Journal of Computational Physics 438 (2021) 110373

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Optimal control of volume-preserving mean curvature flow

Antoine Laurain [a], Shawn W. Walker [b,*]

[a] Department of Applied Mathematics, Universidade de São Paulo, Brazil
[b] Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803-4918, USA

ARTICLE INFO

Article history:
Available online 24 April 2021

Keywords:
Mean curvature flow
Optimal control
Transverse field
PDEs on moving surfaces

ABSTRACT

We develop a framework and numerical method for controlling the full space-time tube of a geometrically driven flow. We consider an optimal control problem for the mean curvature flow of a curve or surface with a volume constraint, where the control parameter acts as a forcing term in the motion law. The control of the trajectory of the flow is achieved by minimizing an appropriate tracking-type cost functional. The gradient of the cost functional is obtained via a formal sensitivity analysis of the space-time tube generated by the mean curvature flow. We show that the perturbation of the tube may be described by a transverse field satisfying a parabolic equation on the tube. We propose a numerical algorithm to approximate the optimal control and show several results in two and three dimensions demonstrating the efficiency of the approach.

1. Introduction

There exists a host of applications that involve large deformations or geometric flows: the motion of droplets [25], fluid structure interaction [6,26], mean curvature flow [3,18,22], deformable elastic bodies [36,39,43], liquid thin films on curved surfaces [37] to name a few. Such problems can often be formulated as the evolution of sets under a gradient flow. A simple yet challenging example of gradient flow is the evolution by mean curvature flow (MCF), for which a rich mathematical theory has been developed. The main focus of the literature on MCF is the proof of existence and regularity of the flow, and the study of its geometric properties.

Controlling these complex motions can further extend these applications or make them more optimal. The theoretical study of this question has seldom been addressed, with the notable exception of the work of Zolésio and his collaborators; see [16,29]. Moreover, the numerical realization of optimal geometric control is sorely lacking, except for the numerical work [10] where a phase-field approach is used; see also [13] where a parameter identification problem using a sharp interface formulation is investigated. The optimal control of gradient flows is of course a challenging topic, considering that even the MCF is still a very active field of research with many open questions.

In this paper, we consider an optimal control problem for the mean curvature flow of a curve or surface (embedded manifold) with a volume constraint on the evolving domain. The goal is to control the evolution of the manifold so that it tracks a desired trajectory; this is accomplished by introducing an appropriate control variable into the motion law of the manifold. Thus, our problem can be seen as the optimal control of a volume-preserving MCF with a forcing term. We note that there exists an extensive literature on volume-preserving MCF, see the regularized mean curvature flow of [24] and also [3,23], and on MCF with forcing term, see [11,22,23]. We employ a time-independent control and consider the tracking

* Corresponding author.
E-mail addresses: laurain@ime.usp.br (A. Laurain), walker@math.lsu.edu (S.W. Walker).

https://doi.org/10.1016/j.jcp.2021.110373
0021-9991/

of both time-independent and time-dependent trajectories in our numerical experiments. We observe in our numerical experiments that the stationarity of the control does not create issues for the control of time-dependent trajectories, but this certainly restricts the set of trajectories that are exactly reachable with such control.

The control of the trajectory of the evolving set may be formulated as the control of the *space-time tube* defined by this evolving set. We seek to solve a time-dependent geometric PDE constrained optimization problem to minimize an appropriate tracking-type functional. The desired trajectory is determined by the zero level set of a given function. The theoretical study of the optimal control of such space-time tubes has been pioneered by Zolésio and its collaborators; see [16,29]. Using the concept of *transverse field*, they performed the sensitivity analysis of the flow with respect to perturbations of the motion law. This concept of transverse field plays a key role in our approach.

The main task in our framework is to relate the perturbation of the control, which appears as a forcing term in the motion law, with the perturbation of the motion law. Compared to the theory and applications developed in [16,29], we are facing an additional fundamental difficulty. Indeed in our problem the motion law is implicitly defined as it depends on the shape of the space-time tube via the mean curvature term. We tackle this issue using a formal perturbation analysis of the flow. In this way we are able to show that the perturbation of the flow can be described by a parabolic PDE on the space-time tube generated by the flow. This allows us to compute the derivative of the tracking-type functional with respect to the control, via the introduction of an appropriate adjoint state.

The type of problem considered in this paper shares many similarities with the optimal control of free boundaries for stationary problems. Recent contributions on this topic have been obtained using tools from shape calculus including [20,38] for the shape controllability of the free boundary of an obstacle problem, [21] where the control of the Bernoulli free boundary problem has been investigated, and [25] for the control of the footprint of a sessile droplet via substrate surface tension. In these works, the free boundary depends implicitly on the control and a perturbation analysis allows to compute the sensitivity of the free boundary with respect to the control. The problem considered in the present paper is similar in the sense that the space-time tube also depends implicitly on the control. Thus, on one hand several ideas of [21,25] may be used, but on the other hand the control of the geometric evolution problem presents many specificities that require a different analysis and novel ideas. To the best of our knowledge this is the first investigation of optimal control of geometric evolution problems and MCF using shape optimization tools.

The paper is organized as follows. In Section 2 we introduce the volume-preserving MCF with forcing term and describe the optimal control problem via the minimization of a tracking-type cost functional. In Section 3 we perform the sensitivity analysis of the cost functional with respect to a small perturbation of the control and obtain the gradient of the cost functional via the introduction of an adjoint transverse field. We also show an alternative expression of the gradient of the cost functional based on a scalar adjoint. In Section 4 we explain the optimization scheme and its discretization. Section 5 presents numerical results in two and three dimensions. We conclude with some remarks in Section 6.

2. Control of the mean curvature flow

2.1. Volume-preserving mean curvature flow with forcing term

Let $\mathfrak{D} \subset \mathbb{R}^n$ be a bounded open set which contains all admissible domains for our problem. We consider vector fields with compact support in $\mathfrak{D}$, satisfying

$$\mathcal{V} := \mathcal{C}([0,t_f];\mathcal{C}_c^\infty(\mathfrak{D},\mathbb{R}^n)), \tag{1}$$

where $t_f > 0$ denotes the final time for our interval of interest $[0,t_f]$. For a given $\boldsymbol{V} \in \mathcal{V}$ and $\boldsymbol{x}^0 \in \mathfrak{D}$, let $\boldsymbol{x}(t)$ be the solution of

$$\begin{aligned} &\dot{\boldsymbol{x}}(t) = \boldsymbol{V}(t,\boldsymbol{x}(t)), \quad \text{for } t \in [0,t_f], \\ &\boldsymbol{x}(0) = \boldsymbol{x}^0. \end{aligned} \tag{2}$$

Thus, for all $t \in [0,t_f]$, we can define a transformation $T^t(\boldsymbol{V}) : \overline{\mathfrak{D}} \to \overline{\mathfrak{D}}$ by $T^t(\boldsymbol{V})(\boldsymbol{x}^0) := \boldsymbol{x}(t)$.

Next, let $\Gamma \subset \mathfrak{D}$ be a smooth $(n-1)$-dimensional embedded manifold, without boundary, whose interior is Ω, i.e. $\Gamma \equiv \partial\Omega$. For $t \in [0,t_f]$, let us consider formally the evolution of sets $\Gamma^t = \Gamma^t(\boldsymbol{V})$ given by the following motion law

$$\boldsymbol{V}|_{\Gamma^t} = (u|_{\Gamma^t} - \kappa(t) - \lambda(t))\boldsymbol{\nu}(t) \quad \text{for } t \in [0,t_f], \tag{3}$$

where $\boldsymbol{\nu}(t)$ is the outer unit normal vector of Γ^t, $\kappa(t)$ is the (additive) curvature of $\Gamma^t(\boldsymbol{V})$, $\lambda(t)$ is a Lagrange multiplier to enforce the constraint that $|\Omega^t|$ is constant for all $t \in [0,t_f]$, where Ω^t is the interior of Γ^t (i.e., $\Gamma^t \equiv \partial\Omega^t$), and $u \in \mathcal{C}^\infty(\mathfrak{D})$ is a given control function. The motion law (3) can be described as a volume-preserving MCF with a forcing term u, and $\boldsymbol{V}$ can be derived from a gradient flow with a volume constraint; see [3]. Indeed, $\boldsymbol{V}$ is the (volume-preserving) L^2-gradient flow of the "energy"

$$\mathcal{E}[\Omega] = \int_\Gamma 1 - \int_\Omega u. \tag{4}$$

Note that (3) is an implicit definition of $\boldsymbol{V}$ since κ, λ and $\boldsymbol{\nu}$ depend on $\boldsymbol{V}$ through $\Gamma^t(\boldsymbol{V})$. The existence of the evolution of sets $\Gamma^t(\boldsymbol{V})$ may be proven in the sense of viscosity solutions; see [7,18]. This approach allows topological changes, but the drawback is that fattening of $\Gamma^t(\boldsymbol{V})$ can occur. Another approach is to use parameterization, this avoids the fattening phenomenon but in this case singularities may appear in finite time. In two dimensions, i.e., when $\Gamma^t(\boldsymbol{V})$ is a curve, a more precise analysis is given in [4,5], but singularity may still appear; see [28] for an example and [27] for a comprehensive discussion of singularities for MCF. In any case, finite-time singularities can be expected with the motion law (3).

In this paper we assume existence and smoothness of the sets $\Gamma^t(\boldsymbol{V})$ for $t \in [0, t_f]$ under the motion law (3), and define the smooth transformation $T^t(\boldsymbol{V})(\Gamma) := \Gamma^t(\boldsymbol{V})$. We also assume that $\boldsymbol{\nu}$ and κ can be extended so that $\boldsymbol{V}|_{\Gamma^t}$ can be extended to all of $\mathfrak{D}$ and that $\boldsymbol{V} = \boldsymbol{V}(u) \in \mathcal{V}$; these extensions are discussed in more details in Section 3.2.

Notations: For $f : \mathbb{R}^n \to \mathbb{R}$, ∇f denotes the gradient of f and $\nabla_\Gamma f := \nabla f - (\nabla f \cdot \boldsymbol{\nu})\boldsymbol{\nu}$ the tangential gradient of f on Γ. For $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, $D\boldsymbol{f}$ denotes the Jacobian matrix of $\boldsymbol{f}$ and $\operatorname{div}_\Gamma \boldsymbol{f} := \operatorname{div} \boldsymbol{f} - (D\boldsymbol{f})\boldsymbol{\nu} \cdot \boldsymbol{\nu}$ the tangential divergence of $\boldsymbol{f}$ on Γ.

2.2. Tracking-type cost functional

Our objective is to determine a control u in (3) such that the evolution of sets $\Gamma^t(\boldsymbol{V})$ is as close as possible to a given target trajectory. A convenient way to achieve this is to minimize a tracking-type functional based on a level set function. Let $\phi : [0, t_f] \times \mathfrak{D} \to \mathbb{R}$ be a given smooth function and define its sublevel set

$$Q_d := \{(t, \boldsymbol{x}) \in [0, t_f] \times \mathfrak{D} \mid \phi(t, \boldsymbol{x}) < 0\}. \tag{5}$$

Then Q_d is called the *target trajectory*. Next, introduce the tracking-type cost functional

$$J(\boldsymbol{V}(u)) := \int_0^{t_f} \int_{\Omega^t(\boldsymbol{V}(u))} \phi, \tag{6}$$

and the *reduced* functional, including a Tikhonov regularization of the control,

$$\mathcal{J}(u) := J(\boldsymbol{V}(u)) + \frac{\alpha}{2}\|\nabla u\|^2_{L^2(\mathfrak{D})}.$$

Consider the minimization problem

$$\text{minimize } \mathcal{J}(u), \quad \text{subject to } u \in \mathcal{C}^\infty(\mathfrak{D}). \tag{7}$$

To understand the relevance of the minimization problem (7) to achieve the control objective, we need to introduce the notion of *tubes*.

Definition 1. For a set $\mathcal{A} \subset \mathbb{R}^n$, let the space-time $\mathcal{A}$-tube be given by

$$Q_{\mathcal{A}}(\boldsymbol{V}) := \bigcup_{0 \le t \le t_f} \{t\} \times T^t(\boldsymbol{V})(\mathcal{A}). \tag{8}$$

In view of (5) and (6), if there exists a control $u \in \mathcal{C}^\infty(\mathfrak{D})$ such that $Q_\Omega(\boldsymbol{V}(u)) = Q_d$, then the minimum of $J(\boldsymbol{V}(u))$ is attained precisely for this control. The Tikhonov regularization in $\mathcal{J}(u)$ allows to find a compromise between reaching the target Q_d and obtaining a smooth control u.

3. Derivative of the cost functional

In this paper, we opt for a steepest descent algorithm to find an approximate solution of (7), hence the gradient of $\mathcal{J}(u)$ needs to be computed first. Two main difficulties arise when considering this computation. The first one is the dependence of Ω^t on $\boldsymbol{V}$ in (6). This dependence is relatively well-understood and can be handled via the concept of *transverse field* introduced in [16,29]. The second difficulty comes from the fact that the motion law (3) is an implicit definition of $\boldsymbol{V}$. This makes the sensitivity analysis of $\boldsymbol{V}$ with respect to u challenging, since a small perturbation of u induces a perturbation of $\boldsymbol{V}$ not only through equation (3) but also through the variation of Γ^t. Such a difficulty is new in the area of control of geometric evolution equations, but we may borrow certain ideas from optimal control of free boundary problems [21,25] to tackle this issue.

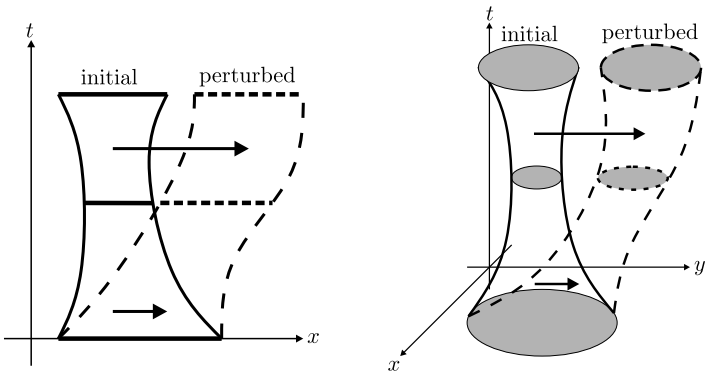

Fig. 1. Diagram of the space-time tube for one dimensional domains (left) and two dimensional domains (right). The time axis is vertical. The initial tube corresponds to $Q_\Omega(\boldsymbol{V})$, while the perturbed tube corresponds to $Q_\Omega(\boldsymbol{V}+s\boldsymbol{W})$; see Definition 1. The transverse field $\boldsymbol{Z}$ is indicated by the horizontal arrows and corresponds to $\partial_s T_s^t(\Omega)|_{s=0}$.

3.1. Derivative of tube functionals via transverse field

In [16,29] the sensitivity analysis of cost functionals $J(\boldsymbol{V})$ of the type (6), sometimes called tube functionals, with respect to variations of $\boldsymbol{V}$ has been investigated. The chief ingredient to compute the derivative of such cost functionals with respect to $\boldsymbol{V}$ is the concept of *transverse field* (see Fig. 1). We briefly explain here how this concept can be applied to our problem.

Let $\boldsymbol{V},\boldsymbol{W}\in\mathcal{V}$ and consider the tube $Q_\Omega(\boldsymbol{V})$, see Definition 1. Let $T_s^t:\overline{\mathfrak{D}}\to\overline{\mathfrak{D}}$ be the transformation generated by the time varying field $\boldsymbol{V}(t,\cdot)+s\boldsymbol{W}(t,\cdot)$, with $T_0^t=T^t$. The perturbed moving set is denoted by $\Omega_s^t:=T_s^t(\Omega)$. The following expression for the derivative of $J(\boldsymbol{V})$ with respect to $\boldsymbol{W}\in\mathcal{V}$ may be found in [29, Section 3.4.1], see also [16]:

$$[\delta J(\boldsymbol{V})](\boldsymbol{W})=\int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V})}\phi(t,\boldsymbol{x})\boldsymbol{Z}(t,\boldsymbol{x})\cdot\boldsymbol{\nu}(t,\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{Z}:[0,t_f]\times\mathfrak{D}\to\mathbb{R}^n$ is the so-called *transverse field*, which depends on $\boldsymbol{W}$ in the following way:

$$\begin{aligned}\partial_t\boldsymbol{Z}+[\boldsymbol{Z},\boldsymbol{V}]&=\boldsymbol{W},\ \text{in }(0,t_f)\times\mathfrak{D},\\ \boldsymbol{Z}(0,\cdot)&=\boldsymbol{0},\ \text{in }\mathfrak{D},\end{aligned} \tag{10}$$

where $[\cdot,\cdot]$ denotes the Lie bracket, i.e., $[\boldsymbol{Z},\boldsymbol{V}]:=(D\boldsymbol{Z})\boldsymbol{V}-(D\boldsymbol{V})\boldsymbol{Z}$. We introduce the notion of material derivative to clarify the question of the solvability of (10).

Definition 2 *(Material Derivative).* Let $f:(0,t_f)\times\mathfrak{D}\to\mathbb{R}$ be $\mathcal{C}^1$, and let $\boldsymbol{V}:\mathfrak{D}\to\mathbb{R}^n$ be a continuous velocity field. Then, we define the *material derivative of* f as

$$D_{\boldsymbol{V}}f\circ T^t(\boldsymbol{V}):=\frac{d}{dt}[f\circ T^t(\boldsymbol{V})].$$

The material derivative for functions defined on subsets of $\mathfrak{D}$ and the material derivative of vector-valued functions is defined in a similar way.

Thus, $\partial_t\boldsymbol{Z}+(D\boldsymbol{Z})\boldsymbol{V}\equiv D_{\boldsymbol{V}}\boldsymbol{Z}$ and the first line of (10) is simply $D_{\boldsymbol{V}}\boldsymbol{Z}-(D\boldsymbol{V})\boldsymbol{Z}=\boldsymbol{W}$. Indeed, if $\boldsymbol{x}(t)$ is the trajectory of a point driven by $\boldsymbol{V}$, then (10) is simply an ordinary differential equation (ODE), i.e., $\boldsymbol{Z}(\boldsymbol{x}(t))'-(D\boldsymbol{V}(\boldsymbol{x}(t)))\boldsymbol{Z}(\boldsymbol{x}(t))=\boldsymbol{W}(\boldsymbol{x}(t))$, whose solvability is guaranteed by the theory of ODEs. Solving the ODE for every possible trajectory yields the unique solution of (10), see [29].

We also recall the concept of shape derivative which is a convenient tool for the forthcoming calculations. We refer to [41, Section 5.4] and [15,19,38] for more details on the notions of material and shape derivatives.

Definition 3 *(Shape Derivative).* Let $f:(0,t_f)\times\mathfrak{D}\to\mathbb{R}^n$ be $\mathcal{C}^1$, and let $\boldsymbol{V}:\mathfrak{D}\to\mathbb{R}^n$ be a continuous velocity field. Then, we define the *shape derivative of* f as

$$\delta_\Omega f(\boldsymbol{V}):=[D_{\boldsymbol{V}}f-\boldsymbol{V}\cdot\nabla f]|_{t=0}\,.$$

The shape derivative for functions defined on subsets of $\mathfrak{D}$ and the shape derivative of vector-valued functions is defined in a similar way.

3.2. Sensitivity analysis of the tube with respect to the control

Considering $J(\boldsymbol{V}(u))$ of (6) as the composition of functions $u \mapsto \boldsymbol{V}(u)$ and $\boldsymbol{V} \mapsto J(\boldsymbol{V})$, we have seen that the differentiability of $\boldsymbol{V} \mapsto J(\boldsymbol{V})$ can be handled via the notion of transverse field in Section 3.1. The differentiability of $u \mapsto \boldsymbol{V}(u)$ on the other hand is a delicate issue since $\boldsymbol{V}$ also depends on $\Gamma^t(\boldsymbol{V})$. Here we perform a formal sensitivity analysis of $u \mapsto \boldsymbol{V}(u)$, borrowing some ideas from [21,25].

Let $\eta \in \mathcal{C}^\infty(\mathfrak{D})$ and consider the following perturbation of the control

$$u_\epsilon := u + \epsilon\eta. \tag{11}$$

The main idea of the sensitivity analysis can be summarized in the following way. The first-order perturbation of $J(\boldsymbol{V}(u))$ with respect to the control u can also be described via the transverse field equation (10), except that the vector $\boldsymbol{W}$ appearing in (10) also depends on $\boldsymbol{Z}$ and η in this case. This provides a new transverse field equation involving only $\boldsymbol{Z}$ and η. Hence, our main task is to determine the dependence of $\boldsymbol{W}$ on $\boldsymbol{Z}$ and η.

In view of the motion law (3), the corresponding perturbed sets $\Gamma^t(\boldsymbol{V}(u_\epsilon))$ and perturbed vector field satisfy

$$\boldsymbol{V}(u_\epsilon) = ((u+\epsilon\eta)|_{\Gamma^t(\boldsymbol{V}(u_\epsilon))} - \kappa(\epsilon) - \lambda(\epsilon))\boldsymbol{\nu}(\epsilon) \quad \text{ on } \Gamma^t(\boldsymbol{V}(u_\epsilon)), \tag{12}$$

where $\boldsymbol{\nu}(\epsilon)$, $\kappa(\epsilon)$ and $\lambda(\epsilon)$ are the corresponding normal vector, curvature and Lagrange multiplier, respectively. The perturbed cost functional $\mathcal{J}(u_\epsilon)$ is defined accordingly and we also define the interior $\Omega^t(\boldsymbol{V}(u_\epsilon))$ of $\Gamma^t(\boldsymbol{V}(u_\epsilon))$. We need to extend $\boldsymbol{V}(u_\epsilon)$ to a neighborhood of $\Gamma^t(\boldsymbol{V}(u_\epsilon))$ since u is defined on all of $\mathfrak{D}$ and not just on the boundary $\Gamma^t(\boldsymbol{V}(u_\epsilon))$. To define such an extension, we first use a unitary extension of $\boldsymbol{\nu}$ along $\boldsymbol{\nu}$, which in turn induces an extension of κ (not necessarily constant along the normal direction $\boldsymbol{\nu}$). Note that choosing a unitary extension of $\boldsymbol{\nu}$ is not necessary but is a convenient choice that simplifies the intermediate calculations. With such an extension, it is known that the shape derivative of κ in a direction $\boldsymbol{Y}$ (see Definition 3) is given by

$$\delta_\Omega \kappa(\boldsymbol{Y}) = -\Delta_\Gamma(\boldsymbol{Y}\cdot\boldsymbol{\nu}), \tag{13}$$

where Δ_Γ denotes the Laplace-Beltrami operator on $\Gamma^t(\boldsymbol{V}(u))$; see [41, lemma 5.6], [25, p. 783] or [12].

We now explain the main idea of the sensitivity analysis. Let us assume that there exists a smooth transformation $\mathcal{T}_\epsilon^t : \overline{\mathfrak{D}} \to \overline{\mathfrak{D}}$ such that $\mathcal{T}_\epsilon^t(\Gamma^t(\boldsymbol{V}(u))) = \Gamma^t(\boldsymbol{V}(u_\epsilon))$. In order to perform the sensitivity analysis of $\boldsymbol{V}(u_\epsilon)$, we suppose that $\boldsymbol{V}(u_\epsilon)$ has the form $\boldsymbol{V}(u_\epsilon)\circ\mathcal{T}_\epsilon^t = \boldsymbol{V}(u) + \widetilde{\boldsymbol{W}}(\epsilon)$ on $\Gamma^t(\boldsymbol{V}(u))$, where $\widetilde{\boldsymbol{W}}(\epsilon) \in \mathcal{V}$ with $\widetilde{\boldsymbol{W}}(0) = \boldsymbol{0}$. Assuming $\widetilde{\boldsymbol{W}}(\epsilon)$ is sufficiently smooth with respect to ϵ, we have the expansion $\widetilde{\boldsymbol{W}}(\epsilon) = \epsilon\widetilde{\boldsymbol{W}}'(0) + O(\epsilon^2)$. The computation of the derivative of $\epsilon \mapsto \mathcal{J}(u_\epsilon)$ at $\epsilon = 0$ only requires to know $\widetilde{\boldsymbol{W}}'(0)$ instead of $\widetilde{\boldsymbol{W}}(\epsilon)$; a similar idea was used in [25]. Thus we introduce the notation $\boldsymbol{W} := \widetilde{\boldsymbol{W}}'(0)$ and in what follows we proceed to determine $\boldsymbol{W}$ as a function of the transverse field $\boldsymbol{Z}$ and η. Combining this with the transverse field equation (10) one obtains an equation involving only $\boldsymbol{Z}$ and η.

