Diffusion-redistanciation schemes for 2D and 3D constrained Willmore flow: application to the equilibrium shapes of vesicles

Thibaut Métivet, Arnaud Sengers, Mourad Ismail, Emmanuel Maitre

Diffusion-redistanciation schemes for 2D and 3D constrained Willmore flow: application to the equilibrium shapes of vesicles

Thibaut Metivet[(1)], **Arnaud Sengers**[(1)], **Mourad Ismaïl**[(2)] **and Emmanuel Maitre**[(1)]

[(1)]: *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*[1], *LJK, 38000 Grenoble, France*
[(2)]: *Laboratoire Interdisciplinaire de Physique, Univ. Grenoble Alpes and CNRS.*

Abstract: In this paper we present a novel algorithm for simulating the Willmore flow with conservation of volume and area. The idea is to adapt the class of algorithms introduced in [17] to the Willmore flow and extend it in dimension three. These algorithms rely on alternating a diffusion of the signed distance function to the interface and a redistanciation step, following Merriman, Bence, and Osher's principle [41]. The constraints are enforced between the diffusion and redistanciation steps via a simple rescaling method. The energy globally decreases at the end of each global step. The algorithm features the computational efficiency of thresholding methods without requiring adaptive remeshing thanks to the use of a signed distance function to describe the interface. This opens its application to dynamic fluid-structure simulations. The methodology is validated by computing the equilibrium shapes of two and three-dimensional vesicles, as well as the Clifford torus.

1. Introduction

Numerous modelling situations can be cast as mathematical problems where an interface motion is driven by the minimisation of a geometric energy under geometric constraints. This is the case of multiphase flows, image segmentation, or elastic interface modeling, to cite a few. The geometric quantities involved in energy and constraints are for instance mean or gaussian curvature of interface, surface area or enclosed volume. The algorithms and methods developed in this article give an efficient framework to address such situations.

Our work was motivated by one of this modeling situations that we now describe. Vesicles are systems of two fluids separated by a bi-layer membrane of phospholipid molecules. These objects can be considered as a simple model for Red Blood Cells (RBC). Since the number of such molecules is constant, this kind of interface has constant area. Therefore, its shape is determined by high order energy, i.e. the mean curvature is minimised. Moreover, there is no exchange across this interface, so that the enclosed volume is constant. Mathematically, the problem of finding a surface minimising its mean curvature is the well known problem of Willmore [54]. In this work however, we are interested in this minimisation under the conservation of area and enclosed volume.

The numerical simulation of vesicles involve the resolution of two-fluid flows (for the inner and outer fluids) and fluid-structure interactions (for the membrane-fluids interaction), which is quite challenging: as a sharp object, a singularity occurs across the membrane bringing stress jump, which should either be dealt with explicit jump conditions, or appropriate numerical spreading. Since the membrane energy is of high gemoetrical order, its gradient involves high order derivatives of the unknowns. In addition, the inextensibility of the immersed interface usually accounted for using elastic tension energies with high modulus makes the resulting numerical problem very stiff.

A lot of such numerical simulations have been carried out by different teams using many numerical methods. To mention some representative works, we can cite the dynamic molecular method [38], the boundary integral method [3], the phase field method (see for example [4, 59, 15, 34, 6]) or level set method [5, 35, 37, 50, 13]. [26] also proposes a model for vesicles implemented using a Lattice Boltzmann Method.

In the context of finite-element methods coupled with level set technique we can cite [28, 13]. We can also mention the work [24] based on a finite-element method where the membrane is modelled as a necklace of small rigid particles.

In the work above, the nonlinear coupling between the fluid flow and the geometric description of interface is usually made explicit, which leads to severe restriction on the time step during simulation. Or it could be solved implicitly by a Newton-type method, which increases dramatically the cost per iteration. In [11], a semi-implicit

[1]Institute of Engineering Univ. Grenoble Alpes

scheme was proposed where an ad-hoc diffusion equation was used as a predictor step for the future position of a drop or a simple elastic interface.

Our approach in this work had as primary aim to propose a systematic way to built a predictor of the position of an interface with constant area and enclosed volume, when it moves to minimise high geometrical order energies, such as its mean curvature. In order to build such a predictor, we extended diffusion-thresholding/redistancing initiated in [41] and extended in [21, 18, 17, 27] to the case of area and enclosed volume conservation.

Outline. The paper is organised as follows: in section 2 we remind diffusion-thresholding and diffusion-redistancing schemes principles, and introduce a new methodology to write higher order motion schemes, such as the Willmore flow, that we found more systematic than in [17]. Then we present a new efficient method to take into account the area and enclosed volume constraints. In section 3 we address the problem of volume and area conservation, introducing an explicit analytic method to project on the constraint set. Then a section is devoted to numerical aspects of the implementation within finite-element approaches. We investigate in detail how to choose an optimal time step and plot numerical convergence curves for a basic scheme and an enhanced one. To evaluate the performance of the diffusion-redistanciation scheme without rescaling, we study the convergence of a torus under unconstrained Willmore flow to the optimal Clifford torus. At last, we present some numerical illustrations of the computation of 2D and 3D equilibrium shapes for vesicles that match well with those obtained with classical numerical schemes.

2. Numerical schemes for higher geometrical motion of interfaces

The class of *diffusion-generated motion* was introduced by the work [41] of Merriman, Bence and Osher. They proposed an efficient algorithm for computing the mean curvature flow of a surface (a curve in two dimensions) without any direct computation of the mean curvature. The algorithm consists in repeating two steps, namely a diffusion step (also named convolution step) and a thresholding step. During the diffusion step, the characteristic function representing the surface is diffused for a certain time step, and the $\frac{1}{2}$ iso-level moves depending on the local curvature. A characteristic function is then recovered by thresholding the resulting function at $\frac{1}{2}$, allowing to iterate the process.

Algorithm 1 Original Convolution-Thresholding scheme

while $t < t_f$ **do**
Solve $\partial_t \phi - \Delta \phi = 0$ with initial condition $\phi_i = \mathbb{1}^{(n)}$ for time δt
Construct the new characteristic function $\mathbb{1}^{(n+1)} = \mathbb{1}_{\{\phi \geq \frac{1}{2}\}}$.

The main advantage of this method is its simplicity and unconditional stability: the convolution step consists in the resolution of a heat equation which can be achieved efficiently, while the standard phase field approach would involve solving a more complicated non-linear equation or computing the curvature and solving a transport equation.

Due to its conceptual simplicity and numerical efficiency, various extensions of the method have been proposed since its introduction. The multiphase case is treated in [47] for junction with equal surface tensions and [16] for arbitrary surface tensions. The idea for the former algorithm is essentially to diffuse each phase separately and assign each point to the phase of maximal value. The later requires to modify the diffusion step to take into account the different surface tensions. An area preserving motion by mean curvature can be found in [49] by changing the threshold value from $\frac{1}{2}$ to a real number λ found using a Newton method. We refer to the next section for more details about area conservation.

Extensions of the convolution-thresholding method to the Willmore flow and other higher order geometric motion, such as surface diffusion, have been proposed independently in [18, 21] in both two and three dimensions. We recall that the Willmore flow is given by the following normal velocity :

$$W = \begin{cases} \Delta_\Gamma H + \dfrac{H^3}{2} & \text{in 2D} \\ \Delta_\Gamma H + 2H\left(H^2 - K\right) & \text{in 3D.} \end{cases} \tag{1}$$

In [18] and [21], a local expansion of the convoluted characteristic function is performed and shows that the velocity W appears at the second order of the diffused interface. More precisely, any point of the interface can be relocated at the origin in a way that the normal to the curve is aligned with the y-axis. The behavior of the point of interest during the convolution step is then described by the following expansion:

$$\phi(\delta t) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y \delta t^{-\frac{1}{2}} + \frac{1}{2\sqrt{\pi}} \kappa \delta^{\frac{1}{2}} - \frac{1}{4\sqrt{\pi}} W \delta t^{\frac{3}{2}} + \mathcal{O}\left(\delta t^{\frac{5}{2}}\right) \tag{2}$$

where $\phi(\delta t)$ denotes the function obtained by diffusing the characteristic function for a time δt. Thresholding the result at $\phi(\delta t) = \frac{1}{2}$ then results in a normal displacement of the point by $\kappa \delta t$, which gives the classical mean curvature scheme. As the velocity term of interest for the Willmore flow can be found at the second order in expansion eq. (2), one can then compute Willmore flow by extracting it with a linear combination of two different solutions of the convolution step taken at different time steps that eliminates the curvature term. If we note $\phi(\delta t)$ the function obtained by diffusing the characteristic function for a time δt, a correct combination to achieve this is:

$$2\phi\left(\sqrt{\delta t}\right) - \phi\left(2\sqrt{\delta t}\right) = -y + W\delta t + \mathcal{O}\left(\delta t^{\frac{3}{2}}\right)$$

At the expense of an additional diffusion at each step and an error of order $\delta t^{\frac{3}{2}}$, we can write an algorithm for the Willmore flow similar to the mean curvature one.

Algorithm 2 Convolution-Thresholding scheme for Willmore flow

while $t < t_f$ **do**

Solve $\partial_t \phi - \Delta \phi = 0$ with initial condition $\phi_i = \mathbb{1}^{(n)}$ for times $t = 2\sqrt{\delta t}$ and $t = \sqrt{\delta t} \to \phi\left(2\sqrt{\delta t}\right), \phi\left(\sqrt{\delta t}\right)$.

Compute $\mathcal{D} = 2\phi\left(\sqrt{\delta t}\right) - \phi\left(2\sqrt{\delta t}\right)$.

Construct the new characteristic function $\mathbb{1}^{(n+1)} = \mathbb{1}_{\{\mathcal{D} \geq \frac{1}{2}\}}$.

This method is highly valuable because it allows to compute a fourth-order flow without any derivation, especially when using low order polynomial approximations. Moreover, the use of regularised interfaces in phase-field methods makes the computation of the Laplace-Beltrami operator particularly difficult.

However the convolution-thresholding method has some indentified drawbacks. Its major issue is its inaccuracy and its ability to get "stuck" when the mesh is not refined enough at the interface. More precisely, as pointed out in [40], given a fixed uniform grid, if the motion during an iteration is smaller than the grid resolution, the thresholding step can reset the interface to its initial configuration and loop indefinitely. A solution to this problem can be found in [48], where the authors use adaptative grids to refine the mesh near the interface where the resolution is more important. Such adaptative strategy is also essential for accuracy considerations, especially in the case of higher order flows, but can become very expensive in the dynamic case or in dimension 3.

An alternative approach to the problem is to allow the function to contain subgrid informations. In [17], the authors replace to that end the characteristic function by a signed distance function and explore the motions that can be obtained through the diffusion and the redistanciation of a signed distance function. A similar expansion as eq. (2) of the solution of the heat equation starting from a distance function shows that the first order term is still a mean curvature term. At the expense of a redistanciation step, though more costly than the thresholding step, one can obtain significantly more accurate computations than the classical convolution thresholding algorithm. However, second order term in this case is not the desired Willmore term anymore and an additional correction is therefore necessary to compute the Willmore flow.

From a theoretical point of view, convergence of the classical algorithm of motion by mean curvature is well understood and various proofs of the consistency of the scheme exist. Some are based on a comparison principle [19, 2, 8], other are based on a gradient flow interpretation [1, 36]. Introduced in the two papers [16, 30], a new interpration of the convolution-thresolding algorithm as a gradient minimising flow has initiated more recent works on the subject [31, 32], while [58] relies on a formal matched asymptotic method. The extension to the multiphase flow can be found in [30] and the case of volume-preserving motion is studied in [33, 43]. The recent results on the gradient flow approach are reviewed in [29]. However, the convergence of higher order motions, especially the Willmore flow, remains an open question.

In this section, we propose a new local expansion of the signed distance function near the interface. It yields similar results to the one in [17] for the 2D case but our approach seems more intrinsic and its generalisation

to higher dimensions and orders is straightforward. We propose diffusion-redistanciation algorithms to compute the Willmore flow in 2D and 3D. As the term in the second order of the expansion is not quite the Willmore term, one has to add a correction term. The 3D version in particular shows a completely new approach and features a correction term using a diffusion of d^2.

2.1. Expansion of a distance function convoluted by the heat kernel

As suggested by the previous work of Esedoglu, Ruuth and Tsai [17], the high-order geometrical motions of a surface can be computed using the convolution of a distance function to this interface by the heat equation kernel. In this section, we thus derive an expansion of the solution of the heat equation with a distance function as its initial condition. In contrast with [17], we compute this expansion for both the two- and three-dimensional cases using a fully implicit approach.

We consider d a signed distance function to some interface Γ, and introduce the heat equation:

$$\partial_t \phi - \Delta \phi = 0. \tag{3}$$

supplemented with the initial condition:

$$\phi(t=0) = d. \tag{4}$$

The Taylor expansion of the heat equation solution with respect to time then reads:

$$\phi(\delta t) = \phi(0) + \partial_t \phi(0)\, \delta t + \partial_{tt}\phi(0)\, \frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right) \tag{5}$$

Recalling eq. (3) and the initial condition eq. (4), we then have $\partial_t \phi(0) = \Delta d$ and $\partial_{tt}\phi(0) = \Delta^2 d$. We denote $w = \frac{\Delta^2 d}{2}$ and can then rewrite eq. (5) as:

$$\begin{aligned} \mathcal{G}_{\delta t} \equiv \phi(\delta t) &= d + \Delta d\, \delta t + \frac{\Delta^2 d}{2}\, \delta t^2 + \mathcal{O}\left(\delta t^3\right) \\ &= d + \Delta d\, \delta t + w\, \delta t^2 + \mathcal{O}\left(\delta t^3\right). \end{aligned} \tag{6}$$

Remark 1. *When time discretisation of (3) will come into play, we will have to consider a second order scheme to get the right expansion. Using a first order Euler scheme, which could be desirable for is stability properties, leads to double the second order term magnitude. We refer to section 4.2 for more details.*

By definition, Δd is the total curvature of the local d iso-surface:

$$\Delta d = \kappa = (n-1)H$$

with n the space dimension and H the mean curvature. We can also express $\Delta^2 d$ close to the interface Γ as a function of the interface geometrical quantities, and in particular the Laplace-Beltrami divergence of the curvature $\Delta_\Gamma \kappa = (n-1)\Delta_\Gamma H$. More specifically, we have $\Delta_\Gamma f = \Delta\left(f(x - d\nabla d)\right)|_\Gamma \equiv \Delta\left(f \circ p\right)|_\Gamma$ for any $f \in \mathcal{C}^2(\Gamma)$ [12], so that:

$$\begin{aligned} \Delta_\Gamma(\Delta d) &= \Delta(\Delta d \circ p)|_\Gamma \\ &= \nabla_i \left(\nabla_j p_i\, \nabla_j \Delta d\right)|_\Gamma \\ &= \nabla_i \left((\delta_{ij} - \nabla_i d\, \nabla_j d - d\nabla_i \nabla_j d)\, \nabla_j \Delta d\right)|_\Gamma \\ &= \left[-\Delta d\, \nabla_N \Delta d - \nabla_N^2 \Delta d + \Delta^2 d - d\nabla_i \left(\nabla_i \nabla_j d\, \nabla_j \Delta d\right)\right]|_\Gamma \end{aligned}$$

where $N \equiv \nabla d$ is the interface normal and we have introduced the normal gradient $\nabla_N \equiv N \cdot \nabla$. Applying the restriction to $\Gamma \equiv \{d = 0\}$, we finally obtain:

$$\Delta^2 d = \Delta_\Gamma \kappa + \nabla_N^2 \kappa + \kappa\, \nabla_N \kappa \tag{7}$$

We can also compute:

$$\begin{aligned} \nabla_N \kappa &= \nabla_i d\, \nabla_i \nabla_k \nabla_k d \\ &= \nabla_i d\, \nabla_k \nabla_k \nabla_i d \\ &= \nabla_k \left(\cancel{\nabla_i d\, \nabla_k \nabla_i d}\right) - \nabla_k \nabla_i d\, \nabla_i \nabla_k d \\ &= -\mathrm{Tr}\left[(\nabla^2 d)^2\right] \end{aligned}$$

since the distance property $|\nabla d| = 1$ gives $\nabla_i d\, \nabla_k \nabla_i d = \nabla_k |\nabla d|^2/2 = 0$. Recalling that $\nabla^2 d$ is the Weingarten map of Γ, we can then write:

$$\nabla_N \kappa = -\sum_i \kappa_i^2 = \begin{cases} -\kappa^2 \\ 2K - \kappa^2 \end{cases} = \begin{cases} -H^2 & \text{in 2D} \\ -2\left(2H^2 - K\right) & \text{in 3D} \end{cases} \tag{8}$$

where we have introduced the principal curvatures $\kappa_i, i = 1, \ldots, n-1$, and the Gauss curvature $K = \prod_i \kappa_i$. The second normal derivative can then be written as:

$$\begin{aligned} \nabla_N^2 \kappa &= \nabla_i d\, \nabla_i \nabla_N \kappa \\ &= -2\, \nabla_i d\, \nabla_i \nabla_j \nabla_k d\, \nabla_k \nabla_j d \\ &= -2\, [\nabla_j (\cancel{\nabla_i d \nabla_k \nabla_i d}\, \nabla_j \nabla_k d) - \nabla_j \nabla_i d\, \nabla_k \nabla_i d\, \nabla_j \nabla_k d - \cancel{\nabla_i d \nabla_k \nabla_i d}\, \nabla_j \nabla_j \nabla_k d] \\ &= 2\, \mathrm{Tr}\left[(\nabla^2 d)^3\right] \end{aligned}$$

that is:

$$\nabla_N^2 \kappa = 2\sum_i \kappa_i^3 = \begin{cases} 2\kappa^3 \\ 2\kappa\left(\kappa^2 - 3K\right) \end{cases} = \begin{cases} 2H^3 & \text{in 2D} \\ 4H\left(4H^2 - 3K\right) & \text{in 3D.} \end{cases} \tag{9}$$

Inserting eq. (8) and eq. (9) into eq. (7), we finally obtain:

$$\Delta^2 d \equiv 2w = \begin{cases} \Delta_\Gamma \kappa + \kappa^3 \\ \Delta_\Gamma \kappa + \kappa(\kappa^2 - 4K) \end{cases} = \begin{cases} \Delta_\Gamma H + H^3 & \text{in 2D} \\ 2\left(\Delta_\Gamma H + 4H(H^2 - K)\right) & \text{in 3D.} \end{cases} \tag{10}$$

As expected, we observe that the second-order term in the expansion of the heat equation solution (6) feature the required high-order geometrical quantities for the geometrical flows we are interested in. In the following, we shall in particular devise specific algorithms for the simulation of Willmore flows, but the method presented here is generic, and can be carried on at higher orders similarly.

