SIAM J. Sci. Comput.
Vol. 36, No. 4, pp. A1384–A1414

MANIFOLD EVOLUTION WITH TANGENTIAL REDISTRIBUTION OF POINTS*

KAROL MIKULA[†], MARIANA REMEŠÍKOVÁ[†], PETER SARKOCI[†], and DANIEL ŠEVČOVIČ[‡]

Abstract. In this paper we propose several techniques for tangential redistribution of points on evolving surfaces. This is an important issue in numerical approximation of any Lagrangian evolution model, since the quality of the mesh has a significant impact on the result of the computation. We explain the volume-oriented and length-oriented tangential redistribution methods in a general setting of an m-dimensional manifold evolving in an n-dimensional manifold. Then, we apply the proposed techniques to several manifold evolution problems. We explain the numerical approximation of the models and present experiments illustrating the performance of our redistribution techniques.

Key words. manifold evolution, tangential redistribution, mean curvature flow, finite volume method, surface evolution

AMS subject classifications. 53C44, 65M08, 65M50

DOI. 10.1137/130927668

1. Introduction.

Given a Riemannian manifold (possibly a manifold with boundary) (X, g_X) of dimension m and a Riemannian manifold (Y, g_Y) of dimension n, $m \leq n$, we define an *evolution of X in Y* as any smooth map $F\colon X \times [0, t_s] \to Y$ such that $F^t = F(\cdot, t)$ is an immersion for every t. Throughout the text, we will also use the notion of an *evolving manifold*—we will speak, for example, of movement of points on an evolving manifold. Here, we have in mind a one-parameter family of subsets of Y, $\{\mathrm{Im}(F^t)\}_{t\geq 0}$. We would like to emphasize that defining F^t as an immersion, we do not exclude (for the sake of generality) self-intersections of $\mathrm{Im}(F^t)$.

Since we speak about evolution, the parameter t can be naturally viewed as time. Within our considerations, we will often deal with maps of the form $f\colon X \times [0, t_s] \to Z$, where Z is some target space. The notation f^t will then always represent a time slice of the a map f, $f(\cdot, t)$. Conversely, given first a map of the form $f^t\colon X \to Z$, we will automatically consider it as a time slice of the map

$$f\colon X \times [0, t_s] \to Z\colon (x, t) \mapsto f^t(x).$$

Since F is assumed to be smooth, for a fixed $x \in X$, the map $t \mapsto F^t(x)$ is a smooth curve in Y. As such it defines, for every t, a vector $v^t(x)$ tangential to Y at the point $F^t(x)$. The map $v\colon X \times [0, t_s] \to TY$, where TY stands for the tangent bundle of Y, represents the velocity field of the evolution. Thus, F is a solution to the evolution equation

$$\partial_t F = v \tag{1.1}$$

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section July 5, 2013; accepted for publication (in revised form) April 15, 2014; published electronically July 1, 2014. This work was supported by grants APVV-0184-10, VEGA 1/0747/12, and VEGA 2/0059/12.

http://www.siam.org/journals/sisc/36-4/92766.html

[†] Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 81368 Bratislava, Slovakia (mikula@math.sk, remesikova@math.sk, sarkoci@math.sk).

[‡] Department of Applied Mathematics and Statistics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská Dolina, 84248 Bratislava, Slovakia (sevcovic@fmph.uniba.sk).

accompanied by an initial condition and, in the case of a manifold with boundary, a boundary condition. In many situations, it is useful to see v as a composition of two orthogonal components with different roles in the evolution process. Let T_xX denote the tangential space to X at x. Pushing T_xX forward along F^t, we obtain a subspace $(F^t)_*(T_xX)$ of $T_{F^t(x)}Y$; it is the tangent space at $F^t(x)$ to $\mathrm{Im}(F^t)$ (or its branch, if $F^t(x)$ happens to be a self-intersection point). This allows for the decomposition $v^t(x) = v^t_N(x)+v^t_T(x)$, where $v^t_T(x)$—the *tangential velocity*—lies in $(F^t)_*(T_xX)$ while $v^t_N(x)$—the *normal velocity*—is its orthocomplement in $T_{F^t(x)}Y$. From this follows $v = v_N + v_T$ and the evolution equation modifies to

$$\partial_t F = v_N + v_T. \tag{1.2}$$

An evolution of this type is present in many models in physics, computer vision and image processing, biology etc. Some typical examples are the evolution of a phase interface [2, 4], forest fire front propagation [3], segmentation of objects in two-, three-, or even four-dimensional (2D, 3D, or 4D) images [1, 14, 24, 25, 28], computation of minimal surfaces, e.g., in architecture [23], or modeling of molecular surfaces [5]. Two basic approaches are used for solving manifold evolution problems: the Lagrangian approach that evolves the manifold directly [6, 13, 18, 19, 20, 24, 30, 39] and the Eulerian (level set) approach that considers the m-dimensional manifold as a level set of a function of $m + 1$ variables [5, 14, 25, 28, 36, 38]. This paper follows the Lagrangian approach.

Having decomposed v according to (1.2), we can see that the velocity field consists of two substantially different parts. While the normal component changes the image of the manifold X, the tangential component only changes the images of its individual points. Thus, it allows us to change the immersion of X in Y without affecting its image.

The question of the choice of the immersion becomes important in numerical realizations of problem (1.2). Since the immersion determines the positions of discretization points, its choice is a key factor for the computation process. Inappropriately placed discretization points can lead to unacceptable numerical errors or even to a crash of the computation. This concerns not only the initial placing of the points; the quality of the mesh can deteriorate during the computation as a consequence of the prescribed movement of the points. Therefore, in many cases it is necessary to redistribute the mesh points along the manifold during the evolution process. In other situations, even if the quality of the numerical approximation is not the main question of interest, a specific distribution of mesh points might be desirable for improving the quality of representation of objects—for example, it can help to have a denser distribution of discretization points in places with high curvature. There are also applications that directly require some sort of uniform or regular distribution of points. An example is the design of shell constructions mentioned later in this paper, where an appropriate placement of the node points optimizes the process of truss manufacturing and the visual impression.

An appropriate choice of the tangential velocity in (1.2) allows us to obtain an immersion that meets our criteria without otherwise affecting the evolution process. Various techniques for tangential redistribution of points have been designed for evolving curves in two dimensions [10, 16, 22, 26, 30, 31, 39, 40, 42], in three dimensions [21, 33], and generally in $\mathbb{R}^d$ or on 2D manifolds [7, 9, 32]. However, the situation becomes more complicated for higher dimensional manifolds—the higher dimensionality of the tangent space in each point of the manifold modifies the character of the problem and gives broader possibilities. Some work has already been done for surfaces in three dimensions. The work of Morigi [35] proposes several possibilities for tangential redistribution focused either on areas, lengths or angles in a discrete mesh. Barrett, Garcke, and Nurnberg [6, 8] introduced an angle-oriented redistribution technique based on discrete conformal parametrizations. Our paper introduces several new possibilities for tangential redistribution based on controlling the volume density during the evolution. We generalize the ideas that have been previously used for curves in two and three dimensions [30, 31, 33]. Our methods allow conservation of relative volumes during the evolution or an asymptotically uniform distribution of points with respect to the volume density. The asymptotically uniform redistribution model also provides a control of the speed of redistribution and, if we want, we can adjust the model to control the limit volume of the manifold in case of manifolds with boundary. Compared with the cited works, all results are formulated in a more general setting.

The redistribution methods are first presented in a general continuous form, not particularly as methods for adjusting a discrete mesh. After, we apply them to three examples of manifold evolution problems—the mean curvature flow of surfaces in $\mathbb{R}^3$, surface evolution in $\mathbb{R}^3$ with an advection term, and curvature driven evolution of curves on a sphere. We explain the discretization of all problems and demonstrate the performance of the method by various examples. Besides test problems, we show how our methods can be applied to practical problems coming from architecture or biological data analysis.

2. Tangential redistribution of points on an evolving manifold.

When we are dealing with an evolution equation of the form (1.2), it is usually seen as a model corresponding to a specific application. For simplicity, let us assume that there is no tangential movement of points given by the character of our problem and the corresponding model is just

$$\partial_t F = v_N. \tag{2.1}$$

This assumption will simplify the explanation without loss of any of the important aspects of the problem. In this setting, the tangential term will be added to our model purely in order to control the distribution of points on the evolving manifold. It will be designed according to our specific criteria.

2.1. The evolution of the induced metric.

By means of the immersion F^t, we can induce a metric on X as the pull-back of g_Y along F^t, $g_{F^t} = (F^t)^*(g_Y)$. Denote by ξ the measure on the Borel sets of X induced by g_X. At every time t there is also a measure χ^t on X induced by the metric tensor g_{F^t}. The measure of an arbitrary measurable set $U \subseteq X$ is computed as

$$\chi^t(U) = \int_U \mathrm{d}\,\chi^t = \int_U G^t \mathrm{d}\,\xi. \tag{2.2}$$

The quantity G^t is the Radon–Nikodým derivative

$$G^t = \frac{\mathrm{d}\chi^t}{\mathrm{d}\xi}$$

and we will call it the *volume density* of F^t. As we can see from (2.2), it expresses how F locally shrinks or expands volumes.

The evolution of the immersion F^t results in evolution of the induced metric g_{F^t} and the induced measure χ^t. These are the tools to examine the properties of the immersion from the point of view of the distribution of points. In the following section, we will formulate some desirable properties of the immersion in terms of the volume density and the global volume of X. In order to be able to use those conditions for construction of the tangential velocity field, we need to know how the volume density and the global volume evolve with the evolving immersion.

According to the work of Bauer, Harms, and Michor [11], the map G satisfies the following evolution equation:

$$\partial_t G = (-g_Y(h, v_N) + \mathrm{div}_{g_F} w_T)\, G. \tag{2.3}$$

Here, h^t is the mean curvature vector of F^t and $w_T^t(x)$ is the pull-back of $v_T^t(x)$ along F^t, $w_T^t(x) = (F^t)^*(v_T^t(x))$, which means w_T^t is a vector field on X. The operator div_{g_F} represents the divergence on X associated to the induced metric g_F.

Now let A^t represent the volume of X measured by the induced measure χ^t. Then from (2.3) follows

$$\partial_t A = \int_X (-g_Y(h, v_N) + \mathrm{div}_{g_F} w_T)\, \mathrm{d}\chi, \tag{2.4}$$

where χ is defined so that $\chi(\cdot, t) = \chi^t$. Applying the divergence theorem we get

$$\partial_t A = \int_X (-g_Y(h, v_N))\, \mathrm{d}\chi + \int_{\partial X} g_F(w_T, \nu)\, \mathrm{d}H_\chi, \tag{2.5}$$

where H_{χ^t} is the $(m-1)$-dimensional Hausdorff measure on ∂X induced by χ^t and for $x \in \partial X$, $\nu(x)$ is the outward unit normal (with respect to g_F) to ∂X in $T_x X$. If $\partial X = \emptyset$ or $w_T\restriction_{\partial X}$ is a tangential vector field on ∂X, i.e., $g_F(w_T, \nu) = 0$, we get

$$\partial_t A = \int_X (-g_Y(h, v_N))\, \mathrm{d}\chi. \tag{2.6}$$

2.2. The volume-oriented tangential redistribution. The volume-oriented redistribution is used when we want to control the volume density of the evolution of X in Y. In the discrete setting, this corresponds to controlling the volumes of the m-dimensional mesh elements—lengths of lines for a discretized curve, areas of polygons for a discretized surface, etc.

A typical requirement for an evolution is to conserve the relative volume of any set throughout the whole time domain. This means that for any $U \subseteq X$ we require

$$\frac{\chi^t(U)}{A^t} = \frac{\chi^0(U)}{A^0}$$

for all $t \geq 0$. This implies

$$\frac{G^t(x)}{A^t} = \frac{G^0(x)}{A^0}$$

for almost all $x \in X$, $t \geq 0$, or, in other words, for $t \geq 0$ we have

$$\partial_t \left(\frac{G}{A}\right) = 0. \tag{2.7}$$

In order to find a corresponding tangential velocity v_T or its pull-back w_T, we combine (2.7) with (2.3) and (2.5) and we get

$$\mathrm{div}_{g_F} w_T = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \frac{1}{A}\int_{\partial X} g_F(w_T, \nu)\, \mathrm{d}H_\chi, \tag{2.8}$$

where $\langle g_Y(v_N, h)\rangle_\chi$ is the mean of $g_Y(v_N, h)$ over X with respect to χ, i.e.,

$$\langle g_Y(v_N, h)\rangle_\chi = \frac{1}{A}\int_X g_Y(v_N, h)\,\mathrm{d}\chi.$$

If $\partial X = \emptyset$ or $g_F(w_T, \nu) = 0$, we have

$$\operatorname{div}_{g_F} w_T = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi. \tag{2.9}$$

If condition (2.7) is satisfied throughout the evolution, we keep the quality of the initial immersion with respect to the relative volumes. However, we might not be satisfied with the initial state; what we usually want is some sort of uniformity which can be difficult to obtain at the starting point. Thinking of the discrete case, we can easily construct a uniform mesh in a subset of $\mathbb{R}^n$. But discretization of an arbitrarily curved shape in some uniform way needs a certain effort even in the one-dimensional (1D) case and becomes more difficult as the dimension increases. Instead of trying to find a uniform immersion at the beginning, we can start with any (reasonable) immersion and let it approach the uniform immersion during the evolution process.