Next, we identify the equation for $\boldsymbol{W}$ by considering the weak formulation of the perturbed gradient flow (12):

$$I(\epsilon) := \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u_\epsilon))} [\boldsymbol{V}(u_\epsilon) - (u + \epsilon\eta - \kappa(\epsilon) - \lambda(\epsilon))\boldsymbol{\nu}(\epsilon)]\cdot\boldsymbol{\xi} = 0, \quad \forall \boldsymbol{\xi} \in \mathcal{V}, \tag{14}$$

with the volume constraint

$$\int_0^{t_f} \mu(t)\left(|\Omega^t(\boldsymbol{V}(u_\epsilon))| - C_V\right) dt = 0, \quad \forall \mu \in L^1([0,t_f]), \tag{15}$$

for some fixed volume constant C_V. Note that μ is independent of ϵ.

Since $\kappa(\epsilon)$, $\lambda(\epsilon)$ and $\boldsymbol{\nu}(\epsilon)$ correspond to the perturbation $\boldsymbol{V}(u_\epsilon)\circ\mathcal{T}_\epsilon^t = \boldsymbol{V}(u) + \widetilde{\boldsymbol{W}}(\epsilon)$, then the results of [16,29] show that the derivatives with respect to ϵ of $\kappa(\epsilon)$, $\lambda(\epsilon)$ and $\boldsymbol{\nu}(\epsilon)$ at $\epsilon = 0$ are equal to the shape derivatives of these quantities with respect to the transverse field $\boldsymbol{Z}$, given by (10), corresponding to the perturbation $\boldsymbol{V}(u_\epsilon)$. For instance we have

$$\boldsymbol{\nu}'(0) = \left.\frac{d\boldsymbol{\nu}(\epsilon)}{d\epsilon}\right|_{t=0} = \delta_\Omega \boldsymbol{\nu}(\boldsymbol{Z}).$$

Since we have assumed a unitary extension of $\boldsymbol{\nu}$ along the normal $\boldsymbol{\nu}$ it is known that $\delta_\Omega\boldsymbol{\nu}(\boldsymbol{Z}) = -\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})$; see [41, lemma 5.5] or [19, Proposition 5.4.14]. Therefore we have

$$\boldsymbol{\nu}'(0) = -\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}). \tag{16}$$

Furthermore, in view of (13) the derivative of $\kappa(\epsilon)$ is given by

$$\kappa'(0) = -\Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}). \tag{17}$$

Let us recall a general formula for the shape derivative of a boundary integral, see for instance [15, Theorem 4.3, Chapter 9] or [41, Eqn. (5.48)]. For a smooth transformation $T^\epsilon(\boldsymbol{Y}) : \overline{\mathfrak{D}} \to \overline{\mathfrak{D}}$, a smooth $(n-1)$-dimensional embedded manifold γ and a sufficiently smooth function f, we have

$$\frac{d}{d\epsilon} \int_{T^\epsilon(\boldsymbol{Y})(\gamma)} f(\epsilon, \boldsymbol{x})\, dS(\boldsymbol{x}) \bigg|_{\epsilon=0} = \int_\gamma \partial_\epsilon f(0,\boldsymbol{x}) + (\partial_\nu f(0,\boldsymbol{x}) + f(0,\boldsymbol{x})\kappa)(\boldsymbol{Y}\cdot\boldsymbol{\nu})\, dS(\boldsymbol{x}). \tag{18}$$

Using (16), (17) and (18) we can now compute the derivative of (14) as

$$\begin{aligned} I'(0) = &\int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} [\boldsymbol{W} - (\eta + \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) - \lambda'(0))\boldsymbol{\nu} + (u-\kappa-\lambda)\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})]\cdot\boldsymbol{\xi} \\ &+ \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \left(\partial_\nu([\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\cdot\boldsymbol{\xi}) + \kappa[\underbrace{\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}}_{=\boldsymbol{0}}]\cdot\boldsymbol{\xi} \right)(\boldsymbol{Z}\cdot\boldsymbol{\nu}). \end{aligned}$$

Note that we have

$$\begin{aligned} \partial_\nu([\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\cdot\boldsymbol{\xi}) &= \nabla([\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\cdot\boldsymbol{\xi})\cdot\boldsymbol{\nu} \\ &= D[\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\boldsymbol{\nu}\cdot\boldsymbol{\xi} + (D\boldsymbol{\xi})\boldsymbol{\nu}\cdot\underbrace{[\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]}_{=\boldsymbol{0}} \\ &= D[\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\boldsymbol{\nu}\cdot\boldsymbol{\xi}. \end{aligned}$$

Introducing the time-varying, spatial constant $\rho := \lambda'(0)$, we obtain for all $\boldsymbol{\xi} \in \mathcal{V}$:

$$\begin{aligned} 0 = I'(0) = &\int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} [\boldsymbol{W} - (\eta + \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) - \rho)\boldsymbol{\nu} + (u-\kappa-\lambda)\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})]\cdot\boldsymbol{\xi} \\ &+ \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} D[\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}]\boldsymbol{\nu}\cdot\boldsymbol{\xi}(\boldsymbol{Z}\cdot\boldsymbol{\nu}). \end{aligned}$$

In fact we also have $D[\boldsymbol{V} - (u-\kappa-\lambda)\boldsymbol{\nu}] = 0$ on $\Gamma^t(\boldsymbol{V}(u))$ due to $\boldsymbol{V} = (u-\kappa-\lambda)\boldsymbol{\nu}$ in a neighborhood of $\Gamma^t(\boldsymbol{V}(u))$, using the extension of $\boldsymbol{V}$ described at the beginning of this section. Furthermore, differentiating (15) yields

$$\int_0^{t_f} \mu(t) \int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{Z}(t,\boldsymbol{x})\cdot\boldsymbol{\nu}(t,\boldsymbol{x}) = 0, \quad \forall \mu \in L^1([0,t_f]). \tag{19}$$

Hence, we arrive at the following equations relating $\boldsymbol{W}$, ρ and $\boldsymbol{Z}$ on Γ^t:

$$[\boldsymbol{W} - (\eta + \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) - \rho)\boldsymbol{\nu} + (u-\kappa-\lambda)\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})] = \boldsymbol{0} \quad \text{on } \Gamma^t(\boldsymbol{V}(u)), \tag{20}$$

$$\int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{Z}(t,\boldsymbol{x})\cdot\boldsymbol{\nu}(t,\boldsymbol{x}) = 0. \tag{21}$$

Note that ρ plays the role of a Lagrange multiplier for the volume constraint (21).

The transverse field $\boldsymbol{Z}$ and the perturbation field $\boldsymbol{W}$ are also related by equation (10). Thus, combining (10) and (20)-(21) yields the new transverse field equation for $\boldsymbol{Z}$. However, (20) shows that $\boldsymbol{W}|_{\Gamma^t}$ only depends on the values of $\boldsymbol{Z}$ on Γ^t, but we need $\boldsymbol{W}$ defined on $\mathfrak{D}$ in order to use (10). Hence, we assume $\boldsymbol{W}$ can be extended from Γ^t to all of $\mathfrak{D}$ in a smooth way, and then we may *restrict* (10) to Γ^t. We obtain the following system of equations for $\boldsymbol{Z}$:

$$\begin{aligned} &\partial_t \boldsymbol{Z} + (D\boldsymbol{Z})\boldsymbol{V} + A\boldsymbol{Z} + \rho\boldsymbol{\nu} = \eta\boldsymbol{\nu}, \text{ on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t \in (0,t_f), \\ &\int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{Z}(t,\boldsymbol{x})\cdot\boldsymbol{\nu}(t,\boldsymbol{x}) = 0, \text{ for all } t \in (0,t_f), \\ &\boldsymbol{Z}(0,\cdot) = \boldsymbol{0}, \text{ on } \Gamma^0(\boldsymbol{V}(u)), \end{aligned} \tag{22}$$

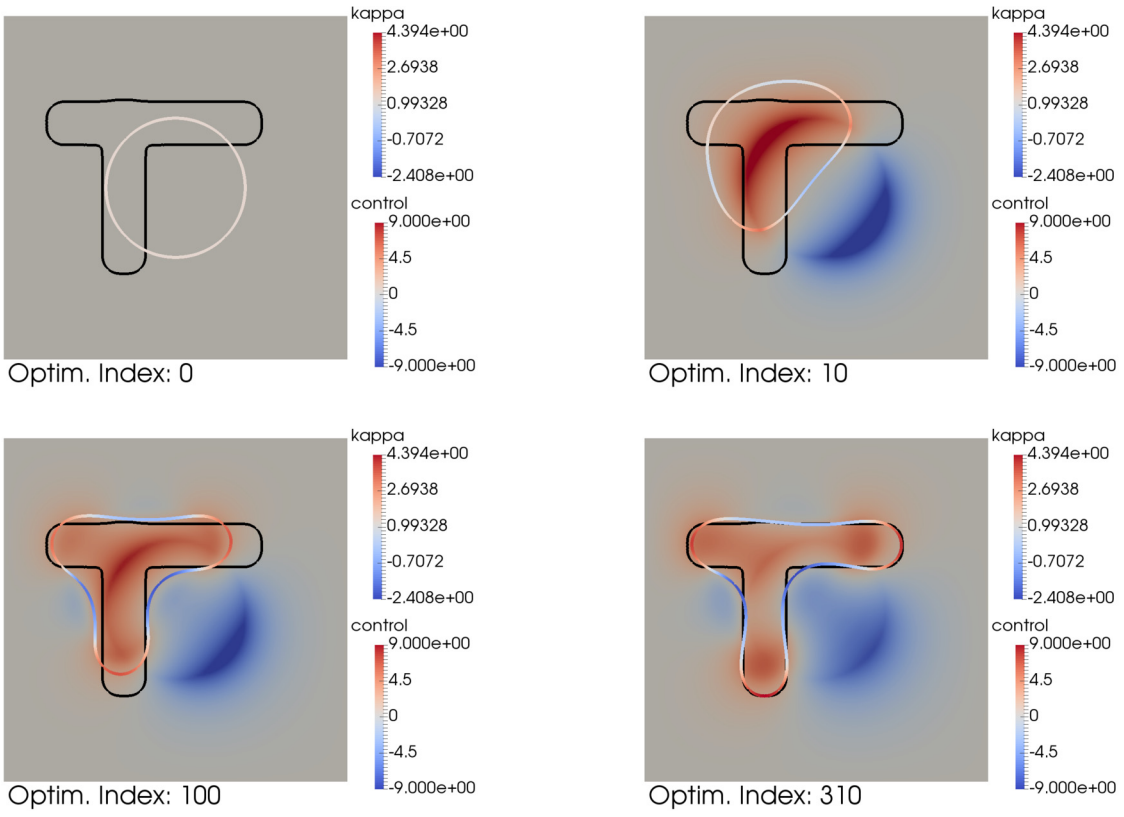

Fig. 2. Evolution of the control u versus optimization iteration (Section 5.1). The black curve is the zero level set of ϕ. The curve Γ is plotted (at the final time of the forward simulation) with color according to κ (the additive curvature). The background control function u is plotted with its own color scale. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where the state operator A is given by

$$A\boldsymbol{Z} := -(D\boldsymbol{V})\boldsymbol{Z} - \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})\boldsymbol{\nu} + (u-\kappa-\lambda)\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) \quad \text{on } \Gamma^t(\boldsymbol{V}(u)). \tag{23}$$

Note that $A\boldsymbol{Z}$ only depends on the values of $\boldsymbol{Z}$ on Γ^t. Also, the material derivative $D_{\boldsymbol{V}}\boldsymbol{Z} := \partial_t\boldsymbol{Z} + (D\boldsymbol{Z})\boldsymbol{V}$ is *independent* of any extension of $\boldsymbol{Z}$ even though the constitutive terms are not, i.e., $\partial_t\boldsymbol{Z}$ and $(D\boldsymbol{Z})\boldsymbol{V}$ do depend on the extension. Hence, (22) fully describes the evolution of $\boldsymbol{Z}$ on Γ^t, in the sense that it is independent of the values of $\boldsymbol{Z}$ outside of Γ^t.

3.3. Derivative of the cost functional and adjoint transverse field

We can now compute the derivative of $u \mapsto J(\boldsymbol{V}(u))$ by formally applying the chain rule to the composition of functions $u \mapsto \boldsymbol{V}(u)$ and $\boldsymbol{V} \mapsto J(\boldsymbol{V})$, using (9) and the equations (22) for the transverse field $\boldsymbol{Z}$. As is usual in PDE-constrained optimization, the gradient of the cost functional becomes apparent through the use of a so-called *adjoint state* $\boldsymbol{R}$. Since $\boldsymbol{R}$ is adjoint to the transverse field $\boldsymbol{Z}$, we refer to it as the *adjoint transverse field*. Also, as $\boldsymbol{Z}$ is a parabolic equation with initial conditions, $\boldsymbol{R}$ is a backward parabolic equation with terminal conditions. This is a well-known property for PDE-constrained optimization involving parabolic equations. We introduce the adjoint transverse field equations as

$$\begin{aligned} &\partial_t\boldsymbol{R} + (D\boldsymbol{R})\boldsymbol{V} + A^\star\boldsymbol{R} + \mu(t)\boldsymbol{\nu} = -\phi\boldsymbol{\nu}, \quad \text{on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t\in(0,t_f),\\ &\int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{R}(t,\cdot)\cdot\boldsymbol{\nu}(t,\cdot) = 0, \quad \text{for all } t\in(0,t_f),\\ &\boldsymbol{R}(t_f,\cdot) = 0, \quad \text{on } \Gamma^{t_f}(\boldsymbol{V}(u)), \end{aligned} \tag{24}$$

where μ is a Lagrange multiplier for the volume constraint in (24). The adjoint operator $A^\star$ is given by

$$A^\star\boldsymbol{R} := \boldsymbol{R}\mathrm{div}_\Gamma(\boldsymbol{V}) + \mathrm{div}_\Gamma[(\boldsymbol{V}\cdot\boldsymbol{\nu})\boldsymbol{R}]\boldsymbol{\nu} + (D\boldsymbol{V})^\mathsf{T}\boldsymbol{R} + \Delta_\Gamma(\boldsymbol{R}\cdot\boldsymbol{\nu})\boldsymbol{\nu} - \kappa\boldsymbol{V}(\boldsymbol{R}\cdot\boldsymbol{\nu}) \text{ on } \Gamma^t(\boldsymbol{V}(u)). \tag{25}$$

Note that, since $D_{\boldsymbol{V}}\boldsymbol{R} \equiv \partial_t\boldsymbol{R} + (D\boldsymbol{R})\boldsymbol{V}$, we do not need $\boldsymbol{R}$ extended, i.e., (24) is well-defined on the moving surface Γ^t. Recalling that

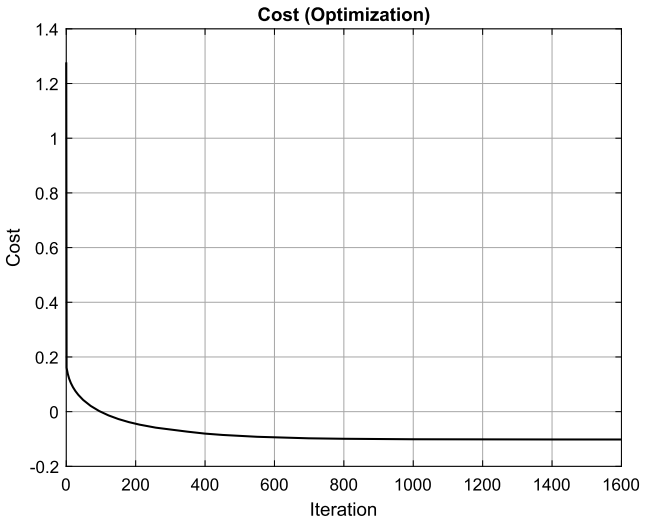

Fig. 3. Decrease of the optimization cost (56) versus optimization iteration (Section 5.1).

$$\mathcal{J}(u) = \int_0^{t_f} \int_{\Omega^t(\boldsymbol{V}(u))} \phi + \frac{\alpha}{2} \|\nabla u\|^2_{L^2(\mathfrak{D})},$$

we obtain the following result.

Theorem 1. *The derivative of $\mathcal{J}$ at u in the direction $\eta \in \mathcal{C}^\infty(\mathfrak{D})$ is given by*

$$\mathcal{J}'(u;\eta) = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \eta \boldsymbol{R} \cdot \boldsymbol{v} + \alpha \int_{\mathfrak{D}} \nabla u \cdot \nabla \eta, \tag{26}$$

where the adjoint transverse field $\boldsymbol{R}$ is the solution to (24).

Proof. Using (9) and the chain rule, we get

$$\mathcal{J}'(u;\eta) = [\delta J(\boldsymbol{V}(u))](\boldsymbol{W}) = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} (\phi \boldsymbol{v}) \cdot \boldsymbol{Z} + \alpha \int_{\mathfrak{D}} \nabla u \cdot \nabla \eta.$$

Note that $\boldsymbol{Z}$ solution of (22) is (indirectly) a function of η. Using (24) we get

$$\mathcal{J}'(u;\eta) = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -[\underbrace{\partial_t \boldsymbol{R} + (D\boldsymbol{R})\boldsymbol{V}}_{\equiv D_{\boldsymbol{V}}\boldsymbol{R}} + A^\star \boldsymbol{R} + \mu(t)\boldsymbol{v}] \cdot \boldsymbol{Z} + \alpha \int_{\mathfrak{D}} \nabla u \cdot \nabla \eta.$$

Let us define

$$\begin{aligned} L_1 &:= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -(D_{\boldsymbol{V}}\boldsymbol{R}) \cdot \boldsymbol{Z} \\ &= -\int_0^{t_f} \frac{d}{dt} \int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{R} \cdot \boldsymbol{Z} + \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{R} \cdot D_{\boldsymbol{V}}\boldsymbol{Z} + (\boldsymbol{R} \cdot \boldsymbol{Z}) \mathrm{div}_\Gamma(\boldsymbol{V}), \\ L_2 &:= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -(A^\star \boldsymbol{R} + \mu(t)\boldsymbol{v}) \cdot \boldsymbol{Z} = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -(A^\star \boldsymbol{R}) \cdot \boldsymbol{Z}, \end{aligned}$$

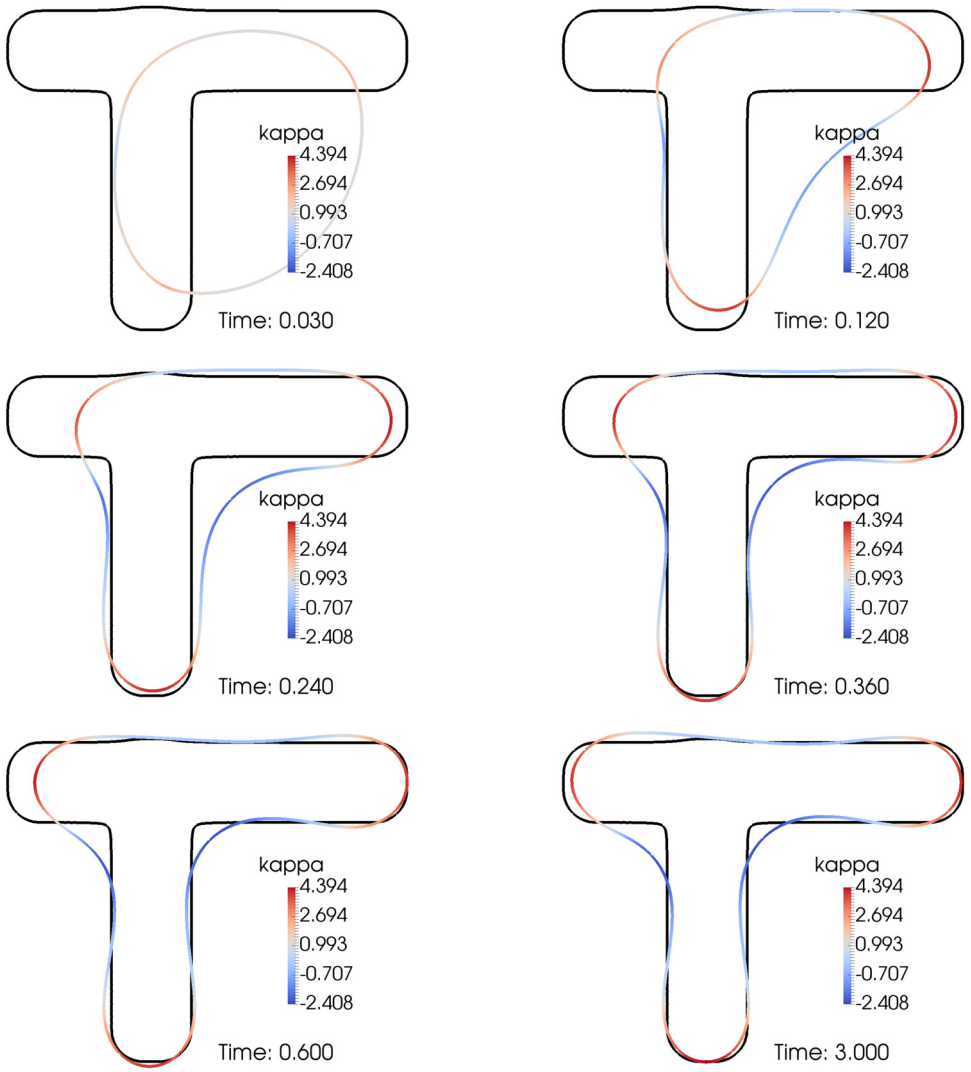

Fig. 4. Evolution of the forward simulation using the final control obtained by the optimization (Section 5.1). The black curve is the zero level set of ϕ. The curve Γ is plotted with color according to κ.

where we used (22) and standard relations for shape derivatives [41]. Proceeding with L_1, and noting that boundary terms drop because of the terminal and initial conditions in (24) and (22), we get

$$
\begin{aligned}
L_1 &= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{R} \cdot D_{\boldsymbol{V}} \boldsymbol{Z} + (\boldsymbol{R} \cdot \boldsymbol{Z}) \operatorname{div}_\Gamma(\boldsymbol{V}) \\
&= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -\boldsymbol{R} \cdot (-\boldsymbol{Z} \operatorname{div}_\Gamma(\boldsymbol{V}) + A\boldsymbol{Z} + \rho \boldsymbol{v} - \eta \boldsymbol{v}) \\
&= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -\boldsymbol{R} \cdot \{-\boldsymbol{Z} \operatorname{div}_\Gamma(\boldsymbol{V}) - (D\boldsymbol{V})\boldsymbol{Z} - \Delta_\Gamma(\boldsymbol{Z} \cdot \boldsymbol{v})\boldsymbol{v} + (u - \kappa - \lambda)\nabla_\Gamma(\boldsymbol{Z} \cdot \boldsymbol{v}) - \eta \boldsymbol{v}\},
\end{aligned}
$$

where we used the mean value zero condition in (24).

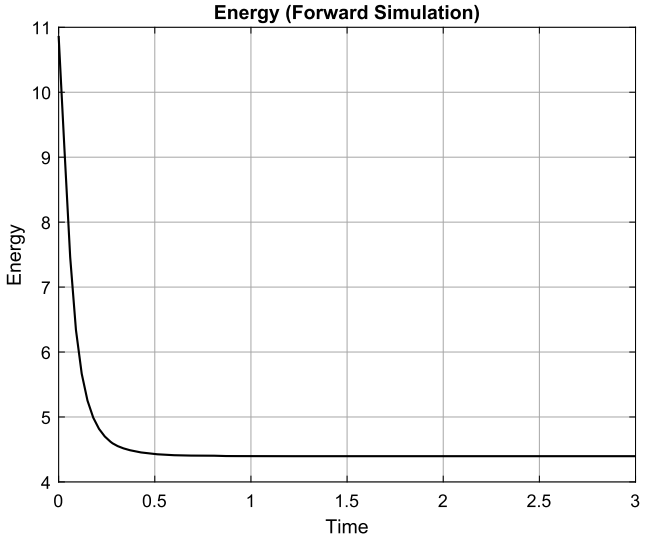

Fig. 5. Energy (4) of the forward simulation versus time (Section 5.1). The control obtained at the final optimization iteration is used here.

