive the actual velocity of the cell starting from the curve found connecting the cell center in every frame of a 2D video [12]. Consequently, we needed to remove the random motion of the cell and correct possible imperfections in cell centers detection while keeping the final trajectory as close to the original as possible. We discuss the results in Section 3.

2. Numerical algorithm

To apply the numerical scheme, the curve is discretized to a set of points, $\mathbf{x}_0,\ \mathbf{x}_1, \ldots,\ \mathbf{x}_{n+1}$ as displayed in Fig. 1.

Let $g = |\mathbf{x}_u| = \sqrt{(\frac{\partial x_1}{\partial u})^2 + (\frac{\partial x_2}{\partial u})^2}$. If we denote by s the unit arc-length parametrization of the curve Γ, then $ds = |\mathbf{x}_u| du = g du$ and $du = \frac{1}{g} ds$. The unit tangent vector $\mathbf{T}$ is defined as $\mathbf{T} = \frac{\partial \mathbf{x}}{\partial s} = \mathbf{x}_s$ while the unit normal vector $\mathbf{N}$ is $\mathbf{N} = \mathbf{x}_s^{\perp}$ such that $\mathbf{T} \wedge \mathbf{N} = -1$. If $\mathbf{T} = (\frac{x_1}{\partial s}, \frac{x_2}{\partial s})$, then $\mathbf{N} = (\frac{x_2}{\partial s}, -\frac{x_1}{\partial s})$.

From the Frenet-Serret formulas we have $\mathbf{T}_s = -k\mathbf{N}$ and $\mathbf{N}_s = k\mathbf{T}$, where k is the curvature. It follows that $-k\mathbf{N} = \mathbf{T}_s = (\mathbf{x}_s)_s = \mathbf{x}_{ss}$. We can rewrite (9) into the form of the so-called intrinsic partial differential equation

$$\mathbf{x}_t = \delta \mathbf{x}_{ss} + \alpha \mathbf{x}_s + \lambda w \mathbf{x}_s^{\perp} \tag{10}$$

which is suitable for numerical discretization. Since $\mathbf{x} = (x_1, x_2)$, (10) represents a system of two partial differential equations for the two components of the position vector $\mathbf{x}$.

2.1. Suitable choice of tangential velocity

In [3, 5, 8, 9], it has been shown how to choose the tangential velocity to have an asymptotically uniform redistribution of points, and in this section, we recall it for completeness. If we want to redistribute the points along the curve, we have to consider the ratio $\frac{g}{L}$, where $g = |\partial_u \mathbf{x}|$ represents the local length of the curve while L is the global length. In the discrete form, we obtain

$$\frac{g}{L} \approx \frac{\frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{h}}{L} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{Lh} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{\frac{L}{n+1}},$$

where $h = \frac{1}{n+1}$. We define the i-th element $\mathbf{h}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ with length $h_i = |\mathbf{h}_i|$. In the formula above, $n+1$ is the number of elements used in the spatial discretization based on the flowing finite volume method. The numerator represents the distance between two grid points, and on the other hand, the denominator represents the distance that would occur if we had uniformly distributed points. It is then clear that if we want uniformly distributed points, we will have to ask for that ratio to tend to 1. In continuous setting, we have to fulfill the condition

$$\frac{g}{L} \to 1.$$

For the time evolution of the ratio, we obtain

$$\left(\frac{g}{L}\right)_t = \frac{g_t L - L_t g}{L^2}. \tag{11}$$

First, we observe that the time derivative of the local length is obtained using the Frenet-Serret formulas and is given by

$$\begin{aligned} g_t = |\mathbf{x}_u|_t &= \frac{\mathbf{x}_u}{|\mathbf{x}_u|} \cdot (\mathbf{x}_u)_t = \frac{g\mathbf{x}_s}{g} \cdot (\mathbf{x}_t)_u \\ &= \mathbf{T} \cdot (\beta \mathbf{N} + \alpha \mathbf{T})_u = \mathbf{T} \cdot g(\beta \mathbf{N} + \alpha \mathbf{T})_s \\ &= \mathbf{T} \cdot g(\beta_s \mathbf{N} + \beta \mathbf{N}_s + \alpha_s \mathbf{T} + \alpha \mathbf{T}_s) \\ &= \mathbf{T} \cdot g(\beta_s \mathbf{N} + k\beta \mathbf{T} + \alpha_s \mathbf{T} - k\alpha \mathbf{N}) \\ &= gk\beta + g\alpha_s = gk\beta + \alpha_u. \end{aligned} \tag{12}$$

To get the formula for the global length, we have to integrate $g = |\mathbf{x}_u|$ over the curve Γ

$$L = \int_{\Gamma} g du = \int_0^1 g du,$$

and then

$$L_t = \int_0^1 g_t du = \int_0^1 gk\beta du + \int_0^1 \alpha_u du = \int_0^1 k\beta ds + \alpha(1) - \alpha(0). \tag{13}$$

Now, since we fixed the endpoints of the open curve, we get $\alpha(1) = \alpha(0) = 0$, and equation (13) reduces to

$$L_t = \int_0^1 k\beta ds = L\langle k\beta\rangle_\Gamma, \tag{14}$$

where $\langle k\beta\rangle_\Gamma = \frac{1}{L}\int_\Gamma k\beta ds$. Substituting (12) and (14) in (11), we obtain

$$\left(\frac{g}{L}\right)_t = \frac{(gk\beta + \alpha_u)L - Lg\langle k\beta\rangle_\Gamma}{L^2} = \frac{g}{L}(k\beta + \alpha_s - \langle k\beta\rangle_\Gamma). \tag{15}$$

If we now impose that $\left(\frac{g}{L}\right)_t = \omega\left(1 - \frac{g}{L}\right)$, where ω is a parameter that determines the speed of convergence of the ratio, we get the desired condition $\frac{g}{L} \to 1$. Finally, we obtain the formula for the tangential velocity

$$\alpha_s = \langle k\beta\rangle_\Gamma - k\beta + \omega\left(\frac{L}{g} - 1\right). \tag{16}$$

2.2. Numerical discretization

Let us consider the following form of the intrinsic PDE (10)

$$\mathbf{x}_t - \alpha\mathbf{x}_s = \delta\mathbf{x}_{ss} + \lambda w\mathbf{x}_s^\perp, \tag{17}$$

where α and w are given by (16) and (5), respectively. First, we perform the spatial discretization based on the flowing finite volume method [7, 9], then we discuss the semi-implicit time discretization, which is implicit in the intrinsic diffusion term and uses the inflow-implicit/outflow-explicit strategy for the intrinsic advection term [6].

Consider an open curve discretized into $n + 2$ grid points. We fixed the two points $\mathbf{x}_0$ and $\mathbf{x}_{n+1}$ at the endpoints of the curve. Consequently, for these two points, the velocity is zero. We define $h_i = |\mathbf{x}_i - \mathbf{x}_{i-1}|$ and consider the finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$, where $\mathbf{x}_{i+\frac{1}{2}}$ represents the middle point between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, i.e.,

$$\mathbf{x}_{i+\frac{1}{2}} = \frac{\mathbf{x}_i + \mathbf{x}_{i+1}}{2}.$$

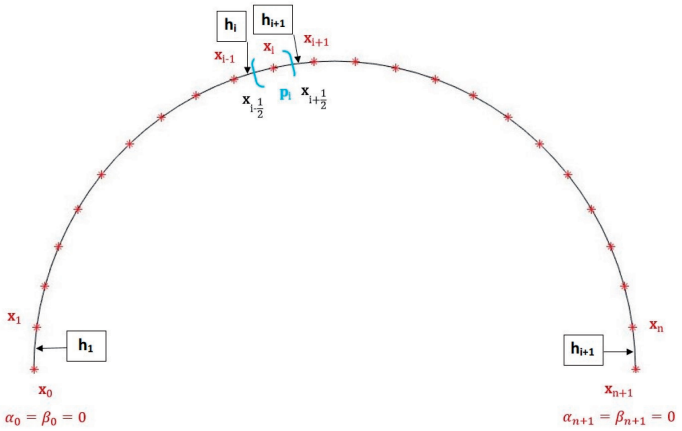


FIGURE 1. Open curve discretization. The finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$ is highlighted in blue.