The immersion F^t is *volume-uniform with respect to* g_X if its volume density is constant on its domain. This is equivalent to the practically more convenient dimensionless condition

$$\frac{G^t(x)}{A^t} = C$$

for almost all $x \in X$, where $C \in \mathbb{R}_+$ is a constant. An asymptotically uniform evolution F satisfies the condition

$$\frac{G^t}{A^t} \underset{t\to\infty}{\longrightarrow} C$$

uniformly with respect to x. This can be achieved, for example, if $\frac{G}{A}$ is a solution of the equation

$$\partial_t\left(\frac{G}{A}\right) = \left(C - \frac{G}{A}\right)\omega, \tag{2.10}$$

where $\omega\colon [0, t_s] \to \mathbb{R}_+$. This condition can be further modified—besides the immersion converging to a uniform immersion, we might want the global volume A to converge to a given value A_∞. In that case we get

$$\frac{\partial_t G}{A_\infty} = \left(C - \frac{G}{A_\infty}\right)\omega. \tag{2.11}$$

Combining (2.10) with (2.3) and (2.5) we get the condition for w_T

$$\operatorname{div}_{\mathrm{g_F}} w_T = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \frac{1}{A}\int_{\partial X} g_F(w_T, \nu)\,\mathrm{d}H_\chi + \left(C\frac{A}{G} - 1\right)\omega \tag{2.12}$$

and, in case $\partial X^t = \emptyset$ or $v_T \cdot \nu = 0$,

$$\operatorname{div}_{\mathrm{g_F}} w_T = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \left(C\frac{A}{G} - 1\right)\omega. \tag{2.13}$$

Moreover, if we want to prescribe the limit of the volume of A, we combine (2.11) with (2.3). This results in

$$\operatorname{div}_{\mathrm{g_F}} w_T = g_Y(v_N, h) + \left(C\frac{A_\infty}{G} - 1 \right) \omega. \tag{2.14}$$

If we were able to compute w_T directly from (2.8)–(2.9) or (2.12)–(2.14), we could obtain $v_T(x)$ by pushing $w_T(x)$ forward along F. However, these conditions do not uniquely determine the tangential field w_T; there are infinitely many possibilities for how to choose it. One possibility is to assume that w_T is a gradient field, which means that

$$w_T^t = \nabla_{g_{F^t}} \psi^t,$$

where ψ^t is a function on X, $\psi^t \colon X \to \mathbb{R}$. The gradient $\nabla_{g_{F^t}} \psi^t$ is a vector field on X defined by

$$g_{F^t}(\nabla_{g_{F^t}} \psi^t, u) = \frac{\partial \psi^t}{\partial u} \quad \forall u \in TX.$$

Using this assumption, we get equations for the Laplace–Beltrami operator of ψ^t. For the conservation of relative volumes

$$\Delta_{g_F} \psi = g_Y(v_N, h) - \langle g_Y(v_N, h) \rangle_\chi + \frac{1}{A} \int_{\partial X} g_F(\nabla_{g_F} \psi, \nu) \, \mathrm{d}H_\chi \tag{2.15}$$

and for the asymptotically uniform redistribution

$$\Delta_{g_F} \psi = g_Y(v_N, h) - \langle g_Y(v_N, h) \rangle_\chi + \frac{1}{A} \int_{\partial X} g_F(\nabla_{g_F} \psi, \nu) \, \mathrm{d}H_\chi + \left(C\frac{A}{G} - 1 \right) \omega. \tag{2.16}$$

Equation (2.14) is reduced to

$$\Delta_{g_F} \psi = g_Y(v_N, h) + \left(C\frac{A_\infty}{G} - 1 \right) \omega. \tag{2.17}$$

If (2.8)–(2.9) or (2.12)–(2.14) are accompanied by an appropriate boundary condition, ψ^t is uniquely determined. In the case of manifolds with boundary, we have to prescribe $\psi^t\restriction_{\partial X}$. In the case of closed manifolds, we have to prescribe the value of ψ^t at one selected point.

2.3. The length-oriented tangential redistribution. In some cases, the volumes or relative volumes of mesh elements are not the main object of interest and we rather want to control distances between selected points on the manifold. In the section devoted to applications, we will mention a problem arising in architecture where the lengths of the 1D mesh elements are a crucial factor.

A natural way to control distances on a manifold is to focus on the evolution of selected curves on the manifold and redistribute the points on these curves as the manifold is evolving. Looking at the problem in this way, we can apply (2.8)–(2.9) or (2.12)–(2.14) and find an appropriate tangential velocity for evolving curves in a higher dimensional manifold Y.

First, let us think of an evolution of a single curve Γ in Y. Since we are dealing with a 1D manifold, the procedure described in section 2.2 can be somewhat simplified. Namely, we can directly obtain the tangential velocity v_T given the divergence of w_T and an appropriate boundary condition.

We can write

$$v_T = \|v_T\|_{g_Y} \frac{v_T}{\|v_T\|_{g_Y}} = \alpha T_Y,$$

where $\|\cdot\|_{g_F}$ is the norm associated with g_F, $\alpha\colon \Gamma \times [0, t_s] \to \mathbb{R}$ and $T_Y = F_*(T_F)$, T_F being a unit vector field on Γ with respect to the metric g_F. The divergence of w_T can then be expressed simply as the directional derivative of the function α

$$\operatorname{div}_{g_F} w_T = \frac{\partial \alpha}{\partial T_F}.$$

Now we can rewrite the conditions for the tangential velocity. If we want to keep relative lengths on the curve, we get

$$\frac{\partial \alpha}{\partial T_F} = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \frac{\alpha(x_E) - \alpha(x_S)}{A}, \tag{2.18}$$

where x_S and x_E are the boundary points of Γ (if any) and A is the length of Γ with respect to g_F. The mean curvature vector h is simply the curvature vector of F. If $\alpha(x_S) = \alpha(x_E)$, then

$$\frac{\partial \alpha}{\partial T_F} = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi. \tag{2.19}$$

For the asymptotically uniform redistribution, (2.12) reduces to

$$\frac{\partial \alpha}{\partial T_F} = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \frac{\alpha(x_E) - \alpha(x_S)}{A} + \left(C\frac{A}{G} - 1\right)\omega \tag{2.20}$$

and (2.13) reduces to

$$\frac{\partial \alpha}{\partial T_F} = g_Y(v_N, h) - \langle g_Y(v_N, h)\rangle_\chi + \left(C\frac{A}{G} - 1\right)\omega. \tag{2.21}$$

Finally, for the asymptotically uniform redistribution with a prescribed limit length A_∞ we have

$$\frac{\partial \alpha}{\partial T_F} = g_Y(v_N, h) + \left(C\frac{A_\infty}{G} - 1\right)\omega. \tag{2.22}$$

As we can see, to obtain a unique solution to (2.18)–(2.22), it is enough to set the value of α in x_S and, if necessary, x_E. If there is no boundary, we pick any point on Γ and set the value of α there.

A curve on an evolving higher dimensional manifold is obtained as the push-forward of a 1D submanifold Γ of a manifold X along the immersion F. The normal velocity of such a curve is obtained by restriction of the normal velocity of X to Γ. Then we can use (2.18)–(2.22) in order to determine the tangential velocity in the points of the curve.

Now we can take our ideas further and consider two intersecting curves Γ_1, Γ_2 on X. The intersection point will be denoted by x_0. We set the tangential velocity of the intersection point to

$$v_T(x_0, t) = \frac{v_{T,\Gamma_1}(x_0) + v_{T,\Gamma_2}(x_0)}{2}, \tag{2.23}$$

where v_{T,Γ_1} and v_{T,Γ_2} are the tangential velocities computed individually for Γ_1 and Γ_2. While this choice of v_T is no more suitable for the relative length preservation, it still can be used if our goal is the asymptotically uniform immersion of the two curves. In fact, if Γ_1 and Γ_2 are linearly independent in x_0, then v_T in this point will never be zero if the two corresponding immersions are not uniform. We are also free to prescribe a limit length for Γ_1 and Γ_2 as in (2.22). Since the linear independence of the curves is a sufficient condition for the reliability of (2.23), we can have up to m curves intersecting at x_0 and still obtain an asymptotically uniform immersion for all of them. Analogously to (2.23), v_T at x_0 would be taken as the arithmetic mean of all v_{T,Γ_i}.

This reasoning extends straightforwardly to a network of curves. It is enough that there are no more than m intersecting curves in one node and that all curves passing through one node are linearly independent. If this condition is satisfied, we can apply the asymptotically uniform redistribution to all curves in the network. In the discrete case, we can obtain (at $t \to \infty$) uniformly distributed discretization points on each curve. Having the possibility of setting the limit length of every curve (and thus the limit length of each linear segment of the curve), we can obtain equal distances between discretization points everywhere in the network. The resulting mesh will thus depend on the curves that we select for redistribution (only the lengths of the edges belonging to one of the selected curves can be directly controlled) and on the method of redistribution that we choose.

3. Examples of manifold evolution problems.

At this point we apply the proposed tangential redistribution techniques to several particular problems. The methods already have been demonstrated to work in the case of curves evolving in $\mathbb{R}^2$ [30, 31] and $\mathbb{R}^3$ [33, 34] and curves evolving on smooth function graphs [32]. Here we discuss the mean curvature flow of surfaces in $\mathbb{R}^3$, the problem of 3D image segmentation where the normal velocity of the evolving surface depends on the image data and the evolution of a curve on a sphere driven by its curvature.

3.1. Mean curvature flow of surfaces.

In this example, X is a 2D manifold and Y is $\mathbb{R}^3$ with the standard Euclidean metric. We also suppose that F^t is a smooth embedding of X in $\mathbb{R}^3$. We tested our methods on two mean curvature flow models. The first model is the standard mean curvature flow of a surface in $\mathbb{R}^3$, which means F satisfies the evolution equation

$$\partial_t F = h, \tag{3.1}$$

where $h = HN$ is the mean curvature vector of F, H is the mean curvature of F (defined as the trace of the second fundamental form of F), and N is the outward unit normal to $\mathrm{Im}(F)$. If we are dealing with an evolving closed surface, this model leads to shrinking of the volume enclosed by the surface. For an evolving surface with boundary, (3.1) is coupled with a Dirichlet boundary condition and $\mathrm{Im}(F^t)$ will converge to the minimal surface for the given boundary curve $\partial\mathrm{Im}(F^0)$. The initial condition F^0 is prescribed so that it is compatible with the boundary condition.

The model (3.1) can be alternatively rewritten as

$$\partial_t F = \Delta_{g_F} F, \tag{3.2}$$

where Δ_{g_F} stands for the Laplace–Beltrami operator with respect to the metric g_F.

The second model is the volume preserving mean curvature flow of closed surfaces. In this case we have

$$\partial_t F = h - \bar{h}, \tag{3.3}$$

where $\bar{h} = \bar{H}N$ and $\bar{H}$ is the average of the mean curvature over X, which means

$$\bar{H} = \frac{1}{\chi(X)} \int_X H \, \mathrm{d}\chi.$$

The alternative formulation in terms of the Laplace–Beltrami operator reads

$$\partial_t F = \Delta_{g_F} F - \bar{h}. \tag{3.4}$$

Equations (3.2) and (3.4) are the ones that we are going to discretize. In order to be able to control the quality of the discretization during the evolution, we add the tangential term to both models. The complete models then read

$$\partial_t F = \Delta_{g_F} F + v_T, \tag{3.5}$$

$$\partial_t F = \Delta_{g_F} F - \bar{h} + v_T. \tag{3.6}$$

In the former case, we might have a surface with boundary. In some cases, the tangential movement of points along the boundary might be desirable. Thus, in general we set

$$\partial_t F = v_T, \quad x \in \partial X, \tag{3.7}$$

with the condition that v_T lies in the tangential space of $\partial(\mathrm{Im}(F))$.

3.1.1. The time discretization. In order to obtain the time discretization of (3.5) and (3.6), we apply the semi-implicit approach. This approach was chosen because it demonstrated better stability properties compared to the explicit discretization. If τ is the time step, $t^n = n\tau$, and $F^n = F(\cdot, t^n)$, we obtain

$$\frac{F^n - F^{n-1}}{\tau} = \Delta_{F^{n-1}} F^n + v_T^{n-1} \tag{3.8}$$

for the mean curvature flow model and

$$\frac{F^n - F^{n-1}}{\tau} = \Delta_{F^{n-1}} F^n - \bar{h}^{n-1} + v_T^{n-1} \tag{3.9}$$

for the volume preserving mean curvature flow. The symbol $\Delta_{F^{n-1}}$ denotes the Laplace–Beltrami operator with respect to the metric $g_{F^{n-1}}$ induced by F^{n-1}.