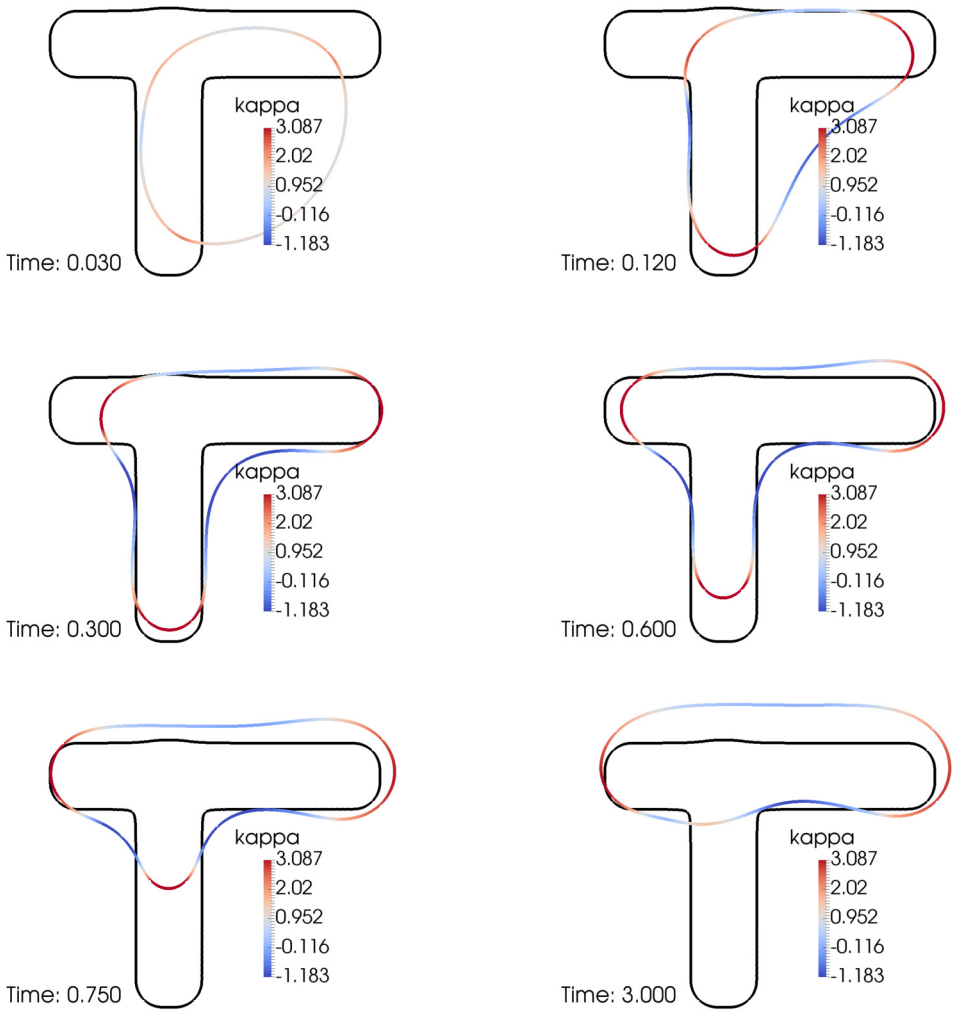

Fig. 6. Evolution of the forward simulation with perturbed initial circle with center at $(0.02, 0.0)$ (Section 5.1). The control u was taken from the final iteration of the optimization.

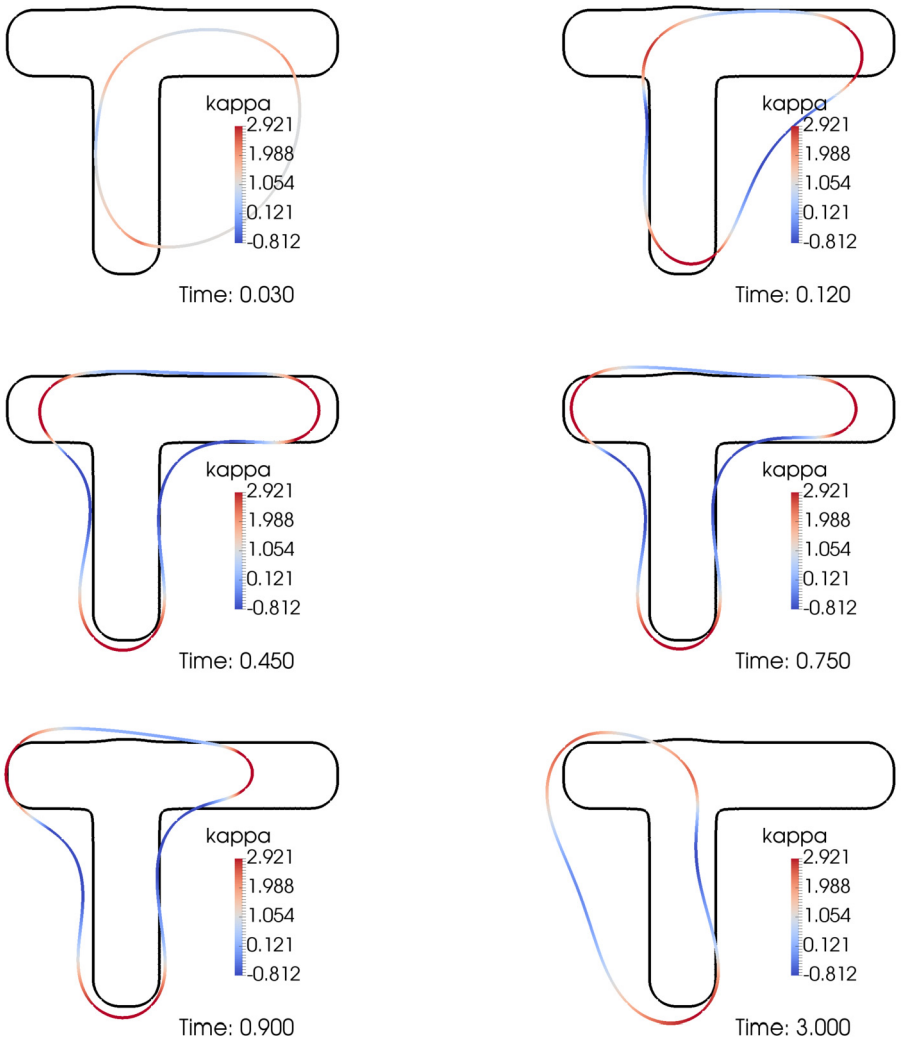

Fig. 7. Evolution of the forward simulation with perturbed initial circle with center at $(0.0, -0.02)$ (Section 5.1). The control u was taken from the final iteration of the optimization.

Using Green's second identity and the fact that $\Gamma^t(\boldsymbol{V}(u))$ is a manifold without boundary, we obtain

$$\int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu})\boldsymbol{R}\cdot\boldsymbol{\nu} = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \Delta_\Gamma(\boldsymbol{R}\cdot\boldsymbol{\nu})\boldsymbol{Z}\cdot\boldsymbol{\nu},$$

see [41, Proposition 16] or [19, Theorem 5.4.13]. Then, integration by parts gives

$$\begin{aligned}
&\int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -(u-\kappa-\lambda)\boldsymbol{R}\cdot\nabla_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) \\
&\quad= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} (\boldsymbol{Z}\cdot\boldsymbol{\nu})\mathrm{div}_\Gamma[(u-\kappa-\lambda)\boldsymbol{R}] - \kappa(\boldsymbol{Z}\cdot\boldsymbol{\nu})(u-\kappa-\lambda)\boldsymbol{R}\cdot\boldsymbol{\nu} \\
&\quad= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \boldsymbol{Z}\cdot[\mathrm{div}_\Gamma[(\boldsymbol{V}\cdot\boldsymbol{\nu})\boldsymbol{R}]\boldsymbol{\nu} - \kappa\boldsymbol{V}(\boldsymbol{R}\cdot\boldsymbol{\nu})],
\end{aligned}$$

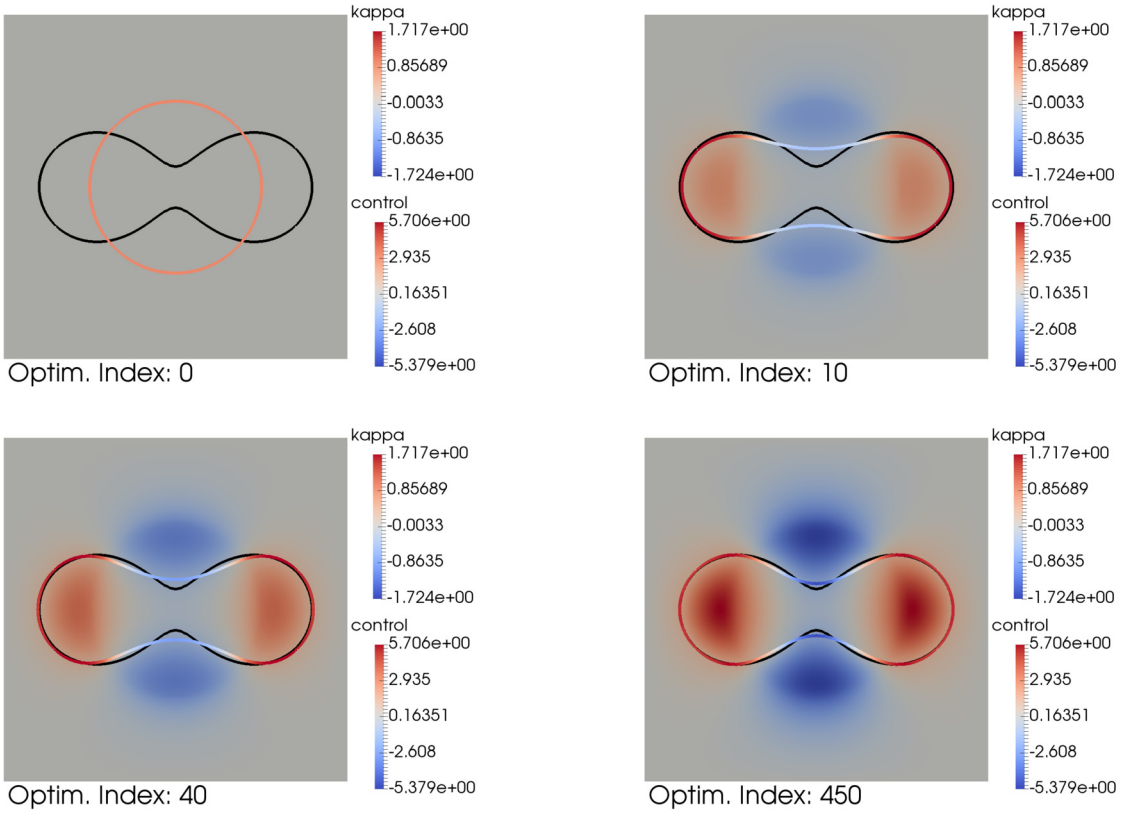

Fig. 8. Evolution of the control u versus optimization iteration (Section 5.2). The black curve is the zero level set of ϕ. The curve Γ is plotted (at the final time of the forward simulation) with color according to κ (the additive curvature). The background control function u is plotted with its own color scale.

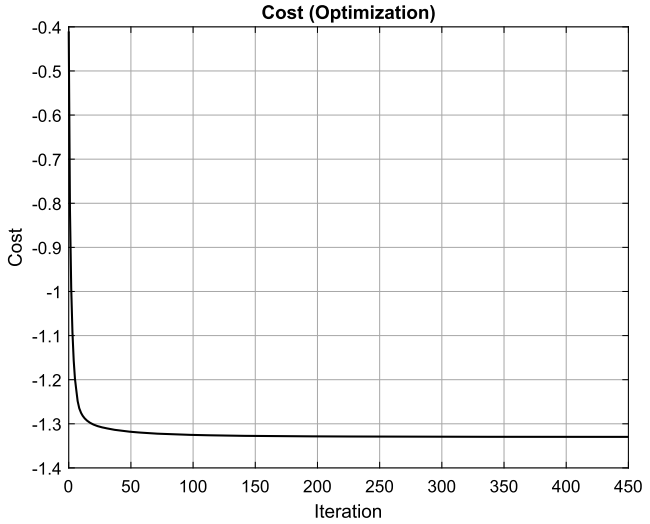

Fig. 9. Decrease of the optimization cost (56) versus optimization iteration (Section 5.2).

where we used $\boldsymbol{V} = (u - \kappa - \lambda)\boldsymbol{\nu}$ on $\Gamma^t(\boldsymbol{V}(u))$. Gathering these results, and using (25), we obtain (26). □

3.4. Normal transverse field and scalar adjoint

In Section 3.2 we have derived the system of equations (22) for the transverse field $\boldsymbol{Z}$ associated with the perturbation of $\mathcal{J}(u)$. Since the derivative (9) only depends on the normal component $\boldsymbol{Z} \cdot \boldsymbol{\nu}$ of $\boldsymbol{Z}$, it is natural to ask wether $\boldsymbol{Z} \cdot \boldsymbol{\nu}$ and the tangential component of $\boldsymbol{Z}$ may be fully decoupled. Indeed, for numerical applications it is more convenient to solve only

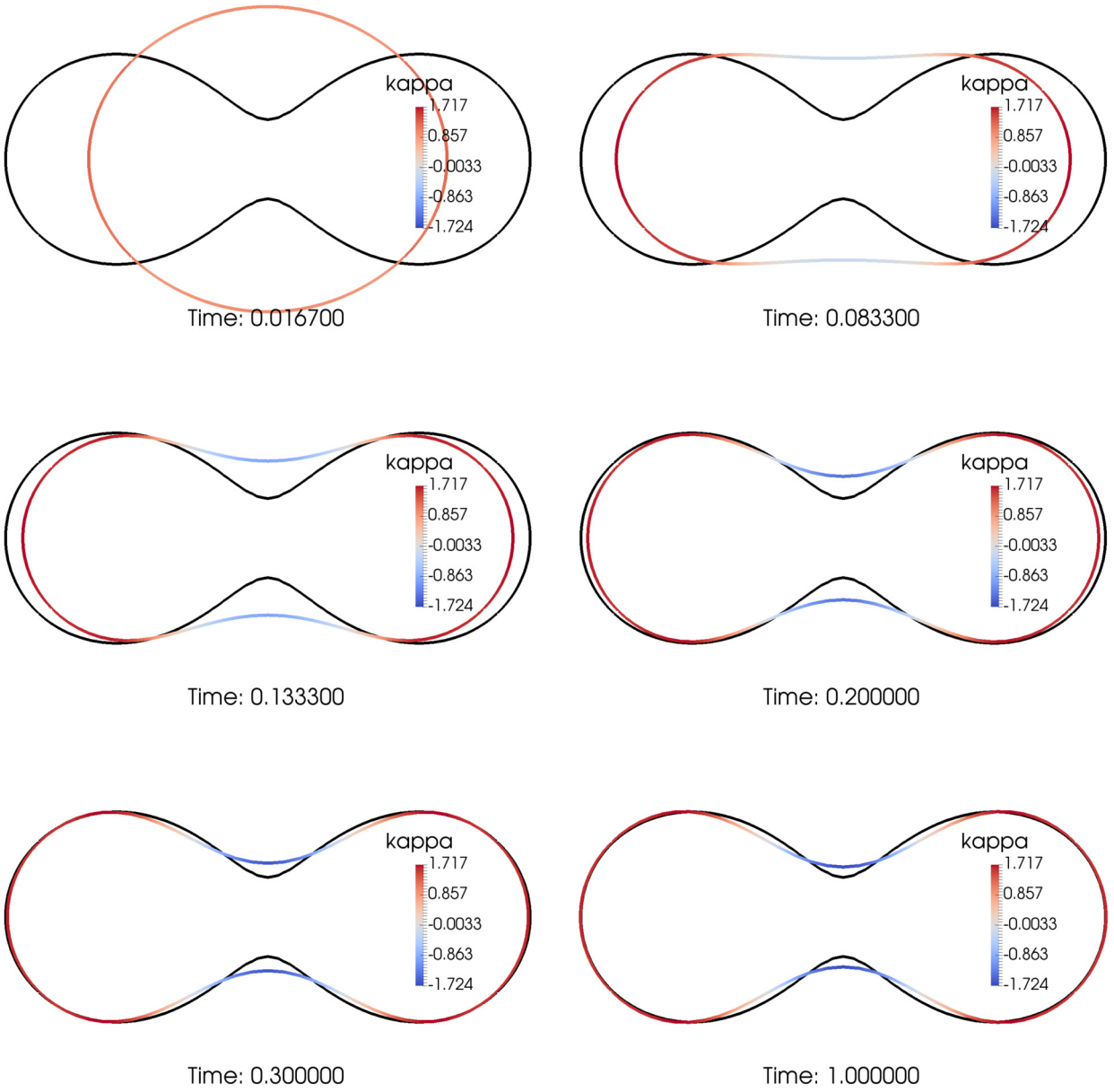

Fig. 10. Evolution of the forward simulation using the final control obtained by the optimization (Section 5.2). The black curve is the zero level set of ϕ. The curve Γ is plotted with color according to κ.

for $\boldsymbol{Z}\cdot\boldsymbol{\nu}$ instead of $\boldsymbol{Z}$. In this section we show that the normal component can indeed be decoupled and that $\boldsymbol{Z}\cdot\boldsymbol{\nu}$ satisfies an evolution equation similar to (22).

We introduce the simpler notation $Z_\nu := \boldsymbol{Z}\cdot\boldsymbol{\nu}$. Referring to (22), (23) and recalling that $D_{\boldsymbol{V}}\boldsymbol{Z} := \partial_t \boldsymbol{Z} + (D\boldsymbol{Z})\boldsymbol{V}$, we derive a stand-alone equation for Z_ν. Projecting the first equation in (22) onto $\boldsymbol{\nu}$, we get

$$\begin{aligned}\eta &= (D_{\boldsymbol{V}}\boldsymbol{Z})\cdot\boldsymbol{\nu} - \boldsymbol{\nu}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{Z} - \Delta_\Gamma(\boldsymbol{Z}\cdot\boldsymbol{\nu}) + \rho \\ &= D_{\boldsymbol{V}}(\boldsymbol{Z}\cdot\boldsymbol{\nu}) - \boldsymbol{Z}\cdot(D_{\boldsymbol{V}}\boldsymbol{\nu}) - \boldsymbol{\nu}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{Z} - \Delta_\Gamma Z_\nu + \rho \\ &= D_{\boldsymbol{V}}(\boldsymbol{Z}\cdot\boldsymbol{\nu}) + \boldsymbol{\nu}^{\mathsf{T}}(D_\Gamma\boldsymbol{V})\boldsymbol{Z} - \boldsymbol{\nu}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{Z} - \Delta_\Gamma Z_\nu + \rho \\ &= D_{\boldsymbol{V}} Z_\nu - (\boldsymbol{\nu}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{\nu})Z_\nu - \Delta_\Gamma Z_\nu + \rho,\end{aligned} \tag{27}$$

where we used that $D_{\boldsymbol{V}}\boldsymbol{\nu} = -(D_\Gamma\boldsymbol{V})^{\mathsf{T}}\boldsymbol{\nu}$ (see [41, Eqn. (5.38)]). Noting that

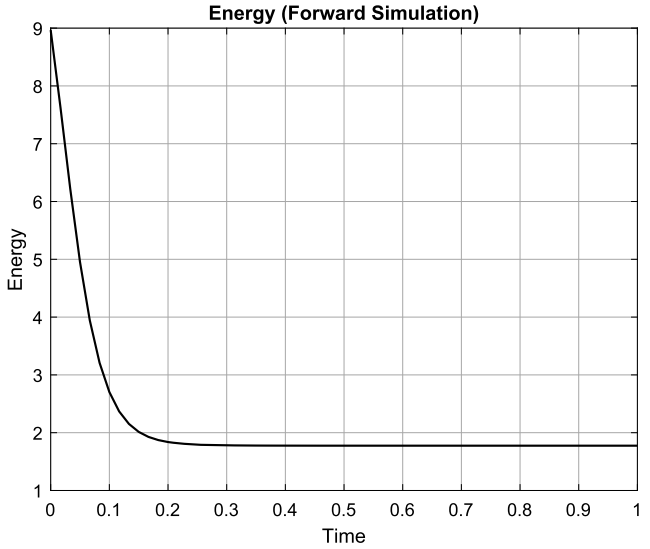

Fig. 11. Energy (4) of the forward simulation versus time (Section 5.2). The control obtained at the final optimization iteration is used here.

$$\partial_{\boldsymbol{v}}(u-\kappa)=\partial_{\boldsymbol{v}}(\boldsymbol{V}\cdot\boldsymbol{v})=\boldsymbol{v}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{v}+\boldsymbol{V}^{\mathsf{T}}\underbrace{(D\boldsymbol{v})\boldsymbol{v}}_{=\boldsymbol{0}}, \tag{28}$$

we obtain the following stand-alone evolution equation for Z_v:

$$\begin{aligned} &D_{\boldsymbol{V}}Z_v-\partial_{\boldsymbol{v}}(u-\kappa)Z_v-\Delta_\Gamma Z_v+\rho=\eta,\ \text{on }\Gamma^t(\boldsymbol{V}(u))\text{ for all }t\in(0,t_f),\\ &\int_{\Gamma^t(\boldsymbol{V}(u))} Z_v(t,\cdot)=0,\ \text{for all }t\in(0,t_f),\\ &Z_v(0,\cdot)=0,\ \text{on }\Gamma^0(\boldsymbol{V}(u)). \end{aligned} \tag{29}$$

In Section 3.3 we have introduced the adjoint transverse field $\boldsymbol{R}$ to compute the derivative (26). Since we have obtained a stand-alone system of equations (29) for the normal transverse field Z_v and since the derivative (9) only depends on Z_v, we just need a scalar adjoint to compute the derivative $\mathcal{J}'(u;\eta)$. Therefore we introduce a scalar adjoint state R satisfying the following backward evolution equation with terminal conditions:

$$\begin{aligned} &D_{\boldsymbol{V}}R+A_\Gamma^\star R+\mu(t)=-\phi,\quad \text{on }\Gamma^t(\boldsymbol{V}(u))\text{ for all }t\in(0,t_f),\\ &\int_{\Gamma^t(\boldsymbol{V}(u))} R(t,\cdot)=0,\quad \text{for all }t\in(0,t_f),\\ &R(t_f,\cdot)=0,\quad \text{on }\Gamma^{t_f}(\boldsymbol{V}(u)), \end{aligned} \tag{30}$$

where μ is a Lagrange multiplier for the volume constraint in (30) and

$$\begin{aligned} A_\Gamma^\star R&:=R\operatorname{div}_\Gamma(\boldsymbol{V})+R\partial_{\boldsymbol{v}}(u-\kappa)+\Delta_\Gamma R\\ &=R\operatorname{div}_\Gamma(\boldsymbol{V})+R\boldsymbol{v}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{v}+\Delta_\Gamma R\\ &=R\operatorname{div}[(\boldsymbol{V}\cdot\boldsymbol{v})\boldsymbol{v}]+\Delta_\Gamma R=R\partial_{\boldsymbol{v}}(u-\kappa)+R(\boldsymbol{V}\cdot\boldsymbol{v})\kappa+\Delta_\Gamma R. \end{aligned} \tag{31}$$

The well-posedness for the equations of Z_v and R may be proven using the abstract framework for parabolic PDEs on evolving spaces developed in [1,2].

We can now compute $\mathcal{J}'(u;\eta)$ using the scalar adjoint transverse field R; this provides an alternative to Theorem 1 for computing $\mathcal{J}'(u;\eta)$.