Notice that the velocities $\alpha_i,\ \beta_i$ will be referred to the points $\mathbf{x}_i, i = 0, \ldots, n+1$ with $\alpha_0 = 0, \beta_0 = 0$ and $\alpha_{n+1} = 0, \beta_{n+1} = 0$. On the other hand, the curvature k_i will be referred to the elements $\mathbf{h}_i,\ i = 1, \ldots, n+1$. Integrating (17) over the finite volume $\mathbf{p}_i$, see Fig. 1, we get

$$\int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_t ds - \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \alpha \mathbf{x}_s ds = \delta \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_{ss} ds + \lambda \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} w \mathbf{x}_s^{\perp} ds, \tag{18}$$

where $\delta,\ \lambda$ are constants and the values α, w are considered to be constant over the finite volume $\mathbf{p}_i$ and will be indicated as $\alpha_i,\ w_i$. The length of the interval $\mathbf{p}_i$ is $\frac{h_i+h_{i+1}}{2}$ consequently, and using the Newton-Leibniz formula we get the following approximation of (18) for $i = 1, \ldots, n$

$$\frac{h_i + h_{i+1}}{2}(\mathbf{x}_i)_t - \alpha_i [\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} = \delta [\mathbf{x}_s]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} + \lambda w_i \left([\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \right)^{\perp}. \tag{19}$$

First of all, notice that

$$\begin{aligned} -\alpha_i [\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} &= -\alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = -\alpha_i, \left(\frac{\mathbf{x}_i + \mathbf{x}_{i+1}}{2} - \frac{\mathbf{x}_i + \mathbf{x}_{i-1}}{2} \right) \\ &= -\alpha_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right) = \frac{\alpha_i}{2}(\mathbf{x}_i - \mathbf{x}_{i+1}) - \frac{\alpha_i}{2}(\mathbf{x}_i - \mathbf{x}_{i-1}). \end{aligned} \tag{20}$$

Then, approximating $\mathbf{x}_s$ by a finite difference, we get

$$[\mathbf{x}_s]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} = \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right). \tag{21}$$

Combining (20), (21), and (19), we obtain

$$\begin{aligned} &\frac{h_i + h_{i+1}}{2}(\mathbf{x}_i)_t + \frac{\alpha_i}{2}(\mathbf{x}_i - \mathbf{x}_{i+1}) - \frac{\alpha_i}{2}(\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + \lambda w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}. \end{aligned} \tag{22}$$

Next we introduce time discretization: let m be the time step index and τ the length of the discrete time step. The basic idea of the inflow-implicit/outflow--explicit strategy is that we treat outflow from a cell explicitly while inflow implicitly [6]. If $\alpha_i^m < 0$, i.e., $(-\alpha)$ in (17) is positive in the finite volume $\mathbf{p}_i$, then there is an inflow into the finite volume through its boundary point $\mathbf{x}_{i-\frac{1}{2}}$ and there is an outflow through the other boundary point $\mathbf{x}_{i+\frac{1}{2}}$. On the other hand, if $\alpha_i^m > 0$, there is outflow through $\mathbf{x}_{i-\frac{1}{2}}$ and inflow through $\mathbf{x}_{i+\frac{1}{2}}$. We therefore define

$$\begin{aligned} b^{in}_{i-\frac{1}{2}} &= \max(-\alpha_i^m, 0), b^{out}_{i-\frac{1}{2}} = \min(-\alpha_i^m, 0), \\ b^{in}_{i+\frac{1}{2}} &= \max(\alpha_i^m, 0), b^{out}_{i+\frac{1}{2}} = \min(\alpha_i^m, 0), \end{aligned} \tag{23}$$

and rewrite equation (22) as

$$\begin{aligned} &\frac{h_i^m + h_{i+1}^m}{2}(\mathbf{x}_i)_t + \frac{1}{2}\left(b^{in}_{i+\frac{1}{2}}\right)\left(\mathbf{x}_i^{m+1} - \mathbf{x}_{i+1}^{m+1}\right) + \frac{1}{2}\left(b^{out}_{i+\frac{1}{2}}\right)\left(\mathbf{x}_i^{m} - \mathbf{x}_{i+1}^{m}\right) \\ &\quad + \frac{1}{2}\left(b^{in}_{i-\frac{1}{2}}\right)\left(\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1}\right) + \left(b^{out}_{i-\frac{1}{2}}\right)\left(\mathbf{x}_i^{m} - \mathbf{x}_{i-1}^{m+1}\right) \\ &= \delta \left(\frac{\mathbf{x}_{i+1}^{m+1} - \mathbf{x}_i^{m+1}}{h_{i+1}^m} - \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1}}{h_i^m} \right) + \lambda w_i^m \left(\frac{\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m}{2} \right)^{\perp}. \end{aligned} \tag{24}$$

Let us approximate the time derivative by the finite difference $\mathbf{x}_t = \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_i^m}{\tau}$. Note that we take the unknowns in the inflow part of the advection term implicitly while taking the ones in the outflow part explicitly. Moreover, we take the diffusion term implicitly and the attracting term explicitly.

Finally, we obtain the fully discrete scheme

$$\begin{aligned}
&\mathbf{x}_{i-1}^{m+1}\left(-\frac{\delta}{h_i^m}-\frac{b_{i-\frac{1}{2}}^{in}}{2}\right)+\mathbf{x}_{i+1}^{m+1}\left(-\frac{\delta}{h_{i+1}^m}-\frac{b_{i+\frac{1}{2}}^{in}}{2}\right)\\
&\quad+\mathbf{x}_i^{m+1}\left(\frac{h_{i+1}^m+h_i^m}{2\tau}+\frac{\delta}{h_i^m}+\frac{\delta}{h_{i+1}^m}+\frac{b_{i-\frac{1}{2}}^{in}}{2}+\frac{b_{i+\frac{1}{2}}^{in}}{2}\right)\\
&=\mathbf{x}_i^m\frac{h_{i+1}^m+h_i^m}{2\tau}-\frac{b_{i+\frac{1}{2}}^{out}}{2}(\mathbf{x}_i^m-\mathbf{x}_{i+1}^m)\\
&\quad-\frac{b_{i-\frac{1}{2}}^{out}}{2}(\mathbf{x}_i^m-\mathbf{x}_{i-1}^m)+\lambda w_i^m\left(\frac{\mathbf{x}_{i+1}^m-\mathbf{x}_{i-1}^m}{2}\right)^{\perp},
\end{aligned}\tag{25}$$

for $i=1,\dots,n$ where n is the number of unknown grid points.

The system (25) is represented by a strictly diagonally dominant matrix, then it is always solvable by the classical Thomas algorithm without any restriction on the time step τ. In the numerical scheme (25), there are two parameters α_i^m and w_i^m given by (16) and (5) for which we have not yet defined the discretization.

Let us first consider the discretization of w_i^m referred to the curve grid point $\mathbf{x}_i^m$. Note that the shortest distance of a point to a curve is in the normal direction to the curve. The algorithm of finding the point $\mathbf{q}$ which realizes the minimum distance is organized as follows: consider a point $\mathbf{x}_i^m$ on the evolving curve. Set $D_{\min}=D$ (where D is a real number to be chosen reasonably) and for $j=1,\dots,n+1$ repeat:

(1) consider the line r passing through the points $\mathbf{x}_{j-1}^0$ and $\mathbf{x}_j^0$ of the original curve,
(2) find the line s perpendicular to r and passing through $\mathbf{x}_i^m$,
(3) find the point $\mathbf{q}$ of intersection between the line r and the line s,
(4) find $d=d(\mathbf{q},\mathbf{x}_i^m)$, where d is the Euclidean distance of $\mathbf{q}$ and $\mathbf{x}_i^m$,
(5) if $\mathbf{q}\in[\mathbf{x}_{j-1}^0,\mathbf{x}_j^0]$ and $d<D_{\min}$ then

$$\mathbf{x}^0-\mathbf{x}_i^m=\mathbf{q}-\mathbf{x}_i^m,\qquad D_{\min}=|\mathbf{x}^0-\mathbf{x}_i^m|\,.\tag{26}$$

Due to the discretization of the curve, it may happen that the algorithm does not find any point $\mathbf{q}$. In this case, for every $j=0,\dots,n+1$, we check the Euclidean distance $d(\mathbf{x}_j^0,\mathbf{x}_i^m)$ and we select the minimum $j=j_{\min}$. Then,

$$\mathbf{x}^0-\mathbf{x}_i^m=\mathbf{x}_{j_{\min}}^0-\mathbf{x}_i^m.$$

This approach is natural since we are trying to find the minimum distance between a point and a segment: it is either the minimum distance found in (26)

or the minimum distance between the point and one of the endpoints of the segment. Therefore, we get

$$w_i^m = (\mathbf{x}^0 - \mathbf{x}_i^m) \cdot \mathbf{N}_i^m, \tag{27}$$

where $\mathbf{N}_i^m = \left(\frac{\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m}{h_i^m + h_{i+1}^m}\right)^{\perp}$.