3.1.2. The space discretization. The space discretization of (3.8) and (3.9) is done by a finite volume technique. The finite volume scheme for evolving surfaces is based on polygonal representation of the surfaces. To this end, we consider a triangulation of X—a simplicial complex homeomorphic to X. The corresponding homeomorphism induces a triangular structure on X consisting of vertices X_i,

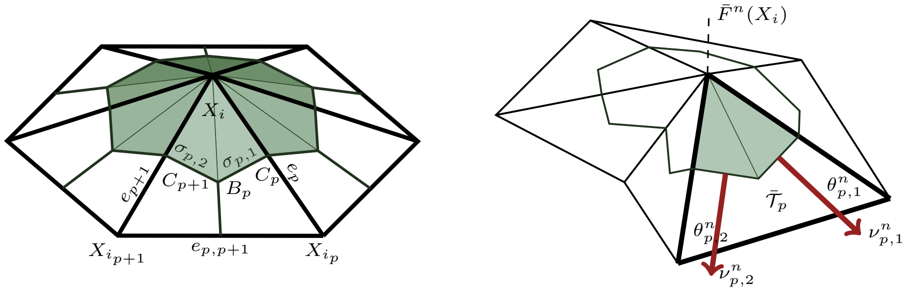

Fig. 1. *The discretization mesh. Left, the triangulation of the abstract manifold X. Right, the corresponding approximation of the embedded manifold $F^n(X)$.*

$i = 1 \dots n_V$, edges e_j, $j = 1 \dots n_E$, and triangles $\mathcal{T}_p$, $p = 1 \dots n_T$; these elements are obtained as the images of the 0,1 and 2-simplices, respectively.

The next step is the construction of a co-volume mesh on X based on the barycentric subdivision of X (Figure 1). We will describe the procedure for a given inner node X_i. For clarity, we will use local indexing of the mesh elements since it is sufficient to explain the idea. Let us suppose that the node X_i is the common vertex of m mesh triangles $\mathcal{T}_1, \dots, \mathcal{T}_m$. Then it is also the common vertex of m edges $e_1, \dots, e_m$, where e_p connects X_i with X_{i_p}. The triangle $\mathcal{T}_p$ admits a barycentric coordinate system—each point of the triangle can be expressed as $P = \lambda_1 X_i + \lambda_2 X_{i_p} + \lambda_3 X_{i_{p+1}}$, where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Let B_p be the barycenter of $\mathcal{T}_p$ and C_p the center of e_p, $p = 1 \dots m$, and let the barycentric subdivision of $\mathcal{T}_p$ be constructed using these points. The co-volume V_i corresponding to X_i is then constructed as the star of X_i—the union of the triangles $\mathcal{V}_{p,1} = X_i C_p B_p$ and $\mathcal{V}_{p,2} = X_i B_p C_{p+1}$ for $p = 1 \dots m$, where we set $C_{m+1} = C_1$. It is a polygon with $2m$ boundary edges $\sigma_{p,1} = C_p B_p$, $\sigma_{p,2} = B_p C_{p+1}$.

The manifold X can be embedded in $\mathbb{R}^3$ in a way that respects its triangular structure. We define the embedding $\bar{F}^n$ as follows. In the vertices X_i, we set $\bar{F}^n(X_i) = F^n(X_i)$. Then, for any triangle $\mathcal{T}_p$ with vertices X_i, X_{i_p}, $X_{i_{p+1}}$, we set

$$\bar{F}^n(\lambda_1 X_i + \lambda_2 X_{i_p} + \lambda_3 X_{i_{p+1}}) = \lambda_1 F^n(X_i) + \lambda_2 F^n(X_{i_p}) + \lambda_3 F^n(X_{i_{p+1}}).$$

This means that $\bar{F}^n(X)$ is a polyhedron with vertices $\bar{F}^n(X_i) = F^n(X_i) = F^n_i$, edges $\bar{F}^n(e_j)$, and triangular faces $\bar{F}^n(\mathcal{T}_p)$. The points F^n_i will be the unknowns of the fully discretized problem.

Since now we have an additional set of embeddings $\bar{F}^n$, the manifold X is equipped with another set of induced metrics g^n—the pull-backs of the Euclidean metric along $\bar{F}^n$. The measure induced by g^n will be referred to as χ^n.

Our approximation will also use some elements of $\bar{F}^n(X)$. First, we will need the outward unit normal ν^n_i to the boundary of $\bar{F}^n(V_i)$ tangent to $\bar{F}^n(X)$ (defined everywhere except the corners of $\bar{F}^n(V_i)$). Since ν^n_i is piecewise constant, we will also use the notation $\nu^n_{p,1}$, $\nu^n_{p,2}$ for the outward unit normals to $\bar{F}^n(\sigma_{p,1})$ and $\bar{F}^n(\sigma_{p,2})$ in the plane of $\bar{\mathcal{T}}_p$. Further, $\theta^n_{p,1}$ and $\theta^n_{p,2}$ will denote the angles of $\mathcal{T}_p$ adjacent to X_{i_p} and $X_{i_{p+1}}$, respectively, measured in the metric g^n.

Integrating (3.8) over V_i, we get

$$\int_{V_i} \frac{F^n - F^{n-1}}{\tau} \, \mathrm{d}\chi_{F^{n-1}} = \int_{V_i} \Delta_{F^{n-1}} F^n \, \mathrm{d}\chi_{F^{n-1}} + \int_{V_i} v_T^{n-1} \, \mathrm{d}\chi_{F^{n-1}}. \tag{3.10}$$

In the same way, for (3.9)

$$\int_{V_i} \frac{F^n - F^{n-1}}{\tau}\,\mathrm{d}\chi_{F^{n-1}} = \int_{V_i} \Delta_{F^{n-1}} F^n\,\mathrm{d}\chi_{F^{n-1}} + \int_{V_i} 2\bar{h}^{n-1}\,\mathrm{d}\chi_{F^{n-1}} + \int_{V_i} v_T^{n-1}\,\mathrm{d}\chi_{F^{n-1}}. \tag{3.11}$$

Here, $\chi_{F^{n-1}}$ stands for the measure induced by the metric $g_{F^{n-1}}$.

Discretization of the Laplace–Beltrami operator. The Laplace–Beltrami operator is discretized by the cotangent scheme presented in Meyer et al. [27]. For any control volume V_i, the Laplace–Beltrami operator term in (3.10) and (3.11) can be rewritten as

$$\begin{aligned} \int_{V_i} \Delta_{F^{n-1}} F^n\,\mathrm{d}\chi_{F^{n-1}} &= \int_{\partial V_i} \nabla_{F^{n-1}} F^n \cdot \nu_i^{n-1}\,\mathrm{d}H_{\chi_{F^{n-1}}} \\ &= \sum_{p=1}^{m} \sum_{q=1,2} \int_{\sigma_{p,q}} \nabla_{F^{n-1}} F^n \cdot \nu_{p,q}^{n-1}\,\mathrm{d}H_{\chi_{F^{n-1}}}, \end{aligned} \tag{3.12}$$

where $\nabla_{F^{n-1}}$ denotes the gradient with respect to the metric $g_{F^{n-1}}$. Now, since $F^n \approx \bar{F}^n$, we have

$$(\nabla_{F^{n-1}} F^n)\restriction_{\mathcal{T}_p} \approx (\nabla_{n-1} \bar{F}^n)\restriction_{\mathcal{T}_p},$$

where ∇_{n-1} is the gradient with respect to g^{n-1}. From the definition of $\bar{F}^n$ follows that its gradient is constant on $\mathcal{T}_p$. This property leads to the equality [27]

$$\sum_{q=1,2} \int_{\sigma_{p,q}} \nabla_{n-1} \bar{F}^n \cdot \nu_{p,q}^{n-1}\,\mathrm{d}H_{\chi^{n-1}} = \frac{1}{2}(\cot\theta_{p,2}^{n-1}(F_{i_p}^n - F_i^n) + \cot\theta_{p,1}^{n-1}(F_{i_{p+1}}^n - F_i^n)) \tag{3.13}$$

for any $p = 1 \dots m$ and further to the approximation (using the full indexing)

$$\int_{V_i} \Delta_{F^{n-1}} F^n\,\mathrm{d}\chi^{n-1} \approx \frac{1}{2} \sum_{p=1}^{m} \left(\cot\theta_{i,p-1,1}^{n-1} + \cot\theta_{i,p,2}^{n-1}\right)(F_i^n - F_{i_p}^n), \tag{3.14}$$

where $\theta_{i,0,1}^{n-1} = \theta_{i,m,1}^{n-1}$.

Discretization of the tangential velocity term. Based on the two possibilities of tangential redistribution presented in section 2, we have to suggest two ways to approach the discretization of the tangential velocity term

$$\int_{V_i} v_T^{n-1}\,\mathrm{d}\chi_{F^{n-1}}. \tag{3.15}$$

In case of the volume-oriented tangential redistribution, v_T is not computed directly but it is assumed to be the gradient of a function ψ that can be found according to (2.15)–(2.17). If we want to use the length-oriented redistribution, we have at our disposal a direct way to obtain v_T (2.18)–(2.22). This determines how we approximate the integral of v_T^{n-1} in (3.10) and (3.11). In addition, we have to suggest an appropriate discretization of (2.15)–(2.22).

Equations (2.15)–(2.17) arising in the volume-oriented redistribution contain the Laplace–Beltrami operator of the unknown function ψ. Just as before, we can integrate them over the volume V_i and then take advantage of the discretization presented in the previous subsection. This leads to

$$\int_{V_i} \Delta_{F^{n-1}} \psi^{n-1} \,\mathrm{d}\chi^{n-1} \approx \frac{1}{2}\sum_{p=1}^{m} \left(\cot\theta^{n-1}_{i,p-1,1} + \cot\theta^{n-1}_{i,p,2}\right)\left(\psi_i^{n-1} - \psi_{i_p}^{n-1}\right), \tag{3.16}$$

where $\psi^{n-1} = \psi(\cdot, t^{n-1})$ and $\psi_i^{n-1} = \psi(X_i, t^{n-1})$. Further, we need to approximate the terms on the right-hand side of the integrated versions of (2.15)–(2.17). We take

$$\int_{V_i} g_Y(v_N^{n-1}, h^{n-1}) \,\mathrm{d}\chi^{n-1} \approx \chi^{n-1}(V_i) g_Y(v_{N,i}^{n-1}, h_i^{n-1})$$

and the remaining integrals can be approximated analogously. Now, $g_Y(v_{N,i}^{n-1}, h_i^{n-1})$ is simply

$$g_Y(v_{N,i}^{n-1}, h_i^{n-1}) = v_{N,i}^{n-1} \cdot h_i^{n-1} = h_i^{n-1} \cdot h_i^{n-1} = (H_i^{n-1})^2$$

for the standard mean curvature flow and

$$g_Y(v_{N,i}^{n-1}, h_i^{n-1}) = (h_i^{n-1} - \bar{h}^{n-1}) \cdot h_i^{n-1} = (H_i^{n-1})^2 - \bar{h}^{n-1} \cdot h_i^{n-1}$$

for the volume preserving mean curvature flow. Since

$$h = \Delta_{g_F} F$$

we can, once more, use the discretization of the Laplace–Beltrami operator (3.14). That means

$$h_i^{n-1} = \frac{1}{2\chi^{n-1}(V_i)} \sum_{p=1}^{m} \left(\cot\theta^{n-1}_{i,p-1,1} + \cot\theta^{n-1}_{i,p,2}\right)\left(F_i^{n-1} - F_{i_p}^{n-1}\right). \tag{3.17}$$

The approximation of the mean curvature H follows directly:

$$H_i^{n-1} = \|h_i^{n-1}\|, \tag{3.18}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^3$. The average of H is approximated as

$$\bar{H}^{n-1} = \frac{1}{A^{n-1}} \sum_{i=1}^{n_V} H_i^{n-1} \chi^{n-1}(V_i), \tag{3.19}$$

where

$$A^{n-1} = \sum_{i=1}^{v_N} \chi^{n-1}(V_i). \tag{3.20}$$

Similarly

$$\langle g_Y(v_N^{n-1}, h^{n-1}) \rangle_{\chi^{n-1}} \approx \frac{1}{A^{n-1}} \sum_{i=1}^{n_V} (H_i^{n-1})^2 \chi^{n-1}(V_i) \tag{3.21}$$

for the mean curvature flow model and

$$\langle g_Y(v_N^{n-1}, h^{n-1}) \rangle_{\chi^{n-1}} \approx \frac{1}{A^{n-1}} \sum_{i=1}^{n_V} \left((H_i^{n-1})^2 - \bar{h}^{n-1} \cdot h_i^{n-1}\right) \chi^{n-1}(V_i) \tag{3.22}$$

for the volume preserving mean curvature flow. Finally, if we want to use the asymptotically uniform redistribution, we have to approximate the term

$$\left(C\frac{A}{G}-1\right)\omega.$$

The question is how to approximate the volume density G. We have

$$A(t^{n-1}) = \int_X G(x,t^{n-1})\,\mathrm{d}\xi \approx \sum_{i=1}^{n_V} G_i^{n-1}\xi(V_i).$$

On the other hand,

$$A(t^{n-1}) \approx \sum_{i=1}^{n_V} \chi^{n-1}(V_i).$$

Since we do not have any particular conditions imposed on the measure ξ, we can assume that $\xi(X) = 1/C$ and $\xi(V_i) = \xi(X)/n_V$ for all $i = 1 \dots n_V$. Then we can set

$$G_i^{n-1} = \chi^{n-1}(V_i)\frac{n_V}{\xi(X)} = Cn_V\chi^{n-1}(V_i). \tag{3.23}$$

Therefore,

$$\left(C\frac{A}{G}-1\right)\omega \approx \left(\frac{A^{n-1}}{n_V\chi^{n-1}(V_i)}-1\right)\omega^{n-1}. \tag{3.24}$$

Equations (3.16)–(3.24) provide all necessary ingredients for the full discretization of (2.15)–(2.17). After an appropriate combination, depending on the type of redistribution, we end up with a linear system with the unknowns ψ_i^n. In case of closed surfaces, we have to prescribe ψ_i^n for one selected index i in order to obtain a system with a unique solution. If we are dealing with a surface with boundary, the value ψ_i^n has to be set for all its boundary vertices.