Theorem 2. *The derivative of $\mathcal{J}$ at u in the direction η is given by*

$$\mathcal{J}'(u;\eta)=\int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V}(u))} R\eta+\alpha\int_{\mathfrak{D}}\nabla u\cdot\nabla\eta, \tag{32}$$

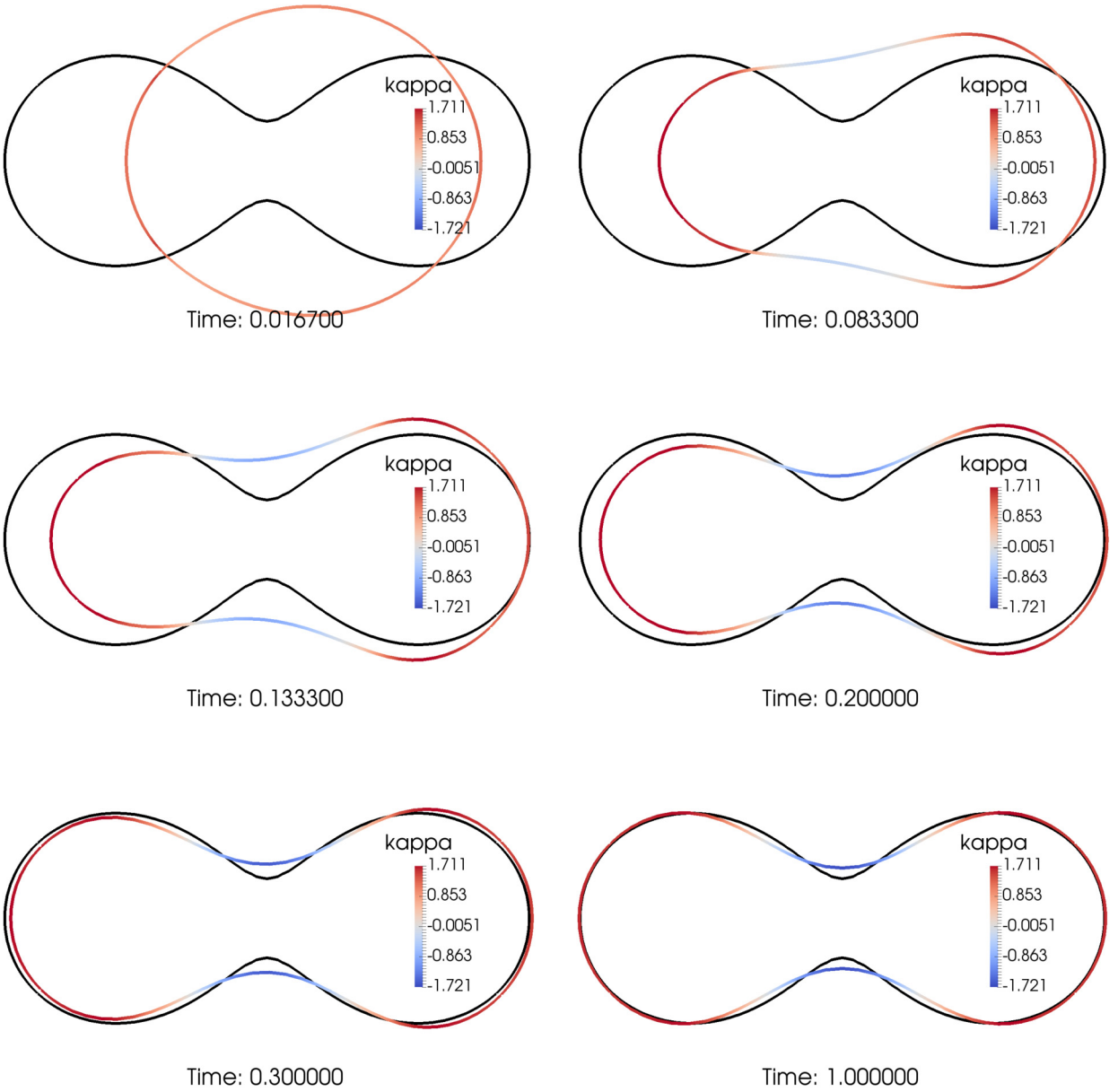

Fig. 12. Evolution of the forward simulation with perturbed initial circle with center at $(0.2, 0.0)$ (Section 5.2). The control u was taken from the final iteration of the optimization.

where the scalar adjoint state R is given by (30).

Proof. Using (9) and the chain rule, we get

$$\mathcal{J}'(u;\eta) = [\delta J(\boldsymbol{V}(u))](\boldsymbol{W}) = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} \phi Z_v + \alpha \int_{\mathfrak{D}} \nabla u \cdot \nabla \eta.$$

Note that Z_v solution of (29) is a function of η. Using (30), we get

$$\mathcal{J}'(u;\eta) = \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} -Z_v[D_{\boldsymbol{V}} R + A_\Gamma^\star R + \mu(t)] + \alpha \int_{\mathfrak{D}} \nabla u \cdot \nabla \eta.$$

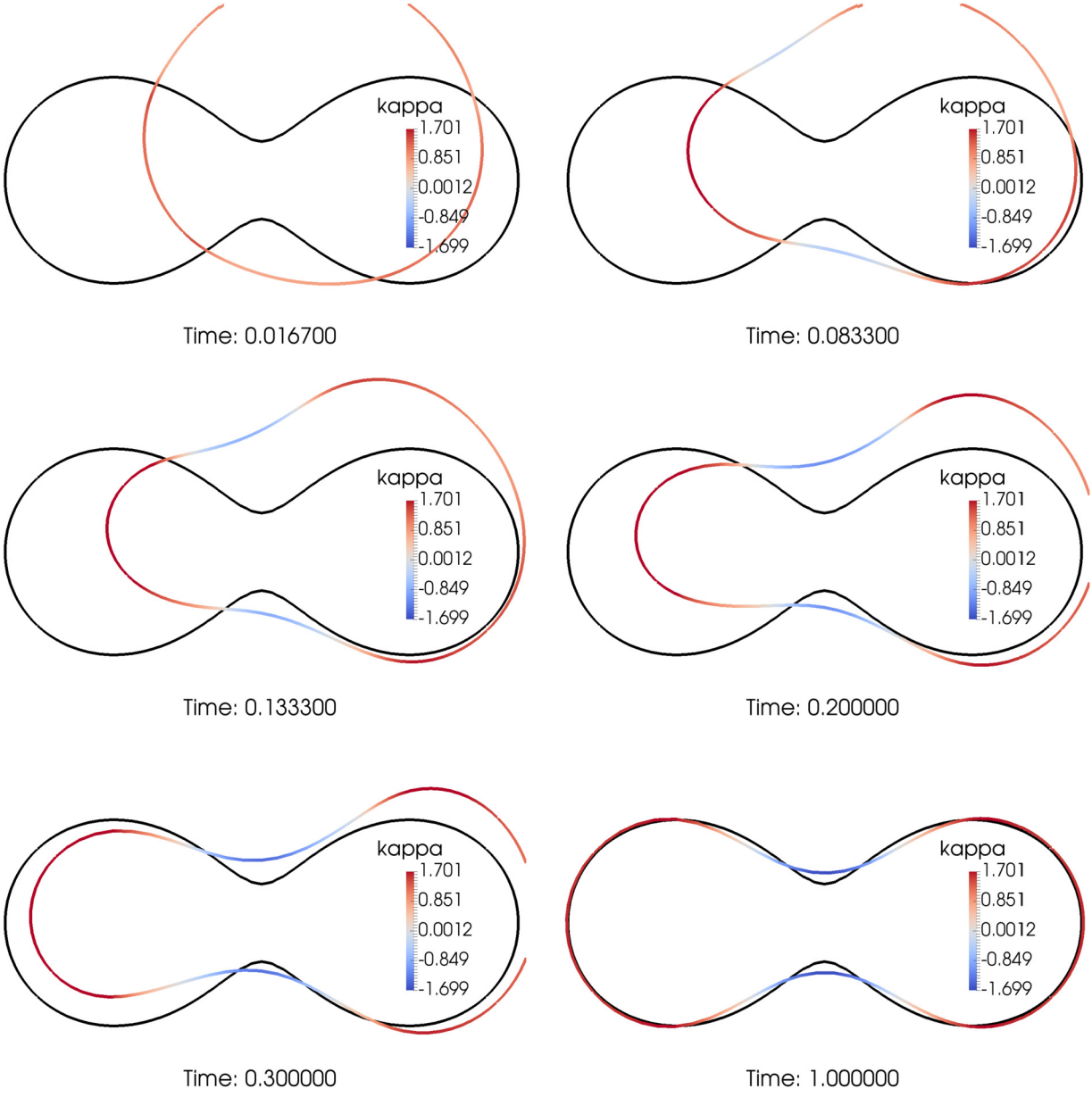

Fig. 13. Evolution of the forward simulation with perturbed initial circle with center at $(0.3, 0.3)$ (Section 5.2). The control u was taken from the final iteration of the optimization. Note that some of the simulation frames have been cropped slightly in order to "zoom-in" as much as possible.

Noting that

$$\int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V}(u))} -Z_\nu D_{\boldsymbol{V}} R = -\int_0^{t_f}\frac{d}{dt}\int_{\Gamma^t(\boldsymbol{V}(u))} Z_\nu R + \int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V}(u))} R D_{\boldsymbol{V}} Z_\nu + R Z_\nu \mathrm{div}_\Gamma(\boldsymbol{V}),$$

using the initial and terminal conditions, and (29), gives

$$\int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V}(u))} -Z_\nu D_{\boldsymbol{V}} R = \int_0^{t_f}\int_{\Gamma^t(\boldsymbol{V}(u))} R[\partial_{\boldsymbol{\nu}}(u-\kappa)Z_\nu + \Delta_\Gamma Z_\nu - \rho(t) + \eta] + R Z_\nu \mathrm{div}_\Gamma(\boldsymbol{V})$$

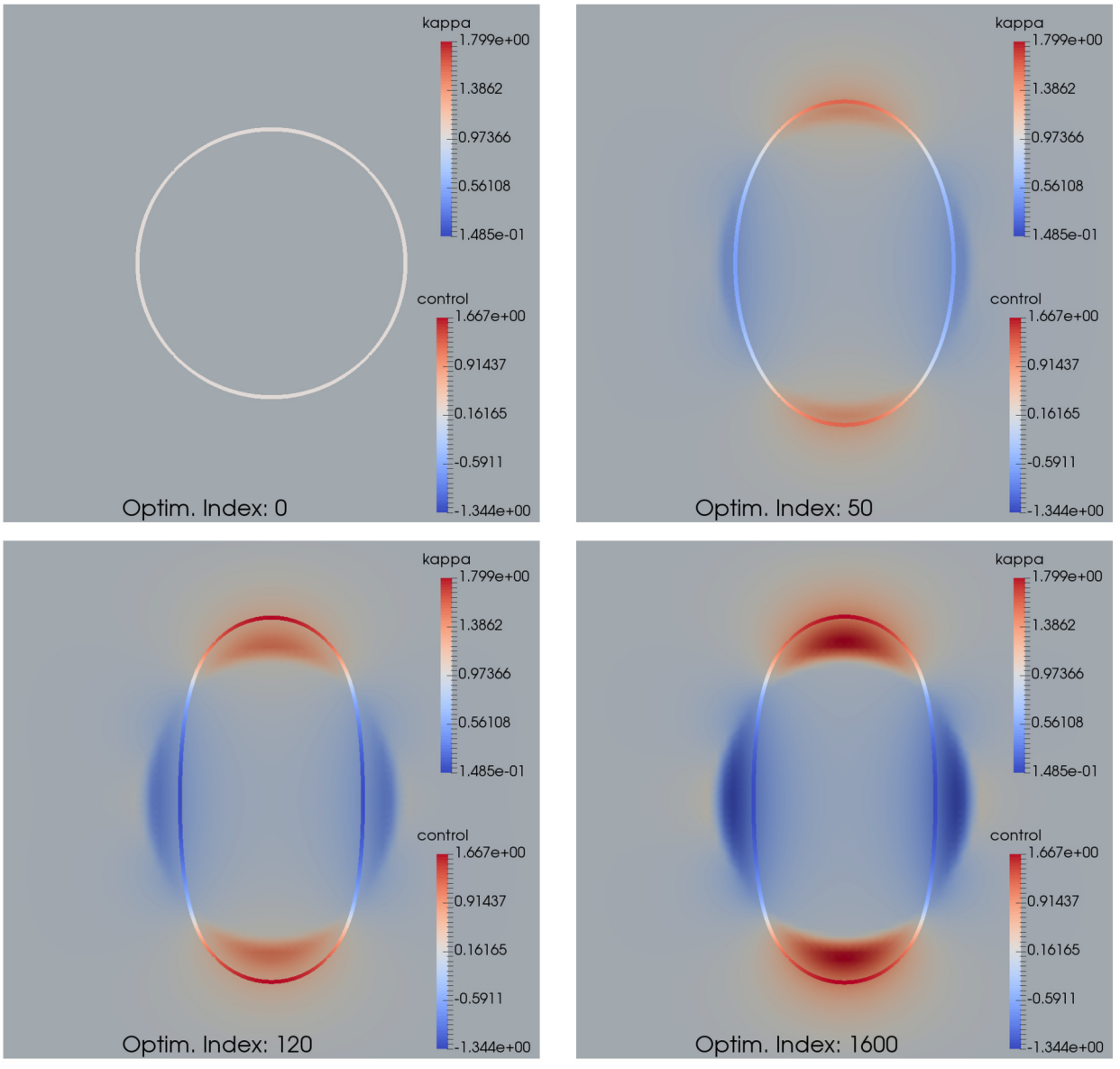

Fig. 14. Evolution of the control u versus optimization iteration (Section 5.3). The zero level set of ϕ is not shown (see Fig. 16). The curve Γ is plotted (at the final time of the forward simulation) with color according to κ (the additive curvature). The background control function u is plotted with its own color scale.

$$= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} Z_\nu [R \mathrm{div}_\Gamma(\boldsymbol{V}) + R\partial_{\boldsymbol{\nu}}(u-\kappa) + \Delta_\Gamma R] + R\eta$$

$$= \int_0^{t_f} \int_{\Gamma^t(\boldsymbol{V}(u))} Z_\nu A_\Gamma^\star R + R\eta,$$

where we used the mean value zero property in (30), and integrated by parts. The result then follows. □

It is worthwhile to check if the perturbation field $\boldsymbol{W}$ solution of (20) is such that $\boldsymbol{W} \cdot \boldsymbol{\nu}$ is mean value zero on Γ^t. Recalling (20), (22), and applying a similar argument as in (27), we have

$$D_{\boldsymbol{V}} Z_\nu - (\boldsymbol{\nu}^\mathsf{T}(D\boldsymbol{V})\boldsymbol{\nu}) Z_\nu = \boldsymbol{W} \cdot \boldsymbol{\nu}, \text{ on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t \in (0, t_f). \tag{33}$$

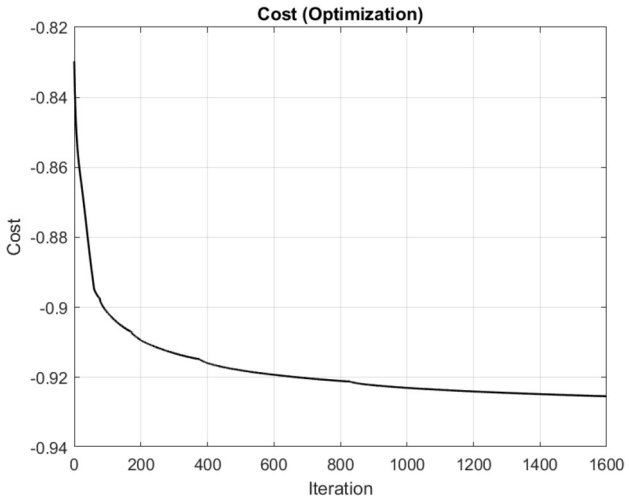

Fig. 15. Decrease of the optimization cost (56) versus optimization iteration (Section 5.3).

Integrating over Γ^t, we then get

$$\int_{\Gamma^t} D_{\boldsymbol{V}} Z_\nu + Z_\nu \mathrm{div}_\Gamma(\boldsymbol{V}) - \int_{\Gamma^t} Z_\nu \left[\mathrm{div}_\Gamma(\boldsymbol{V}) + (\boldsymbol{\nu}^\mathsf{T}(D\boldsymbol{V})\boldsymbol{\nu})\right] = \int_{\Gamma^t} \boldsymbol{W}\cdot\boldsymbol{\nu}, \tag{34}$$

or actually

$$\frac{d}{dt}\int_{\Gamma^t} Z_\nu - \int_{\Gamma^t} Z_\nu \mathrm{div}\,\boldsymbol{V} = \int_{\Gamma^t} \boldsymbol{W}\cdot\boldsymbol{\nu}, \tag{35}$$

for all $t\in(0,t_f)$. Since Z_ν is mean value zero for all $t\in(0,t_f)$, then $\boldsymbol{W}\cdot\boldsymbol{\nu}$ is also mean value zero, *provided* $\boldsymbol{V}$ is extended in a divergence free way. This is certainly possible to do since $\boldsymbol{V}\cdot\boldsymbol{\nu}$ is mean value zero. However, it is not necessary that this be done; we do not explicitly use $\boldsymbol{W}$. In other words, for the numerical scheme we use, it is not necessary that $\boldsymbol{W}$ be mean value zero.

3.5. Decoupling the normal and tangential components of the adjoint

Comparing the two expressions (26) and (32) of the derivative $\mathcal{J}'(u;\eta)$ indicates that $R\equiv\boldsymbol{R}\cdot\boldsymbol{\nu}$ on $\Gamma^t(\boldsymbol{V}(u))$. We show that the normal component $\boldsymbol{R}\cdot\boldsymbol{\nu}$ and the tangential component $\boldsymbol{R}_\Gamma$ of $\boldsymbol{R}$ can indeed be explicitly decoupled, but interestingly the proof is much less straightforward than the similar decoupling for $\boldsymbol{Z}$ performed in Section 3.4. In the case of $\boldsymbol{Z}$, the projection of equations (22) onto the normal component $\boldsymbol{\nu}$ naturally yields an independent system of equations for Z_ν, whereas for $\boldsymbol{R}$ the same projection does not reveal the decoupling of $\boldsymbol{R}\cdot\boldsymbol{\nu}$ and $\boldsymbol{R}_\Gamma$. Nevertheless, we are able to show that the tangential component $\boldsymbol{R}_\Gamma$ vanishes which yields the expected decoupling.

Define the time-dependent tangent space projection $\boldsymbol{P}=\boldsymbol{I}-\boldsymbol{\nu}\otimes\boldsymbol{\nu}$ on Γ^t, and set $\boldsymbol{R}_\Gamma := \boldsymbol{P}\boldsymbol{R}$ where $\boldsymbol{R}$ solves (24). Next, we derive an evolution equation for $\boldsymbol{R}_\Gamma$. Preliminary calculations give

$$\begin{aligned}\boldsymbol{P}D_{\boldsymbol{V}}\boldsymbol{R} &= D_{\boldsymbol{V}}\boldsymbol{R}_\Gamma + [D_{\boldsymbol{V}}(\boldsymbol{\nu}\otimes\boldsymbol{\nu})]\boldsymbol{R} = D_{\boldsymbol{V}}\boldsymbol{R}_\Gamma + D_{\boldsymbol{V}}\boldsymbol{\nu}(\boldsymbol{R}\cdot\boldsymbol{\nu}) + \boldsymbol{\nu}[(D_{\boldsymbol{V}}\boldsymbol{\nu})\cdot\boldsymbol{R}],\\ &= D_{\boldsymbol{V}}\boldsymbol{R}_\Gamma - (D_\Gamma\boldsymbol{V})^\mathsf{T}\boldsymbol{\nu}(\boldsymbol{R}\cdot\boldsymbol{\nu}) - \boldsymbol{\nu}[\boldsymbol{\nu}^\mathsf{T}(D_\Gamma\boldsymbol{V})\boldsymbol{R}_\Gamma],\end{aligned} \tag{36}$$

where we used $D_{\boldsymbol{V}}\boldsymbol{\nu} = -(D_\Gamma\boldsymbol{V})^\mathsf{T}\boldsymbol{\nu}$ (see [41, Eqn. (5.38)]). Using (25), we have

$$\boldsymbol{P}(A^\star\boldsymbol{R}) = \boldsymbol{R}_\Gamma\mathrm{div}_\Gamma(\boldsymbol{V}) + (D_\Gamma\boldsymbol{V})^\mathsf{T}\boldsymbol{R} - \kappa(\boldsymbol{R}\cdot\boldsymbol{\nu})\boldsymbol{P}\boldsymbol{V}. \tag{37}$$

Using (37) and $\boldsymbol{P}\boldsymbol{V}=\boldsymbol{0}$, applying $\boldsymbol{P}$ to (24) yields

$$\begin{aligned}D_{\boldsymbol{V}}\boldsymbol{R}_\Gamma + (D_\Gamma\boldsymbol{V})^\mathsf{T}\boldsymbol{P}\boldsymbol{R} - [\boldsymbol{\nu}^\mathsf{T}(D_\Gamma\boldsymbol{V})\boldsymbol{R}_\Gamma]\boldsymbol{\nu} + \boldsymbol{R}_\Gamma\mathrm{div}_\Gamma(\boldsymbol{V}) = \boldsymbol{0},\ \text{on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t\in(0,t_f),\\ \boldsymbol{R}_\Gamma(t_f,\cdot) = \boldsymbol{0},\ \text{on } \Gamma^{t_f}(\boldsymbol{V}(u)),\end{aligned} \tag{38}$$

or equivalently

$$\begin{aligned}D_{\boldsymbol{V}}\boldsymbol{R}_\Gamma + \left[(D_\Gamma\boldsymbol{V})^\mathsf{T} - (\boldsymbol{\nu}\otimes\boldsymbol{\nu})(D_\Gamma\boldsymbol{V}) + \boldsymbol{I}\mathrm{div}_\Gamma(\boldsymbol{V})\right]\boldsymbol{R}_\Gamma = \boldsymbol{0},\ \text{on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t\in(0,t_f),\\ \boldsymbol{R}_\Gamma(t_f,\cdot) = \boldsymbol{0},\ \text{on } \Gamma^{t_f}(\boldsymbol{V}(u)),\end{aligned} \tag{39}$$

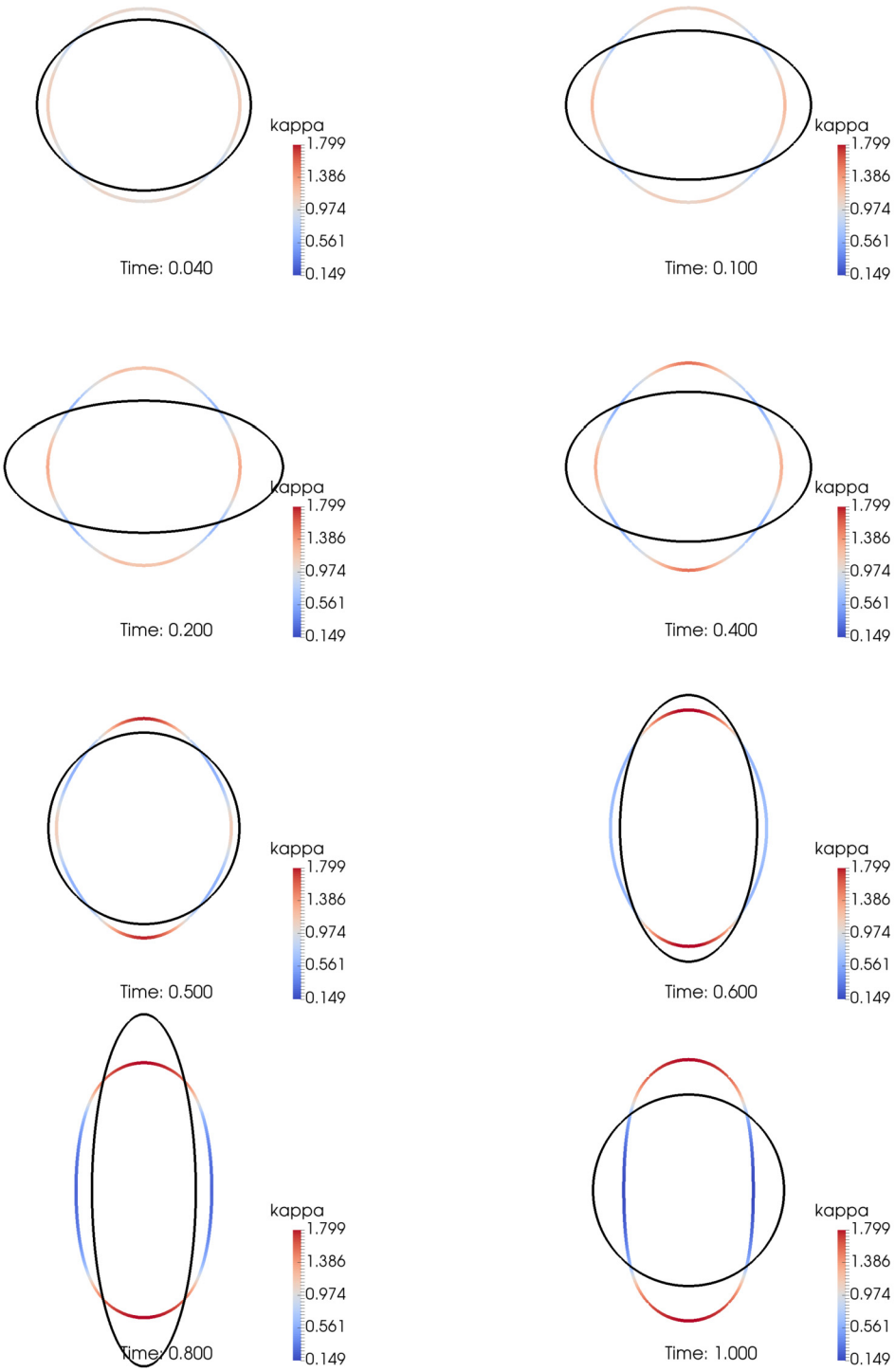

Fig. 16. Evolution of the forward simulation using the final control obtained by the optimization (Section 5.3). The black curve is the zero level set of the time-varying ϕ. The curve Γ is plotted with color according to κ.

which can be viewed as a simple ODE for $\boldsymbol{R}_\Gamma$ with zero right-hand-side and vanishing initial condition. Thus, $\boldsymbol{R}_\Gamma \equiv \boldsymbol{0}$ for all $\boldsymbol{x} \in \Gamma^t$ and all $t \in (0, t_f)$.