Consider now the discretization of α_i^m. Note that, since we fixed the first and the last point, we set $\alpha_0^m = 0$ and $\alpha_{n+1}^m = 0$ for every $m \geq 0$. We get α_i^m for $i = 1, \ldots, n$ by

$$\alpha_i^m = \alpha_{i-1}^m + h_i^m \langle k\beta \rangle_\Gamma^m - h_i^m k_i^m \beta_i^m + \omega \left(\frac{L^m}{n+1} - h_i^m \right). \tag{28}$$

The curvature k_i^m, the normal velocity β_i^m, the mean value $\langle k\beta \rangle_\Gamma^m$ and the total length L^m are given by the following formulas [1,9,10]

$$\begin{aligned} k_i^m &= \operatorname{sgn}(\mathbf{h}_{i-1}^m \wedge \mathbf{h}_{i+1}^m) \tfrac{1}{2h_i^m} \arccos\left(\tfrac{\mathbf{h}_{i-1}^m \cdot \mathbf{h}_{i+1}^m}{h_{i-1}^m h_{i+1}^m}\right), \\ \beta_i^m &= -\delta k_i^m + \lambda w_i^m, \\ \langle k\beta \rangle_\Gamma^m &= \tfrac{1}{L^m} \textstyle\sum_{l=1}^{n+1} h_l^m k_l^m \beta_i^m, \\ L^m &= \textstyle\sum_{l=1}^{n+1} h_l^m, \end{aligned} \tag{29}$$

where

$$\mathbf{h}_i^m = \mathbf{x}_i^m - \mathbf{x}_{i-1}^m, \quad |\mathbf{h}_i^m| = h_i^m \quad \text{and} \quad \mathbf{h}_{i-1}^m \wedge \mathbf{h}_{i+1}^m$$

is the wedge product, i.e., the determinant of the matrix with columns

$$\mathbf{h}_{i-1}^m \quad \text{and} \quad \mathbf{h}_{i+1}^m.$$

For the first and last element, since we don't have the values of h_{i-1}^m, respectively h_{i+1}^m, we set $k_1^m = k_2^m$ and $k_{n+1}^m = k_n^m$.

3. Numerical experiments

3.1. Smoothed trajectories

First, we considered the simple curve shown in Figs. 2-4 to see the influence of the term which attracts the curve to the initial condition. In Fig. 2, we considered the evolution of the curve driven by the PDE (1) with $\lambda = 0$ and $\lambda = 1$, $\delta = 0.001$, $\omega = 1$, and $\tau = 0.001$ for 10000 time steps. The red line represents the initial condition while the blue lines are the results of the smoothing plotted every 1000-th time step. The curve evolved only by the curvature influence goes to the straight line, while if we consider $\lambda = 1$, the curve is attracted to the initial condition and stays closer to it. For this example, we observed that the numerical solution of the model approaches the stationary solution.

In Fig. 3 we considered the same initial curve as in Fig. 2 and we plotted the results of the smoothing after 400 time steps for $\lambda = 0$, $\lambda = 1$, $\lambda = 5$ and $\lambda = 10$, $\delta = 0.001$, $\omega = 1$, and $\tau = 0.001$: we observe that considering bigger λ, the smoothed curve stays closer to the original one. To understand better the influence of the attracting term in Fig. 4, we plotted the vectors driving one of the discrete points of the evolving curve: the blue arrow is the vector $(\mathbf{x}_0 - \mathbf{x})$, the red arrow is its projection in the normal direction to the evolving curve, and the green one is the diffusion term. We observe that the attracting vector always points towards the original curve: it is clear that, if we consider bigger λ, the result will stay closer to the initial condition.

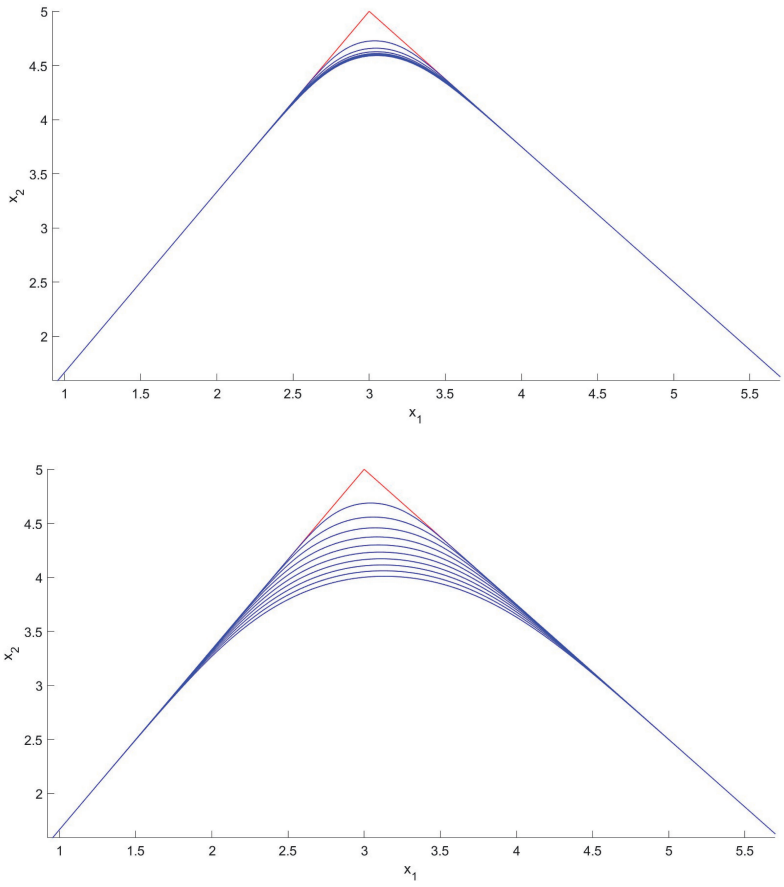


FIGURE 2. Evolution of the initial curve (red) with $\lambda = 1$ (top) and $\lambda = 0$ (bottom), $\delta = 0.001$, $\omega = 1$, and $\tau = 0.001$. The blue curves are the evolved curves plotted every 1000-th time step, from the time step 1000 to 10000.

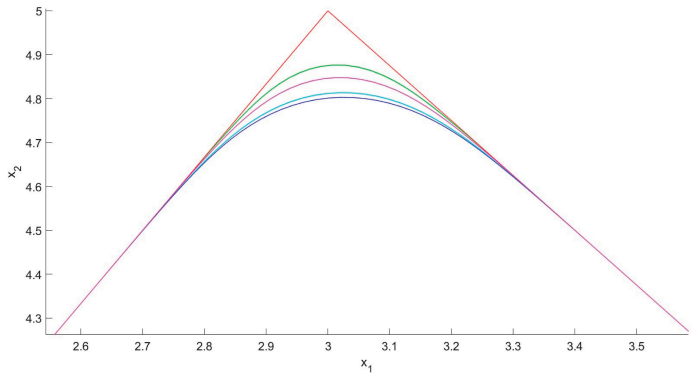


FIGURE 3. Comparison of the evolution of the initial curve (red) for different values of λ after 400 time steps. Results are shown for $\lambda = 0$ (blue), $\lambda = 1$ (light blue), $\lambda = 5$ (pink) and $\lambda = 10$ (green), $\delta = 0.001$, $\omega = 1$, and $\tau = 0.001$.

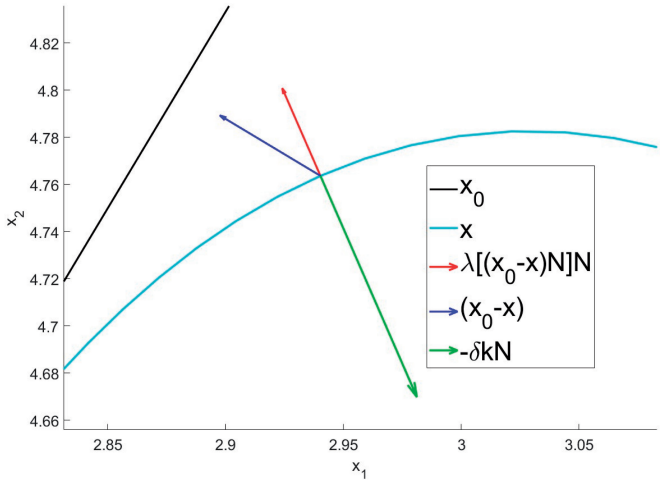


FIGURE 4. Plotting of vectors driving one of the curve grid points during the evolution. $-\delta k\mathbf{N}$ (green), $(\mathbf{x}_0 - \mathbf{x})$ (blue), and its projection in normal direction to the curve $[(\mathbf{x}_0 - \mathbf{x}) \cdot \mathbf{N}]\mathbf{N}$ (red).