Now we are ready for discretizing the integral of the tangential velocity v_T^{n-1} (3.15). Recall that we supposed

$$(F^{n-1})^*\left(v_T^{n-1}(x)\right) = w_T^{n-1}(x) = (\nabla_{F^{n-1}}\psi^{n-1})(x)$$

for any $x \in X$. This implies

$$v_T^{n-1}(x) = (F^{n-1})_*\left(w_T^{n-1}(x)\right) = (F^{n-1})_*\left((\nabla_{F^{n-1}}\psi^{n-1})(x)\right).$$

This means that $\mathrm{Im}(v_T^{n-1}) = \nabla_{\mathrm{Im}(F^{n-1})}\left((F^{n-1})_*\psi^{n-1}\right)$, where $\nabla_{\mathrm{Im}(F^{n-1})}$ is defined so that the diagram

$$\begin{tikzcd}
\mathcal{F}\left(\mathrm{Im}(F^{n-1})\right) \arrow[r, dotted, "\nabla_{\mathrm{Im}(F^{n-1})}"] & \mathcal{V}\left(\mathrm{Im}(F^{n-1})\right) \\
\mathcal{F}(X) \arrow[u, "(F^{n-1})_*"] \arrow[r, "\nabla_{F^{n-1}}"'] & \mathcal{V}(X) \arrow[u, "(F^{n-1})_*"']
\end{tikzcd}$$

commutes ($\mathcal{F}$ denoting the space of functions and $\mathcal{V}$ the space of vector fields). Obviously, if $\mathrm{Im}(F^{n-1})$ is considered to be a submanifold of $\mathbb{R}^3$ with the metric $g_{\mathrm{Im}(F^{n-1})}$ pulled back from $\mathbb{R}^3$, then $\nabla_{\mathrm{Im}(F^{n-1})}$ represents the gradient on $\mathrm{Im}(F^{n-1})$ with respect to $g_{\mathrm{Im}(F^{n-1})}$. It is also usual [17] to define $\nabla_{\mathrm{Im}(F^{n-1})}\varphi$ for a function $\varphi\colon \mathrm{Im}(F^{n-1}) \to \mathbb{R}$ as

$$\nabla_{\mathrm{Im}(F^{n-1})}\varphi = \nabla\bar{\varphi} - (\nabla\bar{\varphi}\cdot N)N,$$

where $\bar{\varphi}$ is any smooth extension of φ to the neighborhood of $\mathrm{Im}(F^{n-1})$ in $\mathbb{R}^3$. In this case, $\nabla_{\mathrm{Im}(F^{n-1})}$ is usually referred to as the tangential gradient or the surface gradient.

Having v_T^{n-1} of this specific form, the following identity holds [17]:

$$\int_{V_i} v_T^{n-1}\,\mathrm{d}\chi_{F^{n-1}} = \int_{\partial V_i} \psi^{n-1}\nu_i^{n-1}\,\mathrm{d}H_{\chi_{F^{n-1}}} - \int_{V_i} \psi^{n-1}h^{n-1}\,\mathrm{d}\chi_{F}{}^{n-1}. \tag{3.25}$$

The approximation follows straightforwardly,

$$\begin{aligned}\int_{V_i} v_T^{n-1}\,\mathrm{d}\chi_{F^{n-1}} \approx\ & \sum_{p=1}^{m}\left(\|\sigma_{i,p,1}\|_{n-1}\psi_{i,p,1}^{n-1}\nu_{i,p,1}^{n-1} + \|\sigma_{i,p,2}\|_{n-1}\psi_{i,p,2}^{n-1}\nu_{i,p,2}^{n-1}\right)\\ & -\chi^{n-1}(V_i)\psi_i^{n-1}h_i^{n-1},\end{aligned} \tag{3.26}$$

where $\psi_{i,p,1}^{n-1}$, $\psi_{i,p,2}^{n-1}$ are the values of ψ^{n-1} in the midpoints of $\sigma_{i,p,1}$ and $\sigma_{i,p,2}$. They are obtained from the values of ψ^{n-1} in the vertices X_i by linear interpolation, which means

$$\begin{aligned}\psi_{i,p,1}^{n-1} &= \frac{5\psi_i^{n-1} + 5\psi_{i_p}^{n-1} + 2\psi_{i_{p+1}}^{n-1}}{12},\\ \psi_{i,p,2}^{n-1} &= \frac{5\psi_i^{n-1} + 2\psi_{i_p}^{n-1} + 5\psi_{i_{p+1}}^{n-1}}{12}.\end{aligned}$$

The other option for the tangential redistribution—the length-oriented redistribution—leads to a much simpler approximation of the tangential velocity term. Since in this case we obtain v_T explicitly, we simply use

$$\int_{V_i} v_T^{n-1}\,\mathrm{d}\chi_{F^{n-1}} \approx \chi^{n-1}(V_i)v_{T,i}^{n-1}, \tag{3.27}$$

where $v_{T,i}^{n-1}$ is the tangential velocity in X_i.

The principle of the length-oriented redistribution is to consider the point X_i situated on a curve Γ, which means, in the discretized setting, $X_i = \Gamma_j$. Since Γ is one-dimensional, we can assume that the points Γ_j, $j = 0\dots n_p$, are ordered (indexed) according to their distance from a boundary point, if any, or from an arbitrarily chosen point Γ_0 in case there is no boundary. The velocity $v_{T,i}^{n-1}$ is obtained as

$$v_{T,i}^{n-1} = \alpha_j^{n-1}T_{Y,j}^{n-1},$$

where $\alpha_j^{n-1} = \alpha^{n-1}(\Gamma_j)$ and $T_{Y,j}^{n-1} = F_*^{n-1}(T_{\Gamma,j}^{n-1})$, $T_{\Gamma,j}^{n-1}$ being the unit tangent vector to Γ in Γ_j with respect to $g_{F^{n-1}}$. The speed α_j^{n-1} is obtained from (2.18)–(2.22), where the directional derivative $\frac{\partial\alpha^{n-1}}{\partial T_{\Gamma,j}^{n-1}}$ is approximated as

$$\frac{\partial\alpha^{n-1}}{\partial T_{\Gamma,j}^{n-1}} \approx \frac{\alpha_j^{n-1} - \alpha_{j-1}^{n-1}}{d_{n-1}(\Gamma_{j-1},\Gamma_j)}. \tag{3.28}$$

Here, $d_{n-1}(\Gamma_{j-1}, \Gamma_j)$ is the distance between Γ_{j-1} and Γ_j with respect to the metric g_{n-1}. The value α_0^{n-1} is set according to the specific requirements of the problem that we are solving.

Further, we approximate the vector $T_{Y,j}^{n-1}$,

$$\begin{aligned} T_{Y,0}^{n-1} &\approx \frac{F^{n-1}(\Gamma_1) - F^{n-1}(\Gamma_0)}{d_{n-1}(\Gamma_0, \Gamma_1)}, \\ T_{Y,j}^{n-1} &\approx \frac{\frac{F^{n-1}(\Gamma_{j+1}) - F^{n-1}(\Gamma_j)}{d_{n-1}(\Gamma_{j+1}, \Gamma_j)} + \frac{F^{n-1}(\Gamma_j) - F^{n-1}(\Gamma_{j-1})}{d_{n-1}(\Gamma_{j-1}, \Gamma_j)}}{2}, \quad j = 1 \dots n_p - 1, \\ T_{Y,n_p}^{n-1} &\approx \frac{F^{n-1}(\Gamma_{n_p}) - F^{n-1}(\Gamma_{n_p-1})}{d_{n-1}(\Gamma_{n_p-1}, \Gamma_{n_p})}. \end{aligned} \tag{3.29}$$

In order to discretize the curvature vector h, we use the difference

$$h(\Gamma_j) \approx \frac{T_{Y,j+1}^{n-1} - T_{Y,j}^{n-1}}{\frac{d_{n-1}(\Gamma_{j-1}, \Gamma_j) + d_{n-1}(\Gamma_j, \Gamma_{j+1})}{2}} \tag{3.30}$$

for $j = 1 \dots n_p - 1$. In the first and the last point, we can set

$$\begin{aligned} h(\Gamma_0) &\approx \frac{T_{Y,1}^{n-1} - T_{Y,0}^{n-1}}{d_{n-1}(\Gamma_1, \Gamma_0)}, \\ h(\Gamma_{n_p}) &\approx \frac{T_{n_p} - T_{n_p-1}}{d_{n-1}(\Gamma_{n_p}, \Gamma_{n_{p-1}})} \end{aligned}$$

if we need the approximation of the curvature there.

If the point X_i is the intersection point of two curves, then $v_{T,i}^{n-1}$ is taken as the arithmetic mean of the two corresponding tangential velocities according to (2.23).

3.1.3. The fully discrete formulation. At this point, only a few details are missing for the full discretization of the mean curvature flow models. First, we take

$$\int_{V_i} \frac{F^n - F^{n-1}}{\tau} \, \mathrm{d}\chi_{F^{n-1}} \approx \chi^{n-1}(V_i) \frac{F_i^n - F_i^{n-1}}{\tau}. \tag{3.31}$$

If we are dealing with the volume preserving version of the mean curvature flow, we also have to include the approximation

$$\int_{V_i} 2\bar{h}^{n-1} \, \mathrm{d}\chi_{F^{n-1}} \approx 2\chi^{n-1}(V_i)\bar{h}^{n-1}, \tag{3.32}$$

where $\bar{h}_i^{n-1}$ is obtained from (3.19). Finally, combining (3.31), (3.14), (3.26) or (3.27), and, if appropriate, (3.32), we obtain a linear system with unknowns F_i^n. The solution of this system is the numerical solution of our problem.

Remark 3.1. In our implementation, we multiplied each linear equation corresponding to the fully discretized evolution problem by the factor $\tau/\chi^{n-1}(V_i)$. Therefore, if s_i is the sum of all off-diagonal coefficients in the ith row of the system matrix, the diagonal coefficient is equal to $s_i + 1$. For an appropriately chosen time step τ

(depending on the properties of the triangulation), the system will be diagonally dominant. We solved this system using the BiCGStab method [41] and usually only a few iterations were needed to obtain a sufficiently precise solution in all our experiments. We also tested the SOR method that, too, never failed to converge though the number of iterations was much higher even if the method was optimally tuned. When using the volume-oriented redistribution, we have to solve another system corresponding to the discretization of (2.15)–(2.17). Here we do not have the possibility to guarantee the diagonal dominance. However, both BiCGStab and SOR methods converged in all experiments that we performed though the number of necessary iterations was usually one order of magnitude higher than in the case of the system for F_i^n. The number of iterations depends on the discretization of the evolving manifold. A more precise investigation of the properties of the two matrices and of the convergence of particular numerical methods that can be applied could be a subject of further research.

Remark 3.2. Within the description of the discretization mesh, we did not mention the boundary nodes X_i. The difference here is that the node with m neighboring mesh triangles has $m+1$ neighboring edges $e_1, \dots, e_{m+1}$ and the point X_i is one of the vertices of the co-volume V_i—see Figure 2. However, the discretization of the Laplace–Beltrami operator is not necessary in the boundary points; what we need is just the area of the boundary co-volume.

3.1.4. Experiments and applications. In this section, we demonstrate the properties and performance of the numerical scheme and the tangential redistribution techniques. We present several test examples as well as some practical applications.

The experimental order of convergence of the finite volume scheme. If $\mathrm{Im}(F^0)$ is a standard Euclidean sphere, the exact solution to (3.1) is known. If the radius of the sphere is r_0, then $\mathrm{Im}(F^t)$ is a sphere with radius

$$r(t) = \sqrt{r_0^2 - 4t}.$$

This allows us to examine the experimental order of convergence (EOC) of the numerical method described in the previous section.

In our case, we set $r_0 = 1.0$ and we stopped the computation when $t_s = 0.06$. The discretization of the sphere was based on an octahedron with isosceles triangular faces and the mesh refinement was done by dividing each triangle into four equal triangles. The coarsest grid used for the evaluation of EOC contained 66 vertices and 128 triangles (see Figure 3). We used the coupling $\tau \sim l^2$, where l characterizes the size of the triangle sides. The EOC was computed as

$$\log_2\left(\frac{\delta_l}{\delta_{l/2}}\right),$$

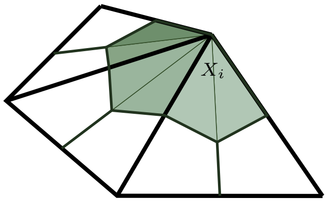

FIG. 2. *The co-volume corresponding to a boundary node X_i.*

where δ_l is the L_2 error,

$$\delta_l = \sum_{j=1}^{t_s/\tau} \left(\sum_{i=1}^{n_V} \left| \|F_i^j\| - r(t^j) \right| \chi^{n-1}(V_i) \right) \tau.$$

The error $\delta_{l/2}$ is computed in the same way after applying one step of space discretization refinement and changing τ accordingly.

We tested the method without any tangential redistribution and then with the asymptotically uniform volume-oriented tangential redistribution with $\omega = 1.0$, $\omega = 10.0$, $\omega = 100.0$. The results are presented in Tables 1–4. The first two tables also show the CPU time (in seconds) needed for the computation on a single 2.4-GHz processor. The most time-consuming part of the procedure is updating the mesh-related elements (control volumes, normals, etc.) in each time step. Computing the tangential component of the velocity takes relatively a lot of time with respect to solving the global linear system due to the higher number of iterations needed for the BiCGStab method to converge.