2.2. Convolution of a generic function of the distance

For practical reasons that will prove useful thereafter, we consider in this section the expansion of the heat equation solution eq. (3), but with the initial condition:

$$\phi(t = 0) = f(d)$$

where f is some generic $\mathcal{C}^\infty(\mathbb{R})$ function. Similarly to eq. (6), we denote $\mathcal{G}_{\delta t}$ the corresponding solution, and write as before:

$$\mathcal{G}_{\delta t}^{(f)} = f(d) + \Delta f(d)\, \delta t + \Delta^2 f(d)\, \frac{\delta t^2}{2} + \mathcal{O}\left(\delta t^3\right). \tag{11}$$

Recalling that $|\nabla d| = 1$, and using the usual derivation rules, we can then compute:

$$\Delta f(d) = \nabla \cdot (f^{(1)} \nabla d) = f^{(2)} + f^{(1)} \Delta d$$

and:

$$\begin{aligned} \Delta^2 f(d) &= f^{(4)}(d) + 2f^{(3)}(d)\Delta d + f^{(2)}(d)\left[(\Delta d)^2 + 2\nabla_N \Delta d\right] + f^{(1)}(d)\Delta^2 d \\ &= f^{(4)}(d) + 2f^{(3)}(d)\kappa + f^{(2)}(d)\left[\kappa^2 + 2\nabla_N \kappa\right] + 2f^{(1)}(d) w \\ &\equiv f^{(4)}(d) + 2f^{(3)}(d)\kappa + 2f^{(2)}(d) c + 2f^{(1)}(d) w \end{aligned}$$

where we have introduced:

$$c \equiv \frac{\kappa^2}{2} + \nabla_N \kappa = \begin{cases} -\dfrac{\kappa^2}{2} \\ 2K - \dfrac{\kappa^2}{2} \end{cases} = \begin{cases} -\dfrac{H^2}{2} & \text{in 2D} \\ -2(H^2 - K) & \text{in 3D.} \end{cases} \tag{12}$$

We eventually get:

$$\mathcal{G}_{\delta t}^{(f)} = f(d) + \delta t \left(f^{(2)}(d) + f^{(1)}(d)\kappa\right) + \delta t^2 \left(\frac{f^{(4)}(d)}{2} + f^{(3)}(d)\kappa + f^{(2)}(d) c + f^{(1)}(d) w\right) + \mathcal{O}\left(\delta t^3\right). \tag{13}$$

2.3. Expansion order selection

We now turn to the use of combinations of $\mathcal{G}$ for different time-steps in order to select the appropriate order in the expansion eq. (6). To minimise assembly and preconditioner computation costs, we restrict our analysis to the case of two different time-steps denoted $\alpha\delta t$ and $\beta\delta t$ respectively. To simplify the notations, we denote in the following $\mathcal{G}_\alpha \equiv \mathcal{G}_{\alpha\delta t}$ and $\mathcal{G}_\beta \equiv \mathcal{G}_{\beta\delta t}$. We then consider the general linear combination $a\,\mathcal{G}_\alpha + b\,\mathcal{G}_\beta$ which expansion writes:

$$a\,\mathcal{G}_\alpha + b\,\mathcal{G}_\beta = (a+b)\,d + (a\alpha + b\beta)\,\kappa\delta t + (a\alpha^2 + b\beta^2)\,w\delta t^2$$

To obtain the first- or second-order terms of eq. (6) independently, we need to choose a_1, b_1, a_2, b_2, α, and β such that:

$$\text{Order 1}: \quad a_1\alpha + b_1\beta = 1, \quad a_1\alpha^2 + b_1\beta^2 = 0 \tag{14}$$

$$\text{Order 2}: \quad a_2\alpha + b_2\beta = 0, \quad a_2\alpha^2 + b_2\beta^2 = -1 \tag{15}$$

With the additional constraint that $\alpha > 0$, $\beta > 0$, the general solution is:

$$\beta = \alpha\frac{\sqrt{-a_1}}{\sqrt{b_1}}, \quad b_2 = a_2\frac{\sqrt{b_1}}{\sqrt{-a_1}}.$$

We thus choose the following parameters:

$$\begin{aligned} &\alpha = \sqrt{2}, \quad \beta = \frac{\sqrt{2}}{2} \\ &a_1 = -\frac{\sqrt{2}}{2}, \quad b_1 = 2\sqrt{2} \\ &a_2 = -1, \quad b_2 = 2 \end{aligned}$$

so that:

$$a_1\,\mathcal{G}_\alpha + b_1\,\mathcal{G}_\beta = \frac{3\sqrt{2}}{2}d + \kappa\delta t + \mathcal{O}\left(\delta t^3\right) \tag{16}$$

and:

$$a_2\,\mathcal{G}_\alpha + b_2\,\mathcal{G}_\beta = d - w\delta t^2 + \mathcal{O}\left(\delta t^3\right) \tag{17}$$

2.4. Willmore flow

In this section, we derive an algorithm which relies on solving heat equation problems starting from a distance function to compute high-order motions of an interface in a two- or three-dimensional domain. We focus our analysis more specifically to the Willmore flow, but the approach can be straightforwardly extended to a large class of surface diffusion motions.

We consider the Willmore energy functional [61]:

$$\mathcal{E}_W(\Gamma) \equiv \int_\Gamma H^2. \tag{18}$$

The surfaces which minimise eq. (18) over the set of one- or two-dimensional surfaces can be obtained from the *Willmore flow* gradient descent, which moves iteratively some initial surface with the normal velocity WN where (c.f. [61]):

$$W = \begin{cases} \Delta_\Gamma H + \dfrac{H^3}{2} & \text{in 2D} \\ \Delta_\Gamma H + 2H(H^2 - K) & \text{in 3D.} \end{cases}$$

For a surface Γ implicitely defined by the signed distance function as $\Gamma = \left\{\vec{x} \in \mathbb{R}^3, d(\vec{x}) = 0\right\}$, the Willmore flow corresponds to a usual transport equation:

$$\partial_t d + WN \cdot \nabla d = 0. \tag{19}$$

Recalling that $\nabla d = N$, this amounts to solving $\partial_t d + W = 0$, or if we consider a first-order time-discretisation with time-step δt,

$$d^{(p+1)} = d^{(p)} - W\delta t. \tag{20}$$

We can then observe that the right-hand side of eq. (20) can be obtained from the expansions of solutions to the heat equation derived above. More precisely, the second order term $w\delta t^2$ (eq. (10)) provides the required high-order surface term $\Delta_\Gamma H$. This is reminiscent of the approach of [18, 21], where convolutions of the characteristic function defining the interface are combined to evolve the interface according to the Willmore flow. However, while the second-order terms appearing in the expansion of the convolution of the characteristic function by the heat kernel provide the exact Willmore velocity, this is no more the case for the distance function, and we need to add some correction terms in order to retrieve the correct Willmore flow.

We also stress that $d^{(n)} - W\delta t$ is not in general a distance function, so that we need to add a redistantiation step to iterate the process. This redistantiation is the counterpart to the thresholding step in the usual convolution-thresholding algorithms for mean curvature [41] or higher-order [18, 21] motions using a characteristic function. In the following, we denote **Redist** this redistantiation step, and will use a fast-marching algorithm to actually compute it. We provide more details about our implementation in section 4.

2D case In 2D, the correction is rather simple, and only involves the curvature H^3, namely:

$$W_{2D} = 2w - \frac{H^3}{2} = \frac{2}{\delta t^2}\left(d - a_2\,\mathcal{G}_\alpha - b_2\,\mathcal{G}_\beta\right) - \frac{H^3}{2}. \tag{21}$$

The curvature H can be computed with different methods, including using the order-1 selection eq. (16), the c correction term introduced above in eq. (12) or direct methods using Δd. In our computations we will use either the order-1 diffusion combination method or the c correction term computed with diffusions of d^2 (c.f. eq. (23)) as for the three-dimensional case. We refer to section 4.3 for a discussion on the efficiency of each method.

As pointed out in remark 1 and discussed in more details in section 4.2, the choice of the numerical scheme used to solve the heat equation has an impact on the developement eq. (5). We present here algorithms using either an implicit Euler or a Crank-Nicolson scheme and choose our combination accordingly.

Our diffusion-redistantiation algorithm for the Willmore flow of a curve in two dimensions finally writes:

Algorithm 3 2D Willmore flow

while $t < t_f$ **do**

Solve $\partial_t \phi - \Delta\phi = 0$ with initial condition $\phi_i = d^{(n)}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}_{\sqrt{2}\delta t}$, $\mathcal{G}_{\delta t/\sqrt{2}}$.

Compute the curvature H.

Compute $\mathcal{D} = \begin{cases} \mathcal{D}_E = -\frac{1}{2}\mathcal{G}_{\sqrt{2}\delta t} + \mathcal{G}_{\delta t/\sqrt{2}} + \frac{H^3}{2}\delta t^2 \\ \mathcal{D}_{\mathrm{CN}} = -\mathcal{G}_{\sqrt{2}\delta t} + 2\mathcal{G}_{\delta t/\sqrt{2}} + \frac{H^3}{2}\delta t^2 \end{cases}$.

Compute the new signed distance function $d^{(n+1)} = \mathbf{Redist}(\mathcal{D})$.

During our simulations, we however preferred to use an algorithm that computes the correction the same as in the 3D case method using the convolution of d^2, presented in the next paragraph, as it proved more robust in some cases. The algorithm reads: Despite two additional convolutions, this algorithm seems to produce better

Algorithm 4 2D Willmore flow with alternative correction

while $t < t_f$ **do**

Solve $\partial_t \phi - \Delta\phi = 0$ with initial condition $\phi_i = d^{(n)}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}_{\sqrt{2}\delta t}$, $\mathcal{G}_{\delta t/\sqrt{2}}$.

Solve $\partial_t \phi - \Delta\phi = 0$ with initial condition $\phi_i = d^{(n)2}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}^{d^2}_{\sqrt{2}\delta t}$, $\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}$.

Compute the curvature H.

Compute $\mathcal{D} = \begin{cases} \mathcal{D}_E = \frac{Hd^{(n)2}}{2} + \left(-\frac{1}{2}\mathcal{G}_{\sqrt{2}\delta t} + \mathcal{G}_{\delta t/\sqrt{2}}\right)\cdot\left(1 - Hd^{(n)}\right) + \left(-\frac{1}{2}\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + \mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right)\cdot\frac{H}{2} \\ \mathcal{D}_{\mathrm{CN}} = \frac{Hd^{(n)2}}{2} + \left(-\mathcal{G}_{\sqrt{2}\delta t} + 2\mathcal{G}_{\delta t/\sqrt{2}}\right)\cdot\left(1 - Hd^{(n)}\right) + \left(-\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + 2\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right)\cdot\frac{H}{2} \end{cases}$

Compute the new signed distance function $d^{(n+1)} = \mathbf{Redist}(\mathcal{D})$.

results for our numerical experiments. It has the advantage of using a correction term which is already of order two. We suspect that multiplying the correction by δt^2 in the first version can have some detrimental numerical effects due to the space discretisation errors.

3D case In 3D, the correction term is more complicated, as it involves both the mean and the gaussian curvatures. More precisely, we have:

$$W_{3D} = w - 2H(H^2 - K) = w + Hc$$

where c is the term introduced in eq. (12). To compute this correction term c, we note from eq. (13) that we need to choose f such that its second derivative does not vanish. To this end, we choose arbitrarily $f(d) = d^2$ and introduce

$$\mathcal{G}_\alpha^{d^2} = d^2 + 2\alpha\delta t\,(1 + d\kappa) + 2(\alpha\delta t)^2\,(c + dw) + \mathcal{O}\left(\delta t^3\right). \tag{22}$$

so that we can extract the correction term from $H\mathcal{G}_\alpha^{d^2}$.

However, using only $\mathcal{G}_\alpha^{d^2}$ to compute c requires to compensate the zero- and first-order terms in eq. (22), which can lead to large errors in particular regarding the $2\delta t\, d\,\kappa$ term. Note that while this latter term disappears as $h \to 0$ on the interface (as $d|_\Gamma = 0$ by definition), the discrete term brings up $\mathcal{O}(h)$ errors which can be large when in the δt order term. To prevent this, we choose to also select directly the second-order term in eq. (22) using a linear combination as introduced in eq. (17). We thus use:

$$a_2\,\mathcal{G}_\alpha^{d^2} + b_2\,\mathcal{G}_\beta^{d^2} = d^2 - 2\delta t^2\,(c + dw) + \mathcal{O}\left(\delta t^3\right). \tag{23}$$

Recalling $W_{3D} = w + Hc$, and combining eq. (17) and eq. (23), we finally obtain:

$$W_{3D} = \frac{1}{\delta t^2}\left[d - \frac{Hd^2}{2} - (a_2\,\mathcal{G}_\alpha + b_2\,\mathcal{G}_\beta)\,(1 - H\,d) - \frac{H}{2}\left(a_2\,\mathcal{G}_\alpha^{d^2} + b_2\,\mathcal{G}_\beta^{d^2}\right)\right] \tag{24}$$

and our algorithm for the three-dimensional Willmore flow writes:

Algorithm 5 3D Willmore flow

while $t < t_f$ **do**

Solve $\partial_t\phi - \Delta\phi = 0$ with initial condition $\phi_i = d^{(n)}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}_{\sqrt{2}\delta t}$, $\mathcal{G}_{\delta t/\sqrt{2}}$.

Solve $\partial_t\phi - \Delta\phi = 0$ with initial condition $\phi_i = d^{(n)2}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}^{d^2}_{\sqrt{2}\delta t}$, $\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}$.

Compute the curvature H.

Compute $\mathcal{D} = \begin{cases} \mathcal{D}_E = \dfrac{Hd^{(n)2}}{2} + \left(-\dfrac{1}{2}\mathcal{G}_{\sqrt{2}\delta t} + \mathcal{G}_{\delta t/\sqrt{2}}\right)\cdot\left(1 - Hd^{(n)}\right) + \left(-\dfrac{1}{2}\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + \mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right)\cdot\dfrac{H}{2} \\ \mathcal{D}_{\mathrm{CN}} = \dfrac{Hd^{(n)2}}{2} + \left(-\mathcal{G}_{\sqrt{2}\delta t} + 2\mathcal{G}_{\delta t/\sqrt{2}}\right)\cdot\left(1 - Hd^{(n)}\right) + \left(-\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + 2\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right)\cdot\dfrac{H}{2} \end{cases}$

Compute the new signed distance function $d^{(n+1)} = \mathbf{Redist}(\mathcal{D})$.