Then, we focused on real data of macrophage motion. In the following experiments, we applied the proposed model to macrophage trajectories. The piecewise linear initial condition was found connecting the cell center of the macrophage for every frame of the 2D+time data [12]. First, we added grid points in the original trajectory so that we had almost uniformly distributed points. For every trajectory, the parameters were chosen as follows:

$$\lambda = 1, \quad \delta = 0.005, \quad \omega = 1, \quad \text{and} \quad \tau = 0.0001.$$

Note that, due to the tangential redistribution of points, in order to get a stable solution we should consider the time step $\tau < \frac{h_{\min}}{2\alpha_{max}}$, where $h_{\min} = \min_i h_i$ and $\alpha_{max} = \max_i \alpha_i$. Figs. 5–9 show the result of the smoothing for two different trajectories. The red lines in Figs. 5 and 8 are the original trajectories for two different macrophages. As one can observe, due to the random motion of the cells, these trajectories usually have noisy parts: Figs 6, 7, and 9 show the details of these parts until the curve is completely unknotted. If one wants to keep the curve closer to the original trajectory, then it is necessary to choose a bigger value for λ. The blue lines are the smoothed trajectories after different numbers of time steps: the evolving curve is kept close to the original trajectory by the attracting term but we also observe the smoothing of the noisy parts due to the diffusion term. In Figs. 6 and 7, we show the result of the smoothing until the random part of the trajectory is completely smoothed: the black dot represents the last point of the trajectory.

As a stopping criterion, we considered the mean Hausdorff distance between two discrete curves. Consider the discrete sets of points:

$$\mathcal{A} = \{a_0, \ldots, a_{n+1}\}, \qquad \mathcal{B} = \{b_0, \ldots, b_{n+1}\}$$

we will indicate the elements of the discrete curves by

$$\mathbf{a}_i = a_i - a_{i-1} \quad \text{and} \quad \mathbf{b}_i = b_i - b_{i-1}, \quad i = 1, \ldots, n+1,$$

respectively. Let us indicate by $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_{n+1}\}$ and $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}\}$ the sets of elements. The mean Hausdorff distance $\bar{d}_H(\mathcal{A}, \mathcal{B})$ is then defined as

$$\begin{aligned} \bar{\delta}_H(\mathcal{A}, \mathcal{B}) &= \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{b} \in B} d(a_i, \mathbf{b}), \\ \bar{\delta}_H(\mathcal{B}, \mathcal{A}) &= \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{a} \in A} d(b_i, \mathbf{a}), \\ \bar{d}_H(\mathcal{A}, \mathcal{B}) &= \frac{\bar{\delta}_H(\mathcal{A}, \mathcal{B}) + \bar{\delta}_H(\mathcal{B}, \mathcal{A})}{2}, \end{aligned} \tag{30}$$

where $d(a_i, \mathbf{b})$ is the distance between a point $a_i \in \mathcal{A}$ and an element $\mathbf{b} \in B$ defined in Section 2.2. Since in our model the endpoints $a_0,\ a_{n+1},\ b_0,\ b_{n+1}$ are fixed, we do not consider them in the computation of the mean Hausdorff distance. In order not to slow down the computation too much, this distance was calculated only every p number of time steps. We chose the tolerance $\epsilon = 0.000065$ and stopped iterating when $d_H(\mathbf{x}_n, \mathbf{x}_{n+p}) < \epsilon$. For the trajectory in Fig. 5, the number of time steps needed to stop the evolution was 580, while for the one in Fig. 8, it was 340.

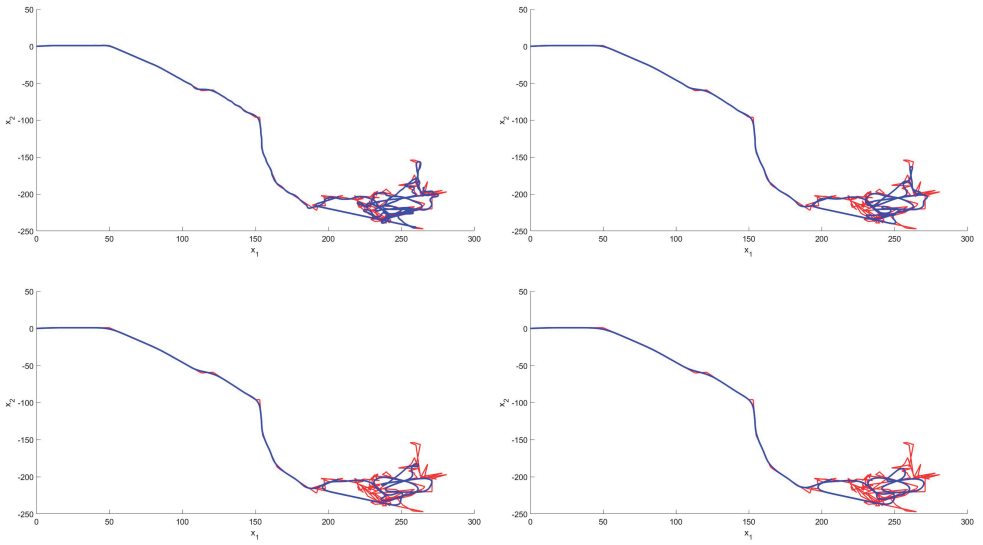


FIGURE 5. The original trajectory (red line) and smoothed trajectory (blue line). Results (from top to bottom) after 50, 200, 400 and 580 time steps.

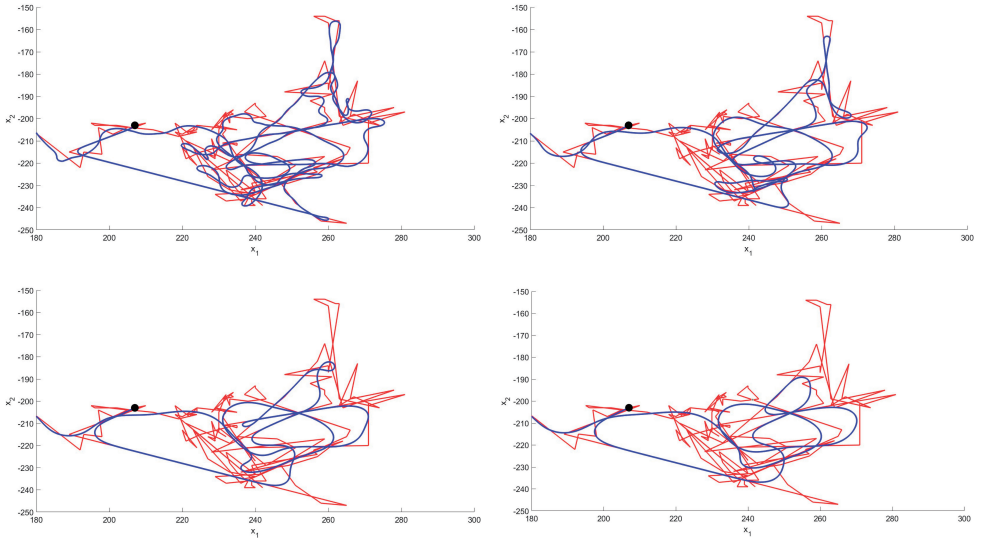


FIGURE 6. A detail of noisy part of trajectory in Fig. 5. Original trajectory (red line) and smoothed trajectory (blue line). Results (from top to bottom) after 50, 200, 400 and 580 time steps. The black dot represents the last point of the trajectory.

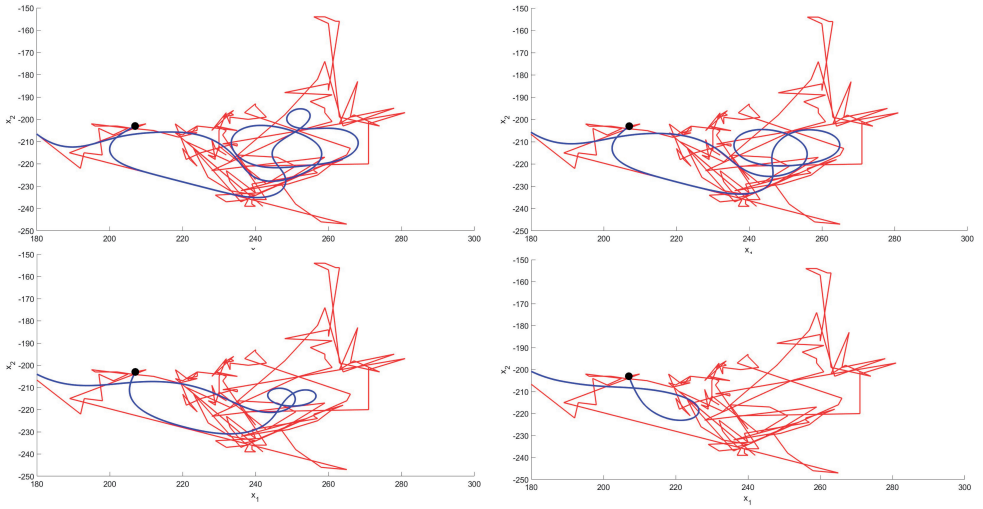


FIGURE 7. A detail of noisy part of trajectory in Fig. 5. Original trajectory (red line) and smoothed trajectory (blue line). Results (from top to bottom) after 1000, 1500, 2500, and 5000 time steps. The black dot represents the last point of the trajectory.