Looking at the tables, we observe the second order accuracy in all four cases. This order of convergence is due to the space discretization that is second order accurate and due to the coupling $\tau \sim l^2$ that is natural for parabolic problems. Investigating the convergence in time alone, we would observe the first order accuracy. Comparing the results, we can see that the L_2 error is higher when we use the tangential redistribution and it is increasing with increasing ω. This is because the approximation of the tangential movement is not purely tangential anymore; this leads to a certain deviation of the mesh nodes from the surface where they should be situated. However, the method still converges without any loss of the convergence rate. Moreover, the presented test example is an evolution of a sphere which is a surface of constant mean curvature. The experiments that follow will demonstrate that the tangential movement might be crucial in evolution of surfaces with a significant variation of the mean curvature.

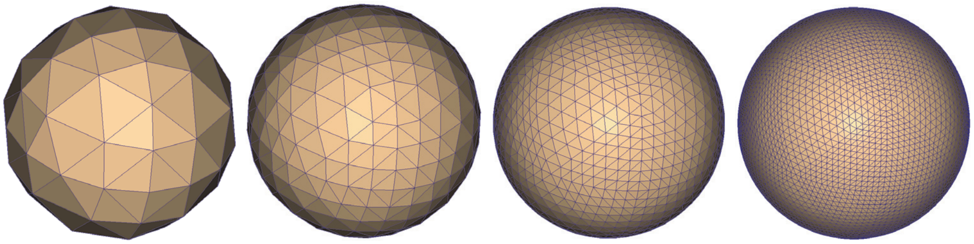

FIG. 3. *The discretization of the sphere, from left to right:* $n_V = 66$, 258, 1026, 4098.

TABLE 1
The EOC for the case with no tangential redistribution.

n_V	τ	L_2 error	CPU	EOC
66	0.01	3.49211e-3	4.03e-3	
258	0.0025	8.93590e-4	6.94e-2	1.96642
1026	0.000625	2.21598e-4	2.04	2.01167
4098	0.00015625	5.53971e-5	109.94	2.00006

TABLE 2
The EOC for the case with the asymptotically uniform tangential redistribution, $\omega = 1.0$.

n_V	τ	L_2 error	CPU	EOC
66	0.01	3.72866e-3	5.27e-3	
258	0.0025	9.27154e-4	9.78e-2	2.00778
1026	0.000625	2.28472e-4	2.83	2.02079
4098	0.00015625	5.70027e-5	198.89	2.00291

TABLE 3
The EOC for the case with the asymptotically uniform tangential redistribution, $\omega = 10.0$.

n_V	τ	L_2 error	EOC
66	0.01	4.83682e-3	
258	0.0025	1.19628e-3	2.01549
1026	0.000625	2.96323e-4	2.01332
4098	0.00015625	7.38617e-5	2.00427

TABLE 4
The EOC for the case with the asymptotically uniform tangential redistribution, $\omega = 100.0$.

n_V	τ	L_2 error	EOC
66	0.01	1.48276e-2	
258	0.0025	2.54933e-3	2.54010
1026	0.000625	5.53847e-4	2.20256
4098	0.00015625	1.33084e-4	2.05715

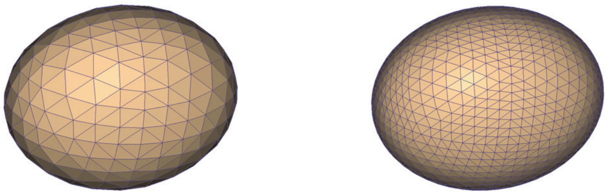

FIG. 4. *The discretization of the ellipsoid with semiprincipal axes of lengths $a = 1.2$, $b = 1.0$, $c = 0.8$. Left, $n_V = 258$; right, $n_V = 1026$.*

The role of tangential redistribution in the computation process. Here, we present several experiments demonstrating the effects and importance of the tangential redistribution in the process of evolution of discrete surfaces.

In the first experiment, we applied the mean curvature flow model and the evolving surface was an approximation of the ellipsoid with the semiprincipal axes of lengths $a = 1.2$, $b = 1.0$, $c = 0.8$. The grid was constructed in the same way as described in section 3.1.2 (Figure 4). Tables 5, 6, and 7 illustrate the effect of the volume-oriented tangential redistribution measured in terms of the ratio $r_A = \frac{A_{min}}{A_{max}}$, where A_{min} is the minimum and A_{max} the maximum co-volume area in the mesh. The first table displays the evolution of r_A in the case with no tangential redistribution. The second table shows how r_A evolves when we apply the redistribution preserving relative areas. The third table contains the results for the asymptotically uniform redistribution. The number of node points, the time step, and the redistribution speed ω are listed in each column of the tables. As we can see, in the case with no redistribution

Table 5

Evolution of r_A in the case with no tangential redistribution.

t	n_V=258	n_V=1026
	τ=3.0e-3	τ=7.5e-4
0.000	0.281745	0.256451
0.045	0.224503	0.206107
0.090	0.182472	0.167306
0.135	0.146004	0.130404
0.180	0.110983	0.094962
0.225	0.053318	0.046544

Table 6

Evolution of r_A in the case with the relative area preserving redistribution.

t	n_V=258	n_V=1026
	τ=3.0e-3	τ=7.5e-4
0.000	0.281745	0.256451
0.045	0.273905	0.252824
0.090	0.267797	0.250508
0.135	0.261826	0.249062
0.180	0.255447	0.248805
0.225	0.247222	0.251225

Table 7

Evolution of r_A in the case with the asymptotically uniform redistribution.

t	n_V=258	n_V=258	n_V=258	n_V=1026	n_V=1026	n_V=1026
	τ=3.0e-3	τ=3.0e-3	τ=3.0e-3	τ=7.5e-4	τ=7.5e-4	τ=7.5e-4
	ω=1.0	ω=10.0	ω=100.0	ω=1.0	ω=10.0	ω=100.0
0.000	0.281745	0.281745	0.281745	0.256451	0.256451	0.256451
0.045	0.282313	0.354639	0.729898	0.263034	0.351779	0.721432
0.090	0.284454	0.419558	0.804872	0.270825	0.437894	0.781555
0.135	0.286607	0.476460	0.847488	0.279512	0.500058	0.817059
0.180	0.288372	0.528390	0.880004	0.289722	0.547976	0.844647
0.225	0.289122	0.587350	0.842915	0.304414	0.600890	0.853963

of points, we can observe a decrease of r_A. If we use the redistribution preserving relative areas, there is only a small change in r_A due to the discretization error. For the asymptotically uniform redistribution, r_A increases with a rate depending on the redistribution speed ω.

In the second experiment, we followed the mean curvature flow of the discretized ellipsoid with semiprincipal axes of lengths $a = 3.0$, $b = 1.0$, $c = 0.8$. This surface has regions with relatively high curvature and these parts become problematic in the course of the evolution process. It is demonstrated in Figure 5, 6, 7, and 8, where we can see how the mesh triangles and control volumes contract in regions with high curvature. We can observe that for both presented discretizations of the ellipsoid the area of some co-volumes shrinks so that it leads to obviously wrong results. As we could see in section 3.1.4, the tangential redistribution introduces some extra numerical error in the earlier stages of the evolution, but in the later stages the approximation is much more correct than in the case with no tangential movement.

Finally, we present an experiment concerning the volume preserving mean curvature flow. This time we used a cymling-like shape with 1026 discretization points. This shape is parametrized as

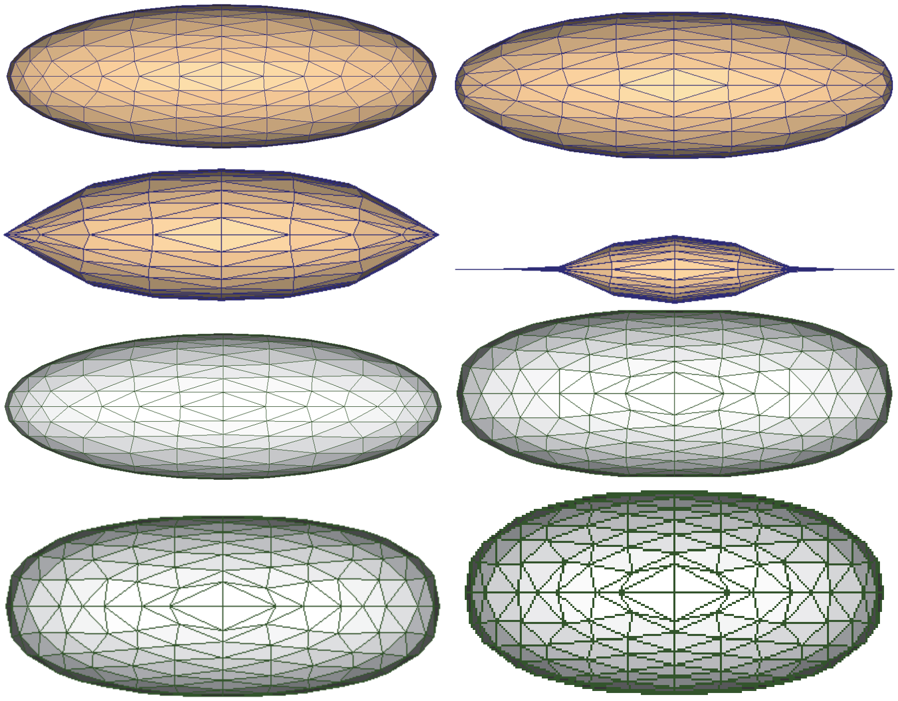

FIG. 5. *Evolution of a discretized ellipsoid with* 258 *grid points,* $\tau = 3.0e-3$*; the selected time steps are* $n = 0$, $n = 40$, $n = 80$, $n = 110$. *The first two rows correspond to the case with no tangential redistribution; the other images represent the evolution with the asymptotically uniform tangential redistribution with* $\omega = 100.0$.

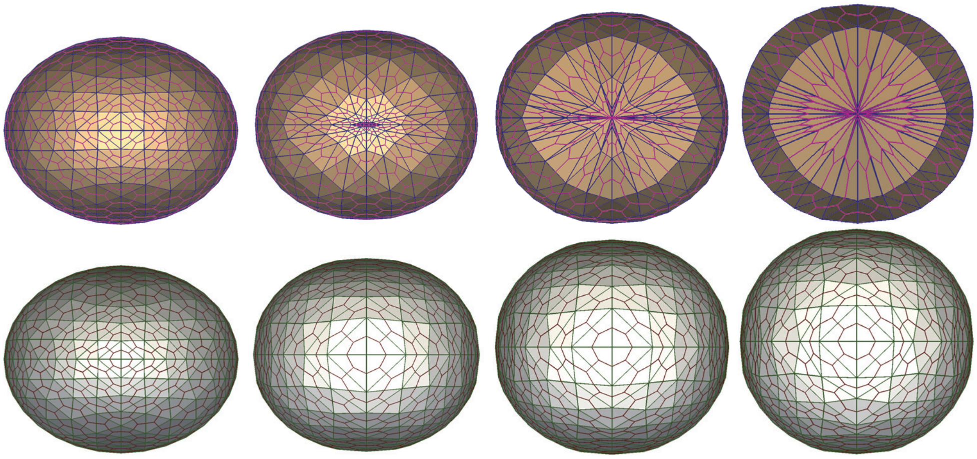

FIG. 6. *Evolution of a discretized ellipsoid with* 258 *grid points,* $\tau = 3.0e-3$, *with both triangle and co-volume mesh displayed. The selected time steps are* $n = 0$, $n = 40$, $n = 80$, $n = 110$. *The first row corresponds to the case with no tangential redistribution, and the second row represents the evolution with the asymptotically uniform tangential redistribution with* $\omega = 100.0$. *The view is set so that we can observe the point with the highest initial mean curvature. The images are scaled in order to show more details.*

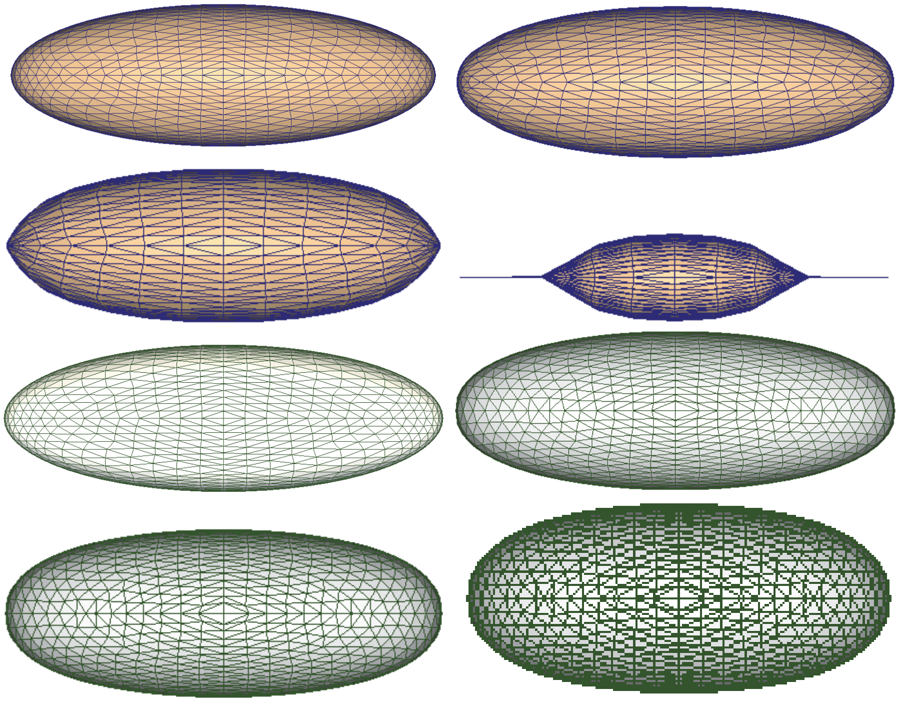

Fig. 7. *Evolution of a discretized ellipsoid with* 1026 *grid points,* $\tau = 7.5\mathrm{e}-4$*; the selected time steps are* $n = 0$, $n = 160$, $n = 320$, $n = 440$. *The first two rows correspond to the case with no tangential redistribution, and the other images represent the evolution with the asymptotically uniform tangential redistribution with* $\omega = 100.0$.