Remark 2. *Note that the flow algorithms actually compute a gradient descent with step δt^2. However, the evolution and diffusion time-steps can easily be decoupled involving the ratio $\frac{\delta t_{flow}}{\delta t^2_{diff}}$, but our numerical tests suggest that keeping this ratio close to 1 leads to the more stable simulations. This decoupling can nevertheless prove useful to implement efficient backtracking methods for the choice of the flow time-step while avoiding unnecessary resolutions of the diffusion equation.*

3. Diffusion-redistancing schemes with volume and area conservation

Since we are interested in computing shapes in $\mathbb{R}^n$, $n = 2$ or $n = 3$, we will in this chapter call *volume* the measure with respect to the Lebesgue mesure of $\mathbb{R}^n$ and *area* the surface measure of hypersurfaces of $\mathbb{R}^n$.

3.1. Review of existing methods

Problems involving area or volume conservation were addressed in the framework of diffusion-redistanciation schemes by [27], and their convergence properties in [33]. In [27], a new algorithm for area preserving flows in two dimensions is introduced by considering normal velocities

$$v_N = \kappa - \bar{\kappa} + S$$

where κ is the curvature, $\bar{\kappa}$ its average on the interface, and S is an additional term which depends on the application. The algorithm relies in 2D on a property linking the mean curvature with the surface area and the genus number, which in case of a connected hypersurface S of genus 0 enclosing an open bounded set Ω, boils down to

$$\bar{\kappa} = \frac{2\pi}{|S|}$$

. Using the divergence theorem, one can then compute $|S|$ with a volume integral as

$$|S| = \int_{\Omega} \Delta d \, dx.$$

In the case where $S = 0$, to ensure an accurate volume conservation, a Newton method is applied to find a real number λ^* close to $\bar{\kappa}$ computed above (which is used as initialisation) to correct the diffusion generated motion by raising or lowering the convolution of the signed distance function (as was introduced the framework of convolution-theresholding schemes by [49]). In the case $n = 3$, while the trick to compute efficiently the mean curvature is not anymore valid, we could still apply the method by computing numerically the mean of the mean curvature.

The level-set community has also developed several techniques to address volume conservation [57, 56, 46] in the modelling of multiphase flows. This corresponds to correct the Hamilton-Jacobi equation of the redistanciation step by a term which ensures that conservation. Namely, the following equation:

$$\partial_\tau \phi = \mathrm{sgn}(\phi_0)(1 - \|\nabla\phi\|) + \lambda f(\phi)$$

is solved with λ computed so that $\int_{\Omega_{ij}} H(\phi)dx$ is conserved on each cell Ω_{ij} of the grid, and f chosen to localise around the interface.

An even more straightforward approach is a post- or preprocessing trick due to Smolianski [55] in the level-set framework, where a raising parameter for the level-set function is explicitly computed to restore the target volume. The trick relies on an expansion of the volume enclosed by a level-set in terms of the height of this level-set:

$$|\{\phi - \delta c < 0\}| := \int_{\{\phi < \delta c\}} 1dx = |\{\phi < 0\}| + \delta c \int_{\{\phi = 0\}} 1d\sigma + o(\delta c)$$

Therefore starting from a reference volume V_0, one can define δc so that $|\{\phi - \delta c < 0\}| = V_0 + o(\delta c)$ by setting:

$$\delta c = \frac{V_0 - |\{\phi < 0\}|}{|\{\phi = 0\}|}.$$

Regarding the conservation of the surface area, one approach, proposed in the context of vesicle or red blood cell simulations is to relax this conservation by introducing an area change energy with a high stiffness. In [9, 10] the authors showed that such an energy could be expressed thanks to $\|\nabla\phi\|$ which records the area change of level-sets of ϕ when ϕ is advected by a divergence free vector field. By taking a high areal tension stiffness, $\|\nabla\phi\|$ is kept close to 1 in a neighbourhood of the interface. In practice, this however induces a high stiffness of the numerical method, which leads to severe time step constraints. Another approach is to enforce exact zero surface divergence of the velocity field which advects the level-set function. This can be done using Lagrange multipliers [13, 28], leading to good area conservation, but at the expense of a bad conditioning of the underlying linear systems to be solve at each iterations, which increases the computational cost.

In this work, we introduce a correction in the spirit of Smolianki's trick to conserve both the volume and the area. From an optimisation point of view, our method can be regarded as a projection method, and the whole contrained Willmore flow as some kind of projected gradient method.

3.2. Raising a level-set function to achieve some given area and enclosed volume

Obviously, we cannot in general fullfil both the area and volume constraints by adding a constant to a level-set function. So let us look at the case where we would add a non-constant function $\delta c : \Omega \to \mathbb{R}$ to an arbitrary level-set function ϕ. Let $\mathcal{H}$ denote a one dimensional smoothed Heaviside function and ζ its derivative. A typical choice is:

$$\mathcal{H}(r) = \begin{cases} \frac{1}{2}\left(r + 1 + \frac{1}{\pi}\sin(\pi r)\right) & |r| < 1 \\ 0 & r \leq -1 \\ 1 & r \geq 1 \end{cases} \text{ and } \zeta(r) = \begin{cases} \frac{1}{2}\left(1 + \cos(\pi r)\right) & |r| < 1 \\ 0 & |r| \geq 1 \end{cases} \tag{25}$$

We can then compute the enclosed volume of the raised level-set:

$$\begin{aligned} |\{\phi - \delta c < 0\}| &= \lim_{\varepsilon \to 0} \int_\Omega \mathcal{H}\left(-\frac{\phi - \delta c}{\varepsilon}\right) dx \\ &= \lim_{\varepsilon \to 0} \int_\Omega \mathcal{H}\left(-\frac{\phi}{\varepsilon}\right) dx + \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \delta c \, dx + \mathcal{O}\left(\delta c^2\right) \end{aligned} \tag{26}$$

Note that if one adds to a level-set function a δc which averages to zero on $\{\phi = 0\}$, the enclosed volume does not change.

Let us have know a look at the area, where we have to take care that $\phi - \delta c$ is not in general a distance function:

$$\begin{aligned} |\{\phi - \delta c = 0\}| &= \lim_{\varepsilon \to 0} \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi - \delta c}{\varepsilon}\right) |\nabla(\phi - \delta c)| dx + \mathcal{O}\left(\delta c^2\right) \\ &= \lim_{\varepsilon \to 0} \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| dx - \int_\Omega \frac{1}{\varepsilon^2}\zeta'\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \delta c \, dx - \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi \cdot \nabla\delta c}{|\nabla\phi|} dx + \mathcal{O}\left(\delta c^2\right) \end{aligned}$$

Integrating by parts in the last term:

$$\begin{aligned} -\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi \cdot \nabla\delta c}{|\nabla\phi|} dx &= \int_\Omega \operatorname{div}\left(\frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi}{|\nabla\phi|}\right) \delta c \, dx \\ &= \int_\Omega \frac{1}{\varepsilon^2}\zeta'\left(\frac{\phi}{\varepsilon}\right) \frac{\nabla\phi \cdot \nabla\phi}{|\nabla\phi|} \delta c \, dx + \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \delta c \, dx \\ &= \int_\Omega \frac{1}{\varepsilon^2}\zeta'\left(\frac{\phi}{\varepsilon}\right) |\nabla\phi| \delta c \, dx + \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \kappa \delta c \, dx \end{aligned}$$

where κ is the total curvature as introduced above. We finally get:

$$|\{\phi - \delta c = 0\}| = |\{\phi = 0\}| + \lim_{\varepsilon \to 0} \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) \kappa \delta c \, dx + \mathcal{O}\left(\delta c^2\right) \tag{27}$$

Note that when ϕ is a distance function $\phi = d$, the volume and area of the perturbed level-set $d - \delta c$ simplify to formulae involving only "surface" integrals :

$$|\{d - \delta c < 0\}| = |\{d < 0\}| + \int_{\{d=0\}} \delta c \, d\sigma + \mathcal{O}\left(\delta c^2\right)$$

which boils down to the Smolianski formula when δc is constant, and

$$|\{d - \delta c = 0\}| = |\{d = 0\}| + \int_{\{d=0\}} \kappa \delta c \, d\sigma + \mathcal{O}\left(\delta c^2\right).$$

Given a target volume V_0 and area A_0, we propose the following fast method to correct a level-set function so that its zero level-set achieves those targets.

1. Raise ϕ by a constant λ to achieve the right enclosed volume:

$$\lambda = \frac{V_0 - |\{\phi < 0\}|}{\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) dx} \tag{28}$$

2. Raise $\tilde{\phi} = \phi + \lambda$ by $\mu(\kappa - \overline{\kappa})$, which does not change the enclosed volume, where the constant μ and $\overline{\kappa}$ are given by:

$$\begin{aligned} \mu &= \frac{A_0 - |\{\tilde{\phi} = 0\}|}{\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\tilde{\phi}}{\varepsilon}\right)\kappa(\kappa - \overline{\kappa})\, dx} = \frac{A_0 - |\{\tilde{\phi} = 0\}|}{(\overline{\kappa^2} - \overline{\kappa}^2)\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\tilde{\phi}}{\varepsilon}\right) dx} \\ \overline{\kappa} &= \frac{\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\tilde{\phi}}{\varepsilon}\right)\kappa}{\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\tilde{\phi}}{\varepsilon}\right)} \end{aligned} \tag{29}$$

Remark 3. *In the practical numerical computations, the "thickness" ε of the interface is kept finite and proportional and close to the mesh size h: $\varepsilon \in [h; 3h]$, following the usual level-set approach.*

Clearly, the method fails if $\overline{\kappa^2} = \overline{\kappa}^2$, which is the case for a circle (resp. sphere). But in that case the perimeter (resp. area) and enclosed area (resp. volume) are linked and one cannot set one independently of the other.

Let us investigate how the former restoring of volume and area change the energy we are minimising. Assuming an energy of the form:

$$\mathcal{E}[\phi] = \int_\Omega E d\nu_\varepsilon$$

where $d\nu_\varepsilon = \frac{1}{\varepsilon}\zeta(\frac{\phi}{\varepsilon})dx$ and, with a gradient which can be written as

$$d\mathcal{E}[\phi](\psi) = \int_\Omega F\psi d\nu_\varepsilon.$$

Proposition 1. *Let us consider a gradient-corrected method to minimise $\mathcal{E}$ iteratively by alternating steepest descent with corrections stages* 1 *and* 2 *described above. Then, at first order in the descent parameter, the energy does not increase during this iteration.*

Proof. A steepest descent method to minimise this energy amounts to change ϕ to $\phi + \psi$ with $\psi = -\rho F$, which decreases the energy, at first order, by $\rho\int_\Omega F^2 d\nu_\varepsilon$.

But this motion changed the enclosed volume and the area of interface. The volume is now $|\{\phi + \psi < 0\}|$ which correspond at first order to a change of $\rho\int_\Omega F d\nu_\varepsilon$. Therefore the λ parameter computed above is equal to:

$$\lambda = \rho\frac{\int_\Omega F d\nu_\varepsilon}{\int_\Omega d\nu_\varepsilon} =: \rho \fint_\Omega F d\nu_\varepsilon$$

where from now on, we will denote with a dashed integral sign the mean value, i.e. the integral divided by the measure of the set (with respect to ν_ε) onto which the integral is taken:

$$\fint_\Omega F d\nu_\varepsilon = \frac{1}{\nu_\varepsilon(\Omega)}\int_\Omega F d\nu_\varepsilon$$

The descent motion and this volume correction change the area by:

$$\rho\int_\Omega F\kappa d\nu_\varepsilon - \lambda\int_\Omega \kappa d\nu_\varepsilon = \rho\int_\Omega F(\kappa - \overline{\kappa})d\nu_\varepsilon.$$

Thus the second step, area correction, computes a correction $\nu(\kappa - \overline{\kappa})$ with:

$$\mu = \rho\frac{\fint_\Omega F(\kappa - \overline{\kappa})d\nu_\varepsilon}{\overline{\kappa^2} - \overline{\kappa}^2}.$$

Those two corrections raise the energy by:

$$\lambda\int_\Omega F d\nu_\varepsilon + \mu\int_\Omega F(H - \overline{H})d\nu_\varepsilon = \rho\nu_\varepsilon(\Omega)\left[\left(\fint_\Omega F d\nu_\varepsilon\right)^2 + \frac{\left(\fint_\Omega F(H - \overline{H})d\nu_\varepsilon\right)^2}{\overline{H^2} - \overline{H}^2}\right].$$

Finally, the two successive corrections would not raise the energy more than it has been decreased by the descent method provided that the following inequality holds:

$$\left(⨍_\Omega F d\nu_\varepsilon\right)^2 + \frac{\left(⨍_\Omega F(H-\overline{H})d\nu_\varepsilon\right)^2}{\overline{H^2}-\overline{H}^2} \leq ⨍_\Omega F^2 d\nu_\varepsilon.$$

This is the object of the following lemma. We therefore justified that our descent-correction method does not increase energy, at least in this first order analysis. □

Lemma 1. *Let (X,Σ,μ) be a measure space, and A a measurable subset of X with $\mu(A)<+\infty$. Let $f,g\in L^2(A)$ with $⨍_A g d\mu = 0$. Then:*

$$\left(⨍_A f d\mu\right)^2 ⨍_A g^2 d\mu + \left(⨍_A fg d\mu\right)^2 \leq ⨍_A f^2 d\mu ⨍_A g^2 d\mu \tag{30}$$

Proof. This is an easy extension of the Cauchy-Schwarz inequality. Indeed applying the latter for $f - ⨍_A f d\mu$ and g we have:

$$\left(⨍_A \left(f - ⨍_A f dx\right) g d\mu\right)^2 \leq ⨍_A \left(f - ⨍_A f d\mu\right)^2 d\mu ⨍_A g^2 d\mu$$

Since g is of zero mean on A, we have for the left hand side of the former inequality,

$$⨍_A \left(f - ⨍_A f d\mu\right) g d\mu = ⨍_A fg d\mu$$

while concerning the right hand side we observe that:

$$⨍_A \left(f - ⨍_A f d\mu\right)^2 d\mu + \left(⨍_A f d\mu\right)^2 = ⨍_A f^2 d\mu$$

which leads to eq. (30). □

Remark 4. *In order to prevent undesired effects from outside of the neighbourhood of the interfacen we can slightly alter the method during the step recovering the area. We rescale with the function $\mu(\kappa-\bar{\kappa})\frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)$ rather than $\mu(\kappa-\bar{\kappa})$. This amounts to choose $\delta c = (\kappa-\bar{\kappa})\frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)$ in eq. (27) and to redefine $\bar{\kappa}$ (resp. $\bar{H}$ as:*

$$\bar{\kappa} = \frac{\int_\Omega \frac{1}{\varepsilon^2}\zeta^2\left(\frac{\phi}{\varepsilon}\right)\kappa}{\int_\Omega \frac{1}{\varepsilon^2}\zeta^2\left(\frac{\phi}{\varepsilon}\right)}$$

In our simulations, we fix the value of the rescaling term $\mu(H-\bar{H})$ outside of the neighbourhood of the interface in order to have a neutral effect.

Remark 5. *An alternative option is to rescale the volume and the area during the same step by solving the following system:*

$$\begin{pmatrix} \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right) & \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)H \\ \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)H & \int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)H^2 \end{pmatrix} \cdot \begin{pmatrix}\lambda\\ \mu\end{pmatrix} = \begin{pmatrix} V_0 - |\{\phi<0\}| \\ A_0 - |\{\phi=0\}| \end{pmatrix}$$

The first equation (resp. the second) corresponds to the volume expansion eq. (26) (the area expansion eq. (27)) with $\delta c = \lambda + \mu H$. It is important to note that the constant part $\mu\bar{H}$ of the area term is absorbed inside λ. Both versions are equivalent in theory and yield similar numerical results.

3.3. Willmore flow with volume and area conservation constraints

Combining the algorithms algorithms 3 and 5 for the Willmore flow derived in the previous section and the above method to recover volume and area, we can obtain an algorithm to compute the Willmore flow in both dimension 2 and 3 with conservation of volume and area.

The principle is to alternate one diffusion step moving according the Willmore flow and recovering the volume and area constraints before the redistanciation. Using proposition 1 we can ensure that the energy has globally decreased at the end of one whole step of the method.