Hence $\boldsymbol{R} = r\boldsymbol{v}$ for some scalar valued function r, so then (24) reduces to

$$D_{\boldsymbol{V}}(r\boldsymbol{v}) + A^\star(r\boldsymbol{v}) + \mu(t)\boldsymbol{v} = -\phi\boldsymbol{v}, \quad \text{on } \Gamma^t(\boldsymbol{V}(u)) \text{ for all } t \in (0, t_f),$$

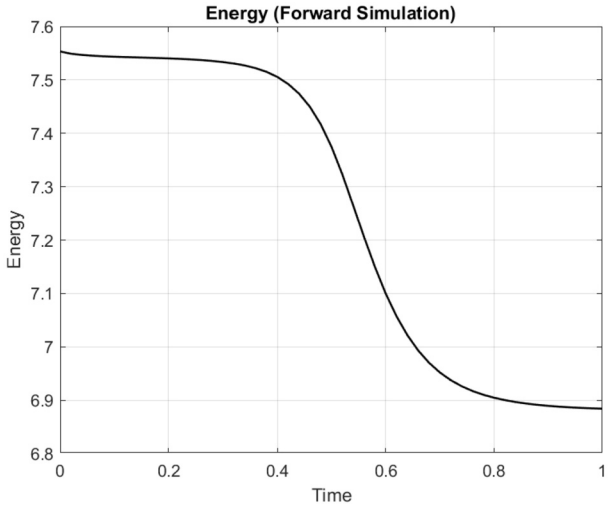

Fig. 17. Energy (4) of the forward simulation versus time (Section 5.3). The control obtained at the final optimization iteration is used here.

$$\begin{aligned} \int_{\Gamma^t(\boldsymbol{V}(u))} r(t,\cdot) &= 0, \quad \text{for all } t \in (0,t_f), \\ r(t_f,\cdot) &= 0, \quad \text{on } \Gamma^{t_f}(\boldsymbol{V}(u)). \end{aligned} \tag{40}$$

Upon noting that $\operatorname{div}_\Gamma[(\boldsymbol{V}\cdot\boldsymbol{\nu})r\boldsymbol{\nu}] = (\boldsymbol{V}\cdot\boldsymbol{\nu})r\operatorname{div}_\Gamma(\boldsymbol{\nu}) = (\boldsymbol{V}\cdot\boldsymbol{\nu})r\kappa$, the first line of (40) becomes

$$\begin{aligned} -\phi\boldsymbol{\nu} &= \boldsymbol{\nu}D_{\boldsymbol{V}}r + rD_{\boldsymbol{V}}\boldsymbol{\nu} + r\boldsymbol{\nu}\operatorname{div}_\Gamma(\boldsymbol{V}) + (\boldsymbol{V}\cdot\boldsymbol{\nu})r\kappa\boldsymbol{\nu} + r(D\boldsymbol{V})^{\mathsf{T}}\boldsymbol{\nu} + \boldsymbol{\nu}\Delta_\Gamma r - \kappa\boldsymbol{V}r + \mu(t)\boldsymbol{\nu}, \\ &= \boldsymbol{\nu}D_{\boldsymbol{V}}r - r(D_\Gamma\boldsymbol{V})^{\mathsf{T}}\boldsymbol{\nu} + r\boldsymbol{\nu}\operatorname{div}_\Gamma(\boldsymbol{V}) + (\boldsymbol{V}\cdot\boldsymbol{\nu})r\kappa\boldsymbol{\nu} + r(D\boldsymbol{V})^{\mathsf{T}}\boldsymbol{\nu} + \boldsymbol{\nu}\Delta_\Gamma r - \kappa\boldsymbol{V}r + \mu(t)\boldsymbol{\nu}, \end{aligned} \tag{41}$$

on $\Gamma^t(\boldsymbol{V}(u))$ for all $t \in (0,t_f)$. Taking the dot product with $\boldsymbol{\nu}$, and using (28), we arrive at

$$\begin{aligned} -\phi &= D_{\boldsymbol{V}}r + r\operatorname{div}_\Gamma(\boldsymbol{V}) + (\boldsymbol{V}\cdot\boldsymbol{\nu})r\kappa + r\boldsymbol{\nu}^{\mathsf{T}}(D\boldsymbol{V})\boldsymbol{\nu} + \Delta_\Gamma r - \kappa(\boldsymbol{V}\cdot\boldsymbol{\nu})r + \mu(t) \\ &= D_{\boldsymbol{V}}r + r\operatorname{div}_\Gamma(\boldsymbol{V}) + r\partial_{\boldsymbol{\nu}}(u-\kappa) + \Delta_\Gamma r + \mu(t), \end{aligned} \tag{42}$$

on $\Gamma^t(\boldsymbol{V}(u))$ for all $t \in (0,t_f)$. Therefore, r solves the scalar adjoint equation in (30). Since the solution is unique we obtain $r \equiv R$ and consequently $R \equiv \boldsymbol{R}\cdot\boldsymbol{\nu}$.

4. Optimization method

We present the basic method for computing a local minimizer of (7). This discussion is formal in the sense that we assume no topological changes in the forward problem, that the objective functionals are sufficiently smooth, etc. Algorithm 1 gives the optimization scheme for the continuous problem: it is based on Theorem 2 and on a standard steepest descent method. In the subsequent sections, we take the optimize then discretize approach, i.e., we follow Algorithm 1 and simply replace each step by the fully discrete finite element version.

4.1. Discretization

For simplicity, we will have a uniform time discretization, but this is not necessary. Thus, we have a set of discrete time points $\{t_i\}_{i=0}^N$, where $t_0 = 0$, $t_N = t_f$, and $t_{i+1} - t_i = \delta t$ (for all i). Moreover, the surface evolution will be captured by a finite set of surfaces $\{\Gamma^i\}_{i=0}^N$, the evolution of which is specified in Section 4.2. Each Γ^i is a triangulated surface with curved triangles. For simplicity of notation, we do not use a subscript h for Γ, u, η, and R. In other words, all solution variables here are discrete, finite element functions.

4.1.1. Standard finite element spaces

Let $\mathcal{T}_h$ be a conforming, shape regular triangulation of $\mathfrak{D}$ (that is not fitted to Γ^i, for all i), and let $\mathcal{G}_h^i$ be a conforming, shape regular curved *surface* triangulation of Γ^i.

Let $\mathbb{S}_h^q$ be the space of continuous Lagrange finite elements over $\mathcal{T}_h$ of degree $q \geq 1$, with standard interpolation operator I_h^q; moreover, let $\mathbb{S}_h^0$ be the space of piecewise constants on $\mathcal{T}_h$ with projection operator I_h^0. In addition, let $S_h^q(\Gamma^i)$ be the space of continuous Lagrange finite elements over $\mathcal{G}_h^i$ of degree $q \geq 1$, with standard interpolation operator $\pi_h^{i,q}$; moreover,

Algorithm 1: Continuous gradient flow algorithm.

Define ϕ, $\alpha > 0$, $t_f > 0$, TOL > 0, and initialize Ω_0^0 or (equivalently) Γ_0^0. Set an initial guess for the control u_0, e.g., $u_0 := 0$. For $k \geq 0$, do the following.

1. Forward problem. Solve (3) on the interval $(0, t_f]$ using u_k, i.e., compute Γ_k^t for all $t \in (0, t_f]$.
2. Adjoint problem. Solve (30) on Γ_k^t to obtain $R_k(t)$ for all $t \in [0, t_f)$.
3. Descent Direction. Solve the following variational problem: find $\eta_{k+1} \in H_0^1(\mathfrak{D})$ such that

$$a\left(\eta_{k+1}, \xi\right) = -\mathcal{J}'(u_k; \xi) = -\int_0^{t_f} \int_{\Gamma_k^t} R_k \xi - \alpha \int_{\mathfrak{D}} \nabla u_k \cdot \nabla \xi, \quad \forall \xi \in H_0^1(\mathfrak{D}), \tag{43}$$

   where $a(\cdot, \cdot)$ is some useful positive-definite symmetric bilinear form (inner product); see (62).
4. Backtracking line-search:
   (a) Initialize the step-size $\rho_{k+1} := 1$.
   (b) Trial update. $u_{k+1} := u_k + \rho_{k+1} \eta_{k+1}$.
   (c) Check for cost decrease. If

$$\mathcal{J}(u_{k+1}) < \mathcal{J}(u_k) \tag{44}$$

   then we accept u_{k+1} and go to step 5. Otherwise, replace $\rho_{k+1} \leftarrow \rho_{k+1}/2$, and go back to step 4(b), and repeat.
5. Terminate. If $|\mathcal{J}(u_{k+1}) - \mathcal{J}(u_k)| <$ TOL, then stop. Otherwise, $k \leftarrow k+1$, and go to step 1.

let $S_h^0(\Gamma^i)$ be the space of piecewise constants on $\mathcal{G}_h^i$ with projection operator $\pi_h^{i,0}$. We usually simplify notation and drop the polynomial degree, i.e., $\pi_h^{i,0} \equiv \pi_h^i$. We also have the L^2 projection operator: $\mathrm{P}_h^q : L^2(\mathfrak{D}) \to \mathbb{S}_h^q$, for any $q \geq 0$.

Next, we have the following discrete, bulk finite element spaces:

$$\mathbb{U}_h(g) := \mathbb{S}_h^q \cap [g + H_0^1(\mathfrak{D})], \tag{45}$$

and the following discrete surface finite element spaces:

$$Y_h(\Gamma^i) := S_h^q(\Gamma^i), \tag{46}$$

where $q \geq 1$. In dealing with the evolving surfaces, we require the "obvious" transfer operator: $\Pi_{i-1}^i : Y_h(\Gamma^{i-1}) \to Y_h(\Gamma^i)$, i.e., if $\boldsymbol{v} \in [Y_h(\Gamma^{i-1})]^n$, then $\widetilde{\boldsymbol{v}} := \Pi_{i-1}^i \boldsymbol{v} := \pi_h^i[\boldsymbol{v} \circ \boldsymbol{\Phi}^{-1}] \in [Y_h(\Gamma^i)]^n$, where $\boldsymbol{\Phi} : \Gamma^{i-1} \to \Gamma^i$.

4.1.2. Geometric quantities

We have access to the additive curvature κ, as well as the normal velocity through our method for the forward problem (Section 4.2). We also need the so-called *shape operator* $D_\Gamma \boldsymbol{v}$ due to the following identity (48). Suppose $\boldsymbol{v}$ is extended constant in the normal direction, then $\partial_{\boldsymbol{v}} \boldsymbol{v} = (D\boldsymbol{v})\boldsymbol{v} = \boldsymbol{0}$ and $\kappa = \operatorname{div} \boldsymbol{v}$; see [41]. Furthermore,

$$\begin{aligned} 0 &= \operatorname{div}[(D\boldsymbol{v})\boldsymbol{v}] = \partial_i[(\partial_j v_i) v_j] = (\partial_j \partial_i v_i) v_j + (\partial_j v_i)(\partial_i v_j) \\ &= \partial_{\boldsymbol{v}}(\operatorname{div} \boldsymbol{v}) + (D\boldsymbol{v})^{\mathsf{T}} : (D\boldsymbol{v}) = \partial_{\boldsymbol{v}} \kappa + (D\boldsymbol{v}) : (D\boldsymbol{v}) \\ &= \partial_{\boldsymbol{v}} \kappa + |D_\Gamma \boldsymbol{v}|^2, \end{aligned} \tag{47}$$

where we used that $D\boldsymbol{v} = D_\Gamma \boldsymbol{v}$ (the shape operator) is a symmetric matrix. Therefore,

$$\partial_{\boldsymbol{v}} \kappa = -|D_\Gamma \boldsymbol{v}|^2. \tag{48}$$

In other words, we do not need an extension to compute $\partial_{\boldsymbol{v}} \kappa$. Note that $|D_\Gamma \boldsymbol{v}|^2$ is also equal to the sum of squares of the principal curvatures; see [12, p. 317].

4.2. Discretization of the forward problem

We use a variation of the Barrett-Garcke-Nürnberg method [9,8]. The sequence of $\{\Gamma^i\}$ will be represented by the following set of finite element functions: $\{\boldsymbol{x}^i\}_{i=0}^N$, where $\boldsymbol{x}^0 \equiv \mathrm{id}_{\Gamma^0} \in [Y_h(\Gamma^0)]^n$, $\boldsymbol{X}^{i+1} \in [Y_h(\Gamma^i)]^n$, for $0 \leq i \leq N-1$, and $\Gamma^{i+1} := \pi_h^i[\boldsymbol{X}^{i+1}(\Gamma^i)]$, for $0 \leq i \leq N-1$.

4.2.1. Statement of the method

More specifically, given $u \in \mathbb{U}_h(0)$, Γ^i, we find $\boldsymbol{X}^{i+1} \in [Y_h(\Gamma^i)]^n$, $\hat{\kappa}^{i+1} \in Y_h(\Gamma^i)$, $\lambda^{i+1} \in \mathbb{R}$ such that

$$\left(D_\Gamma \boldsymbol{X}^{i+1}, D_\Gamma \boldsymbol{Y}\right)_{\Gamma^i} - \left(\hat{\kappa}^{i+1}, \boldsymbol{Y} \cdot \boldsymbol{\nu}\right)_{\Gamma^i} + \lambda^{i+1} (1, \boldsymbol{Y} \cdot \boldsymbol{\nu})_{\Gamma^i} = 0, \quad \forall \boldsymbol{Y} \in [Y_h(\Gamma^i)]^n,$$

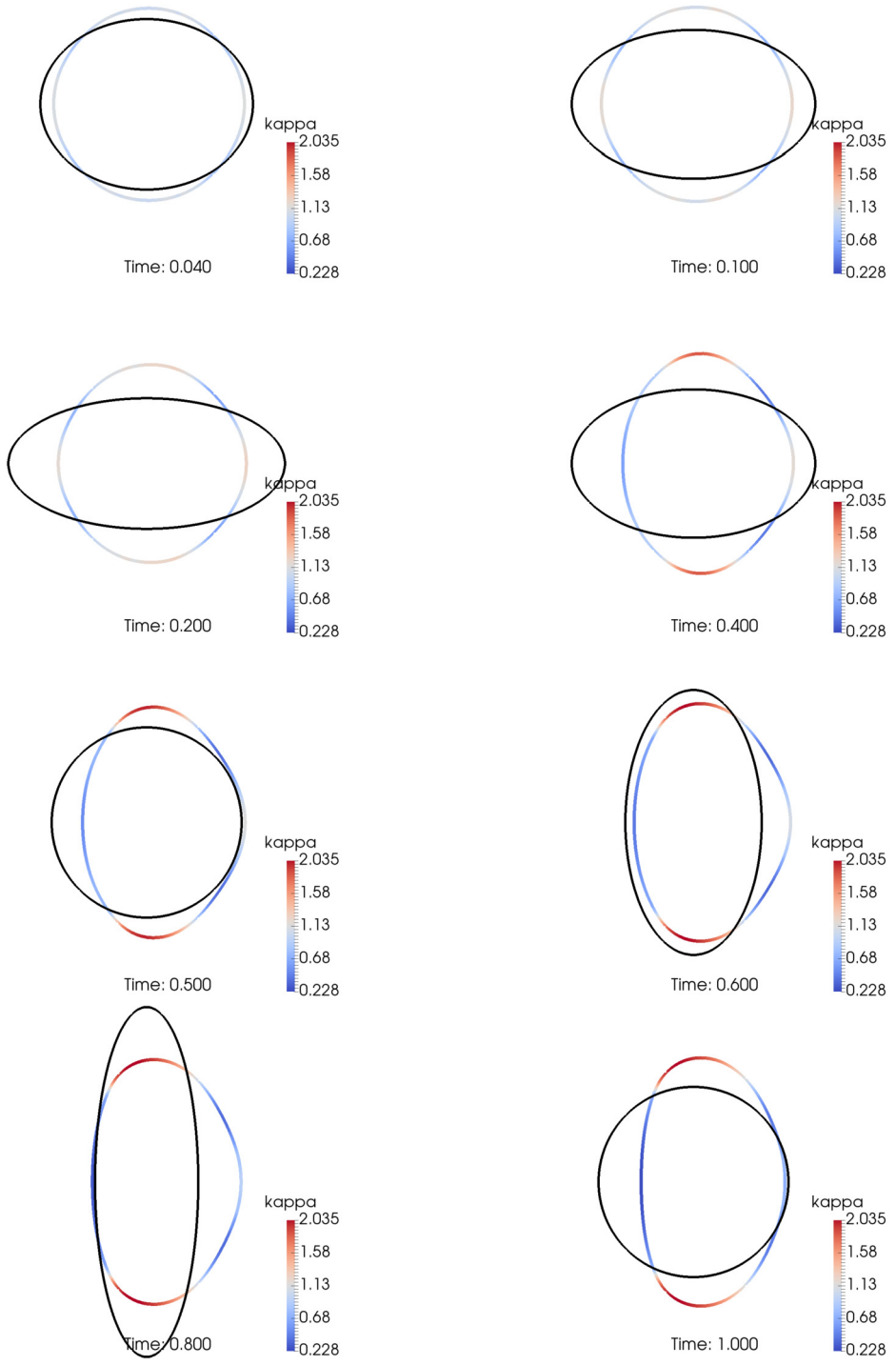

Fig. 18. Evolution of the forward simulation with perturbed initial circle with center at $(0.02, 0.0)$ (Section 5.3). The control u was taken from the final iteration of the optimization.

$$
\begin{aligned}
-\left(\boldsymbol{X}^{i+1}\cdot\boldsymbol{\nu},v\right)_{\Gamma^i}-\delta t\left(\hat{\kappa}^{i+1},v\right)_{\Gamma^i}&=-\left(\mathrm{id}_{\Gamma^i}\cdot\boldsymbol{\nu},v\right)_{\Gamma^i}-\delta t\left(\pi_h^i u,v\right)_{\Gamma^i},\quad \forall v\in Y_h(\Gamma^i),\\
\left(\boldsymbol{X}^{i+1}\cdot\boldsymbol{\nu},1\right)_{\Gamma^i}&=\left(\mathrm{id}_{\Gamma^i}\cdot\boldsymbol{\nu},1\right)_{\Gamma^i},\\
\kappa^{i+1}&:=\hat{\kappa}^{i+1}-\lambda^{i+1}\in Y_h(\Gamma^i),
\end{aligned}
$$

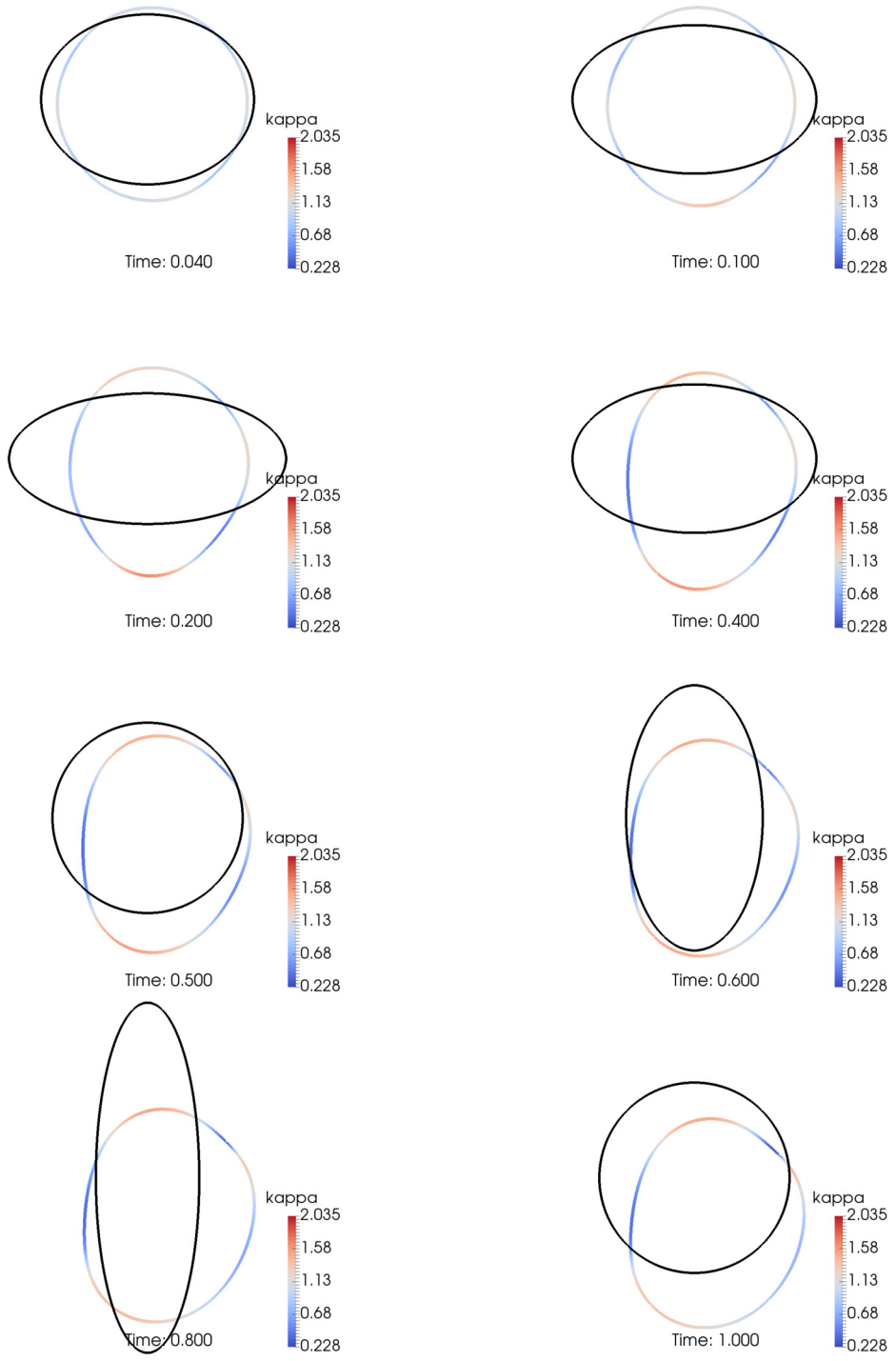

Fig. 19. Evolution of the forward simulation with perturbed initial circle with center at $(0.04, -0.04)$ (Section 5.3). The control u was taken from the final iteration of the optimization.

$$\boldsymbol{V}^{i+1} := \pi_h^i \left[\frac{\boldsymbol{X}^{i+1} - \mathrm{id}_{\Gamma^i}}{\delta t} \right] \in [Y_h(\Gamma^i)]^n, \tag{49}$$

where κ^{i+1} is an approximation of the additive curvature of Γ^i. The main distinction here between this method and the various methods of Barrett-Garcke-Nürnberg [9,8], is that we *do not* use mass-lumped inner products. The reason for this is

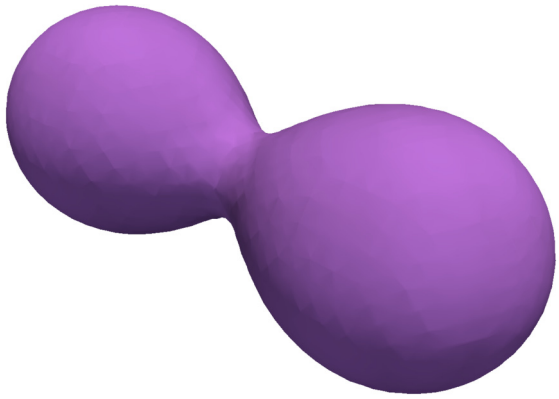

Fig. 20. Illustration of the zero level set of (65) (Section 5.4).