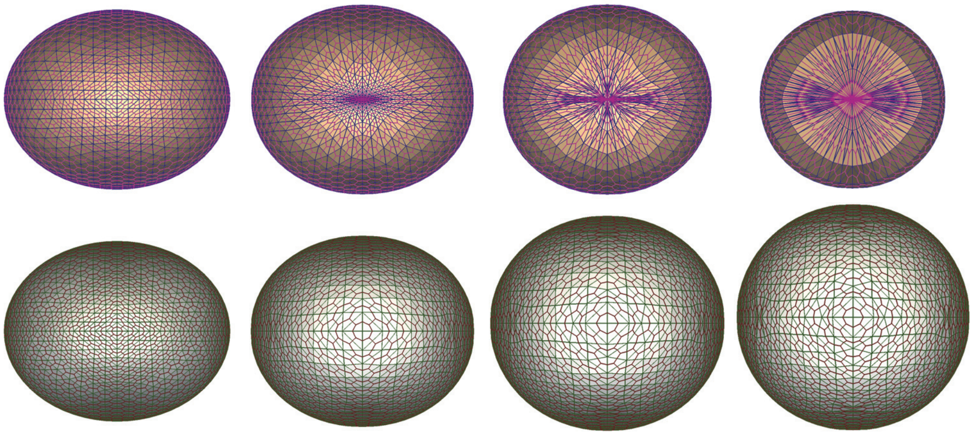

Fig. 8. *Evolution of a discretized ellipsoid with* 1026 *grid points,* $\tau = 7.5\mathrm{e}-4$, *with both triangle and co-volume mesh displayed. The selected time steps are* $n = 0$, $n = 160$, $n = 320$, $n = 440$. *The first row corresponds to the case with no tangential redistribution, and the second row represents the evolution with the asymptotically uniform tangential redistribution with* $\omega = 100.0$. *The view is set so that we can observe the point with the highest initial curvature. The images are scaled in order to show more details.*

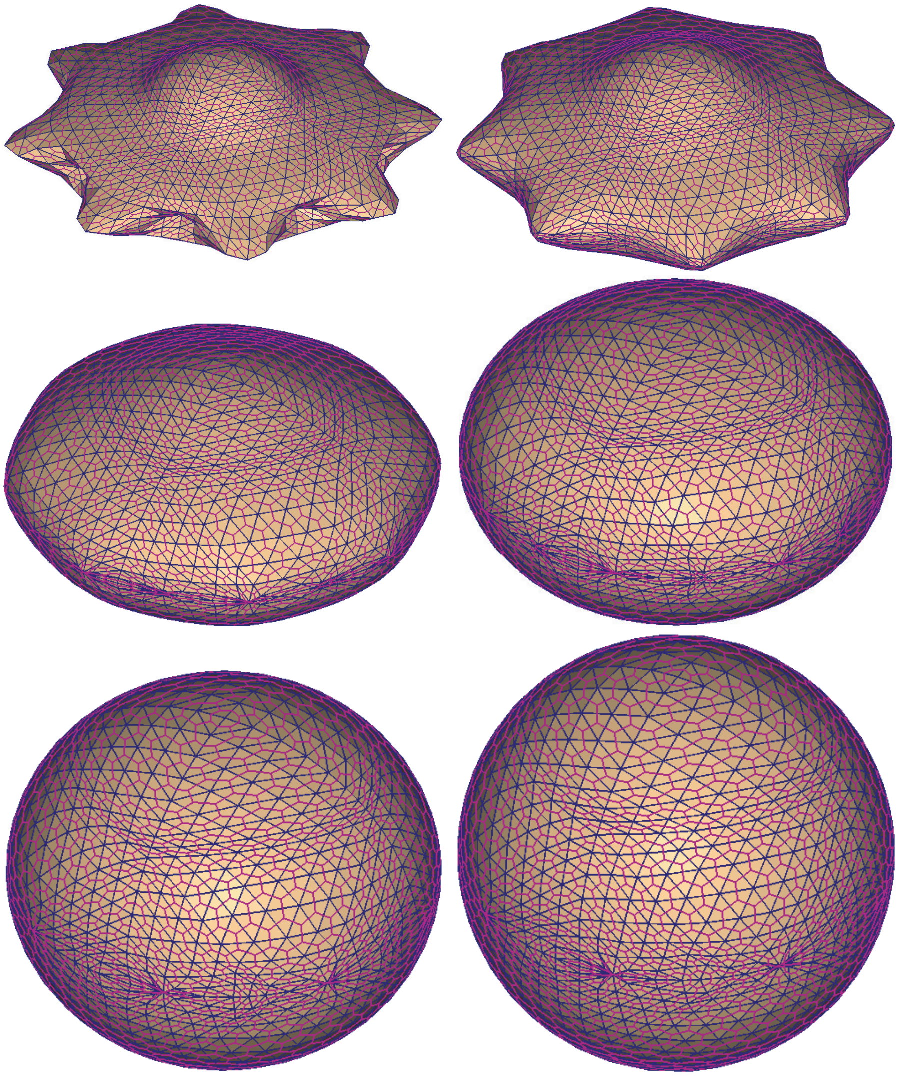

FIG. 9. *Evolution of a discretized cymling-like shape with* 1026 *grid points,* $\tau = 7.5e-4$*, the volume preserving mean curvature flow. Both triangle and co-volume mesh are displayed. The selected time steps are* $n = 0$*,* $n = 10$*,* $n = 40$*,* $n = 100$*,* $n = 250$*,* $n = 1000$*. No tangential redistribution was used.*

$$\begin{aligned}
x(u,v) &= \left(0.15\cos(4u) + \frac{0.5\sin(8v)}{\sqrt{2\pi}}e^{-50u^2} + 0.85\right)\cos u \cos v,\\
y(u,v) &= \left(0.15\cos(4u) + \frac{0.5\sin(8v)}{\sqrt{2\pi}}e^{-50u^2} + 0.85\right)\cos u \sin v,\\
z(u,v) &= -0.5\left(0.15\cos(4u) + \frac{0.5\sin(8v)}{\sqrt{2\pi}}e^{-50u^2} + 0.85\right)\sin u
\end{aligned}$$

and it evolves (for $t \to \infty$) into a sphere with the same volume. We show several selected steps of the evolution with no tangential movement and with the asymptotically uniform redistribution with $\omega = 100$ (Figures 9 and 10). We also present a table

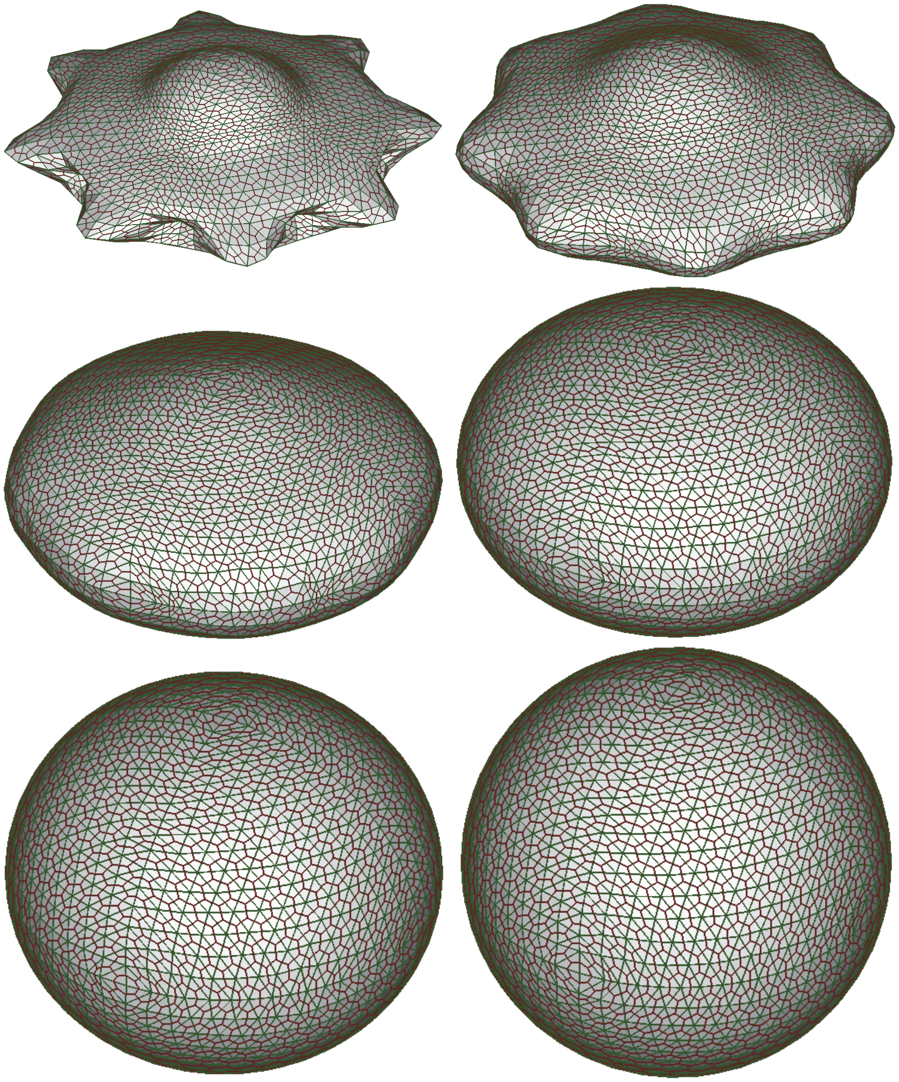

FIG. 10. *Evolution of a discretized cymling-like shape with* 1026 *grid points,* $\tau = 7.5e - 4$, *the volume preserving mean curvature flow. Both triangle and co-volume mesh are displayed. The selected time steps are* $n = 0$, $n = 10$, $n = 40$, $n = 100$, $n = 250$, $n = 1000$. *The asymptotically uniform tangential redistribution with* $\omega = 100.0$ *was used.*

illustrating the evolution of r_A (Table 8) and a detail of the mesh at the end of the evolution (Figure 11). We set the time step $\tau = 7.5e - 4$. Even though this time we do not observe any completely incorrect result, the quality of the mesh during the evolution is much better in the case when we use the tangential redistribution.

Remark 3.3. The redistribution methods are designed so that the redistribution speed ω can be time-dependent. However, in all presented experiments as well as in the examples that follow, $\omega(t)$ was set to be a constant function. This choice was sufficient for demonstrating the behavior of the methods and to obtain results of desired quality. Using a time-dependent redistribution speed would be necessary, for example, if we wanted to achieve a uniform distribution of points at a finite time [31].

TABLE 8

Evolution of r_A for the cymling-like shape, the volume preserving mean curvature flow.

t	No redistribution	Asymptotically uniform redistribution, $\omega = 100.0$
0.0000	0.294552	0.294552
0.0075	0.133083	0.414416
0.0300	0.010791	0.469153
0.0750	0.016281	0.688545
0.1875	0.004780	0.775162
0.7500	0.006693	0.809300

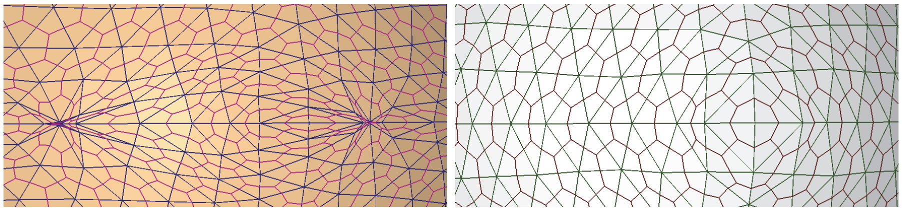

FIG. 11. *A detail of the mesh of the cymling-like shape at $t = 1000$. Left, the case with no tangential redistribution. Right, the same part of the shape but with the asymptotically uniform redistribution applied.*

A practical application of mean curvature flow with tangential redistribution. In some cases we may deal with problems that directly require some sort of uniform discretization of surfaces. Here we present a problem coming from architecture—the design of shell structures. These structures have become very popular in the last decades because they are lightweight and flexible—they can have almost an arbitrary shape. A shell structure consists of two types of elements—1D truss elements usually made of metal or wood and 2D shell elements that can be made, for example, of glass, plastic, or wood. In order to optimize the manufacturing process, it is important to have as many equal elements as possible. In many cases, some sort of uniformity can also help to meet the aesthetic criteria.