The algorithms in 2D and 3D write:

Algorithm 6 2D Willmore flow with constant volume and area

while $t < t_f$ **do**

Solve $\partial_t \phi - \Delta \phi = 0$ with initial condition $\phi_i = d^{(n)}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}_{\sqrt{2}\delta t}$, $\mathcal{G}_{\delta t/\sqrt{2}}$.

Compute the curvature H.

Compute $\mathcal{D} = \begin{cases} \mathcal{D}_E = -\dfrac{1}{2}\mathcal{G}_{\sqrt{2}\delta t} + \mathcal{G}_{\delta t/\sqrt{2}} + \dfrac{H^3}{2}\delta t^2 \\ \mathcal{D}_{\mathrm{CN}} = -\mathcal{G}_{\sqrt{2}\delta t} + 2\mathcal{G}_{\delta t/\sqrt{2}} + \dfrac{H^3}{2}\delta t^2 \end{cases}$

Compute $H(\mathcal{D})$ and its mean $\overline{H}(\mathcal{D})$.

Compute λ:

$$\lambda = \frac{V_0 - |\{\mathcal{D} < 0\}|}{|\{\mathcal{D} = 0\}|}.$$

Compute μ:

$$\mu = \frac{A_0 - |\{\mathcal{D} = 0\}|}{\int_\Omega H(H - \overline{H})\frac{1}{\varepsilon}\zeta\left(\frac{\mathcal{D}}{\varepsilon}\right) d\sigma}.$$

Construct the new signed distance function $d^{(n+1)} = \mathbf{Redist}\left(\mathcal{D} + \lambda + \mu(\mathcal{H} - \overline{\mathcal{H}})\right)$.

Algorithm 7 3D Willmore flow with volume and area constraints

while $t < t_f$ **do**

Solve $\partial_t \phi - \Delta \phi = 0$ with initial condition $\phi_i = d^{(n)}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}_{\sqrt{2}\delta t}$, $\mathcal{G}_{\delta t/\sqrt{2}}$.

Solve $\partial_t \phi - \Delta \phi = 0$ with initial condition $\phi_i = d^{(n)2}$ for times $t = \sqrt{2}\delta t$ and $t = \delta t/\sqrt{2} \to \mathcal{G}^{d^2}_{\sqrt{2}\delta t}$, $\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}$.

Compute the curvature H.

Compute $\mathcal{D} = \begin{cases} \mathcal{D}_E = \dfrac{Hd^{(n)2}}{2} + \left(-\dfrac{1}{2}\mathcal{G}_{\sqrt{2}\delta t} + \mathcal{G}_{\delta t/\sqrt{2}}\right) \cdot \left(1 - Hd^{(n)}\right) + \left(-\dfrac{1}{2}\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + \mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right) \cdot \dfrac{H}{2} \\ \mathcal{D}_{\mathrm{CN}} = \dfrac{Hd^{(n)2}}{2} + \left(-\mathcal{G}_{\sqrt{2}\delta t} + 2\mathcal{G}_{\delta t/\sqrt{2}}\right) \cdot \left(1 - Hd^{(n)}\right) + \left(-\mathcal{G}^{d^2}_{\sqrt{2}\delta t} + 2\mathcal{G}^{d^2}_{\delta t/\sqrt{2}}\right) \cdot \dfrac{H}{2} \end{cases}$

Compute $H(\mathcal{D})$ and its mean $\overline{H}(\mathcal{D})$.

Compute λ:

$$\lambda = \frac{V_0 - |\{\mathcal{D} < 0\}|}{|\{\mathcal{D} = 0\}|}.$$

Compute μ:

$$\mu = \frac{A_0 - |\{\mathcal{D} = 0\}|}{\int_\Omega H(H - \overline{H})\frac{1}{\varepsilon}\zeta\left(\frac{\mathcal{D}}{\varepsilon}\right) d\sigma}.$$

Construct the new signed distance function $d^{(n+1)} = \mathbf{Redist}\left(\mathcal{D} + \lambda + \mu(H - \overline{H})\right)$.

4. Numerical method and practical implementation

4.1. Finite Element library

The equations and integrals introduced above are solved or evaluated within a finite-element framework, using the FEEL++–finite-element C++ library [45, 44], and in particular the *LevelSet* framework from the FEEL++ toolboxes [42], which features a comprehensive and seamless parallel set of tools for this kind of surface tracking methods.

More precisely, we use a continuous Galerkin variationnal approach, and discretise the resulting equations in space with Lagrange polynomials. Introducing $\mathcal{T}_h \equiv \{K_e, 1 \leq e \leq N_{elt}\}$ a compatible tessellation of the computational domain Ω, and the corresponding discrete – unstructured – mesh $\Omega_h = \bigcup_{e=1}^{N_{elt}} K_e$, we define $\mathcal{P}_h^k \equiv \mathcal{P}_h^k(\Omega_h)$ as the finite-element space on Ω_h spanned by Lagrange polynomials of order k.

4.2. Solving the diffusion equation

From a numerical perspective, our Willmore flow algorithms 3 and 5 mainly involve solving the diffusion equation $\partial_t \phi - \Delta\phi = 0$ with appropriate initial and boundary conditions. To this end, we use the $\mathcal{P}_h^k$ finite-element space discretisation introduced above, and a – second order inconditionnally stable – Crank-Nicolson scheme for the time discretisation. The corresponding discrete variational problem then writes:

$$\begin{aligned} &\textit{Find } \phi^{(n+1)} \in \mathcal{P}_h^k \textit{ s.t. } \forall \psi \in \mathcal{P}_h^k, \\ &\int_{\Omega_h} \left(\frac{\phi^{(n+1)}}{\delta t}\,\psi + \frac{1}{2}\,\nabla\phi^{(n+1)} \cdot \nabla\psi \right) = \int_{\Omega_h} \left(\frac{\phi^{(n)}}{\delta t}\,\psi - \frac{1}{2}\,\nabla\phi^{(n)} \cdot \nabla\psi \right) + \int_{\partial\Omega_h} \left(N \cdot \nabla\phi^{(n)} \right) \psi \end{aligned} \tag{31}$$

where the superscript indices denote the time iterations. Note that the last term in eq. (31) comes from the integration by parts of the diffusive terms, and is somehow similar to an explicit discrete Neumann boundary condition preserving the normal gradient of ϕ, namely $N \cdot \nabla\phi^{(n+1)} = N \cdot \nabla\phi^{(n)}$ with N the exterior normal at the boundary of Ω_h. In practice, the best choice of boundary conditions for our distance diffusion problem is an open question, since no natural condition emerges from our analysis, which focuses on the neighborhood of the interface Γ, far from the domain boundary.

As pointed out in remark 1, we would also like to use in some situations an Euler scheme, which is low order and diffusive, which may be more adapted in this context than a dispersive scheme. In that case one should directly use the discrete expansion, which differs at order two (since the scheme is order one):

$$\phi^{n+1} - \delta t \Delta \phi^{n+1} = \phi^n$$

indeed gives

$$\phi^{n+1} = (\mathrm{id} - \delta t \Delta)^{-1}(\phi^n) = \phi^n + \delta t \Delta \phi^n + \delta t^2 \Delta^2 \phi^n + o(\delta t^2) \tag{32}$$

which differs from of (5) by a factor 2 in the second order term. The corresponding discrete variational problem then simplifies to:

$$\begin{aligned} &\textit{Find } \phi^{(n+1)} \in \mathcal{P}_h^k \textit{ s.t. } \forall \psi \in \mathcal{P}_h^k, \\ &\int_{\Omega_h} \left(\frac{\phi^{(n+1)}}{\delta t}\,\psi + \nabla\phi^{(n+1)} \cdot \nabla\psi \right) = \int_{\Omega_h} \frac{\phi^{(n)}}{\delta t}\,\psi + \int_{\partial\Omega_h} \left(N \cdot \nabla\phi^{(n)} \right) \psi \end{aligned} \tag{33}$$

In our numerical results, both schemes give qualitatively the same equilibrium shapes. However the Crank-Nicolson scheme seems more accurate, while the Euler scheme was preferred in dimension 3 where its diffusive behavior brings more stability and smoothness of the interface.

4.3. Computation of the curvature

Our algorithms for the Willmore flow in both 2D algorithm 3 and 3D algorithm 5 require the computation of the curvature of the surface Γ in the correction term. We first present the direct methods currently used in FEEL++. We then introduce our diffusion methods to compute the curvature, which are essentially the classical convolution/thresholding algorithms of order 1 and 2. Using the direct methods as references, we discuss the efficiency of our method and focus on the choice of the time step selection for the diffusion steps.

4.3.1. Direct methods

From a usual level-set perspective, the curvature of Γ can be computed directly using the divergence of the level-set function Δd. For low-order discretisations however, this requires the use of specific strategies, as standard finite-element derivation decreases the discretisation polynomial order by 1, which forbids derivations of degrees higher than the polynomial order. To circumvent this issue, we use a classical Galerkin projection in $L^2(\Omega_h)$ to maintain the element in $\mathcal{P}_h^1$. Namely the computation of the gradient ∇d is as following:

$$\begin{aligned} &\textit{Find } g \in \mathcal{P}_h^1 \textit{ s.t. } \forall v \in \mathcal{P}_h^1, \\ &\int_{\Omega_h} g \cdot v = \int_{\Omega_h} \nabla d \cdot v \end{aligned}$$

We can then obtain the curvature with two successive derivation/projection steps by computing the divergence of the gradient of the distance function:

$$\begin{aligned} &(1) : \textit{Find } g \in \mathcal{P}_h^1 \textit{ s.t. } \forall v \in \mathcal{P}_h^1, \\ &\int_{\Omega_h} g \cdot v = \int_{\Omega_h} \nabla d \cdot v \\ &(2) : \textit{Find } H \in \mathcal{P}_h^1 \textit{ s.t. } \forall w \in \mathcal{P}_h^1, \\ &\int_{\Omega_h} Hw = \int_{\Omega_h} \operatorname{div} g \; w \end{aligned} \tag{34}$$

However the L^2 projection fills the missing information of the lower order elements with noisy values, so that one usually resort to a smoothed L^2 projection method, where a small diffusion term $-\eta\Delta H$ is added:

$$\begin{aligned} &(1) : \textit{Find } g \in \mathcal{P}_h^1 \textit{ s.t. } \forall v \in \mathcal{P}_h^1, \\ &\int_{\Omega_h} g \cdot v = \int_{\Omega_h} \nabla d \cdot v \\ &(2) : \textit{Find } H \in \mathcal{P}_h^1 \textit{ s.t. } \forall w \in \mathcal{P}_h^1, \\ &\int_{\Omega_h} Hw + \eta \int_{\Omega_h} \nabla H \cdot \nabla w - \eta \int_{\partial\Omega_h} \nabla H \cdot Nw = \int_{\Omega_h} \operatorname{div} g \; w. \end{aligned} \tag{35}$$

where the smoothing coefficient η must be adjusted depending on the situation. We will refer to eq. (34) as the L^2 projection and to eq. (35) as the Smoothed projection method.

4.3.2. Using the diffusion of the signed distance function

As mentioned above, we can also retrieve the curvature of the surface represented with the level-set using the diffusion of the signed distance function. Recalling the analytical expansion of the diffusion solution eq. (6), we observe that the curvature of the surface shows up in the first order term in δt, and can thus be retrieved as:

$$H = \frac{\mathcal{G}_{\delta t} - d}{(n-1)\delta t} + \mathcal{O}(\delta t). \tag{36}$$

As explained in section 2.3, at the expense of an additional resolution of the diffusion equation, we can improve the order of accuracy and compute the combination eq. (16) which eliminates the second order term. We can then compute H at order δt^2 as:

$$H = \frac{a_1\mathcal{G}_\alpha + b_1\mathcal{G}_\beta - \frac{3\sqrt{2}}{2}d}{(n-1)\delta t} + \mathcal{O}(\delta t^2) \tag{37}$$

In the following we shall refer to the $\mathcal{O}(\delta t)$ method eq. (36) as the Order 1 diffusion and the $\mathcal{O}(\delta t^2)$ method eq. (37) as the Order 2 diffusion. Keep however in mind that these approximation orders refer to the auxiliary diffusion time-step and not to the usual space discretisation order.

Before comparing these diffusion methods to the direct ones mentioned above 4.3.1, we shall first study in more details the role of the diffusion time-step. Figure 1 shows the evolution of the L^2 error of the curvature computed with both diffusion methods as a function of the diffusion time-step. The numerical test was performed on a two-dimensional circular surface of curvature 1 with a mesh size $h \approx 0.02$, and the error was computed as

$$\mathrm{err}(H;\Gamma) = \frac{\int_\Omega (H - H_{th})^2 \ \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)}{\int_\Omega \frac{1}{\varepsilon}\zeta\left(\frac{\phi}{\varepsilon}\right)}. \tag{38}$$

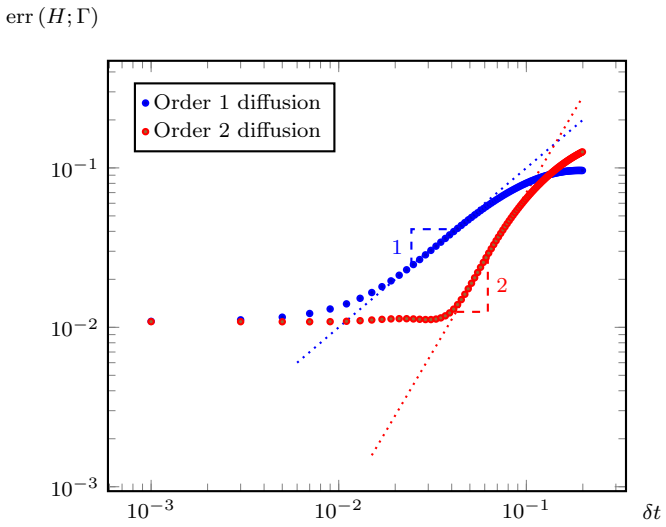


Fig 1: L^2 error on curvature estimates using the Order 1 eq. (36) and Order 2 eq. (37) diffusion methods as a function of the diffusion time-step δt. We observe the expected convergence orders as the time-step decreases, and a saturation of the error for low time-steps which is related to space discretisation error terms.

As expected, we observe convergence of both methods at order $\mathcal{O}(\delta t)$ and $\mathcal{O}(\delta t^2)$ respectively, as well as a saturation of the error for small time-steps, which is due to the spatial discretisation errors. We also note that the Order 2 diffusion method features an optimal error for larger time-steps, which can prove useful when using the same time-step for the flow and the diffusions (c.f. remark 2).

These numerical observations, as well as the analytic expansion of $\mathcal{G}_{\delta t}$ also suggest to choose the time-step following the heuristic strategy:

$$\tilde{H}\delta t \approx h \tag{39}$$

with $\tilde{H}$ an *a priori* estimate of the typical curvature of the surface. From a numerical point of view, this corresponds to having the diffusion move the $d = 0$ level-set a few mesh elements. This heuristic can also be adapted to the case of geometrical flow simulations, and naturally formulates an adaptive time-stepping method, as presented in section 4.5.

4.3.3. Comparaison of the different methods

We now turn to the comparison of the "direct" and "diffusion" curvature methods presented above. We again use a fixed two-dimensional circular interface of radius 1, and vary the characteristic mesh size h used to compute the L^2 error on the curvature estimate along the surface as a function of h for the four methods. We set the smoothing parameters $\eta = 0.03h$ and the diffusion time-steps as $\delta t = 0.7h$. The resulting errors are plotted in log-log scale in fig. 2.

We find that the Order 1 and Order 2 diffusion methods behave similarly, and compare accurately with the Smoothed projection method. These three methods give $\mathcal{O}(h)$ convergence, which is satisfactory for our low-order discretisation. As anticipated, the L^2 projection method does not seem to converge.

In our simulations, we shall therefore use the Order 2 diffusion method, as the two diffusion equation resolutions for $\mathcal{G}_\alpha$ and $\mathcal{G}_\beta$ are needed anyway for the computation of the Willmore flow, and this method provides the best curvature estimate at no additional cost.