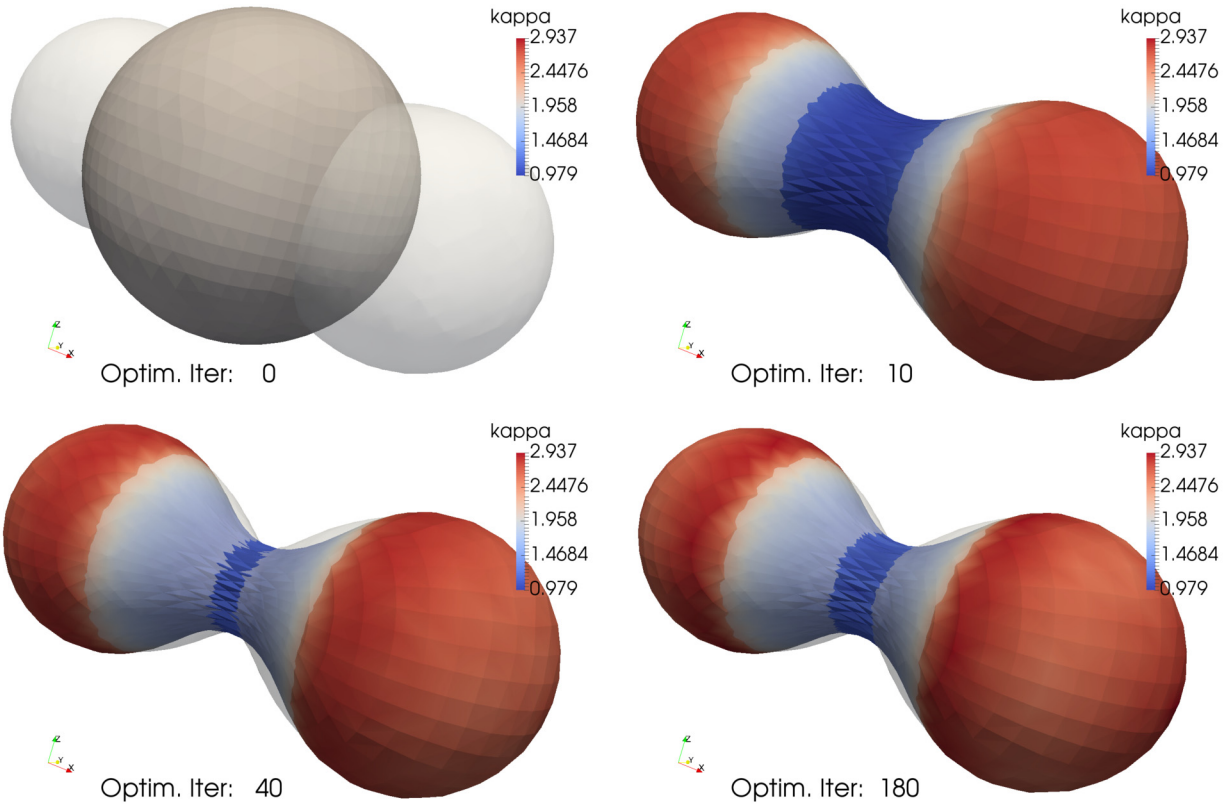

Fig. 21. Evolution of Γ^{t_f} versus optimization iteration (Section 5.4). The white (transparent) surface is the zero level set of ϕ. The surface Γ is plotted (at the final time of the forward simulation) with color according to κ (the additive curvature). Note that the coloring is affected by the transparent zero level set.

we compute the shape operator in (48) explicitly using the curved surface mesh (this is for solving the adjoint problem). The methods in [9,8] are entirely formulated on piecewise linear surface triangulations.

Numerical experiments demonstrate that the solvability of (49) is quite robust; see [14,17,40] for similar methods with proven well-posedness and stability results. The enclosed volume is well-preserved, even when fairly large time-steps are used. In addition, the mesh quality does not degenerate, i.e., the ratio of largest to smallest element remains uniformly bounded. Indeed, the time-step does not appear to greatly affect this. A full theoretical justification of (49) is beyond the scope of this paper, but is a point of future work.

4.2.2. Relation to continuous formulation

Let us now relate the formulation (49) to the continuous problem. Taking the mesh size to zero, and integrating the first equation in (49) by parts, we see that

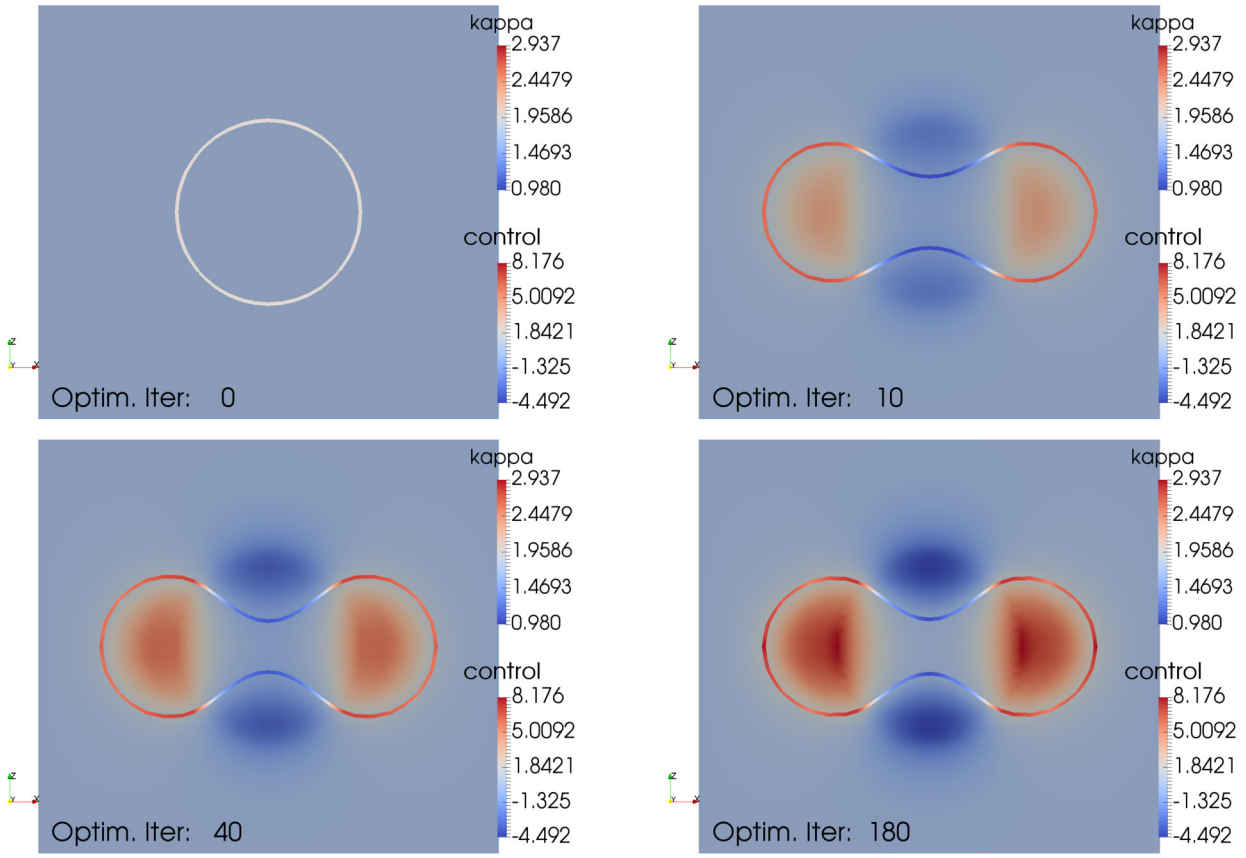

Fig. 22. Evolution of the control u versus optimization iteration (Section 5.4); a 2-D slice is shown. The slice of Γ, which is a curve, is plotted (at the final time of the forward simulation) with color according to κ. The background control function u is plotted with its own color scale.

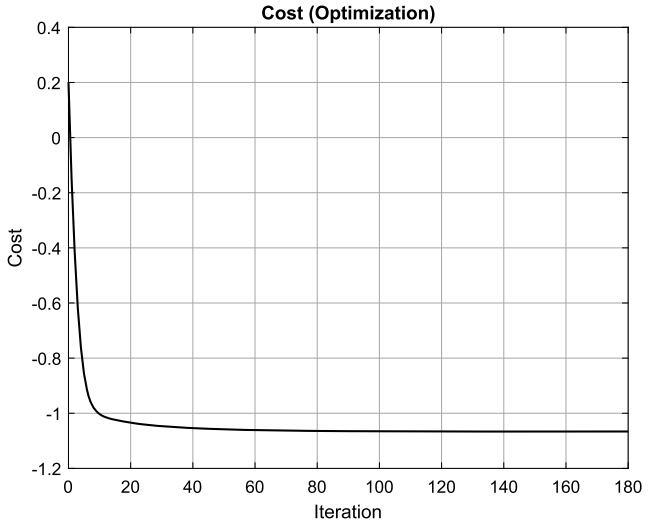

Fig. 23. Decrease of the optimization cost (56) versus optimization iteration (Section 5.4).

$$-\left(\Delta_\Gamma \boldsymbol{X}^{i+1}, \boldsymbol{Y}\right)_{\Gamma^i} = \left(\hat{\kappa}^{i+1} - \lambda^{i+1}, \boldsymbol{Y}\cdot\boldsymbol{\nu}\right)_{\Gamma^i}, \quad \forall \boldsymbol{Y} \in [Y_h(\Gamma^i)]^n. \tag{50}$$

Next, formally taking $\delta t \to 0$, this becomes

$$(\kappa, \boldsymbol{Y}\cdot\boldsymbol{\nu})_{\Gamma^t} = (-\Delta_\Gamma \mathrm{id}_{\Gamma^t}, \boldsymbol{Y})_{\Gamma^t} = \left(\hat{\kappa} - \lambda, \boldsymbol{Y}\cdot\boldsymbol{\nu}\right)_{\Gamma^t}, \quad \forall \boldsymbol{Y} \in [H^1(\Gamma^t)]^n, \tag{51}$$

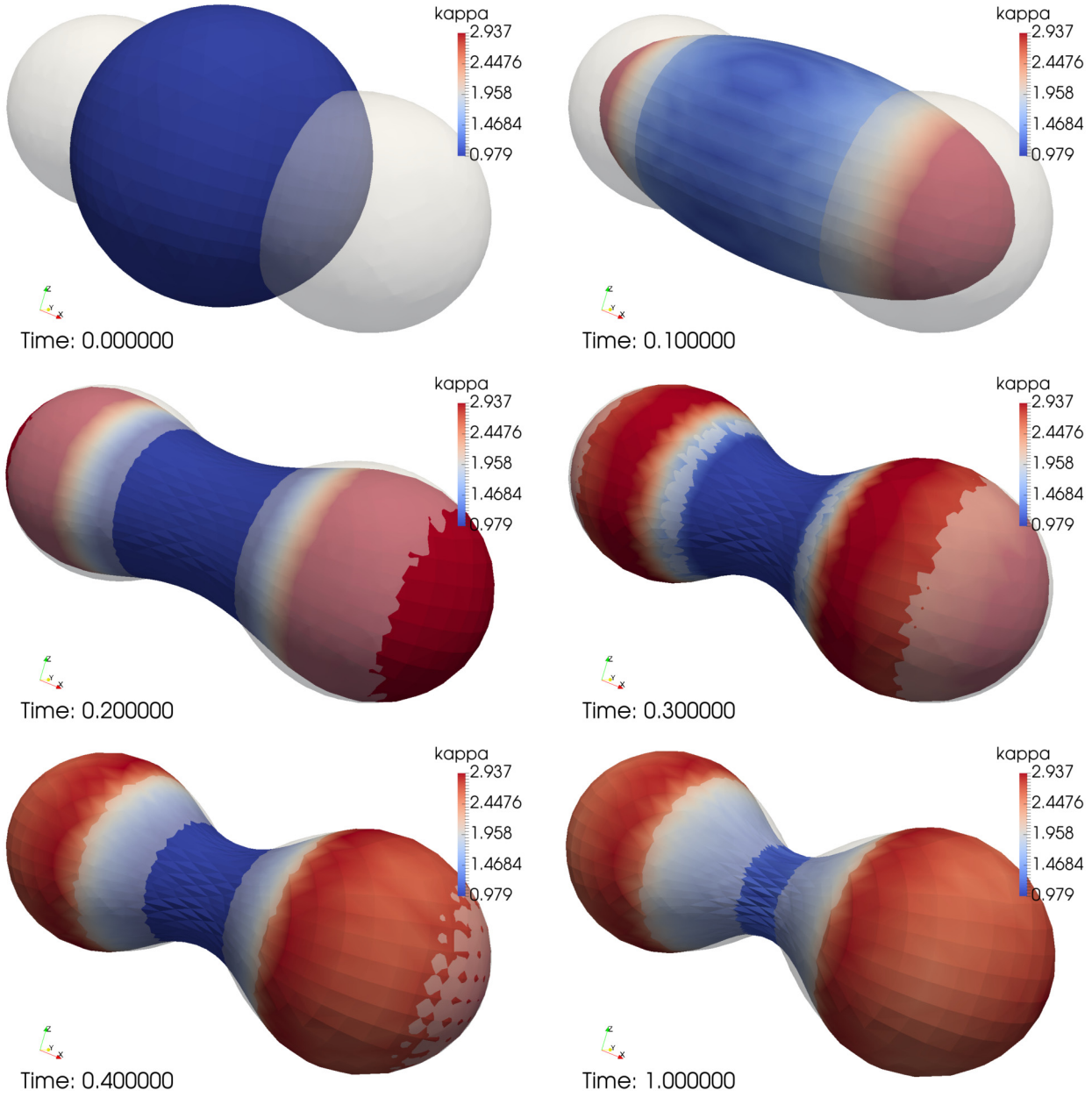

Fig. 24. Evolution of the forward simulation using the final control obtained by the optimization (Section 5.4). The white (transparent) surface is the zero level set of ϕ. The surface Γ is plotted with color according to κ (the additive curvature).

where we used the identity $-\Delta_\Gamma \mathrm{id}_{\Gamma^t} \equiv \kappa \boldsymbol{\nu}$. Thus, $\kappa = \hat{\kappa} - \lambda$. In addition, taking the mesh size to zero and $\delta t \to 0$, the third equation in (49) yields

$$(\boldsymbol{V} \cdot \boldsymbol{\nu}, 1)_{\Gamma^t} = 0, \quad \forall t \in [0, t_f]. \tag{52}$$

Lastly, the second equation in (49) becomes

$$(\boldsymbol{V} \cdot \boldsymbol{\nu}, v)_{\Gamma^t} = \left(u - \hat{\kappa}, v\right)_{\Gamma^t} = (u - \kappa - \lambda, v)_{\Gamma^t}, \quad \forall v \in H^1(\Gamma^t), \tag{53}$$

which matches (3).

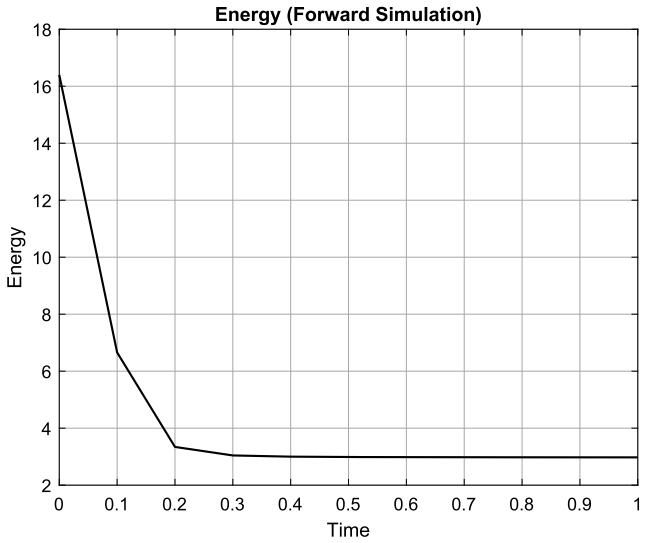

Fig. 25. Energy (4) of the forward simulation versus time (Section 5.4). The control obtained at the final optimization iteration is used here.

4.3. Discretization of the adjoint

Now that we have the sequence $\{\Gamma^i\}$, solving the adjoint problem (numerically) is relatively straightforward. Note that the adjoint evolution problem is solved *backward* in time; see (30). We begin by discretizing the material derivative $D_{\boldsymbol{V}} R$:

$$(D_{\boldsymbol{V}} R)(t_i,\cdot) \approx \frac{R^i - \widetilde{R}^{i+1}}{-\delta t}, \text{ on } \Gamma^i, \text{ with } \widetilde{R}^i = \Pi_i^{i-1} R^i, \tag{54}$$

where $\widetilde{R}^i \in Y_h(\Gamma^{i-1})$, $R^{i-1} \in Y_h(\Gamma^{i-1})$, for all $1 \le i \le N$. Next, let $\boldsymbol{g} := \pi_h^i \mathrm{P}_h(\nabla u) \in \mathbb{U}_h$ (i.e. the L^2 projection). Then, given $\widetilde{R}^N := 0$, we do the following for $i = N-1, \ldots, 2, 1, 0$. Find $R^i \in Y_h(\Gamma^i)$ and $\mu^i \in \mathbb{R}$ such that

$$\begin{aligned} &\left(\frac{R^i - \widetilde{R}^{i+1}}{-\delta t}, Y\right)_{\Gamma^i} + \left(R^i \mathrm{div}_{\Gamma^i}(\boldsymbol{V}^{i+1}), Y\right)_{\Gamma^i} - \left(\nabla_\Gamma R^i, \nabla_\Gamma Y\right)_{\Gamma^i} \\ &+ \left(R^i \boldsymbol{v}^i \cdot \boldsymbol{g}, Y\right)_{\Gamma^i} + \left(R^i |D_{\Gamma^i} \boldsymbol{v}^i|^2, Y\right)_{\Gamma^i} + \mu^i (1, Y)_{\Gamma^i} = -\left(\pi_h^i \phi(t_i,\cdot), Y\right)_{\Gamma^i}, \quad \forall Y \in Y_h(\Gamma^i), \\ &\left(R^i, 1\right)_{\Gamma^i} = 0. \end{aligned} \tag{55}$$

Note that the Laplace term seems to have the "wrong sign", but the time-step is negative since the evolution problem is solved backward in time.

4.4. Discrete optimization method

The discrete optimization algorithm method mirrors Algorithm 1. The discrete cost is defined by

$$\mathcal{J}_h(u) := \sum_{i=0}^{N} \omega_i \int_{\Omega^i(u)} \phi(t_i,\cdot) + \frac{\alpha}{2} \|\nabla u\|^2_{L^2(\mathfrak{D})}, \quad \forall u \in \mathbb{U}_h(0), \tag{56}$$

where ω_i are quadrature weights for the time integral (e.g., one can use the trapezoidal rule). Algorithm 2 gives the fully discrete method.

4.5. Integrating over unfitted domains

The computation of $\int_{\Omega^i(u)} \phi(t_i,\cdot)\, d\boldsymbol{x}$ over a discrete, unfitted grid is problematic. A simple method, that avoids dealing with mesh intersections, could be to define a smoothed characteristic function χ^i that represents Ω^i, or project the smooth characteristic onto piecewise constant functions over the background grid $\mathcal{T}_h$. Then, simply replace $\int_{\Omega^i(u)} \phi(t_i,\cdot)\, d\boldsymbol{x}$ with

$$\int_{\mathfrak{D}} \chi^i \phi(t_i,\cdot)\, d\boldsymbol{x},$$

Algorithm 2: Discrete gradient flow algorithm.

Define ϕ, $\alpha > 0$, $t_f > 0$, $N = t_f/\delta t$, TOL > 0, and initialize Ω_0^0 or (equivalently) Γ_0^0. Set an initial guess for the control u_0, e.g., $u_0 := 0$. For $k \geq 0$, do the following.

1. Forward problem. Solve (49) using u_k, i.e., compute $\{\Gamma_k^i\}_{i=0}^N$.
2. Adjoint problem. Solve (55) on $\{\Gamma_k^i\}_{i=0}^N$ to obtain $\{R_k^i\}_{i=0}^N$.
3. Descent Direction. Solve the following variational problem: find $\eta_{k+1} \in \mathbb{U}_h(0)$ such that

$$a\left(\eta_{k+1}, \xi\right) = -\sum_{i=0}^{N} \omega_i \int_{\Gamma_k^i} R_k^i(\pi_h^i \xi) - \alpha \int_{\mathfrak{D}} \nabla u_k \cdot \nabla \xi, \quad \forall \xi \in \mathbb{U}_h(0), \tag{57}$$

   where $a(\cdot,\cdot)$ is a positive-definite symmetric bilinear form (inner product); see (62).
4. Backtracking line-search:
   (a) Initialize the step-size $\rho_{k+1} := 1$.
   (b) Trial update. $u_{k+1} := u_k + \rho_{k+1}\eta_{k+1}$.
   (c) Check for cost decrease. If

$$\mathcal{J}_h(u_{k+1}) < \mathcal{J}_h(u_k) \tag{58}$$

   then we accept u_{k+1} and go to step 5. Otherwise, replace $\rho_{k+1} \leftarrow \rho_{k+1}/2$, and go back to step 4(b), and repeat.
5. Terminate. If $|\mathcal{J}_h(u_{k+1}) - \mathcal{J}_h(u_k)| < \text{TOL}$, then stop. Otherwise, $k \leftarrow k+1$, and go to step 1.

and integrate in a standard way over $\mathcal{T}_h$. However, the accuracy of this approach could be an issue.