The mean curvature flow model combined with the tangential redistribution of mesh points is an efficient tool for designing structures that represent minimal surfaces. Given a boundary curve Γ, the corresponding minimal surface is obtained by applying the mean curvature flow equation (3.1), where F^0 is a smooth surface with the boundary Γ and it is a topological disk. The truss structure naturally represents the discretization of the surface. In order to be able to control the quality of the grid, we can use the model (3.5) with the tangential redistribution term. Our goal in the design process is to obtain a structure with as many equally sized truss elements as possible. Therefore, this time we use the length-oriented redistribution.

The topic of the truss structure design was elaborated in detail in a recent work [23]. Here we show one example. In this case the boundary curve Γ consisted of four segments,

$$\begin{aligned}
\Gamma_W(z) &= (0.0, z, -(z-0.5)^2 + 0.25), && z \in \langle 0, 1\rangle,\\
\Gamma_E(z) &= (1.0, z, -(z-0.5)^2 + 0.25), && z \in \langle 0, 1\rangle,\\
\Gamma_N(z) &= (z, 1.0, 0.0), && z \in \langle 0, 1\rangle,\\
\Gamma_S(z) &= (z, 0.0, 0.0), && z \in \langle 0, 1\rangle.
\end{aligned}$$

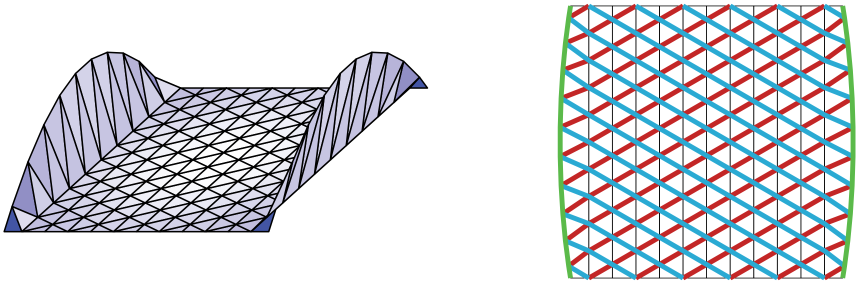

FIG. 12. *The shell structure design—two different views of the initial condition. Left, an axonometric projection; right, a perspective projection. The curves that were used for the length-oriented redistribution are highlighted by thicker red, green, and blue lines.*

At the beginning of the evolution, all grid points were situated in the xy-plane except the boundary points that were placed on the boundary curve (Figure 12). The network of curves γ_k that were used for the length-oriented redistribution consisted of the diagonal curves. Moreover, the points were also allowed to move on the boundary segments Γ_W and Γ_E. We used the redistribution with the prescribed limit length, which means that the tangential velocity was obtained from (2.22). However, since the movement of the endpoints is limited, the limit length was prescribed only for the parts of the curves between their second point and the last but one point. It was set to $A_{\infty,k} = d_{\infty,k}(p_k - 3)$, where p_k is the number of node points of the curve. The parameter $d_{\infty,k}$ represents the length of a single truss segment of the kth curve and it was set to $d_{\infty,k} = 0.101$ for the inner curves and $d_{\infty,k} = 0.12$ for the boundary curves. As for the other parameters, the time step was set to $\tau = 6.25e - 3$ and we stopped the computation after 800 time steps. The redistribution speed was $\omega = 800.0$. The parameter ω was large in order to obtain equally sized truss elements (within the tolerance that allows length differences of about 1‰ of the element length). The tangential speed in the endpoints of the curves was set to zero.

The final question is how to set the value of α in the second points ($\gamma^n_{1,k}$) of the curves. For aesthetic reasons we decided that the first and the last segment of a curve should have the same length at the end of the evolution. This can be achieved by the following choice of $\alpha^n_{1,k}$:

$$\alpha^n_{1,k} = d^n_{1,k} h^n_{1,k} \cdot \kappa^n_{1,k} + (d^n_{p_k,k} - d^n_{1,k})\omega,$$

where $h^n_{1,k}$ is the mean curvature vector of the surface at the point $\gamma^n_{1,k}$, $\kappa^n_{1,k}$ is the curvature vector of γ^n_k at $\gamma^n_{1,k}$, and $d^n_{1,k}$, $d^n_{p_k,k}$ are the lengths of the first and the last segment of γ^n_k.

The results are shown in Figures 13 and 14. Of 401 truss elements, 180 have the same length (within the allowed tolerance). That represents 44.89% of the total number of trusses. For the number of segments tending to infinity, this ratio would approach $\frac{2}{3} = 66.67\%$ (since the curve network is chosen so that two of three sides of almost all triangles would have the prescribed length). The result is also satisfactory from the aesthetic point of view.

3.2. Surface evolution with an advection component. Now we will present an example of surface evolution that, besides the mean curvature movement, contains an advection component. One of the typical applications where such evolution occurs is 3D image segmentation. In this case, the evolution of the surface depends on the

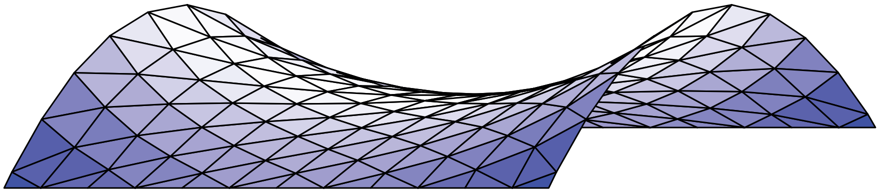

FIG. 13. *The shell structure design—the computed triangulated minimal surface in axonometric projection.*

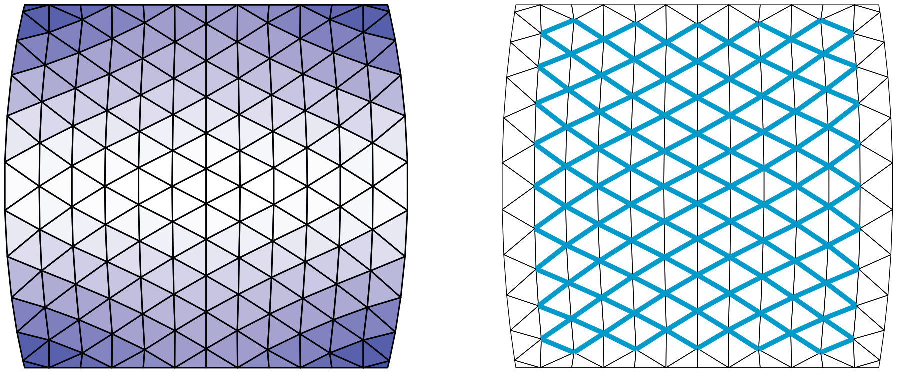

FIG. 14. *The shell structure design—the computed triangulated minimal surface in perspective projection, top view. The equally sized truss elements are highlighted on the right picture.*

image intensity function. Let $I\colon \mathbb{R}^3 \supset \Omega \to \mathbb{R}$ be an image intensity function. The manifold X will be a 2D Riemannian sphere and $F\colon X \to \Omega \times \langle 0, t_s \rangle$ its time-dependent embedding in Ω. We used the following surface evolution model [29]:

$$\partial_t F = a \left(\nabla e \cdot N \right) N + be \Delta_{g_F} F + v_T, \tag{3.33}$$

where the function $e\colon \Omega \to \mathbb{R}$ is an edge detector of the image I. This equation is a 3D analogue of the geodesic active contour model [14] enriched with two parameters $a \in \mathbb{R}_+$, $b \in \mathbb{R}_+$. These parameters are used to better control the evolution process consisting of mean curvature flow and the edge detector driven flow. Assigning different weights to the two normal velocities, we are able to find their most appropriate combination for a particular image.

In order to construct a numerical approximation of (3.33), we apply a semi-implicit time discretization similar to (3.8),

$$\frac{F^n - F^{n-1}}{\tau} = a \left(\nabla e \cdot N^{n-1} \right) N^{n-1} + b \Delta_{F^{n-1}} F^n + v_T^{n-1}. \tag{3.34}$$

For the space discretization, we need two grids—the triangulation representing the evolving surface and the voxel grid of the 3D image. The surface triangulation has been described in section 3.1, as well as the discretization of the Laplace–Beltrami operator. What we need to approximate now is the edge detector e and its gradient.

Let us suppose the image domain Ω is a box subdivided into cubic voxels of side length l and that the image intensity I is constant on any voxel. Since X is embedded in Ω, $F^n(X_i)$ is situated in a voxel P_j, where $j = (x, y, z)$ represents a vector of integer coordinates. The representative value of e in P_j will be denoted by e_j. Further, v_1,

v_2 and v_3 are the standard basis vectors in $\mathbb{R}^3$. The voxel faces will be denoted by $S_j^{\pm p}$, $p = 1, 2, 3$.

First, we construct the approximation of ∇e in the barycenter $c_j^{\pm p}$ of $S_j^{\pm p}$. The derivative in the direction of v_p is discretized by

$$D^{\pm p} I_j = \pm \left(I_{j \pm v_p} - I_j \right) / l. \tag{3.35}$$

For the other two directions v_q, $q \neq p$, we will use the values of e in the centers of the voxel edges $S_j^{\pm p, \pm q}$ denoted by $e_{j \pm \frac{1}{2} v_p \pm \frac{1}{2} v_q}$. We set

$$D^{\pm p,q} e_j = \frac{e_{j \pm \frac{1}{2} v_p + \frac{1}{2} v_q} - e_{j \pm \frac{1}{2} v_p - \frac{1}{2} v_q}}{l}, \tag{3.36}$$

where

$$e_{j \pm \frac{1}{2} v_p \pm \frac{1}{2} v_q} = \frac{e_j + e_{j \pm v_p} + e_{j \pm v_q} + e_{j \pm v_p \pm v_q}}{4}.$$

Finally, the gradient of e in the barycenter of the voxel P_j can be approximated by taking the arithmetic mean of the approximate gradients on its faces.

In our experiment, the edge detector was of the form

$$e(x, y, z) = \frac{1}{1 + K \| \nabla I(x, y, z) \|^2}, \tag{3.37}$$

where K is a positive real constant. So, in addition, we need to approximate ∇I. Here, we apply again (3.35) and (3.36) with e substituted by I. Afterward, we compute the norm of the approximate gradient on each face and then take the arithmetic mean of the six obtained values.

The described segmentation technique was applied to a zebrafish cell nucleus image. Segmentation of cell nuclei plays an important role in cell image analysis [12]. Particularly, having the segmented object in the form of a triangulated surface allows us to directly estimate the surface of the nucleus or evaluate its shape by principal component analysis. Before segmenting, the image was presmoothed by the geodesic mean curvature flow filtering algorithm [14, 15, 25]. The results of the segmentation are shown in Figures 15 and 16. The values of the model parameters were set to $n_V = 258$, $\tau = 0.001$, $l = 1.0$, $a = 1.0$, $b = 200.0$ for time steps $1 \ldots 200$ and $b = 1.0$ after. We used the asymptotically uniform volume-oriented tangential redistribution with $\omega = 100.0$. The initial condition was a sphere centered in a manually estimated nucleus center. Figure 15 shows a 2D slice of the segmented nucleus surface. Figure 16 demonstrates the effect of the tangential redistribution by showing the quality of the resulting triangulation compared to the mesh obtained without any tangential movement of grid points.

3.3. Curvature driven flow of a curve on a sphere. Finally, in order to provide an example when Y is not a Euclidean space, we show an example of a curve evolving on a sphere. The metric g_Y is induced by an embedding of Y in $\mathbb{R}^3$ given by the parametrization

$$S(\vartheta, \varphi) = (\cos\vartheta \cos\varphi, \cos\vartheta \sin\varphi, \sin\vartheta), \quad \vartheta \in \langle -\pi/2, \pi/2 \rangle, \quad \varphi \in \langle -\pi, \pi \rangle.$$

The evolution equation is again

$$\partial_t F = h + v_T. \tag{3.38}$$

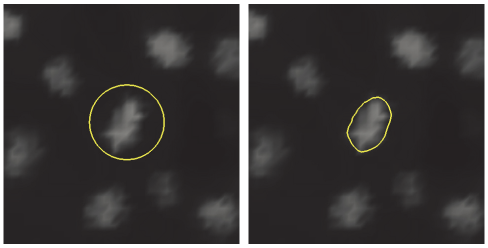

FIG. 15. *Cell nucleus segmentation. A 2D slice of the initial condition and the segmented surface are shown together with the corresponding slice of the image.*

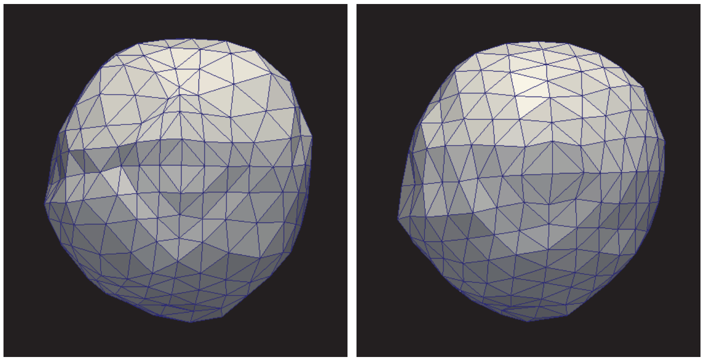

FIG. 16. *Cell nucleus segmentation. Left, the segmented nucleus surface obtained with no tangential redistribution. Right, the surface obtained with tangential redistribution of mesh points,* $\omega = 100.0$.