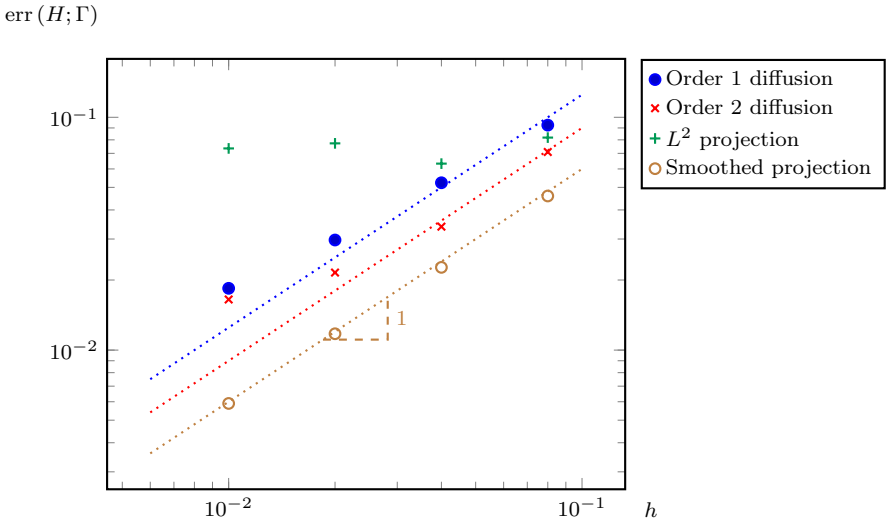


Fig 2: Evolution of the L^2 error as a function of the mesh size for the different methods. The L^2 projection shows no evident convergence while the Order 1 diffusion, the Order 2 diffusion and the Smoothed projection methods yield linear convergence.

4.4. Level-set redistanciation

The redistanciation step is a crucial aspect of our algorithm, as it allows the iteration to proceed while preserving the correct expansions of the diffusion solutions close to the interface. This step can moreover become the main bottleneck of the whole algorithm in terms of both efficiency and stability if not processed carefully. From a general point of view, performing the **Redist** step in algorithms 3 and 5 amounts to computing the distance to an interface Γ located by the 0-level of some function. Stated differently, we need to solve the eikonal equation:

$$|\nabla\phi| = 1, \quad \phi(\Gamma) = 0 \tag{40}$$

which is a challenging boundary-value non-linear equation which has received much attention since the work of Sethian [53] who proposed an iterative upwind *fast-marching* algorithm to solve the eikonal equation starting from the interface and propagating the information outward.

In our finite-element framework, we use a parallel fast-marching algorithm inspired from [62] but adapted to arbitrary – possibly unstructured – meshes. The local – element-wise – eikonal equations are solved with a QR decomposition, and the fast-marching alternates between local-domain solves and ghost inter-domain updates until global convergence is obtained.

In order to ensure good stability properties of our method, and in particular to prevent the fast-marching method from introducing spurious motion of the interface, the initialisation of the algorithm near the implicitly defined interface is crucial. To this end, we rescale the level-set function ϕ by $\frac{1}{|\nabla\phi|}$ on the elements which intersect the interface, which exactly solves the local eikonal equation in the linear – $\mathcal{P}_h^1$ – case, before applying the fast-marching algorithm both inward ($\phi < 0$) and outward ($\phi > 0$) starting from the values encompassing the interface.

4.5. Adaptive time-stepping

As can be seen from eq. (17), given some δt in the resolution of the heat equations, our diffusion-redistanciation schemes actually solves the corresponding Willmore flow with time-step δt^2, namely

$$d^{(n+1)} = d^{(n)} - W\delta t^2.$$

The flow time-step is therefore not constrained by any stability issue, but controls the accuracy of the solution, since the expansions performed in section 2.1 are valid up to terms of order $\mathcal{O}(\delta t^3)$.

The $\mathcal{O}(h)$ errors pertaining to the computation of $\phi^{(n+1)}$ in eq. (31) using $\mathcal{P}_h^1$ Lagrange elements and the expansion (6) however suggest to use a time-step like

$$\delta t \sim \frac{h}{\kappa}$$

in the heat equation. This obviously requires the *a priori* computation of the interface curvature, which is not desirable in our framework. Instead, we therefore use an explicit adaptive time-stepping strategy by setting

$$\delta t^{(n+1)} = \min_{\Gamma}\left(\frac{h}{\kappa^{(n)}}\right) \tag{41}$$

where $\kappa^{(n)}$ is the curvature at the previous iteration and h is a measure of the mesh size, here considered for elements crossed by the interface – e.g.the minimal diameter of the elements crossed by the $\{\phi^{(n)} = 0\}$ level-set.

5. Numerical illustrations

In this section, we illustrate the efficiency our methods by applying them to two classical applications. First, we apply our 3D algorithm for the Willmore flow algorithm 5 to the Willmore problem. Then we apply our algorithm for Willmore flow with area and volume conservation algorithm 7 to the computation of an equilibrium shape of a red blood cell.

5.1. Willmore problem

The *Willmore problem* refers to the study of the minimisers of the Willmore energy function $\mathcal{E}_W$. The case of compact surfaces of genus 0 is trivial as the minimisers are the spheres – recall that the Willmore energy being scale invariant, all the spheres have the same energy.

In the case of compact surfaces of genus 1, the Clifford torus, the torus with a ratio $\sqrt{2}$ between its radii, minimises the Willmore energy. The proof of this conjecture made by Willmore [60] in 1965 has been established recently in [39]. The conjectures for higher genus order [23] are still to be proven.

In the following, we present 2D and 3D simulations of such simple surfaces which evolve according to the Willmore flow using our diffusion-redistanciation algorithms. The existence of analytic minimisers will then provide solid means to evaluate our numerical approach.

5.1.1. 2D Willmore flow of a circle

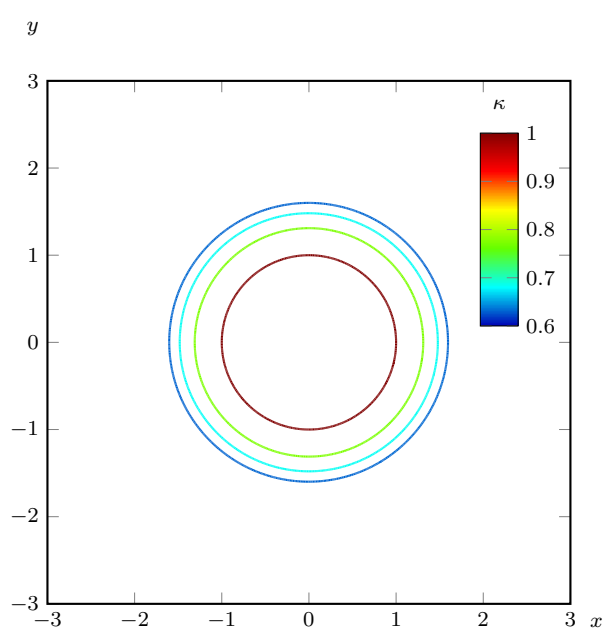
To assess the accuracy of our diffusion-redistanciation scheme, we first simulate an initial two-dimensional circular interface evolving according to the Willmore flow, which can also be computed analytically. As it evolves with the flow, the interface sould remains circular, and increase it radius r as

$$r(t) = (r_0 + 2\,t)^{1/4} \tag{42}$$

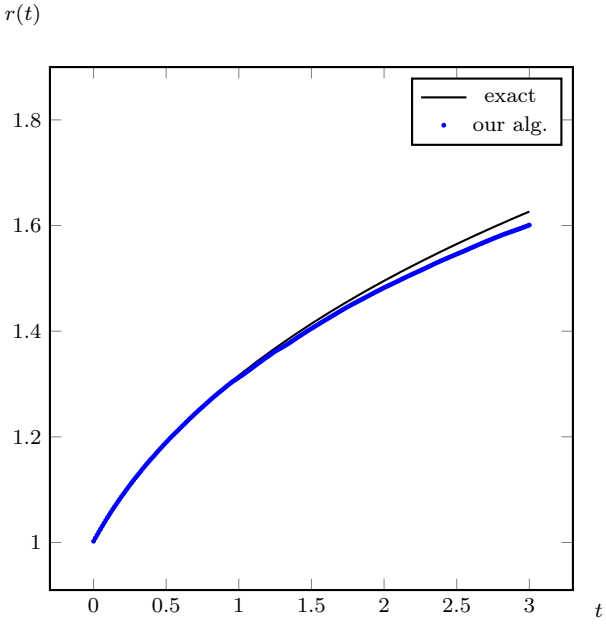
with r_0 the initial radius. Note that this law, which comes from the analytic flow equation $\frac{dr}{dt} = \frac{H^3}{2} = 12r^3$ provides a good test of our algorithm, since the right hand side is actually obtained through a delicate compensation between the second-order term w (c.f. (10)) in the diffusion of d and the correction term c obtained with the diffusion of d^2 (c.f. (12)).

Our simulation was run with $r_0 = 1$ in a square domain of side length 6 with an unstructured mesh of typical size $h \approx 0.02$ and the adaptive time-step strategy presented in section 4.5.

Figure 3 shows some snapshots of the simulation and the evolution of the circle radius. It illustrates the numerical accuracy of our scheme, which succeeds in preserving the symmetry of the shape as it evolves, and compares quantitatively with the exact solution.



(a) Snapshots of the interface at times $t = 0$, 1, 2 and 3.



(b) Evolution of the radius of the circle: comparison with the exact solution.

Fig 3: 2D Willmore flow of a circle.

5.1.2. 3D Willmore flow of a torus

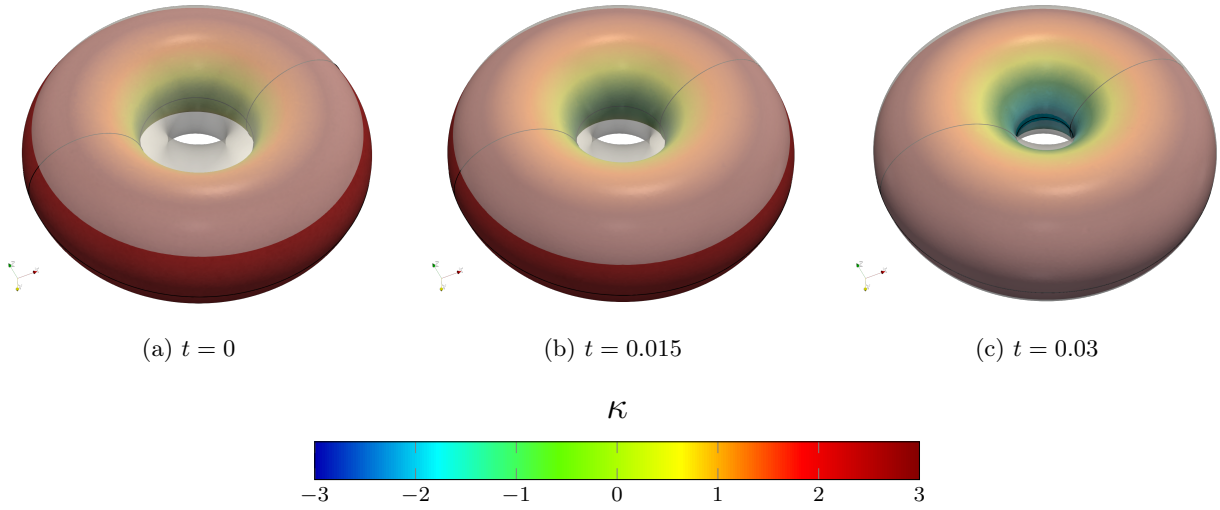
The convergence of a torus toward Clifford torus, whose ratio between major and minor radii equals $\sqrt{2}$, is a good test for our numerical approach, as it challenges both the accuracy of the numerical flow and the stability of the algorithm as it approaches the minimising surface, which features a rather small hole, which can easily be filled by an inaccurate flow to minimise the energy further, as the Willmore energy of any sphere – $4\,\pi$ – falls below the minimal one for torii, which is $2\,\pi^2$.

Simulations have been performed using a phase-field model in [6] but the results are only qualitative. To the extend of our knowledge, there is no other published numerical work to compare with our results.

Figures 4 and 5 show the results of our three-dimensional Willmore flow algorithm 5 starting from a torus with major $a = 2$ and minor $b = 0.5$ radii. The simulation was run in a cuboid with lengths $6 \times 4 \times 6$ and an unstructured mesh of typical size $h \approx 0.02$. We observe that the surface flows to the expected Clifford torus, and that the estimated resulting energy and radii ratio are in good agreement with the expected values for such a torus.

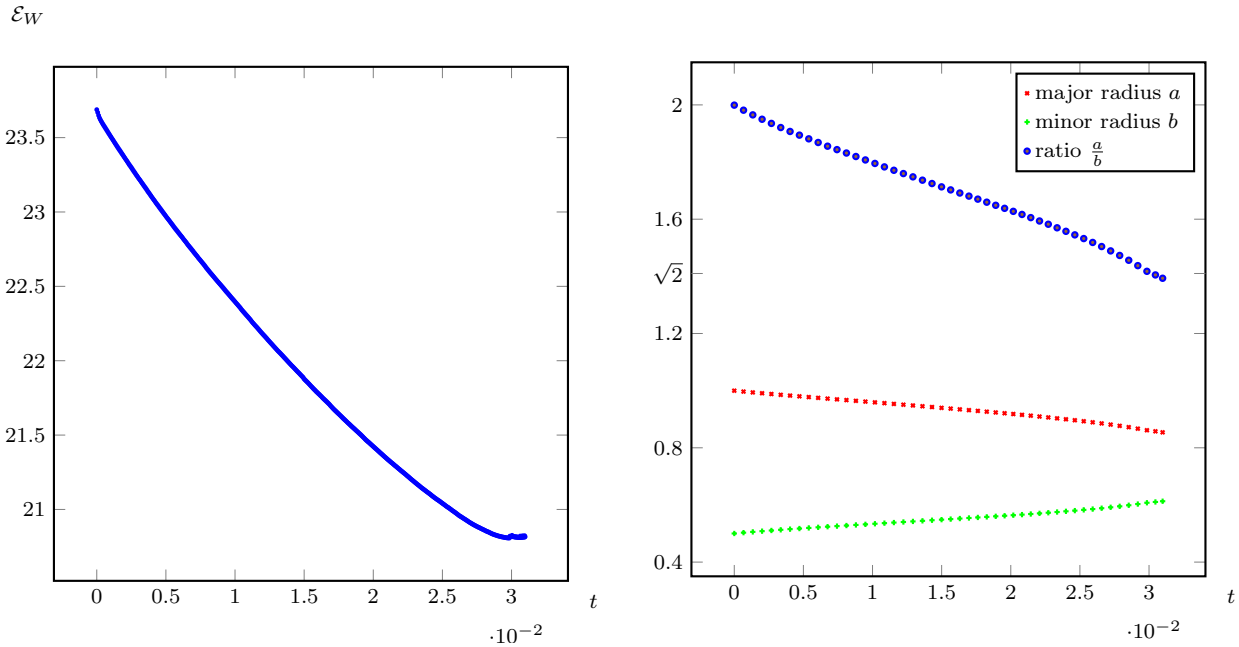
5.2. Vesicles equilibrium shapes

We now consider the simulation of vesicles, which provide the most popular and simple model for capsules, closed thin shell or cells, like human red blood cells. From a physical point of view, the model consists in an elastic thin shell of fixed area enclosing some incompressible fluid, so that the inner volume is also fixed. As such, the vesicle only deforms through bending of the shell, and is thus controlled by a Willmore-like – or



(a) $t = 0$ (b) $t = 0.015$ (c) $t = 0.03$

Fig 4: 3D Willmore flow of a torus toward the Clifford torus. The evolving surface is colored with the value of the curvature while the Clifford torus is represented in semi-transparent white.



(a) Evolution of the Willmore energy $\mathcal{E}_W = \int_\Gamma H^2$.

(b) Evolution of the torus major and minor radii.

Fig 5: Willmore energy and geometrical quantities of the torus. We observe that the flow decreases the energy to about 20.8, which is within 5% from the theoretical Clifford torus energy $2\pi^2 \approx 19.74$. We also observe that at equilibrium, the ratio between the radii is indeed about $\sqrt{2}$.

Canham-Helfrich energy [22, 7] (see e.g. [51] for a presentation of the model in a biological context), which reduces to the standard Willmore energy when the membrane has no spontaneous curvature, which we assume in this work.

In the following, we shall use our diffusion-redistanciation algorithms with the constraints of constant volume and area of the evolving surface to compute the 2D and 3D equilibrium shapes of vesicles with various reduced

volumes. The resulting equilibrium shapes are compared to the ones from the literature when available, and assessed from energetical and stability points of view to illustrate the good properties of our numerical method.