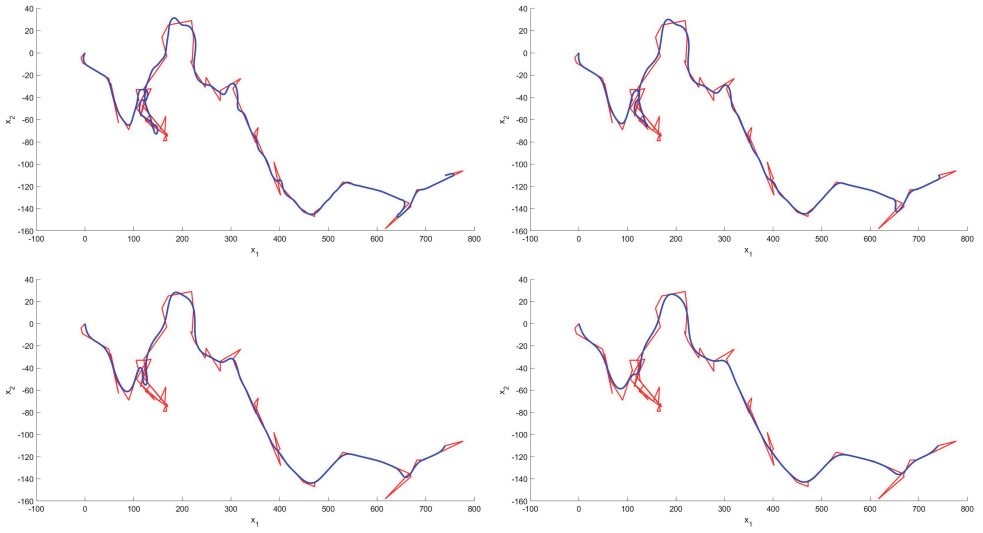


FIGURE 8. The original trajectory (red line) and smoothed trajectory (blue line). Results (from top to bottom) after 50, 100, 200 and 340 time steps.

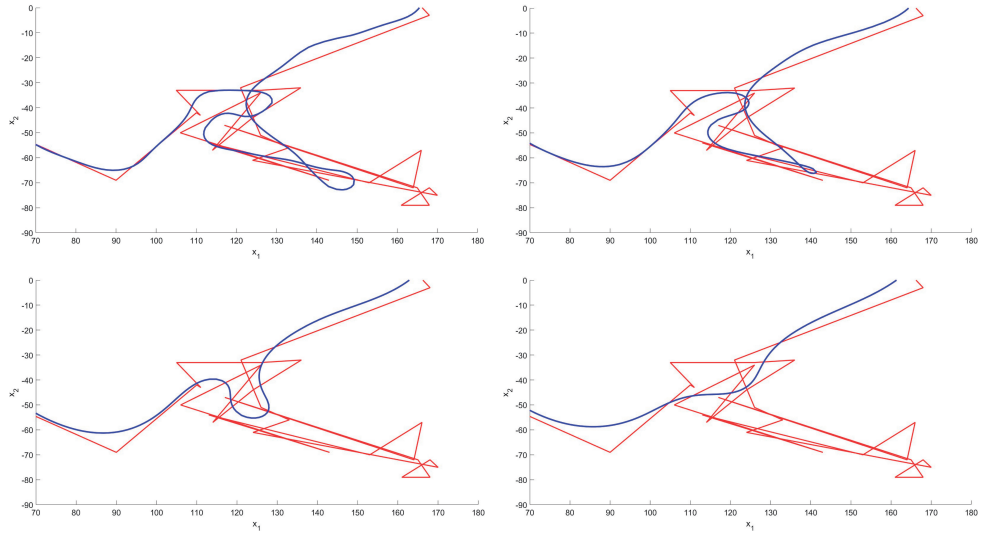


FIGURE 9. A detail of noisy part of trajectory in Fig. 8. Original trajectory (red line) and smoothed trajectory (blue line). Results (from top to bottom) after 50, 100, 200 and 340 time steps.

3.2. Velocities on smoothed curves

Our final goal was to compute the velocity for every grid point on the evolved curve. To do so, we considered the lengths of the initial segments, i.e., the original points without the added grid points. Then, we evolved them according to the formula already shown in Section 2.1 instead of computing the real lengths of the segments on the smoothed curve. This is because we added grid points in the original segments and these points are redistributed along the curve during the evolution, so the real lengths give no meaningful information to calculate velocity.

In the 2D+time data, the real-time between each frame was 1 time unit between the frames, which is equal to 2 minutes. That means, in the original piecewise linear trajectory, the time between two endpoints of a segment was 1 time unit (see Fig. 10). First, when we added the grid points inside the initial segments, we did it in such a way that the new added points were uniformly distributed inside the segment. Consequently, we calculated the original velocity for the grid points as the original length of the segment divided by the time interval Δ_t, that in our case was 1 (time unit), and consider it constant inside the segment. Therefore, in the beginning, all the added grid points inside a segment have the same velocity. We assume this property holds also for the new velocities of the grid points on the evolved curve. Our goal was to derive the velocity

for every grid point on the evolved curve but we couldn't do it directly because, due to the stability of the numerical computations, we added the tangential velocity in our model and this caused an asymptotically uniform redistribution of the points. If the trajectory is not too complicated, the tangential velocity does not move the points too much. But, in our case, some of the trajectories had noisy parts with many points, which were moved a lot during the evolution, so we were losing almost completely the spatial information.

Then, to get the new velocity estimation, first we studied the evolution of the length of the segments. For the total length, we have already shown in Section 2.1 that it holds

$$L_t = \int_0^1 k\beta ds + \int_0^1 \alpha_s ds, = \int_0^1 k\beta ds + \alpha(1) - \alpha(0). \tag{31}$$

In the case of the total length, we had fixed the endpoints of the trajectory so we had $\alpha(1) = \alpha(0) = 0$. Similarly, for the evolution of every segment, we can consider the same formula. We will denote by j the index for the segments and by i the index for the grid points. Consider the jth segment: we will indicate by u_{j-1} and u_j the first and last endpoint of the segment parametrization, respectively. Therefore, for the evolution of the length L_j of the jth segment, it holds

$$(L_j)_t = \int_{u_{j-1}}^{u_j} k\beta ds + \int_{u_{j-1}}^{u_j} \alpha_s ds = \int_{u_{j-1}}^{u_j} k\beta ds + \alpha(u_j) - \alpha(u_{j-1}). \tag{32}$$

Notice that in this case, in general, we have $\alpha(u_j) \neq 0$ and $\alpha(u_{j-1}) \neq 0$. But, as we have already underlined, the tangential velocity is used only for the stability of the numerical computations and it does not influence the shape of the evolving curve. So, to not lose the spatial information, we will calculate the new length of the segment as if we have applied to the trajectory the model with $\alpha = 0$, i.e., $\alpha(u_j) = \alpha(u_{j-1}) = 0$. Consequently, we consider the following model for the evolution of the jth segment

$$(L_j)_t = \int_{u_{j-1}}^{u_j} k\beta ds. \tag{33}$$

Let's now consider the discretization: let m be the time step index and τ the length of the discrete time step. We approximate the time derivative by the finite difference and consider the formulas for k, β defined in (29). To relate the index j of the segments to the index i of the grid points, we will indicate by $\mathcal{I}(u_{j-1})$, $\mathcal{I}(u_j)$ the corresponding i index of the endpoint $\mathbf{x}(u_{j-1})$ and $\mathbf{x}(u_j)$, respectively (see Fig. 10).

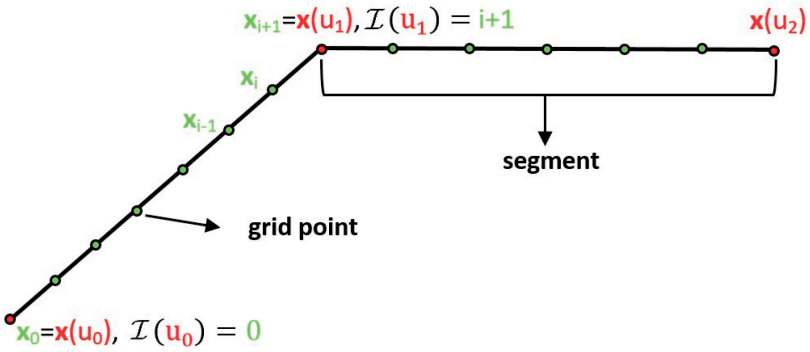


FIGURE 10. Visualization of curve discretization. The green dots represent the grid points and the red dots represent the endpoints of the segments.