Hence, we opted for the following approach, which still avoids mesh intersections but incurs some additional computational cost. For each i, let $\boldsymbol{\sigma}^i \in \mathbb{S}_h^2 \subset [H^1(\mathfrak{D})]^n$ be a piecewise quadratic vector field such that

$$\begin{aligned} \operatorname{div} \boldsymbol{\sigma}^i(\cdot) &= \phi(t_i, \cdot), \quad \text{in } \mathfrak{D}, \\ \boldsymbol{\sigma}^i &= \mathbf{0}, \quad \text{on } \partial\mathfrak{D}, \end{aligned} \tag{59}$$

We implement this construction by solving a discrete Stokes problem with standard Taylor-Hood elements. Since the level set function is predetermined, this can be done offline. Then, we can make the following approximation:

$$\int_{\Omega^i} \phi(t_i, \cdot)\, d\boldsymbol{x} = \int_{\Omega^i} \operatorname{div} \boldsymbol{\sigma}^i \, d\boldsymbol{x} = \int_{\Gamma^i} \boldsymbol{\sigma}^i \cdot \boldsymbol{\nu}\, dS(\boldsymbol{x}) \approx \int_{\Gamma^i} (\pi_h^i \boldsymbol{\sigma}^i) \cdot \boldsymbol{\nu}\, dS(\boldsymbol{x}). \tag{60}$$

Thus, the only geometric approximation error is due to interpolating $\boldsymbol{\sigma}^i$ onto Γ^i; moreover, this only requires finding the location of the vertices of Γ^i within the background grid $\mathcal{T}_h$ in order to evaluate $\pi_h^i \boldsymbol{\sigma}^i$, i.e., we do not need to know how the elements of Γ^i intersect the elements of $\mathcal{T}_h$. We apply the same approach for computing the "energy" (4) of Γ^i in the gradient flow of the forward problem as it depends on the control u integrated over Ω^i.

5. Numerical results

We present examples in two and three dimensions. In both cases, we seek a time-independent control u that drives the space-time tube $Q_\Omega(\boldsymbol{V}(u))$ toward a target trajectory Q_d. In each experiment, the target trajectory Q_d is defined as the sublevel set of a given function ϕ; see (5). In Section 5.3 a *time-dependent* ϕ is used, while in the other experiments ϕ is independent of time. All computations were done using the Matlab/C++ FEM toolbox FELICITY [42]. All 3-D linear system solves use the algebraic multi-grid solver (AGMG) [32,30,31,33]; the Stokes system is solved using the method in [34,35]. All 2-D solves simply use the "backslash" command in Matlab.

5.1. Two-dimensional T-shape

This example seeks to find u such that the curve Γ evolves toward a T-shape. The time-independent level set function is given by

$$\begin{aligned} \phi(x, y) = \widetilde{\min}\Big(& \max(|x + 0.60| - 0.05, 0)^2 + \max(|y| - 0.8, 0)^2 - (0.2)^2, \\ & \max(|x + 0.25| - 1.05, 0)^2 + \max(|y - 0.75| - 0.05, 0)^2 - (0.2)^2 \Big), \end{aligned} \tag{61}$$

where $\widetilde{\min}(a, b) := (a+b)/2 - |a-b|_\epsilon/2$ and $|s|_\epsilon := \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.02$. For the forward problem, the time interval is $[0, t_f]$ with $t_f = 3$ and $N = 100$ time steps. The initial curve Γ^0 is specified to be a circle of radius 0.81. In addition, we used a piecewise quadratic approximation of the curve for the method in (49), with 200 edges in the mesh.

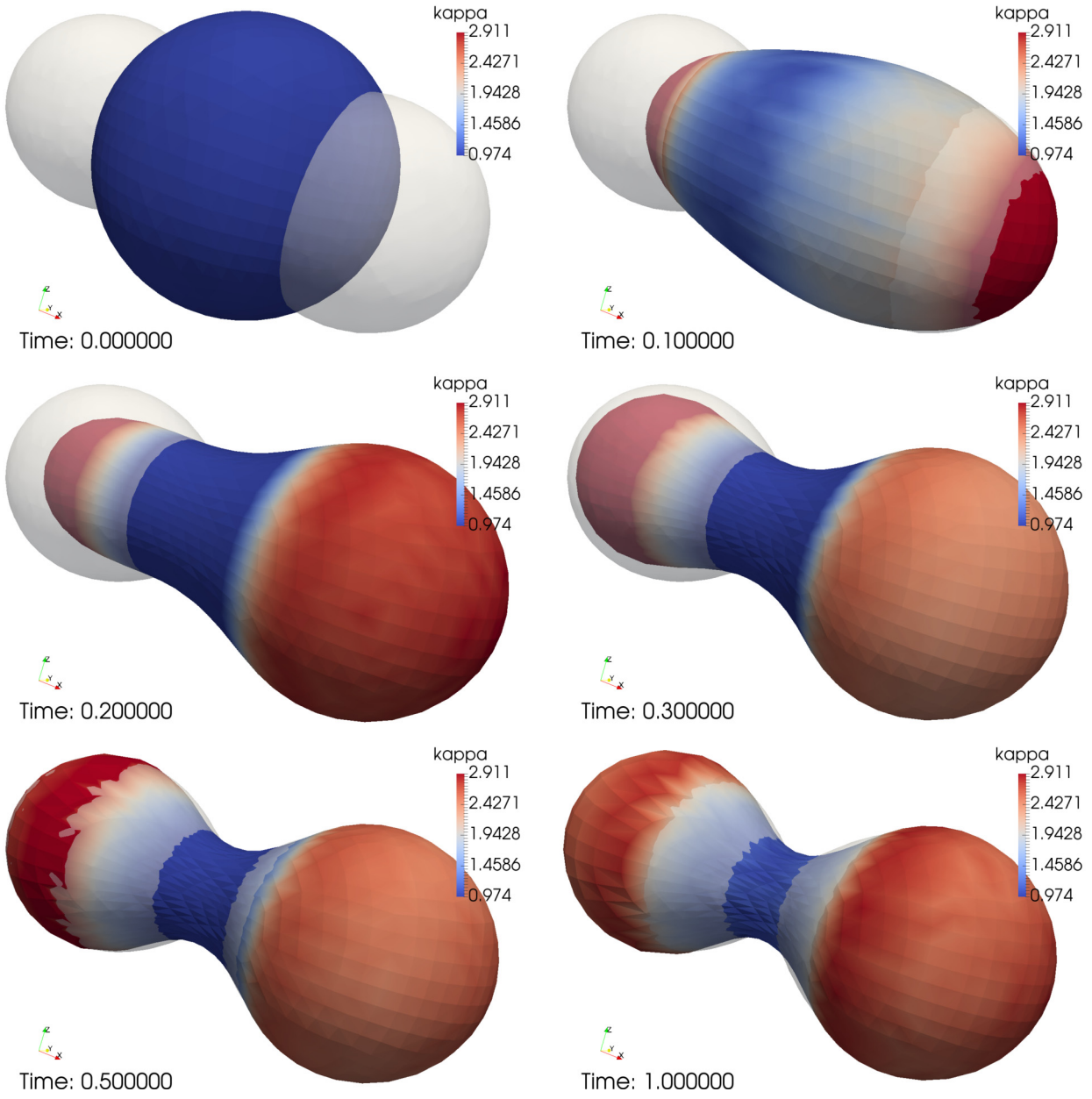

Fig. 26. Evolution of the forward simulation with perturbed initial sphere with center at $(0.2, 0.0, 0.0)$ (Section 5.4). The control u was taken from the final iteration of the optimization.

The optimization parameters are as follows. The "hold-all" domain is $\mathfrak{D} = [-2, 2]^2$, $\alpha = 10^{-4}$, discretized with a uniform triangular mesh consisting of 20201 vertices, and 40000 triangles. The inner product $a(\cdot, \cdot)$ in (57) is given by the following positive-definite bilinear form

$$a(\eta, \xi) = \int_{\mathfrak{D}} \eta\xi \, d\boldsymbol{x} + \beta \int_{\mathfrak{D}} \nabla\eta \cdot \nabla\xi \, d\boldsymbol{x}, \tag{62}$$

where $\beta = 0.1$. The second integral in (62) has a regularizing effect when solving (57) and the coefficient β was chosen heuristically to adjust the regularization level. The initial guess for u is simply $u_0 \equiv 0$, which causes Γ to remain a circle for all time.

Fig. 2 shows how the control u is updated by the optimization. Clearly, the control is chosen to drive Γ to the desired shape represented by the black curve. A plot of the objective functional cost (56) is shown in Fig. 3.

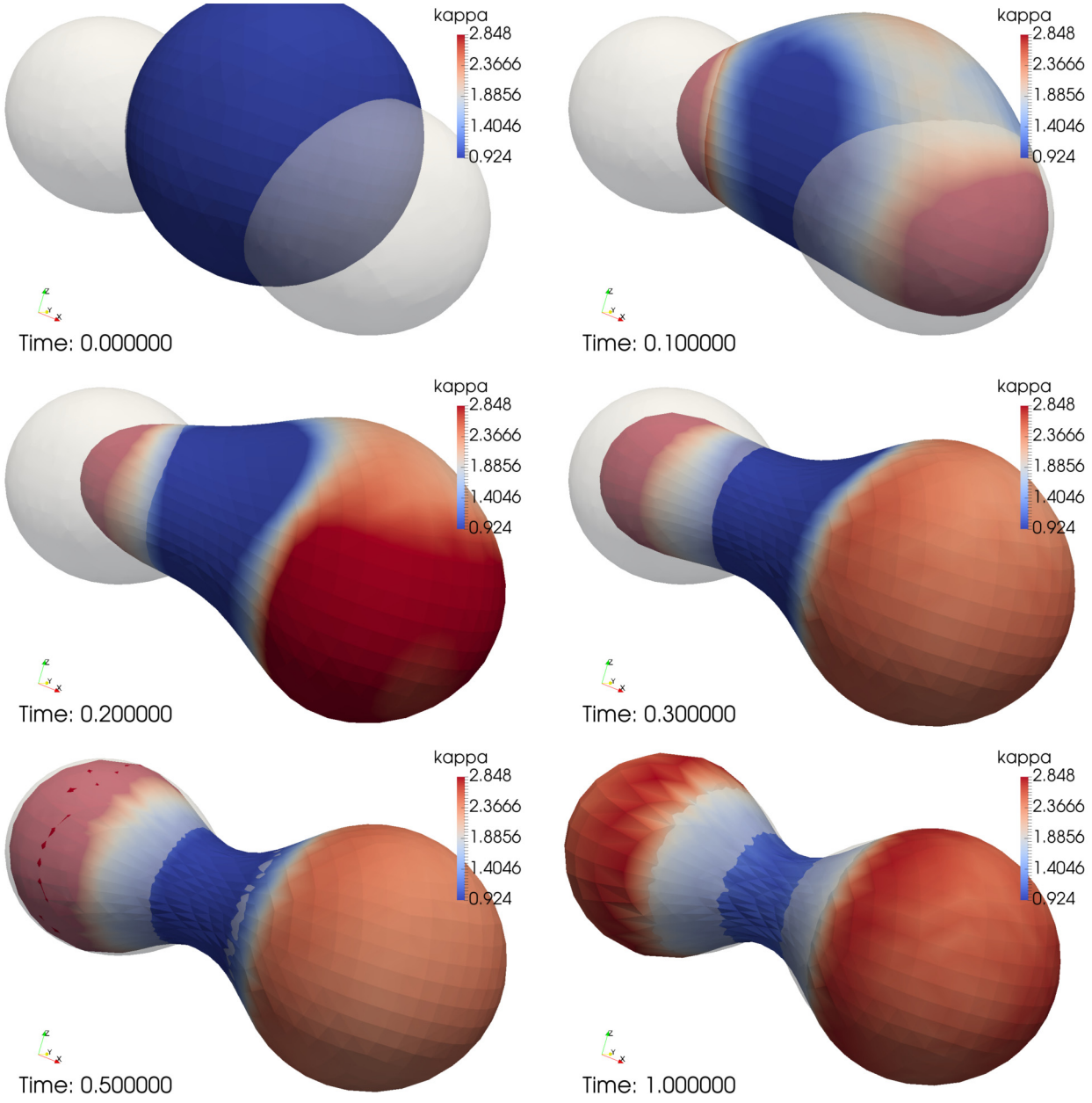

Fig. 27. Evolution of the forward simulation with perturbed initial sphere with center at $(0.22, 0.22, 0.22)$ (Section 5.4). The control u was taken from the final iteration of the optimization.

Fig. 4 shows the evolution of the forward simulation using the control obtained at the final optimization iteration. Fig. 5 shows the "energy" of the forward simulation versus time, where the energy is defined by (4).

Fig. 6 shows the evolution of Γ with a perturbed initial circle, i.e., the center of the circle is $(0.02, 0.0)$. Here, Γ does *not* converge toward the desired shape. However, using an initial center of $(0.01, 0.0)$ yields an evolution similar to Fig. 4. Fig. 7 shows the case where the initial circle center is $(0.0, -0.02)$. Again, Γ does *not* converge toward the desired shape, but using an initial center of $(0.0, -0.01)$ yields an evolution similar to Fig. 4. Hence, the optimal control found is sensitive to perturbations of the initial condition.

5.2. Two-dimensional dumbbell

This example seeks to find u such that the curve Γ evolves toward a dumbbell shape. The time-independent level set function is given by

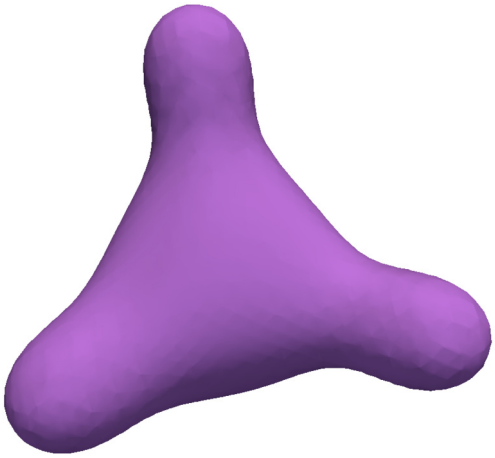

Fig. 28. Illustration of the zero level set of (66) (Section 5.5).

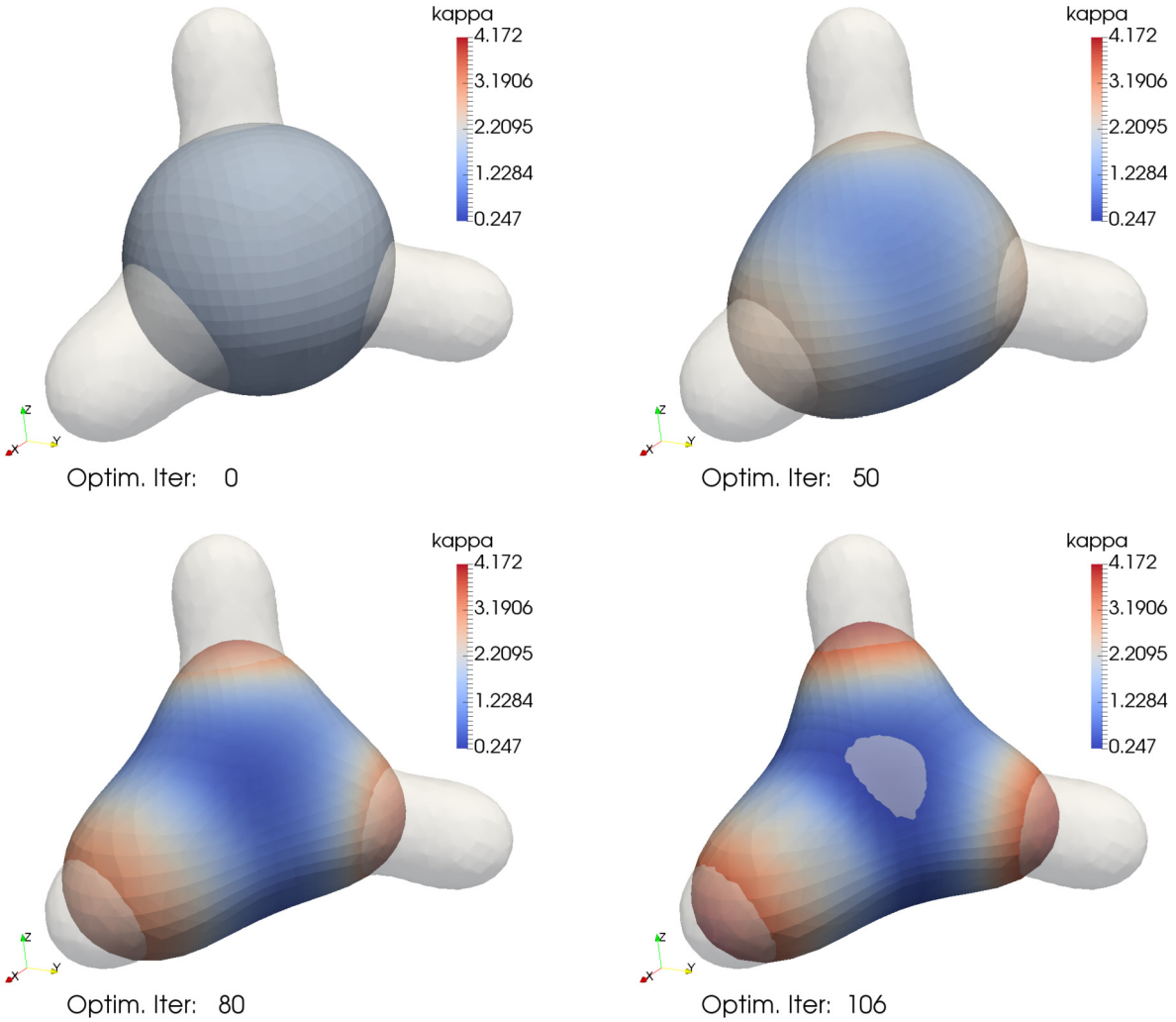

Fig. 29. Evolution of Γ^{t_f} versus optimization iteration (Section 5.5). The white (transparent) surface is the zero level set of ϕ. The surface Γ is plotted (at the final time of the forward simulation) with color according to κ (the additive curvature). Note that the coloring is affected by the transparent zero level set.

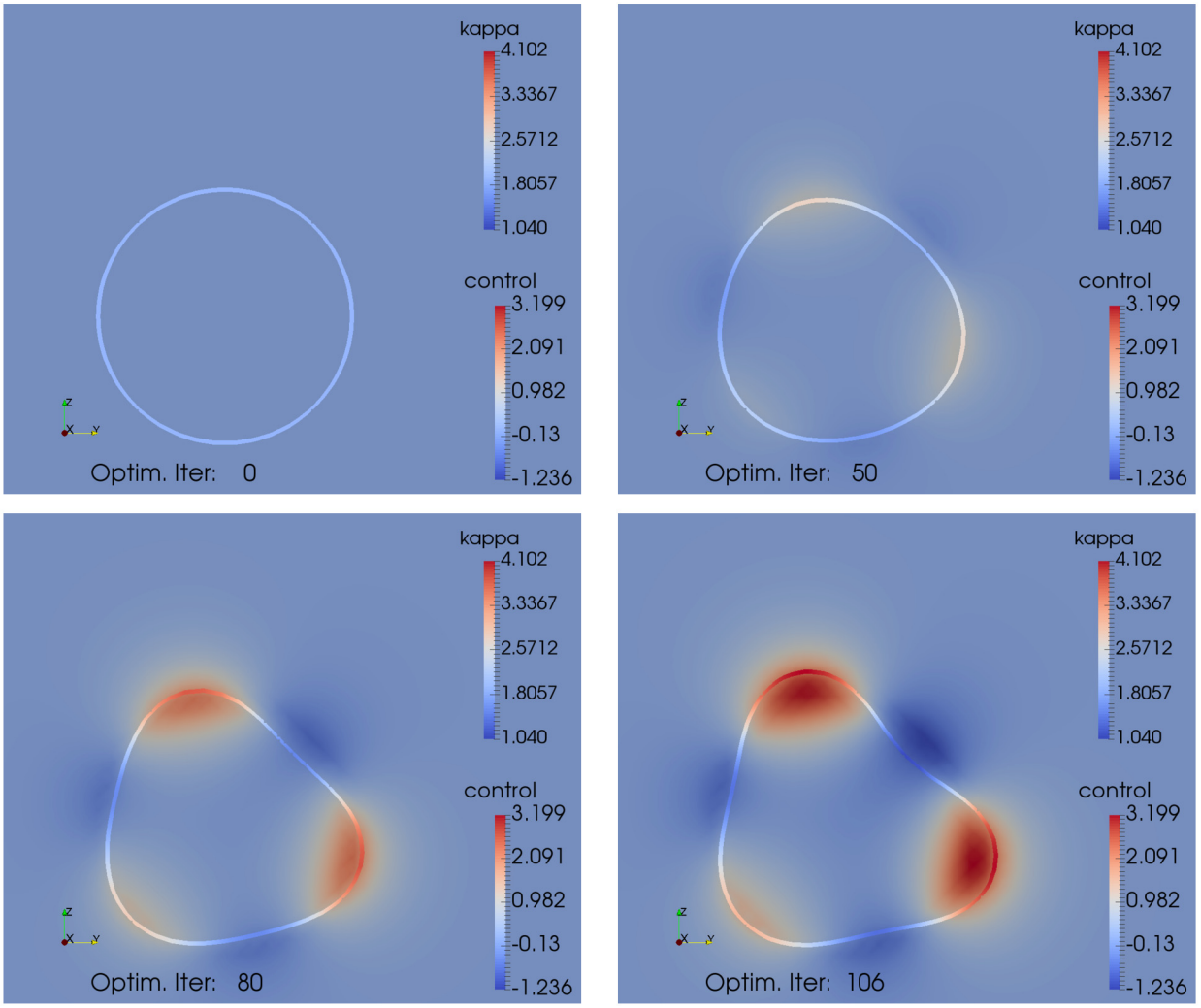

Fig. 30. Evolution of the control u versus optimization iteration (Section 5.5); a 2-D slice is shown, with slice normal $\boldsymbol{n} = (1, 0, 0)$ and passing through the point $(-0.85, 0, 0)$. The slice of Γ, which is a curve, is plotted (at the final time of the forward simulation) with color according to κ. The background control function u is plotted with its own color scale.

$$\phi(x, y) = 2.4 - \sum_{i=1}^{2} \left[\left(\frac{x}{1.35} + a_i \right)^2 + \left(\frac{y}{1.35} + b_i \right)^2 + 0.1 \right]^{-1/2}, \tag{63}$$

where $a_1 = -0.75$, $a_2 = +0.75$, $b_1 = 0.0$, $b_2 = 0.0$. For the forward problem, the time interval is $[0, t_f]$ with $t_f = 1$ and $N = 60$ time steps. The initial curve Γ^0 is specified to be a circle of radius 1.0. In addition, we used a piecewise quadratic approximation of the curve for the method in (49), with 100 edges in the mesh.

The optimization parameters are as follows. The "hold-all" domain is $\mathfrak{D} = [-2, 2]^2$, $\alpha = 10^{-4}$, discretized with a uniform triangular mesh consisting of 7321 vertices, and 14400 triangles. The inner product $a(\cdot, \cdot)$ in (57) is given by (62). The initial guess for u is simply $u_0 \equiv 0$, which causes Γ to remain a circle for all time.

Fig. 8 shows how the control u is updated by the optimization. Clearly, the control is chosen to drive Γ to the desired shape represented by the black curve. A plot of the objective functional cost (56) is shown in Fig. 9.

Fig. 10 shows the evolution of the forward simulation using the control obtained at the final optimization iteration. Fig. 11 shows the "energy" of the forward simulation versus time, where the energy is defined by (4).

Fig. 12 shows the evolution of Γ with a perturbed initial circle, i.e., the center of the circle is $(0.2, 0.0)$. Fig. 13 shows the case where the initial circle center is $(0.3, 0.3)$. Hence, the optimal control found appears to be somewhat robust to perturbations of the initial condition. Of course, a sufficiently large perturbation will cause the curve to evolve to a circle *outside* of the zero level set.