5.2.1. Computation of the equilibrium shapes of 2D vesicles

We first consider the two-dimensional case, and compute the Willmore flow under the constraint of constant inner volume and surface area of an initial ellipse with semi-minor and -major axes adjusted according to a chosen reduced volume

$$\nu = \frac{4\pi V}{A^2} = \frac{V}{\pi \left[\frac{A}{2\pi}\right]^2} \tag{43}$$

The reduced volume represents the ratio between the volume of the ellipse and the volume of a circle of same area. Note that the conservation of both the volume and area of the vesicle naturally entails that the reduced volume is also preserved.

The simulations were run in a square domain $[-4, 4]^2$ with a structured triangle mesh with mesh size $h = 0.04$ and adaptive time-step strategy. We illustrate the convergence of our method to an equilibrium shape in fig. 6 and compare our results with the ones obtained using a lattice-Boltzmann method in [26] in fig. 7. We observe that the shapes computed with our algorithm are in excellent agreement with the ones obtained from direct lattice-Boltzmann simulations. We also illustrate the good volume and area conservation properties of our algorithm in fig. 8. The evolution of the Willmore energy along the flow is shown in fig. 9.

5.2.2. Computation of vesicles equilibrium shapes in 3D

The landscape of vesicle equilibrium shapes is much more rich in three dimensions, as the Gauss curvature comes into play and allows for multiple local minima for shapes with the same topological class and invariants. Axisymmetric equilibrium shapes have been mapped out in a phase diagram in [52], and some non-axisymmetric shapes have been computed in [25, 14, 15, 6], but the general phase diagram for Willmore energy minimisers is still an open question.

In our case, we consider the evolution of ellipsoids under the constant volume and area Willmore flow. As such, we expect to recover the prolate and oblate axisymmetric shapes of zero genus and spontaneous curvature referenced in [52] depending on the initial – conserved – reduced volume,

$$\nu = \frac{6\sqrt{\pi}V}{A^{\frac{3}{2}}} = \frac{V}{\frac{4}{3}\pi \left[\frac{A}{4\pi}\right]^{\frac{3}{2}}}. \tag{44}$$

As shown in [52], the prolate and oblate shapes are both local minimisers for $\nu \approx 0.51$. While the prolate shapes exist for all possible ν, the oblate ones selfintersect below $\nu \approx 0.51$, and the stomatocyte shapes become the only non-prolate feasible minimisers. In this work, we restrict ourselves to $\nu \geq 0.6$, and study both the oblate and prolate cases, as our algorithm appears able to capture the local minimising shapes in a stable way.

Fewer quantitative works are available for comparison in this three dimension case. We can mention [4, 14, 20, 15] where a mix of qualitative and quantitative results are presented. In [4, 14], a phase field method was developed and qualitative results were given. In [15] an adaptive version of the phase field model was proposed with some quantitative results in terms of energy, which are hard to compare with theoretical known values of Seifert [52]. The approach of Feng and Klug [20] is based on surface finite element and is more quantitative in its results as far as the energy is concerned.

Oblate case The oblate case simulations were run in a cuboid with lengths $3.6 \times 6.4 \times 6.4$ and mesh size $h \approx 0.04$. The initial level-set functions were taken as signed distance functions to oblate ellipsoids with semi-minor axis a and semi-major axes $b = c$ adjusted to get a volume $\frac{4\pi}{3}$ for all the simulations and a corresponding reduced volume $\nu = 0.6$, 0.65, 0.7, 0.8 or 0.9.

Figure 10 shows the equilibrium shapes obtain with our constrained diffusion-redistanciation algorithm 7 for the different reduced volumes. We also show two-dimensional cuts of these equilibrium shapes in fig. 11 to illustrate the good symmetry conservation properties of our algorithm. Figures 12 and 13 show the evolution of the Willmore energy, and the area and enclosed volume of the surface as it evolves.

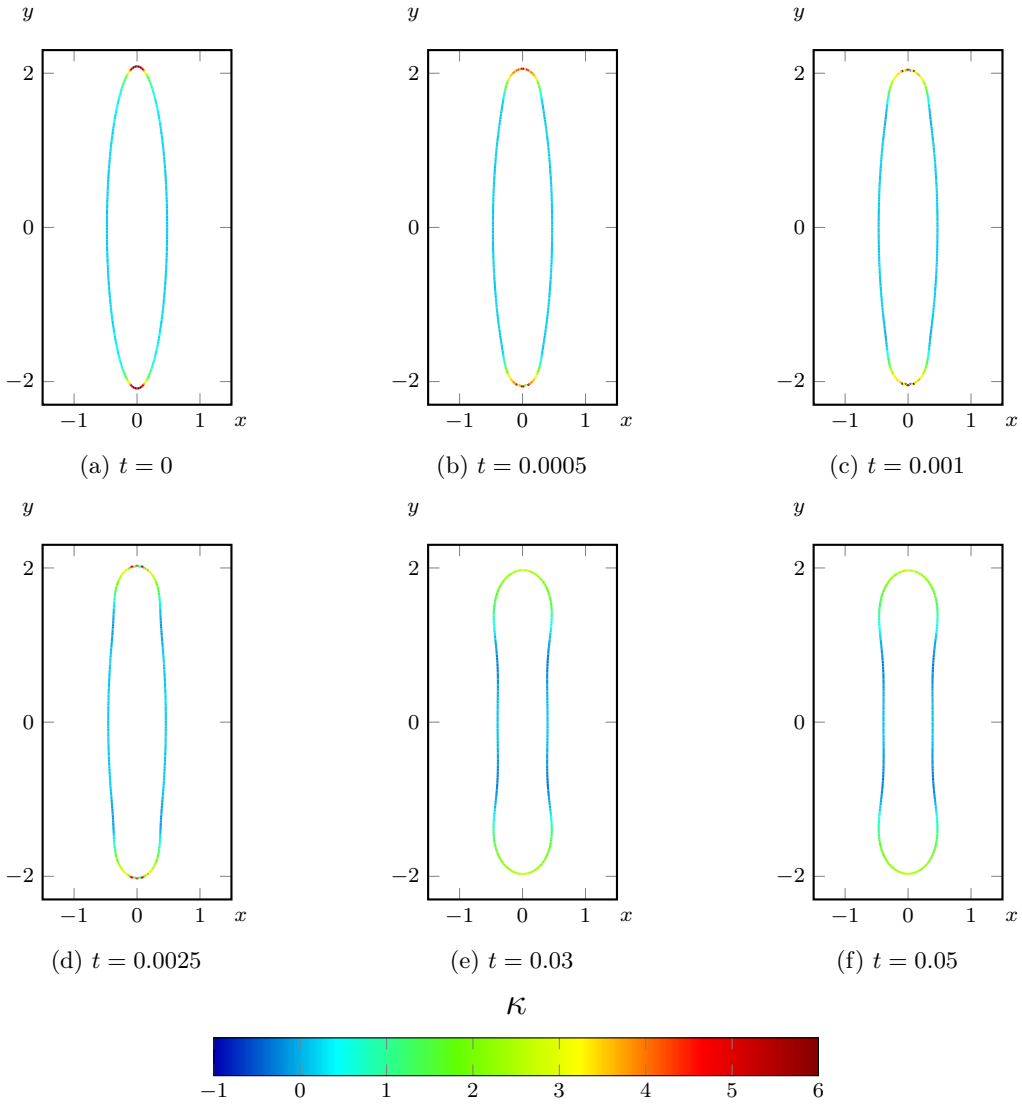


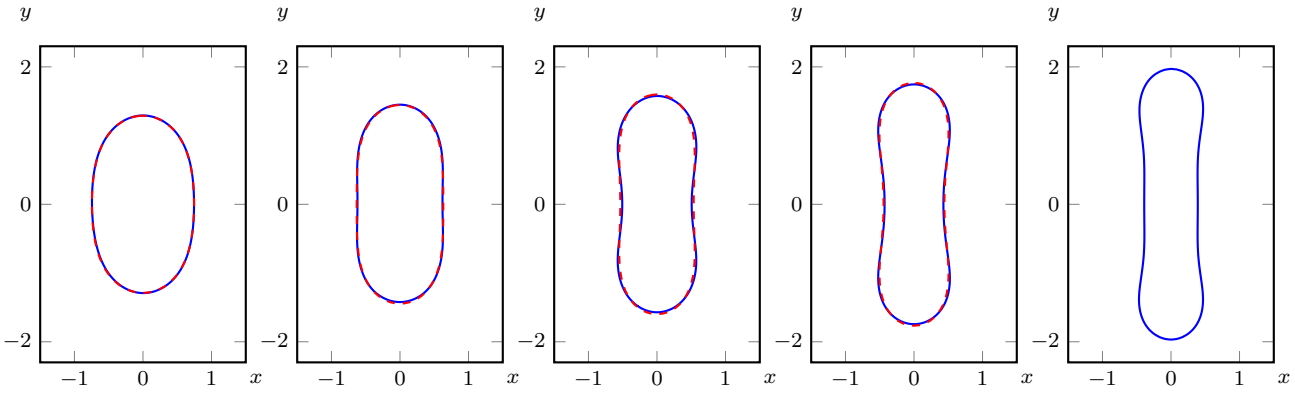
Fig 6: Convergence of algorithm 6 to an equilibrium shape for the $\nu = 0.5$ case, starting from an ellipse with semi-major and -minor axes 2.09021 and 0.478421.

Prolate case The prolate case simulations were run in a cuboid with lengths $9 \times 4 \times 4$ and mesh size $h \approx 0.04$. The initial level-set functions were chosen as in the oblate case, but using prolate ellipsoids.

The resulting equilibrium shapes and cuts are shown in figs. 14 and 15 respectively, while figs. 16 and 17 show the evolution of the surface energy, area and enclosed volume.

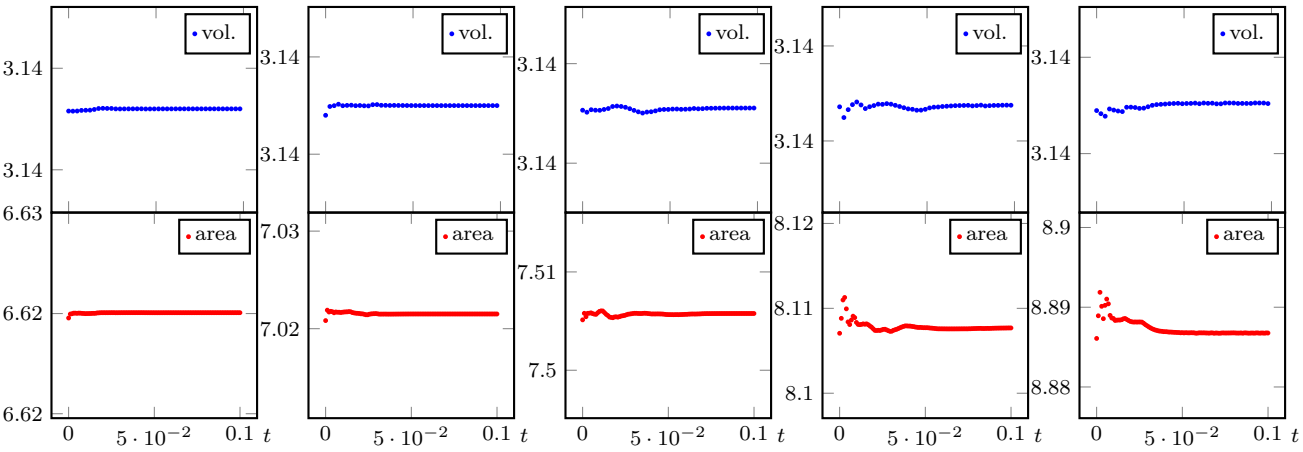
We can observe for all the – oblate and prolate – three-dimensional vesicle simulations that the computed equilibrium shapes are axisymmetric as expected, even though our algorithm does not enforce this property. We also note that our algorithm seems very robust in this case from the energy minimisation point of view, as it reaches the minimum rather quickly and then preserves it for a long time without any numerical artifact.

To compare our results quantitatively, we plot in fig. 18 the equilibrium energies of our vesicles together with the ones obtained in [52] and [20]. We also highlight the numerical convergence of our method in fig. 19 where we plot the relative error of the Willmore energy of our final equilibrium shape (for the reduced volume $\nu = 0.65$) as a function of the mesh size. Our results seem in good agreement with the ones obtained with direct energy minimisation or meshed surface evolution, despite a small $\sim 10\%$ overestimate for the oblate smallest reduced volume cases, which seem related to typical finite-element numerical errors. As shown in fig. 19, the



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.6$ (e) $\nu = 0.5$

Fig 7: Comparison of the 2D equilibrium shapes obtained by algorithm 6 with the ones obtained by [26] using a lattice-Bolzmann method for reduced volumes $\nu = 0.9$, 0.8, 0.7 and 0.6. We also give our result for a reduced volume $\nu = 0.5$.



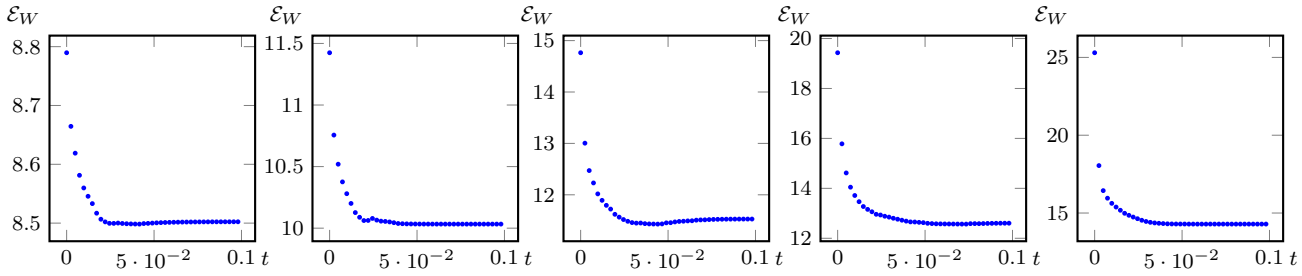
(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.6$ (e) $\nu = 0.5$

Fig 8: Evolution of the 2D vesicle volume and "area" (perimeter in this case) with the volume- and area-preserving algorithm 6. The algorithm displays very good conservation properties : the relative volume and area changes are less than respectively $1 \cdot 10^{-4}$ and $5 \cdot 10^{-4}$ for the most difficult $\nu = 0.5$ case, and the final equilibrium shapes volumes and areas relative changes are $\lesssim 1 \cdot 10^{-5}$ for all the cases.

equilibrium shape energy converges to the expected value (taken from [52] for this axisymmetric case) as $h^{0.84}$, which is rather positive for low-order ($\mathcal{P}^1$) simulations of such high-order effects.

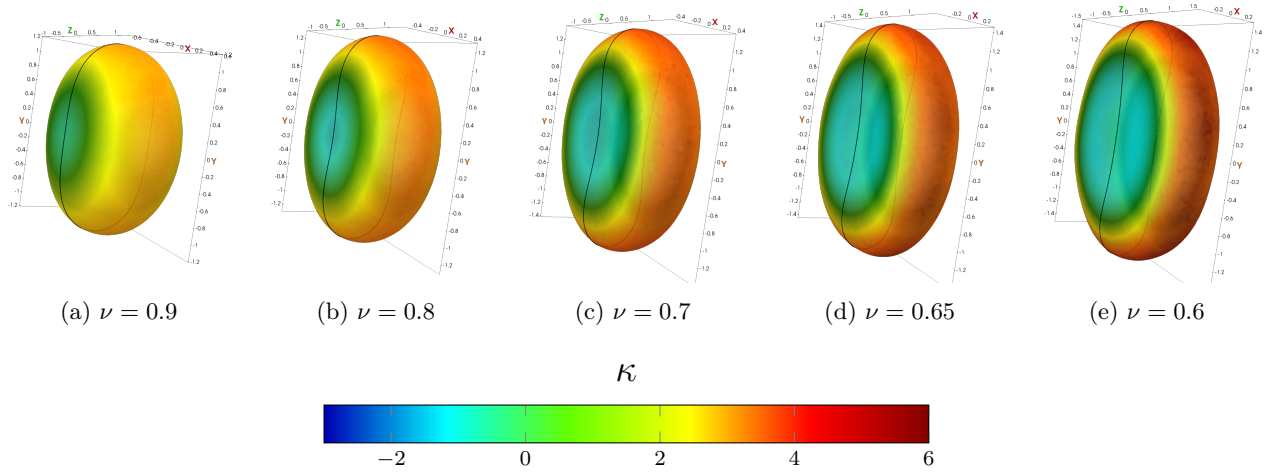
6. Conclusion

We propose in this article a numerical method to predict the position of an interface in 2D or 3D moving according to the gradient of an higher order energy such as the squared mean curvature (Willmore flow) and subject to constraint of area and enclosed volume conservation. This problem is of interest to devise semi-implicit schemes for fluid-structure solvers where the full interaction between immersed vesicles and fluids are involved. As a first step, the present work was restricted to the motion of vesicles membranes minimising their mean curvature at fixed surface area and enclosed volume. The test case considered corresponds to reach an equilibrium shape starting from an ellipsoid, depending on its closeness to a sphere or more elongated shape.