Notice that we consider the i index starting from $\mathcal{I}(u_{j-1})+1$, it is because $h_{\mathcal{I}(u_{j-1})+1}$ is defined as $h_{\mathcal{I}(u_{j-1})+1} = |\mathbf{x}_{\mathcal{I}(u_{j-1})+1} - \mathbf{x}_{\mathcal{I}(u_{j-1})}|$, which is indeed the first element h_i of the segment with endpoints $\mathbf{x}(u_{j-1})$ and $\mathbf{x}(u_j)$. We obtain the fully discrete approximation of equation (33) for the new length of the jth segment

$$L_j^{m+1} = L_j^m + \tau \sum_{i=\mathcal{I}(u_{j-1})+1}^{\mathcal{I}(u_j)} h_i^m k_i^m \beta_i^m. \tag{34}$$

For the first and last element of each segment, we use a different computation of k and β that does not consider information from the previous and next segment.

If we consider the evolution of the curve driven only by the curvature influence, then $\beta_i = -\delta k_i$ and we obtain

$$\sum_{i=\mathcal{I}(u_{j-1})+1}^{\mathcal{I}(u_j)} h_i^m k_i^m \beta_i^m = -\sum_{i=\mathcal{I}(u_{j-1})+1}^{\mathcal{I}(u_j)} h_i^m \delta (k_i^m)^2 \leq 0.$$

Therefore, $L_j^{m+1} \leq L_j^m$ for every segment j and for every time step index m. Experimentally, this property holds also if $\beta_i = -\delta k_i + \lambda w_i$. If the segment is in a part of the trajectory which has high curvature (for example in the noisy parts where the cell shows random motion), the length of that segment will decrease faster than the length of a segment in a part of the trajectory with small curvature (for example in the parts where the cell shows directional motion). It is then clear that some of the segments may eventually disappear and we may obtain $L_j^{m+1} = 0$.

Let us first focus on the simple case, where the length of one segment is decreasing, but the segment is not disappearing. In the first time step, $\mathcal{I}(u_{j-1})$ and $\mathcal{I}(u_j)$ are loaded from the original trajectory and are the indexes of the endpoints of the original segments. Later, the length of every segment changes, so after applying the formula in (34), we need to find the new i indexes for $\mathcal{I}(u_{j-1})$ and $\mathcal{I}(u_j)$. If we indicate by

$$L_s^{m+1} = \sum_{j=1}^{M-1} L_j^{m+1}, \tag{35}$$

where $M-1$ is the number of original segments. Ideally, $L_s^{m+1} = L^{m+1}$, where L^{m+1} is defined in (29). However, in the numerical computations, in later time steps, it holds $L_s^{m+1} \neq L^{m+1}$. We define the ratio of the jth segment as

$$r_j^{m+1} = \frac{L_j^{m+1}}{L_s^{m+1}}, \tag{36}$$

and the discrete length of the segment is defined as

$$L_j^d = r_j^{m+1} L^{m+1}, \tag{37}$$

so that $\sum_{j=1}^{M-1} L_j^d = L^{m+1}$. Once we obtain the new lengths L_j^d, to get the new i indexes $\mathcal{I}(u_{j-1})$ and $\mathcal{I}(u_j)$ for the endpoints of the segments, we apply the following algorithm. Set $i = 1$, $j = 1$, $S = L_j^d$, $S_e = 0$ and repeat until $j = M-1$ and $i = n+1$:

- $S_e = S_e + h_i$
- if $S_e \geq S$
  - if $S_e > S$, then $\mathcal{I}(u_j) = i - 1$
  - if $S_e = S$, then $\mathcal{I}(u_j) = i$
  - update $S = S + L_{j+1}^d$
  - $j = j + 1$.
- $i = i + 1$

Notice that we focused only on the index of the last endpoint of each segment $\mathcal{I}(u_j)$, it is because the last endpoint of the segment j is the first endpoint of the segment $j+1$.

Let us now focus on the disappearing segments. We add the following condition: if $L_j^{m+1} < h_{\min}$, then $L_j^{m+1} = 0$, where

$$h_{\min} = \min_{1 \le i \le n+1} h_i.$$

If we obtain $L_j^{m+1} = 0$, it means the first and the last endpoints of that segment at time step index $m+1$ became the same point, then $\mathcal{I}(u_j) = \mathcal{I}(u_{j-1})$. So we modify the algorithm for finding the new indexes i for $\mathcal{I}(u_j)$ by adding the following condition

- if $r_j^{m+1} = \frac{L_j^{m+1}}{L_s} = 0$ then $\mathcal{I}(u_j) = \mathcal{I}(u_{j-1})$.

This condition will cause that, when we apply formula (34) for every time step index $m+k$, with $k > 0$, we will obtain

$$L_j^{m+k} = L_j^{m+1} = 0.$$

Consider two endpoints of a segment, $\mathbf{x}(u_{j-1})$ and $\mathbf{x}(u_j)$. Note that every endpoint is the position of the cell in a given time and that in our application the initial time between two consecutive endpoints was fixed to 1 time unit. To get the new velocity estimation, consider $(\Delta_t)_j$ the time between the two endpoints $\mathbf{x}(u_{j-1})$ and $\mathbf{x}(u_j)$ of the segment j. In the beginning, every segment has the same value of $(\Delta_t)_j$, later if a segment does not disappear, we consider the same $(\Delta_t)_j$ as in the beginning. On the other hand, if a segment j disappears, we add $\frac{(\Delta_t)_j}{2}$ to the first previous not disappeared segment

$$(\Delta_t)_{j-1} = (\Delta_t)_{j-1} + \frac{(\Delta_t)_j}{2}$$

and $\frac{(\Delta_t)_j}{2}$ to the first following not disappeared segment

$$(\Delta_t)_{j+1} = (\Delta_t)_{j+1} + \frac{(\Delta_t)_j}{2}.$$

To find the new velocity estimation for the grid points on the evolved curve, we apply the following algorithm. Set $j = 1$, $i = 1$ and repeat until $j = M - 1$:

- If $r_j^{m+1} > 0$

$$|\mathbf{v}| = \frac{L_j^d}{(\Delta_t)_j}. \tag{38}$$

- For $i = \mathcal{I}(u_{j-1}), \ldots, \mathcal{I}(u_j)$

$$\mathbf{v}_i^{m+1} = |\mathbf{v}| \frac{(\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1})}{|\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1}|}. \tag{39}$$

If the segment does not disappear, we consider the grid points inside that segment to have the constant velocity equal to the new length of the segment divided by the time interval $(\Delta_t)_j$. On the other hand, if the jth segment disappears, the time $(\Delta_t)_j$ is distributed between the previous and following segments that have not disappeared.

For the numerical experiments, to check our method, we set as initial condition a semi-ellipse and we considered the evolution of the curve with $\lambda = 0$, $\delta = 0.05$, $\omega = 1$ and $\tau = 0.001$. We calculated the velocities using the algorithm described above. The result is shown in Fig. 11 after 1000 time steps, while the vectors are plotted in every 15th grid point. The initial number of segments was 7, then we added grid points so that they were almost uniformly distributed and the number of grid points was 697. Once we got the velocities for every grid point, we normalized the length of the vectors to obtain better visualization, where the color represents the norm. Then, if the arrow is yellow it means in that grid point the velocity is high. On the other hand, if the arrow is blue or light blue, at that point the velocity is slower. As we expected, the length of the segment in the center of the discretized semi-ellipse decreases faster than the others. Indeed, that is the part of the ellipse with the highest curvature. Then, we focused on the real trajectories of the macrophages. We considered the evolution of the trajectories with $\lambda = 1$, $\delta = 0.005$, $\omega = 1$, and $\tau = 0.0001$. Results for three different trajectories are shown in Figs. 12–14. We considered the evolved curve obtained using the mean Hausdorff distance as a stopping criterion as described in the last paragraph of Section 3.1. We observe that the velocity is slowed down in the random parts of the original trajectory that are smoothed during the evolution.

4. Conclusions

We proposed a mathematical method based on the Lagrangian approach to smooth curves. The proposed model contains two terms: the curvature regularization term and the attracting term. The regularizing term smooths the curve, while the attracting term keeps the trajectory close to the initial condition. We described the numerical discretization based on the flowing finite volume method and we proposed a method to find the new velocity on the smoothed curves. We presented and discussed the results for macrophages trajectories for both smoothing of the trajectories and velocity estimation on the smoothed curves.

The future work will be to extend the current model for 3D+time trajectories and to get velocity estimation for those trajectories.