In our case, X will be a curve with boundary and we set $\partial_t F = 0$ in the boundary points. In such setting, the image of X will converge to a straight line—a geodesic—in Y.

In our experiments we used an explicit time discretization and the space discretization was done according to (3.27)–(3.30). The only difference is that the (approximated) curvature vector computed by (3.30) has to be projected on the tangent space of the sphere in the corresponding point of the curve. That means

$$h(X_i, t^{n-1}) \approx \bar{h}_i^{n-1} - (\bar{h}_i^{n-1} \cdot N_{Y,i}^{n-1}) N_{Y,i}^{n-1},$$

where $\bar{h}_i^{n-1}$ is the vector obtained by (3.30) and $N_{Y,i}^{n-1}$ is a unit normal to Y in $F^{n-1}(X_i)$. Since Y is embedded in $\mathbb{R}^3$ as the unit sphere centered at the origin, we have simply $N_{Y,i}^{n-1} = F^{n-1}(X_i)$.

The results of the experiments are shown in Figures 17, 18 and 19. The initial condition is given by the curve γ,

$$\gamma(t) = S(t, 0.3\sin(5t) + (t - \pi/5)^2(t + \pi/5)(-\cos(9t) - 0.1t + 1)), \quad t \in \langle -\pi/5, \pi/5 \rangle.$$

The curve was discretized by 50 grid points. Further, we used $\tau = 2e - 4$ and we performed 3000 time steps. The first figure shows the initial condition and the result of the evolution. The second figure represents a closer view on the evolution, showing several stages of the process together with all discretization points. Here, we used the asymptotically uniform redistribution with $\omega = 100$. The last figure shows what

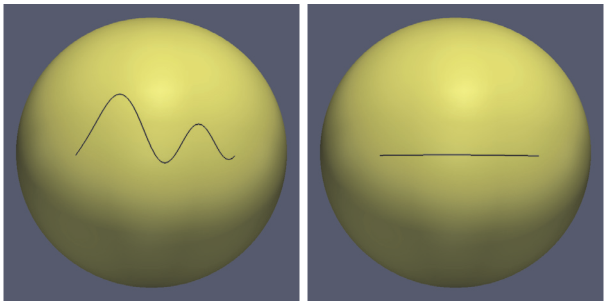

FIG. 17. *Curvature driven evolution of a curve on a sphere with the asymptotically uniform redistribution of grid points. Left, the initial condition. Right, the result of the evolution after* 3000 *time steps—we can see that the curve converges to a geodesic line.*

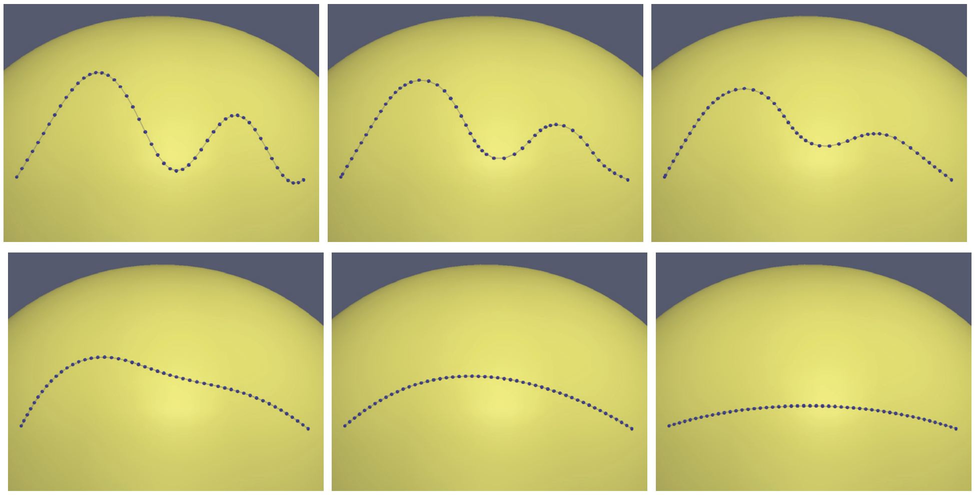

FIG. 18. *Curvature driven evolution of a curve on a sphere with the asymptotically uniform redistribution of grid points. The grid points are marked to demonstrate the effect of the redistribution. Time steps* 0*,* 20*,* 50*,* 150*,* 500*, and* 3000 *are shown.*

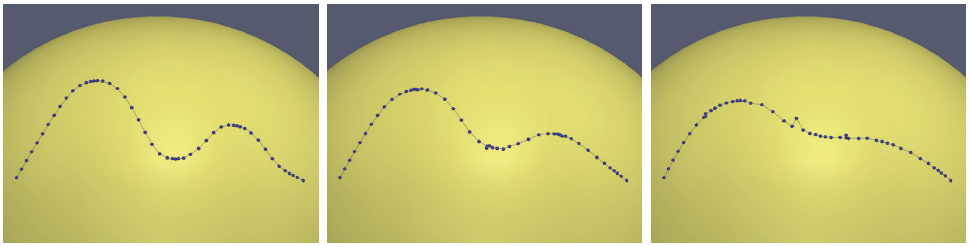

FIG. 19. *Curvature driven evolution of a curve on a sphere without any redistribution. Time steps* 20*,* 50*, and* 110 *are shown.*

happens if no redistribution is applied. Very soon, the algorithm fails to provide a correct result.

4. Conclusions. We presented several possibilities for tangential redistribution of points on evolving manifolds based on controlling the volume density. We demonstrated the performance of the proposed techniques on several examples and a practical application.

There are several issues left for further research. Tangential redistribution makes the Lagrangian approach practically applicable; however, we do not consider topological changes in this paper. Further, the tangential velocity field is considered to be a gradient field. It could be useful to investigate how limiting this assumption is and what other possibilities there are for obtaining this vector field. As far as the numerical scheme is concerned, it is presented for triangular discretizations of surfaces. For the sake of generality, it could be extended to general polygonal meshes.

REFERENCES

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, *Axioms and fundamental equations of image processing*, Arch. Ration. Mech. Anal., 123 (1993), pp. 200–257.

[2] S. B. Angenent and M. E. Gurtin, *Multiphase thermomechanics with an interfacial structure 2. Evolution of an isothermal interface*, Arch. Ration. Mech. Anal., 108 (1989), pp. 323–391.

[3] M. Balažovjech, K. Mikula, M. Petrášová, and J. Urbán, *Lagrangean method with topological changes for numerical modelling of forest fire propagation*, in Proceedings of ALGORITMY 2012, 19th Conference on Scientific Computing, Podbanské, Slovakia, 2012, pp. 42–52.

[4] E. Bänsch and A. Schmidt, *Simulation of dendritic crystal growth with thermal convection*, Interfaces Free Bound., 2 (2000), pp. 95–115.

[5] P. W. Bates, Z. Chen, Y. Sun, G. W. Wei, and S. Zhao, *Geometric and potential driving formation and evolution of biomolecular surfaces*, J. Math. Biol., 59 (2009), pp. 193–231.

[6] J. W. Barrett, H. Garcke, and R. Nurnberg, *On the parametric finite element approximation of evolving hypersurfaces in R^3*, J. Comput. Phys., 227 (2008), pp. 4281–4307.

[7] J. W. Barrett, H. Garcke, and R. Nurnberg, *Numerical approximation of gradient flows for closed curves in $\mathbb{R}^d$*, IMA J. Numer. Anal., 30 (2010), pp. 4–60.

[8] J. W. Barrett, H. Garcke, and R. Nurnberg, *Parametric approximation of surface clusters driven by isotropic and anisotropic surface energies*, Interfaces Free Bound., 12 (2010), pp. 187–234.

[9] J. W. Barrett, H. Garcke, and R. Nurnberg, *Parametric approximation of isotropic and anisotropic elastic flow for closed and open curves*, Numer. Math., 120 (2012), pp. 489–542.

[10] J. W. Barrett, H. Garcke, and R. Nurnberg, *The approximation of planar curve evolutions by stable fully implicit finite element schemes that equidistribute*, Numer. Methods Partial Differential Equations, 27 (2011), pp. 1–30.

[11] M. Bauer, P. Harms, and P. W. Michor, *Sobolev metrics on shape space of surfaces*, J. Geom. Mech., 3 (2011), pp. 389–438.

[12] P. Bourgine, R. Čunderlík, O. Drblíková-Stašová, K. Mikula, N. Peyriéras, M. Remešíková, B. Rizzi, and A. Sarti, *4D embryogenesis image analysis using PDE methods of image processing*, Kybernetika, 46 (2010), pp. 226–259.

[13] F. Cao and L. Moisan, *Geometric computation of curvature driven plane curve evolutions*, SIAM J. Numer. Anal., 39 (2001), pp. 624–646.

[14] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vis., 22 (1997), pp. 61–79.

[15] Y. Chen, B. C. Vemuri, and L. Wang, *Image denoising and segmentation via nonlinear diffusion*, Comput. Math. Appl., 39 (2000), pp. 131–149.

[16] K. Deckelnick, *Weak solutions of the curve shortening flow*, Calc. Var. Partial Differential Equations, 5 (1997), pp. 489–510.

[17] G. Dziuk and C. M. Elliott, *Finite elements on evolving surfaces*, IMA J. Numer. Anal., 27 (2007), pp. 262–292.

[18] G. Dziuk, *Algorithm for evolutionary surfaces*, Numer. Math., 58 (1991), pp. 603–611.

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

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

[21] T. Y. Hou, I. Klapper, and H. Si, *Removing the stiffness of curvature in computing* 3*-d filaments*, J. Comput. Phys., 143 (1998), pp. 628–664.

[22] T. Y. Hou, J. Lowengrub, and M. Shelley, *Removing the stiffness from interfacial flows and surface tension*, J. Comput. Phys., 114 (1994), pp. 312–338.

[23] M. Húska, M. Medľa, K. Mikula, P. Novysedlák, and M. Remešíková, *A new form-finding method based on mean curvature flow of surfaces*, in Proceedings of ALGORITMY 2012, 19th Conference on Scientific Computing, Podbanské, Slovakia, 2012, pp. 120–131.

[24] M. Kass, A. Witkin, and D. Terzopulos, *Snakes: Active contour models*, Int. J. Comput. Vis., 1 (1987), pp. 321–331.

[25] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, *Conformal curvature flows: from phase transitions to active vision*, Arch. Ration. Mech. Anal., 134 (1996), pp. 275–301.

[26] M. Kimura, *Numerical analysis for moving boundary problems using the boundary tracking method*, Japan J. Indust. Appl. Math., 14 (1997), pp. 373–398.

[27] M. Meyer, M. Desbrun, P. Schroeder, and A. H. Barr, *Discrete differential geometry operators for triangulated* 2*-manifolds*, in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., Springer-Verlag, Berlin, 2003, pp. 35–57.

[28] K. Mikula, N. Peyriéras, M. Remešíková, and M. Smíšek, 4*D numerical schemes for cell image segmentation and tracking*, in Proceedings of Finite Volumes in Complex Applications VI, Problems & Perspectives, Springer-Verlag, Berlin, 2011, pp. 693–702.

[29] K. Mikula and M.Remešíková, 3*D Lagrangian segmentation with simultaneous mesh adjustment*, Proceedings of Finite Volumes in Complex Applications VII - Elliptic, Parabolic and Hyperbolic Problems, Springer-Verlag, Berlin (2014), pp. 685-693.

[30] K. Mikula and D. Ševčovič, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473–1501.

[31] K. Mikula and D. Ševčovič, *A direct method for solving an anisotropic mean curvature flow of planar curve with an external force*, Math. Methods Appl. Sci., 27 (2004), pp. 1545–1565.

[32] K. Mikula and D. Ševčovič, *Evolution of curves on surface driven by the geodesic curvature and external force*, Appl. Anal., 85 (2006), pp. 345–362.

[33] K. Mikula and J. Urbán, 3*D curve evolution algorithm with tangential redistribution for a fully automatic finding of an ideal camera path in virtual colonoscopy*, in Proceedings of the Third International Conference on Scale Space Methods and Variational Methods in Computer Vision 2011, Lecture Notes in Comput. Sci. 6667, Springer-Verlag, Berlin, 2011.

[34] K. Mikula and J. Urbán, *A new tangentially stabilized* 3*D curve evolution algorithm and its application in virtual colonoscopy*, Adv. Comput. Math. (2014).

[35] S. Morigi, *Geometric surface evolution with tangential contribution*, J. Comput. Appl. Math., 233 (2010), pp. 1277–1287.

[36] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, Berlin, 2003.

[37] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experiment. Math., 2 (1993), pp. 15–36.

[38] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge University Press, New York, 1999.

[39] D. Ševčovič and S. Yazaki, *Computational and qualitative aspects of motion of plane curves with a curvature adjusted tangential velocity*, Math. Methods Appl. Sci., 35 (2012), pp. 1784–1798.

[40] T. Ushijima and S. Yazaki, *Convergence of a crystalline algorithm for the motion of a closed convex curve by a power of curvature $V = K^{\alpha}$*, SIAM J. Numer. Anal., 37 (2000), pp. 500–522.

[41] H. A. Van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[42] S. Yazaki, *On the tangential velocity arising in a crystalline approximation of evolving plane curves*, Kybernetika, 43 (2007), pp. 913–918.