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.6$ (e) $\nu = 0.5$

Fig 9: Evolution of the Willmore energy $\mathcal{E}_W = \int_\Gamma H^2$ (18) of the 2D vesicle with the volume- and area-preserving algorithm 6.



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.65$ (e) $\nu = 0.6$

Fig 10: 3D oblate vesicles equilibrium shapes obtained with algorithm 7 for the reduced volumes $\nu = 0.9$, 0.8, 0.7, 0.65 and 0.6.

Even in this delimited setting, the problem is tricky since the constraints are highly nonlinear and the energy involved of higher order. Classical approaches through level-set or phase field methods lead to fourth order PDE to solve, while purely Lagrangian methods dealing with a surface mesh are not well suited to be included in a fluid-structure coupling procedure due to the interpolations required between Eulerian and Lagrangian representations. In this work, we build a method where only heat equations are solved to compute the right flow of the mean curvature energy.

Based on diffusion-redistanciation schemes introduced in [17], our first contribution was to provide a more intrinsic formulation of these methods, which opens the way to study more easily other kind of higher order energy. While this study focused on the mean curvature flows, one could for instance also consider problems where the Gaussian curvature is involved. We also extended our geometrical flows algorithms to the evolution of surfaces with conserved area and enclosed volume. The originality of our approach relies on a formula providing explicitly the projection of the unconstrained Willmore flow on motion conserving area and enclosed volume. This is a big advantage in comparison to other methods where the area constraint is penalised (and therefore not exactly fulfilled or leading to stiff problems) or nonlinearly enforced (at a high computational cost).

Our first test case was devoted to the torus, where we observed convergence to the Clifford torus as expected. Our methodology enjoys numerical convergence and compares well with existing numerical results to compute equilibrium shapes of vesicles in dimension 2. In dimension 3, while our results are in good agreement for prolate forms, some discrepancy occurs for oblate vesicle shapes. This is due to the low order of the finite element method used ($P1$ element), in contrast with the C^1 surface finite element representation of [20]. However we observe

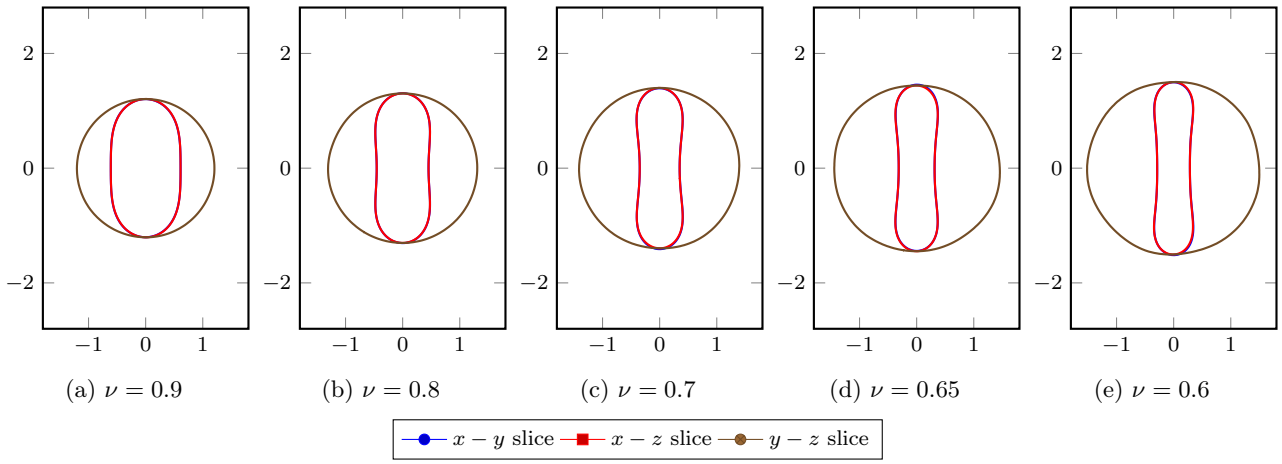


Fig 11: Cut slices in the $x - y$, $x - z$ and $y - z$ planes of the 3D oblate vesicles equilibrium shapes displayed in fig. 10. These cuts highlight the good symmetry preserving property of our algorithm despite the absence of any symmetry enforcing method.

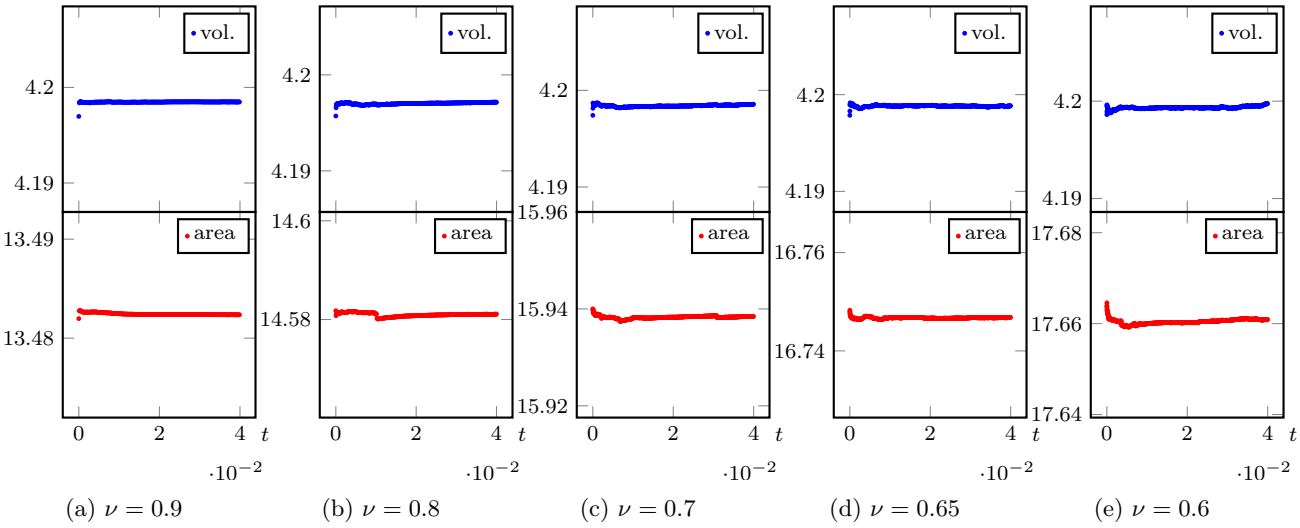
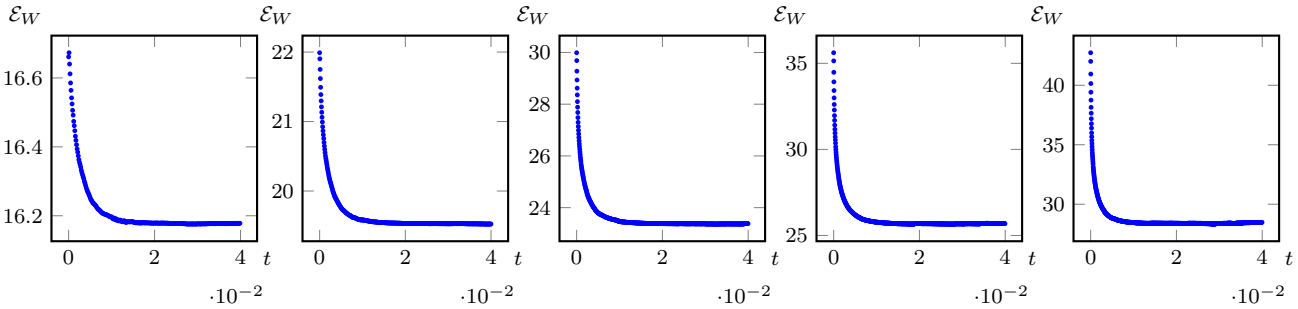


Fig 12: Evolution of the 3D oblate vesicle volume and area with the volume- and area-preserving algorithm 7. The 3D algorithm also displays good conservation properties: the relative volume and area changes are respectively $\lesssim 5 \cdot 10^{-5}$ and $\lesssim 2 \cdot 10^{-4}$ for all the cases.

numerical convergence in dimension 3 as well. The use of polynomial discretisation with higher regularity (such as Hermite elements) would likely cure that problem without any further modification of our algorithm. One remarquable feature of our algorithm is to provide consistent results without any mesh refinement strategy. This is of paramount importance for its possible use in a full three-dimensional fluid-structure problems, where such remeshing could very quickly lead, for realistic situations, to untractable computational complexity.

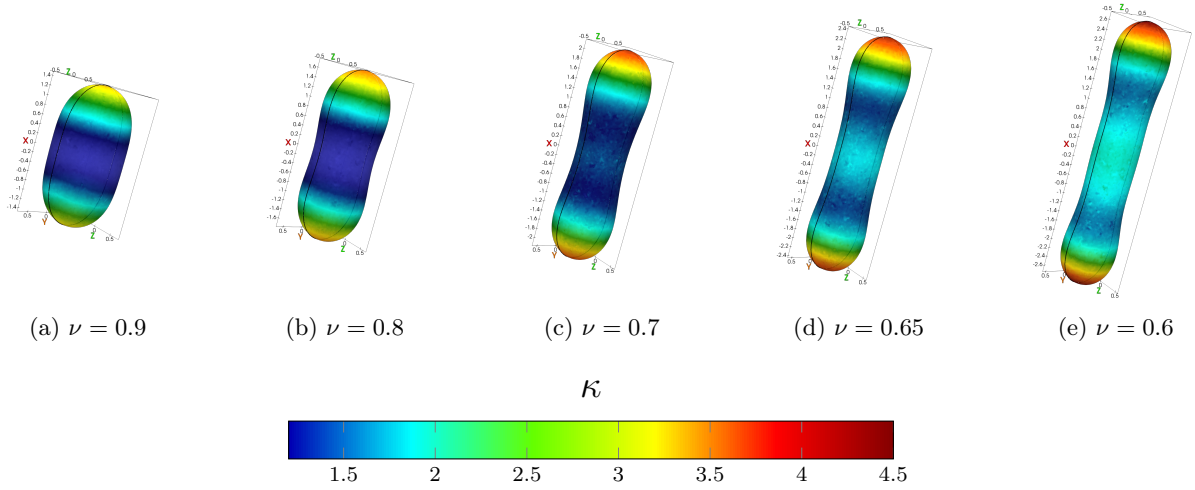
Another big advantage of our approach is to be very easy to implement: high order geometric quantities are computed from diffusion equation of a distance function, which is easily handled by any finite element library.

In a forthcoming work, we will use such a predictor of motion to devise a semi-implicit scheme for the full fluid-membrane coupling problem.



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.65$ (e) $\nu = 0.6$

Fig 13: Evolution of the Willmore energy $\mathcal{E}_W = \int_\Gamma H^2$ (18) of the 3D oblate vesicle with the volume- and area-preserving algorithm 7.



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.65$ (e) $\nu = 0.6$

Fig 14: 3D prolate vesicles equilibrium shapes obtained with algorithm 7 for the reduced volumes $\nu = 0.9$, 0.8, 0.7, 0.65 and 0.6.

References

[1] Fred Almgren, Jean E Taylor, and Lihe Wang. Curvature-driven flows: a variational approach. SIAM Journal on Control and Optimization, 31(2):387–438, 1993. 3

[2] Guy Barles and Christine Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. SIAM Journal on Numerical Analysis, 32(2):484–500, 1995. 3

[3] Julien Beaucourt, Francois Rioual, Thomas Séon, Thierry Biben, and Chaouqi Misbah. Steady to unsteady dynamics of a vesicle in a flow. Physical Review E, 69(1):011906, 2004. 1

[4] Thierry Biben, Klaus Kassner, and Chaouqi Misbah. Phase-field approach to three-dimensional vesicle dynamics. Physical Review E, 72(4):041921, 2005. 1, 21

[5] Thierry Biben and Chaouqi Misbah. Tumbling of vesicles under shear flow within an advected-field approach. Physical Review E, 67(3):031908, 2003. 1

[6] Elie Bretin, Simon Masnou, and Edouard Oudet. Phase-field approximations of the willmore functional and flow. Numerische Mathematik, 131(1):115–171, 2015. 1, 19, 21

[7] Peter B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. Journal of theoretical biology, 26(1):61–81, 1970. 20

[8] Antonin Chambolle and Matteo Novaga. Convergence of an algorithm for the anisotropic and crystalline mean curvature flow. SIAM journal on mathematical analysis, 37(6):1978–1987, 2006. 3

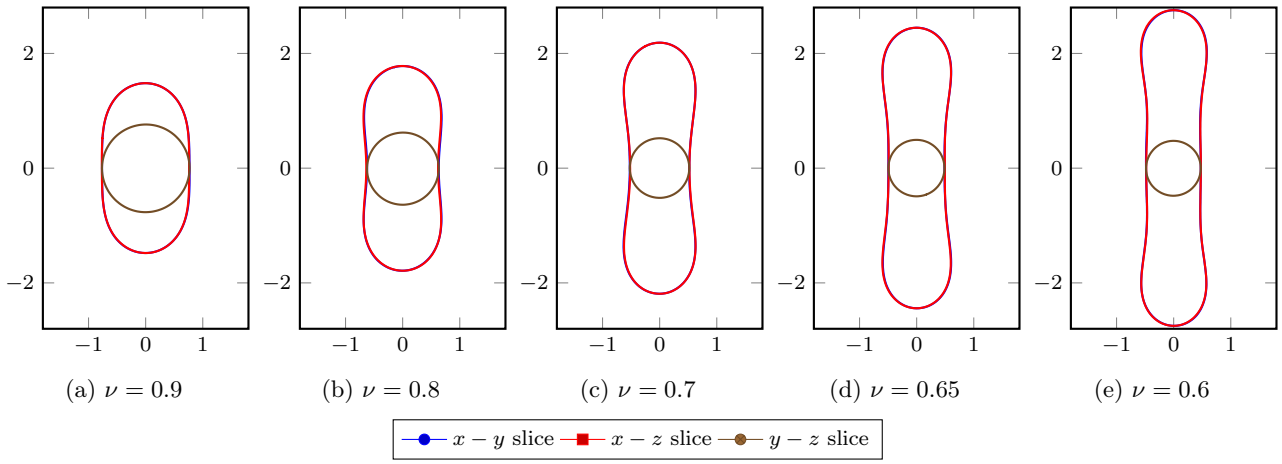


Fig 15: Cut slices in the $x - y$, $x - z$ and $y - z$ planes of the 3D prolate vesicles equilibrium shapes displayed in fig. 10. These cuts highlight the good symmetry preserving property of our algorithm despite the absence of any symmetry enforcing method.

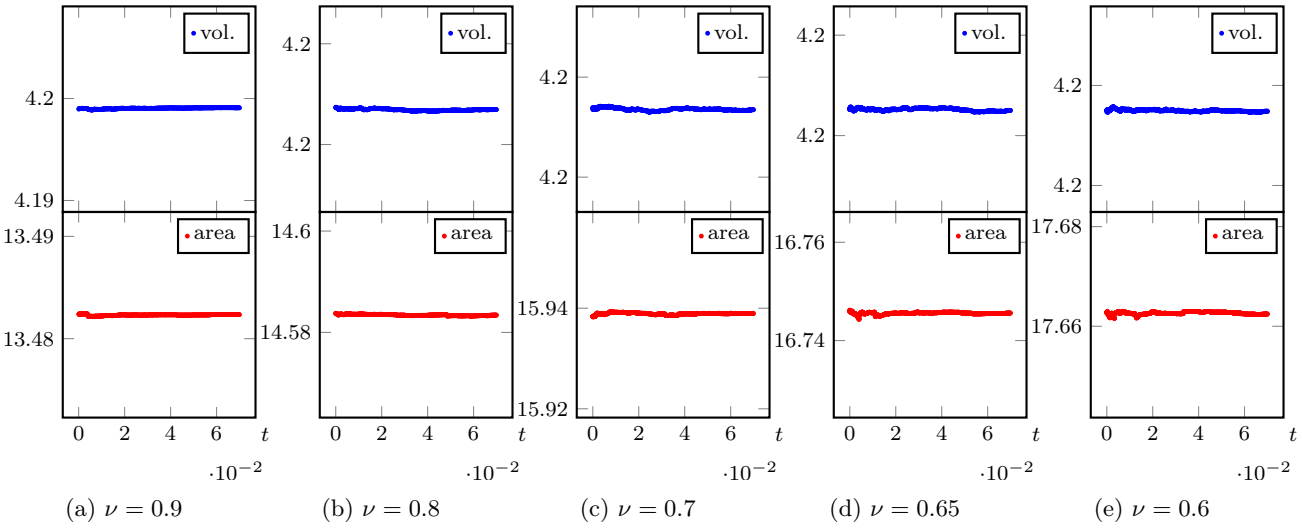


Fig 16: Evolution of the 3D prolate vesicle volume and area with the volume- and area-preserving algorithm 7. The 3D algorithm also displays good conservation properties: the relative volume and area changes are respectively $\lesssim 5 \cdot 10^{-6}$ and $\lesssim 3 \cdot 10^{-5}$ for all the cases.