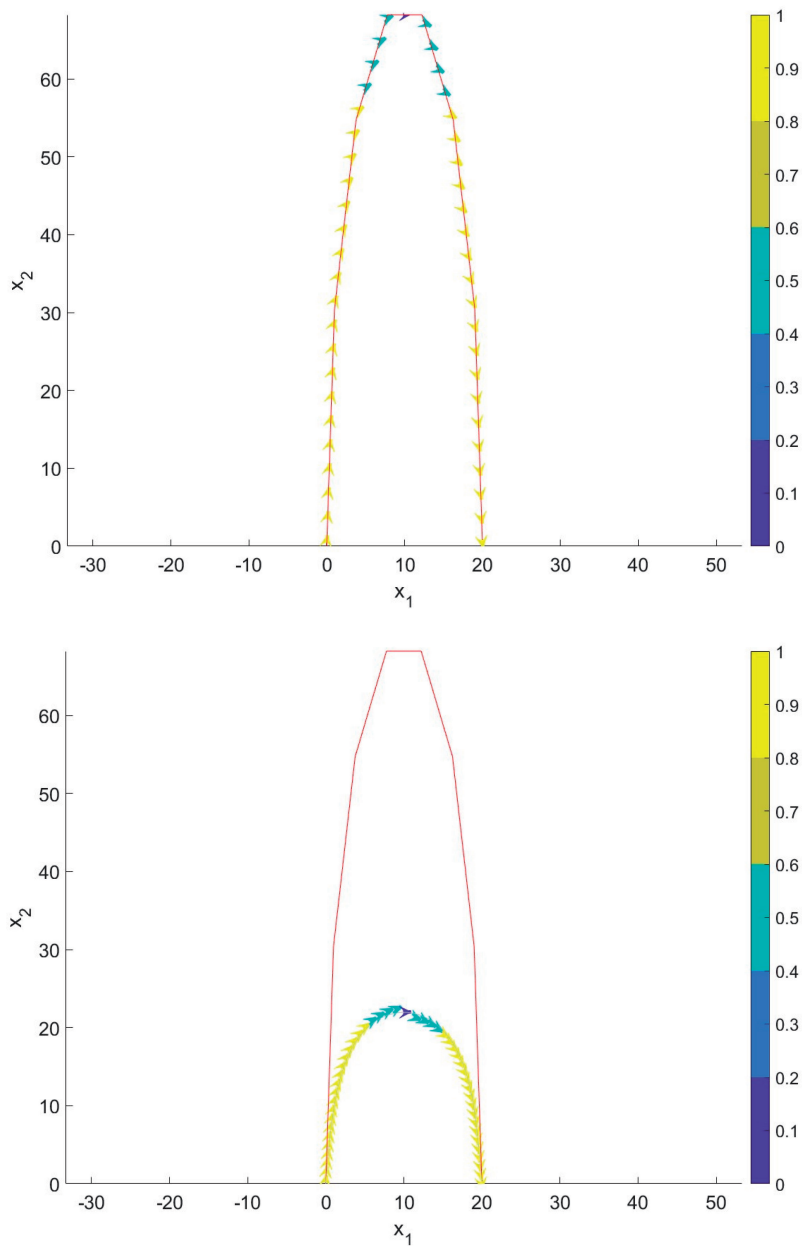


FIGURE 11. Top: initial trajectory and plotting of the velocity vectors. Bottom: initial trajectory (red line), evolved curve, and plotting of the velocity vectors. The vectors are normalized for better visualization, color represents the norm. Vectors are plotted in every 15th grid point.

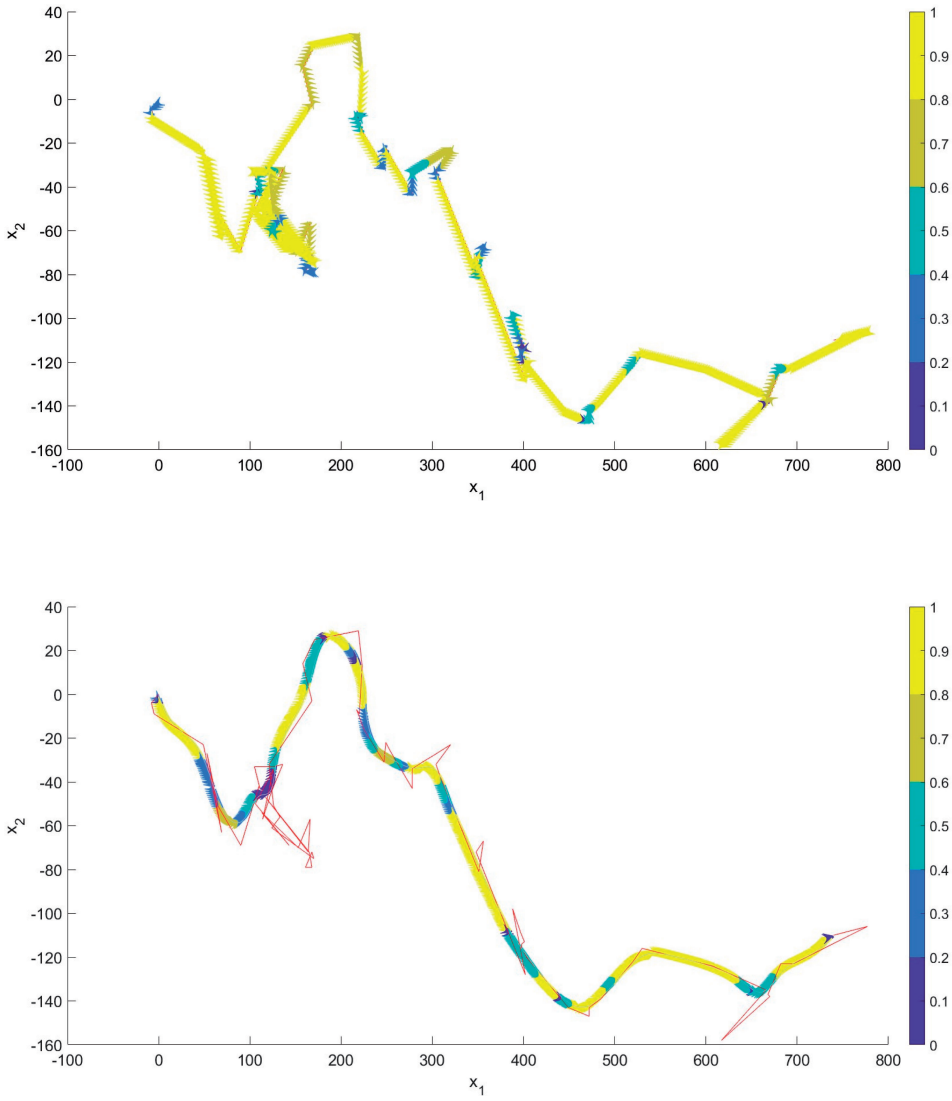


FIGURE 12. Top: initial trajectory and plotting of the velocity vectors. Bottom: initial trajectory (red line), evolved curve, and plotting of the velocity vectors. The vectors are normalized for better visualization, color represents the norm.

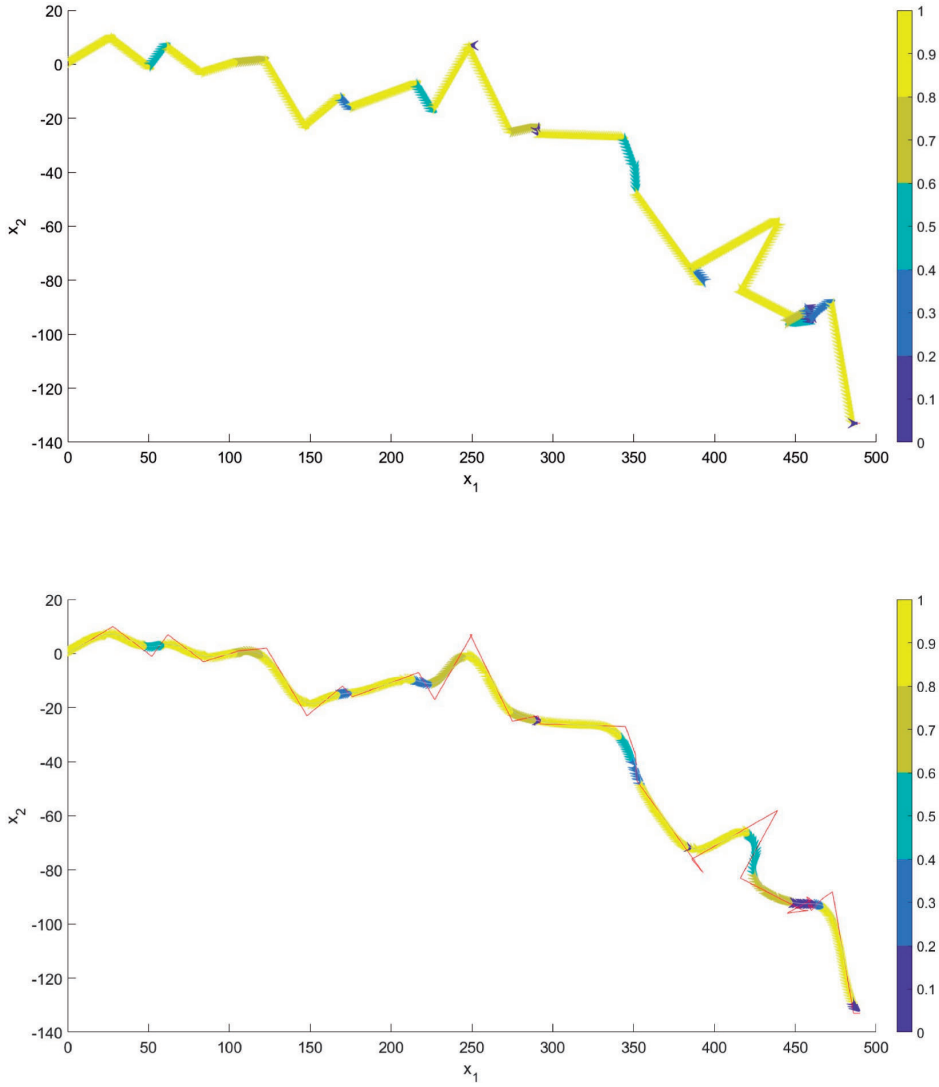


FIGURE 13. Top: initial trajectory and plotting of the velocity vectors. Bottom: initial trajectory (red line), evolved curve, and plotting of the velocity vectors. The vectors are normalized for better visualization, color represents the norm.

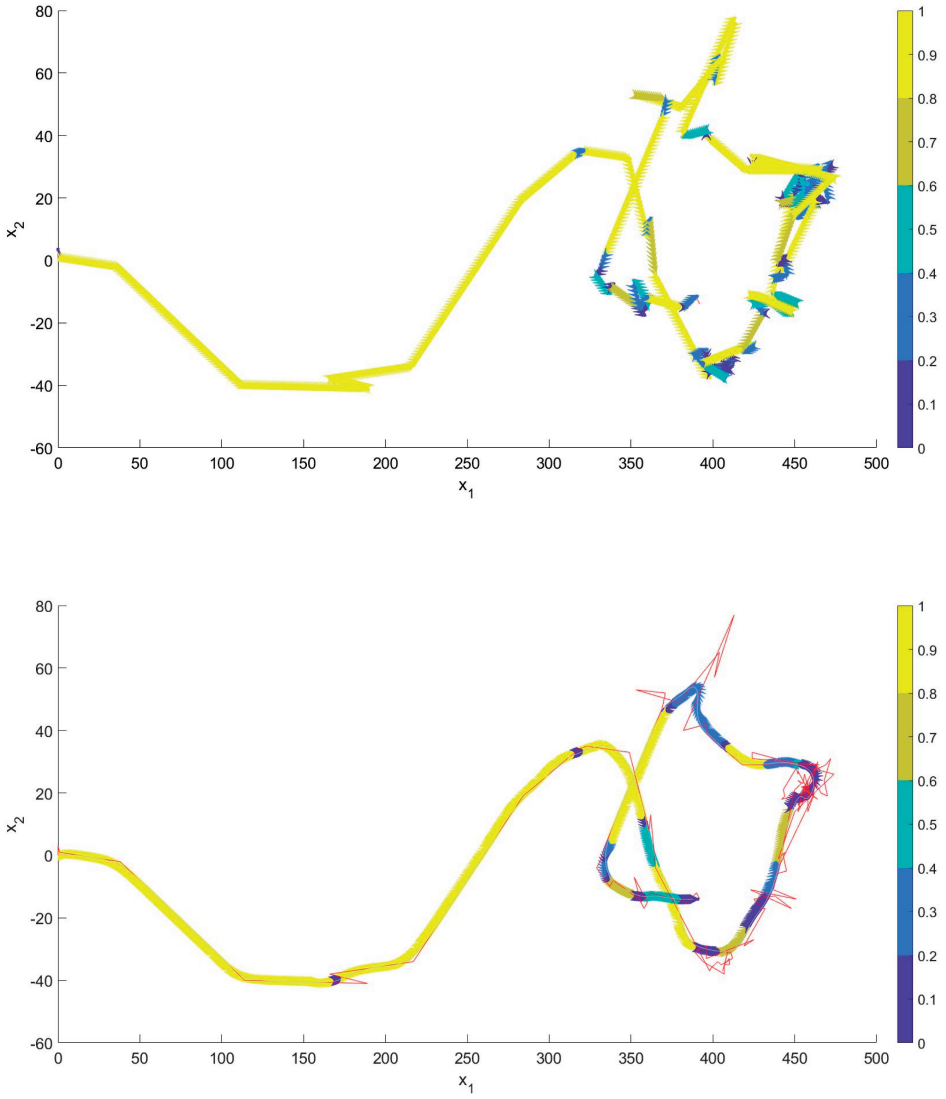


FIGURE 14. Top: initial trajectory and plotting of the velocity vectors. Bottom: initial trajectory (red line), evolved curve, and plotting of the velocity vectors. The vectors are normalized for better visualization, color represents the norm.

5. Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955576 and by the grants APVV-19-0460, VEGA 1/0436/20.

REFERENCES

[1] M.AMBROZ, M.—KOLLÁR, M.—MIKULA, K.: *Semi-implicit scheme for semiautomatic segmentation in NaturaSat software*, 21st Conference on Scientific Computing, Vysoké Tatry - Podbanské, Slovakia, September 10–15 2000, ALGORITMY 2020, (P.Frolkovič, K.Mikula and D.Ševčovič, Eds.) (Proceedings of contributed papers), pp. 171–180.

[2] HENRY, K. M.—PASE, L.—RAMOS-LOPEZ, C. F. —LIESCHKE, G. J.—RENSHAW, S. A.—REYES-ALDASORO, C. C.: *PhagoSight: An open-source MATLAB® package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model*, PLoS ONE, August (2013) e72636. https://doi.org/10.1371/journal.pone.0072636

[3] HOU, T. Y.—LOWENGRUB, J. S.—SHELLEY, M. J.: *Removing the stiffness from interfacial flows with surface tension*, J. Comput. Phys. **114** (1994), no. 2, 312–338.

[4] KADIRKAMANATHAN, V.—ANDERSON, S. R.—BILLINGS, S. A.—ZHANG, X.——HOLMES, G. R.—REYES-ALDASORO, C. C.—ELKS, P. M. —RENSHAW, S. A.: *The neutrophil's eye-view: Inference and visualisation of the chemoattractant field driving cell chemotaxis in vivo*, (2012),PloS ONE, April 26, 2012, DOI:10.1371/journal.pone.0035182, https://doi.org/10.1371/journal.pone.0035182

[5] M.KIMURA, M.: *Numerical analysis of moving boundary problems using the boundary tracking method* Jpn. J. Ind. Appl. Math. **14** (1997), no. 3, 373–398.

[6] MIKULA, K.—OHLBERGER, M.: *Inflow-implicit/outflow-explicit scheme for solving advection equations*. In: Proceedings of the Sixth International Conference on Finite Volumes in Complex Applications, Prague, June 6-10, 2011 (J.Fořt et al. Eds.) *Finite Volumes in Complex Applications VI, Problems & Perspectives, Vol. 1, 2,* (2011) pp. 683–692.

[7] MIKULA, K.—ŠEVČOVIČ, D.: *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math. **61** (2001), no. 5, 1473–1501.

[8] MIKULA, K.—ŠEVČOVIČ, D.: *A direct method for solving an anisotropic mean curvature flow of plane curves with an external force*, Math. Methods Appl. Sci. **27** (2004), no. 13, 1545–1565.

[9] K.MIKULA, —D.ŠEVČOVIČ, —M.BALAZOVJECH,: *A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations*, Commun. Comput. Phys. **7** (2010), no. 1, 195–211.

[10] MIKULA, K.—URBÁN, J.—KOLLÁR, M.—AMBROZ, M.— JAROLÍMEK, I.— ŠIBÍK, J.—ŠIBÍKOVÁ, M.: *An automated segmentation of NATURA 2000 habitats from Sentinel-2 optical data*, Discrete Contin. Dyn. Syst. Ser. S, **14** (2021), no. 3, 1017–1032.

[11] SEOL AH PARK—SIPKA, T.—KRIVÁ, Z.—AMBROZ, M.—KOLLÁR, M. — KÓSA, B.—-NGUYEN-CHI, M.—LUTFALLA, G.—MIKULA, K.: *Macrophage image segmentation by thresholding and subjective surface method*, Tatra Mt. Math. Publ. **75** (2020), 103–120.

[12] SEOL AH PARK—SIPKA, T.—KRIVÁ, Z.— M.NGUYEN-CHI, M. —LUTFALLA, G.—MIKULA, K. : *Segmentation-based tracking of macrophages in 2d+time microscopy movies inside a living animal*, Computers in Biology and Medicine, **153** (2023), Paper no. 106499.

Received November 24, 2022
Revised October 16, 2023
Accepted October 16, 2023
Publ. online October 26, 2023

Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering
Slovak University of Technology in Bratislava
Radlinského 11
810 05 Bratislava
SLOVAKIA

E-mail: giulia.lupi@stuba.sk
karol.mikula@gmail.com
standup1014@gmail.com