[9] Georges-Henri Cottet and Emmanuel Maitre. A level-set formulation of immersed boundary methods for fluid–structure interaction problems. Comptes Rendus Mathematique, 338(7):581–586, 2004. 9

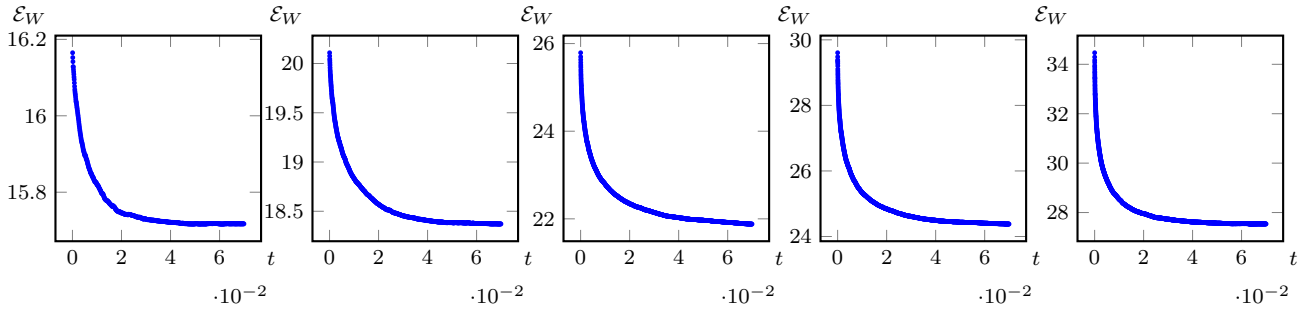
[10] Georges-Henri Cottet and Emmanuel Maitre. A level set method for fluid-structure interactions with immersed surfaces. Mathematical models and methods in applied sciences, 16(03):415–438, 2006. 9

[11] Georges-Henri Cottet and Emmanuel Maitre. A semi-implicit level set method for multiphase flows and fluid–structure interaction problems. Journal of Computational Physics, 314:80–92, 2016. 1

[12] Michel C Delfour and Jean-Paul Zolésio. Shapes and geometries: metrics, analysis, differential calculus, and optimization, volume 22. Siam, 2011. 4

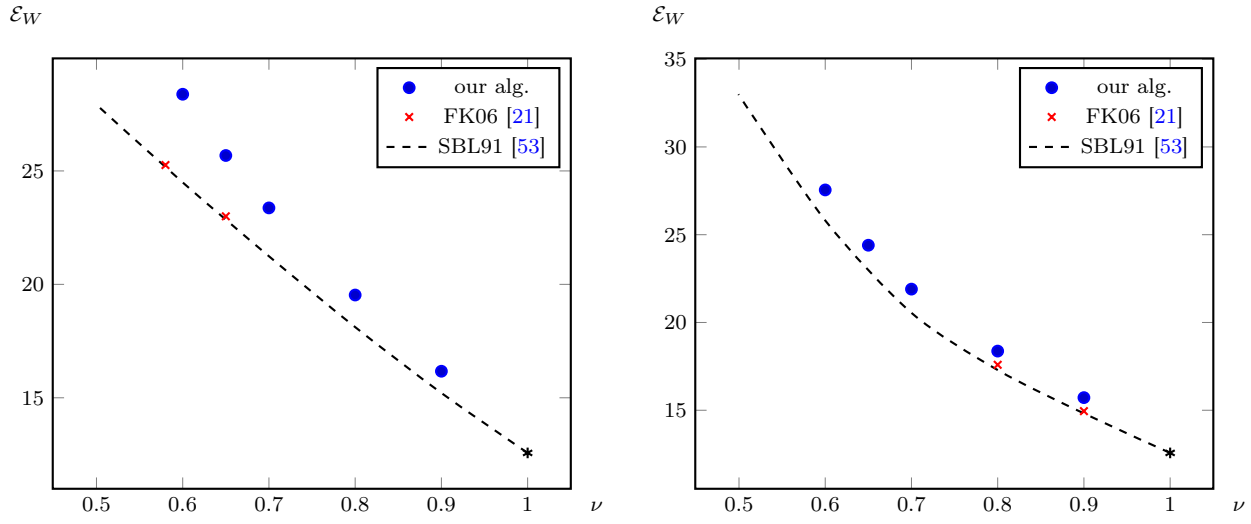
[13] Vincent Doyeux, Yann Guyot, Vincent Chabannes, Christophe Prud'Homme, and Mourad Ismail. Simulation of two-fluid flows using a finite element/level set method. application to bubbles and vesicle dynamics. Journal of Computational and Applied Mathematics, 246:251–259, 2013. 1, 9

[14] Qiang Du, Chun Liu, and Xiaoqiang Wang. Simulating the deformation of vesicle membranes under elastic



(a) $\nu = 0.9$ (b) $\nu = 0.8$ (c) $\nu = 0.7$ (d) $\nu = 0.65$ (e) $\nu = 0.6$

Fig 17: Evolution of the Willmore energy $\mathcal{E}_W = \int_\Gamma H^2$ (18) of the 3D prolate vesicle with the volume- and area-preserving algorithm 7.



(a) Oblate shapes (b) Prolate shapes

Fig 18: Willmore energy of the 3D vesicle equilibrium shapes as a function of the reduced volume, in the oblate and prolate cases. We compare our results to axisymmetric energy minimisation (shooting method) from [52] and to the "subdivision thin shell" vesicle simulations from [20].

bending energy in three dimensions. Journal of Computational Physics, 212(2):757–777, 2006. 21

[15] Qiang Du and Jian Zhang. Adaptive finite element method for a phase field bending elasticity model of vesicle membrane deformations. SIAM Journal on Scientific Computing, 30(3):1634–1657, 2008. 1, 21

[16] Selim Esedoglu and Felix Otto. Threshold dynamics for networks with arbitrary surface tensions. Communications on pure and applied mathematics, 68(5):808–864, 2015. 2, 3

[17] Selim Esedoglu, Steven Ruuth, Richard Tsai, et al. Diffusion generated motion using signed distance functions. Journal of Computational Physics, 229(4):1017–1042, 2010. 1, 2, 3, 4, 24

[18] Selim Esedoglu, Steven J Ruuth, and Richard Tsai. Threshold dynamics for high order geometric motions. Interfaces and Free Boundaries, 10(3):263–282, 2008. 2, 3, 7

[19] Lawrence C Evans. Convergence of an algorithm for mean curvature motion. Indiana University mathematics journal, pages 533–557, 1993. 3

[20] Feng Feng and William S Klug. Finite element modeling of lipid bilayer membranes. Journal of Computational Physics, 220(1):394–408, 2006. 21, 22, 25, 25, 24, 28

[21] Richards Grzhibovskis and Alexei Heintz. A convolution thresholding scheme for the willmore flow. Interfaces and Free Boundaries, 10(2):139–153, 2008. 2, 3, 7

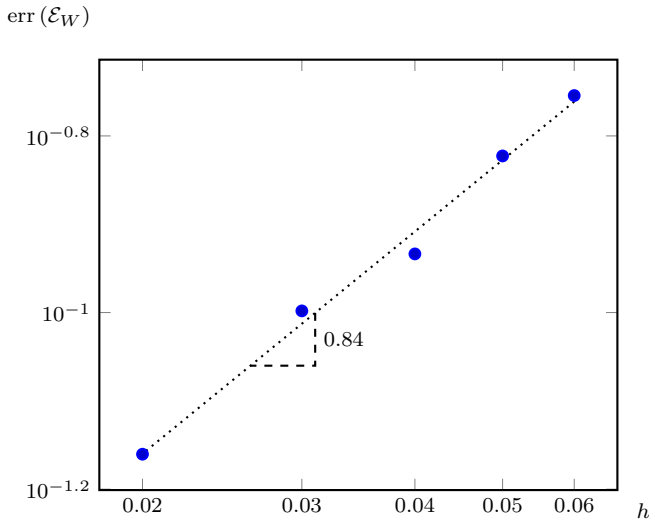


Fig 19: Relative error of the Willmore energy of our three-dimensional vesicle equilibrium shape for the $\nu = 0.65$ case as a function of the mesh size. The error is computed as $\left(\mathcal{E}_W - \mathcal{E}_W^{th}\right)/\mathcal{E}_W^{th}$ with $\mathcal{E}_W^{th}$ taken from [52], and the mesh sizes range from $h \approx 0.02$ to $h \approx 0.06$. The plot is in log-log scale.

[22] Wolfgang Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. Zeitschrift für Naturforschung C, 28(11-12):693–703, 1973. 20

[23] Lucas Hsu, Rob Kusner, and John Sullivan. Minimizing the squared mean curvature integral for surfaces in space forms. Experimental Mathematics, 1(3):191–207, 1992. 18

[24] Mourad Ismail and Aline Lefebvre-Lepot. A necklace model for vesicles simulations in 2d. International Journal for Numerical Methods in Fluids, 76(11):835–854, 2014. 1

[25] Frank Jülicher, Udo Seifert, and Reinhard Lipowsky. Conformal degeneracy and conformal diffusion of vesicles. Physical review letters, 71(3):452, 1993. 21

[26] Badr Kaoui, Jens Harting, and Chaouqi Misbah. Two-dimensional vesicle dynamics under shear flow: Effect of confinement. Physical Review E, 83(6):066319, 2011. 1, 21, 23

[27] Catherine Kublik, Selim Esedoglu, and Jeffrey A Fessler. Algorithms for area preserving flows. SIAM Journal on Scientific Computing, 33(5):2382–2401, 2011. 2, 9

[28] Aymen Laadhari, Pierre Saramito, and Chaouqi Misbah. Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods. Journal of Computational Physics, 263:328–352, 2014. 1, 9

[29] Tim Laux. Gradient-flow techniques for the analysis of numerical schemes for multi-phase mean-curvature flow. Geometric Flows, 3(1):76–89. 3

[30] Tim Laux and Felix Otto. Convergence of the thresholding scheme for multi-phase mean-curvature flow. Calculus of Variations and Partial Differential Equations, 55(5):129, 2016. 3

[31] Tim Laux and Felix Otto. The thresholding scheme for mean curvature flow and de giorgi's ideas for minimizing movements. arXiv preprint arXiv:1910.11442, 2019. 3

[32] Tim Laux and Felix Otto. Brakke's inequality for the thresholding scheme. Calculus of Variations and Partial Differential Equations, 59(1):39, 2020. 3

[33] Tim Laux and Drew Swartz. Convergence of thresholding schemes incorporating bulk effects. arXiv preprint arXiv:1601.02467, 2016. 3, 9

[34] John S Lowengrub, Andreas Rätz, and Axel Voigt. Phase-field modeling of the dynamics of multicomponent vesicles: Spinodal decomposition, coarsening, budding, and fission. Physical Review E, 79(3):031926, 2009. 1

[35] John S Lowengrub, Jianjun Xu, Axel Voigt, et al. Surface phase separation and flow in a simple model of multicomponent drops and vesicles. Fluid Dyn. Mater. Proc, 3(1):1–19, 2007. 1

[36] Stephan Luckhaus and Thomas Sturzenhecker. Implicit time discretization for the mean curvature flow equation. Calculus of variations and partial differential equations, 3(2):253–271, 1995. 3

[37] Emmanuel Maitre, Thomas Milcent, Georges-Henri Cottet, Annie Raoult, and Yves Usson. Applications of level set methods in computational biophysics. Mathematical and Computer Modelling, 49(11-12):2161–2169, 2009. 1

[38] Albert J Markvoort, Rutger A Van Santen, and Peter AJ Hilbers. Vesicle shapes from molecular dynamics simulations. The Journal of Physical Chemistry B, 110(45):22780–22785, 2006. 1

[39] Fernando C Marques and André Neves. Min-max theory and the willmore conjecture. Annals of mathematics, pages 683–782, 2014. 18

[40] Barry Merriman, James K Bence, and Stanley J Osher. Motion of multiple junctions: A level set approach. Journal of Computational Physics, 112(2):334–363, 1994. 3

[41] Barry Merriman, James Kenyard Bence, and Stanley Osher. Diffusion generated motion by mean curvature. Department of Mathematics, University of California, Los Angeles, 1992. 1, 2, 7

[42] Thibaut Metivet, Vincent Chabannes, Mourad Ismail, and Christophe Prud'homme. High-order finite-element framework for the efficient simulation of multifluid flows. Mathematics, 6(10):203, 2018. 14

[43] Luca Mugnai, Christian Seis, and Emanuele Spadaro. Global solutions to the volume-preserving mean-curvature flow. Calculus of Variations and Partial Differential Equations, 55(1):18, 2016. 3

[44] Christophe Prud'homme. A domain specific embedded language in C++ for automatic differentiation, projection, integration and variational formulations. Scientific Programming, 14(2):81–110, 2006. 14

[45] Christophe Prud'Homme, Vincent Chabannes, Vincent Doyeux, Mourad Ismail, Abdoulaye Samake, and Gonçalo Pena. Feel++: A computational framework for galerkin methods and advanced numerical methods. In ESAIM: Proceedings, volume 38, pages 429–455. EDP Sciences, 2012. 14

[46] Giovanni Russo and Peter Smereka. A remark on computing distance functions. Journal of Computational Physics, 163(1):51–67, 2000. 9

[47] Steven J Ruuth. A diffusion-generated approach to multiphase motion. Journal of Computational Physics, 145(1):166–192, 1998. 2

[48] Steven J Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. Journal of Computational Physics, 144(2):603–625, 1998. 3

[49] Steven J Ruuth and Brian TR Wetton. A simple scheme for volume-preserving motion by mean curvature. Journal of Scientific Computing, 19(1-3):373–384, 2003. 2, 9

[50] David Salac and Michael Miksis. A level set projection model of lipid vesicles in general flows. Journal of Computational Physics, 230(22):8192–8215, 2011. 1

[51] Udo Seifert. Configurations of fluid membranes and vesicles. Advances in physics, 46(1):13–137, 1997. 20

[52] Udo Seifert, Karin Berndl, and Reinhard Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous-curvature and bilayer-coupling models. Physical review A, 44(2):1182, 1991. 21, 22, 25, 25, 23, 28, 29

[53] James A Sethian. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences, 93(4):1591–1595, 1996. 17

[54] Leon Simon. Existence of surfaces minimizing the willmore functional. Communications in Analysis and Geometry, 1(2):281–326, 1993. 1

[55] Anton Smolianski. Finite-element/level-set/operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces. International journal for numerical methods in fluids, 48(3):231–269, 2005. 9

[56] Mark Sussman and Emad Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. SIAM Journal on scientific computing, 20(4):1165–1191, 1999. 9

[57] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. An improved level set method for incompressible two-phase flows. Computers & Fluids, 27(5-6):663–680, 1998. 9

[58] Drew Swartz and Nung Kwan Yip. Convergence of diffusion generated motion to motion by mean curvature. Communications in Partial Differential Equations, 42(10):1598–1643, 2017. 3

[59] Xiaoqiang Wang and Qiang Du. Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches. Journal of mathematical biology, 56(3):347–371, 2008. 1

[60] Thomas J Willmore. Note on embedded surfaces. An. Sti. Univ."Al. I. Cuza" Iasi Sect. I a Mat.(NS) B, 11:493–496, 1965. 18

[61] Thomas James Willmore. Riemannian geometry. Oxford University Press, 1996. 6

[62] Jianming Yang and Frederick Stern. A highly scalable massively parallel fast marching method for the

eikonal equation. Journal of Computational Physics, 332:333–362, 2017. 17