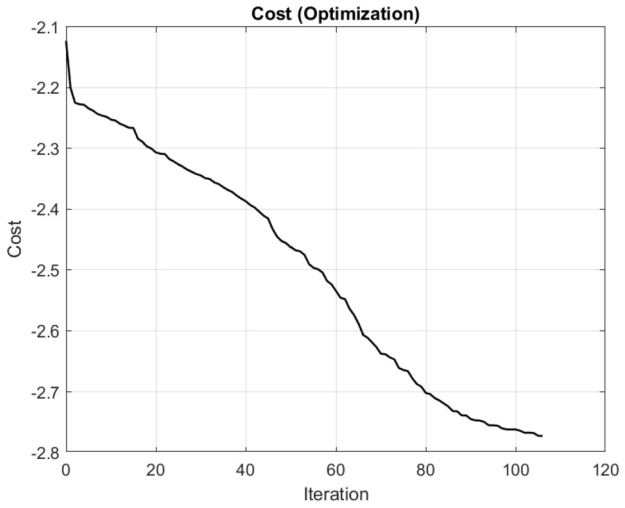

Fig. 31. Decrease of the optimization cost (56) versus optimization iteration (Section 5.5).

5.3. Two-dimensional oscillating ellipse

This example looks for a control u (fixed in time) so that the curve Γ evolves so as to match an oscillating ellipse as well as possible. The time-varying level set function is given by

$$\phi(t,x,y) = \left(\left(\frac{x}{a(t)} \right)^2 + \left(\frac{y}{b(t)} \right)^2 \right)^{1/2} - 1, \quad a(t) = 1 + r_0 \sin(\omega_0 t), \quad b(t) = \frac{1}{a(t)}, \tag{64}$$

where $r_0 = 0.48$, $\omega_0 = 2\pi$. For the forward problem, the time interval is $[0, t_f]$ with $t_f = 1$ and $N = 50$ time steps. The initial curve Γ^0 is specified to be a circle of radius 1.0. In addition, we used a piecewise quadratic approximation of the curve for the method in (49), with 100 edges in the mesh.

The optimization parameters are as follows. The "hold-all" domain is $\mathfrak{D} = [-2, 2]^2$, $\alpha = 10^{-4}$, discretized with a uniform triangular mesh consisting of 8581 vertices, and 16900 triangles. The inner product $a(\cdot, \cdot)$ in (57) is given by (62). The initial guess for u is simply $u_0 \equiv 0$, which causes Γ to remain a circle for all time.

Fig. 14 shows how the control u is updated by the optimization. The control is chosen to drive the evolution of Γ toward a kind of "average" of the desired trajectory dictated by (64). A plot of the objective functional (56) is shown in Fig. 15.

Fig. 16 shows the evolution of the forward simulation using the control obtained at the final optimization iteration. We observe that Γ^t is able to follow the time-dependent trajectory to some extent, even though the control is stationary. Fig. 17 shows the "energy" of the forward simulation versus time, where the energy is defined by (4).

Fig. 18 shows the evolution of Γ with a perturbed initial circle, i.e., the center of the circle is $(0.02, 0.0)$. Here, Γ does *not* match the desired evolution as well as before. Using an initial center of $(0.005, 0.0)$ yields an evolution similar to Fig. 16. Fig. 19 shows the case where the initial circle center is $(0.04, -0.04)$. Here, the match is worse. Using an initial center of $(0.005, -0.005)$ yields an evolution similar to Fig. 16. In this case, the optimal control found is sensitive to perturbations of the initial condition.

5.4. Three-dimensional dumbbell

This example seeks to find u such that the surface Γ evolves toward a dumbbell shape. The time-independent level set function is given by

$$\phi(x,y,z) = 2.4 - \sum_{i=1}^{2} \left[\left(\frac{x}{1.52} + a_i \right)^2 + \left(\frac{y}{1.52} + b_i \right)^2 + \left(\frac{z}{1.52} + c_i \right)^2 + 0.1 \right]^{-1/2}, \tag{65}$$

where $a_1 = -0.75$, $a_2 = +0.75$, $b_1 = 0.0$, $b_2 = 0.0$, $c_1 = 0.0$, $c_2 = 0.0$ (Fig. 20 illustrates the zero level set of (65)). For the forward problem, the time interval is $[0, t_f]$ with $t_f = 1$ and $N = 10$ time steps. The initial surface Γ^0 is specified to be a sphere of radius 1.0. In addition, we used a piecewise quadratic triangular surface mesh for the method in (49), with 2048 triangles and 1026 vertices.

The optimization parameters are as follows. The "hold-all" domain is $\mathfrak{D} = [-2.5, 2.5]^3$, $\alpha = 10^{-4}$, discretized with a uniform tetrahedral mesh consisting of 137720 vertices, and 768000 tetrahedra. The inner product $a(\cdot, \cdot)$ in (57) is similar to (62), where $\beta = 0.1$. The initial guess for u is simply $u_0 \equiv 0$, which causes Γ to remain a sphere for all time.

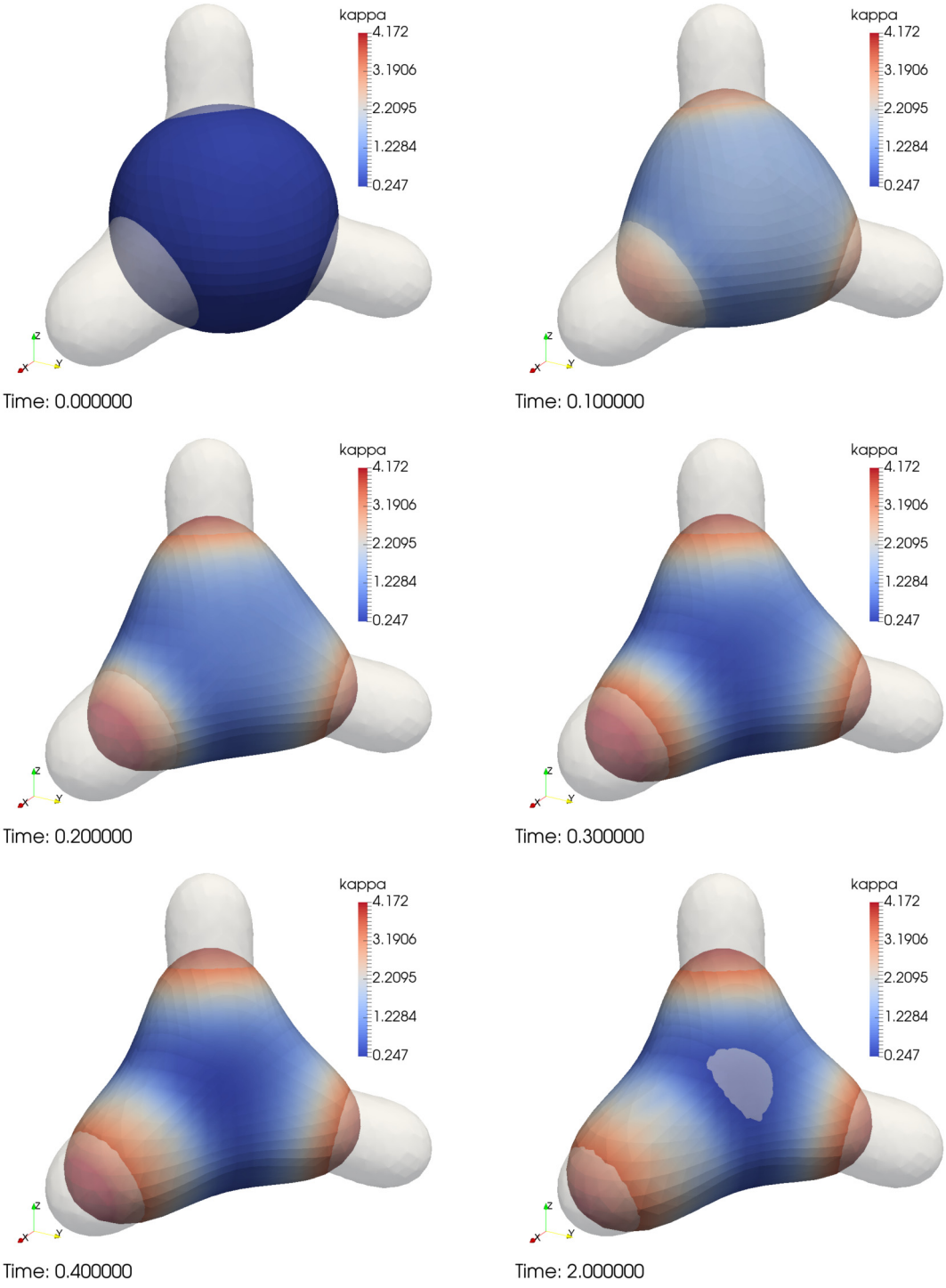

Fig. 32. Evolution of the forward simulation using the final control obtained by the optimization (Section 5.5). The white (transparent) surface is the zero level set of ϕ. The surface Γ is plotted with color according to κ (the additive curvature).

Fig. 21 shows how Γ^{t_f} changes based on the optimization iteration, while Fig. 22 shows how the control u is updated by the optimization. Clearly, the control is chosen to drive Γ to the desired shape (see the transparent white surface). A plot of the objective functional cost (56) is shown in Fig. 23.

Fig. 24 shows the evolution of the forward simulation using the control obtained at the final optimization iteration. Fig. 25 shows the "energy" of the forward simulation versus time.

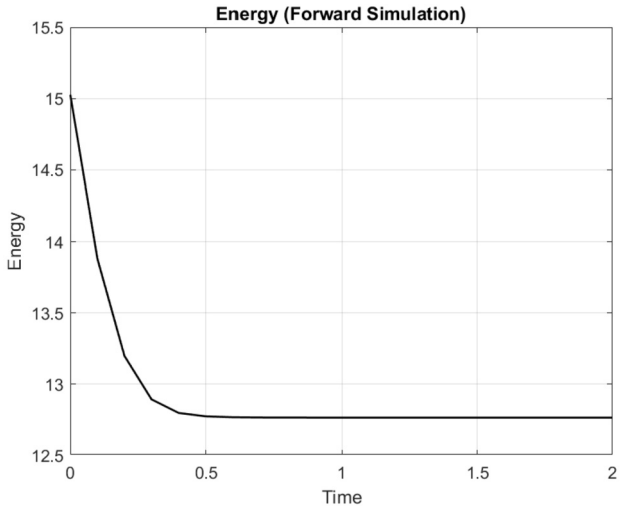

Fig. 33. Energy (4) of the forward simulation versus time (Section 5.5). The control obtained at the final optimization iteration is used here.

Fig. 26 shows the evolution of Γ with a perturbed initial sphere, i.e., the center of the sphere is $(0.2, 0.0, 0.0)$. Fig. 27 shows the case where the initial sphere center is $(0.22, 0.22, 0.22)$. Hence, the optimal control found appears to be somewhat robust to perturbations of the initial condition. Of course, a sufficiently large perturbation will cause the sphere to evolve to a sphere *outside* of the zero level set.

5.5. Three-dimensional smoothed tetrahedron

This example seeks to find u such that the surface Γ evolves toward a smoothed tetrahedral shape. The time-independent level set function is given by

$$\phi(\boldsymbol{x}) = \frac{1}{4.1} F\left(\Pi_{i=1}^{4} |\boldsymbol{x} - \boldsymbol{a}_i| - 6.0\right), \quad F(s) = \begin{cases} s, & \text{if } s < 1, \\ \log(s) + 1, & \text{else}, \end{cases} \tag{66}$$

where $\boldsymbol{x} = (x, y, z)$ and

$$\boldsymbol{a}_1 = (-0.7, -0.9, -0.9), \quad \boldsymbol{a}_2 = (1.1, -0.9, -0.9), \quad \boldsymbol{a}_3 = (-0.9, 1.1, -0.9), \quad \boldsymbol{a}_4 = (-0.9, -0.9, 1.1).$$

Fig. 28 illustrates the zero level set of (66). For the forward problem, the time interval is $[0, t_f]$ with $t_f = 2$ and $N = 20$ time steps. The initial surface Γ^0 is specified to be a sphere, centered at $(-0.5, -0.5, -0.5)$, of radius 1.0656832. In addition, we used a piecewise quadratic triangular surface mesh for the method in (49), with 2048 triangles and 1026 vertices.

The optimization parameters are as follows. The "hold-all" domain is $\mathfrak{D} = [-2.5, 2.5]^3$, $\alpha = 10^{-3}$, discretized with a uniform tetrahedral mesh consisting of 137720 vertices, and 768000 tetrahedra. The inner product $a(\cdot, \cdot)$ in (57) is similar to (62), where $\beta = 0.1$. The initial guess for u is simply $u_0 \equiv 0$, which causes Γ to remain a sphere for all time.

Fig. 29 shows how Γ^{t_f} changes based on the optimization iteration, while Fig. 30 shows how the control u is updated by the optimization. Clearly, the control is chosen to drive Γ to the desired shape (see the transparent white surface), although the match is not perfect. A plot of the objective functional cost (56) is shown in Fig. 31.

Fig. 32 shows the evolution of the forward simulation using the control obtained at the final optimization iteration. Fig. 33 shows the "energy" of the forward simulation versus time.

Fig. 34 shows the evolution of Γ with a perturbed initial sphere, i.e., the center of the sphere is $(-0.5, -0.3, -0.5)$. Fig. 35 shows the case where the initial sphere center is $(-0.3, -0.3, -0.3)$. Hence, the optimal control found appears to be somewhat robust to perturbations of the initial condition. Of course, a sufficiently large perturbation will cause the sphere to evolve to a sphere *outside* of the zero level set.

6. Conclusion

We have developed a methodology for performing optimal PDE control of geometrically driven flows. A central concept in the method is the *transverse field* which accounts for the evolution of the domain and appears in the sensitivity analysis with respect to perturbations of the corresponding space-time tube. An important contribution of the paper is the introduction of an *adjoint transverse field* to compute the derivative of the cost functional, appearing as the solution of a backward parabolic equation with terminal conditions on the boundary of the space-time tube. In this way, we extend the usual adjoint calculus

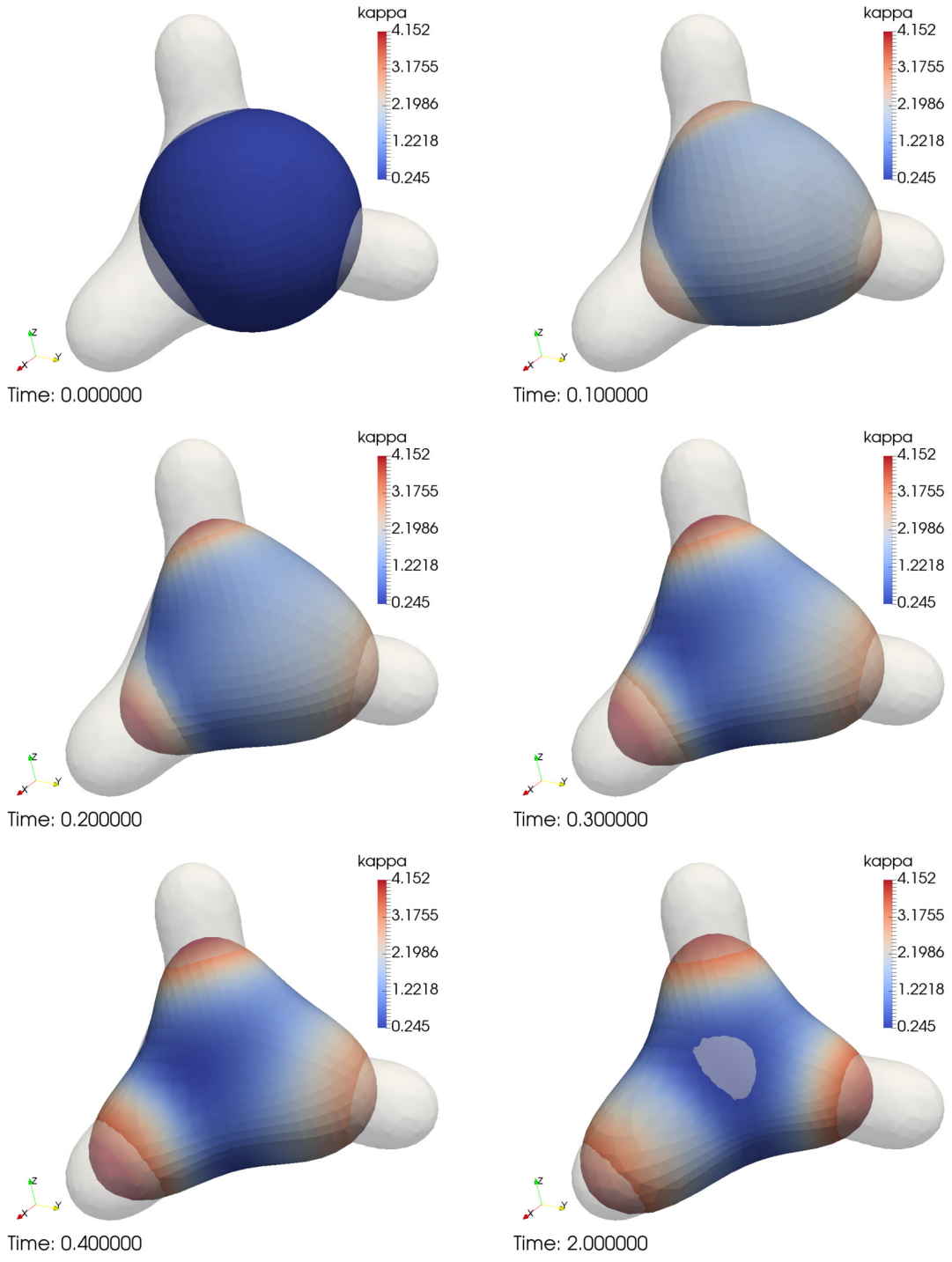

Fig. 34. Evolution of the forward simulation with perturbed initial sphere with center at $(-0.5, -0.3, -0.5)$ (Section 5.5). The control u was taken from the final iteration of the optimization.

from PDE-constrained optimization to optimal control of gradient flows. As a test problem, we consider optimal control of the mean curvature flow equation with volume constraint via a forcing term. The methodology is general and the derivation we present can be extended to incorporate other geometrically driven flows.

Furthermore, we present an effective numerical scheme that captures the geometric evolution of the (forward) problem with reasonable efficiency, as well as a discrete optimization method for finding an optimal control that minimizes our tracking type objective functional. We also demonstrated that our optimal control results can be sensitive to initial pertur-

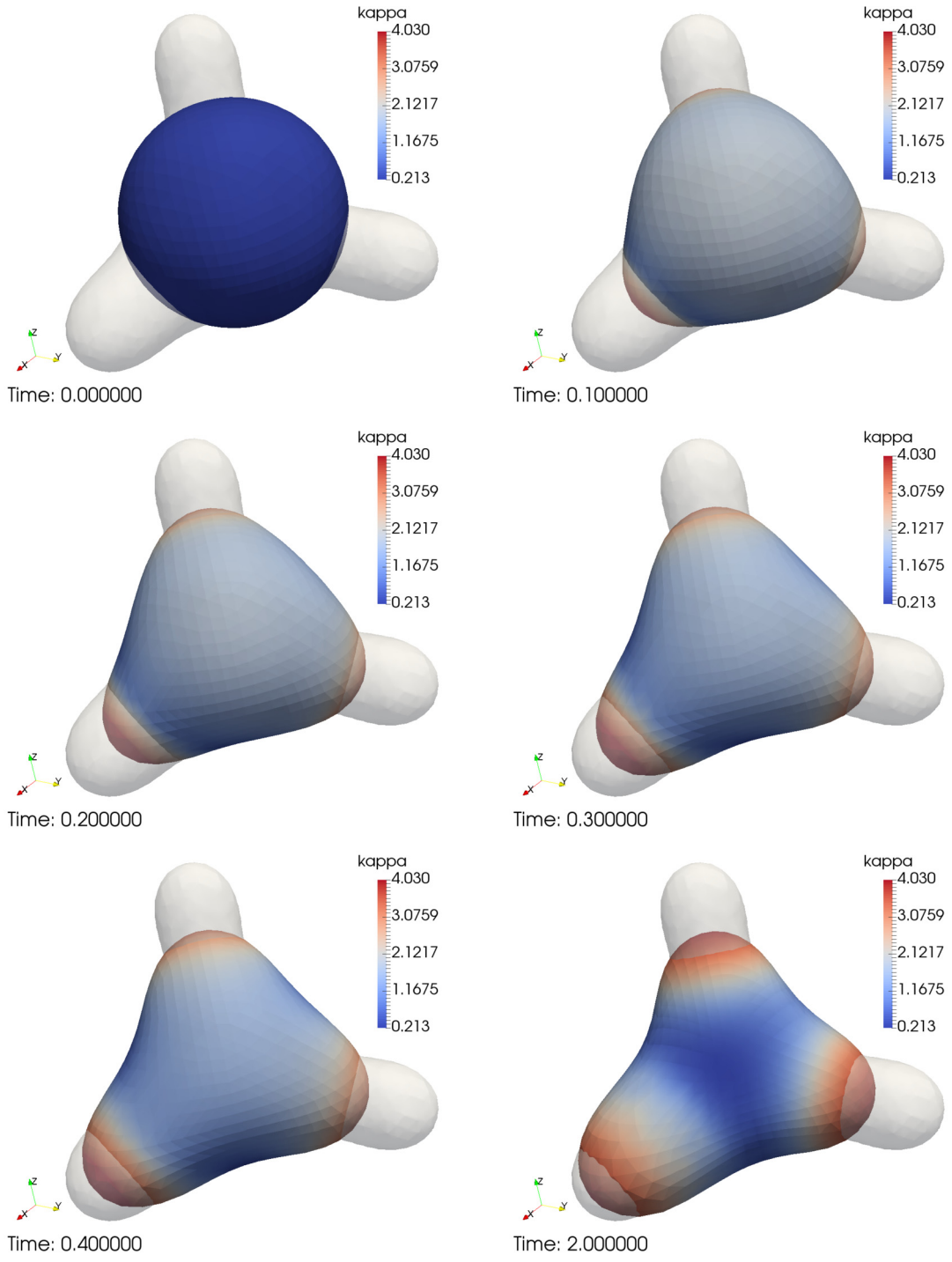

Fig. 35. Evolution of the forward simulation with perturbed initial sphere with center at $(-0.3, -0.3, -0.3)$ (Section 5.5). The control u was taken from the final iteration of the optimization.

bations of the forward problem. In other cases, the results are reasonably robust. The sensitivity is probably linked to the complexity of the desired shape. Improving the robustness of the control, through modification of the optimization scheme, will be a topic for further investigation.

Several questions remain, however, about this framework. Proving rigorous differentiability results about the forward problem, with respect to the control, appears to be extremely challenging. Indeed, this would require advanced regularity results about PDEs on moving surfaces as well as geometric and regularity properties of general gradient flows, which

are still active research topics. In addition, there are many open numerical questions, such as convergence of the forward problem and of the optimal control.

Furthermore, it is not clear what are the "reachable states" of the forward problem, i.e., what kind of stationary states can we drive the evolution toward? Our numerical results indicate that regions with high curvature or geometric singularities may be difficult or impossible to reach. One reason is that MCF is an evolution problem, and it is not obvious that a control, that is time-*independent*, can guide the surface toward an arbitrary shape (even if $\alpha = 0$, i.e., without Tikhonov regularization). Another reason is that MCF is a diffusive equation that smooths out the effect of the control u, which could be limiting. A full investigation of the effects of α is quite involved; even if α is set to zero, the finite size of the mesh will create some regularization. Another relevant question raised by our numerical experiments is to understand how close one can get to time-independent or time-dependent trajectories using a stationary control. It seems that understanding the "exact reachability" or "approximate reachability" of target shapes will require more analysis that remains to be developed.

CRediT authorship contribution statement

Antoine Laurain: Conceptualization, Methodology, Formal analysis, Writing – Original draft preparation, Writing – Review and Editing. **Shawn W. Walker:** Conceptualization, Methodology, Formal analysis, Writing – Original draft preparation, Writing – Review and Editing, Software, Investigation, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments which have helped to improve the paper. Antoine Laurain gratefully acknowledges the support of the Brazilian National Council for Scientific and Technological Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq) through the process: 408175/2018-4 "Otimização de forma não suave e controle de problemas de fronteira livre", and through the program "Bolsa de Produtividade em Pesquisa - PQ 2018", process: 304258/2018-0. Shawn W. Walker acknowledges financial support by the National Science Foundation (NSF) via DMS-1555222 (CAREER).

References

[1] A. Alphonse, C.M. Elliott, B. Stinner, An abstract framework for parabolic PDEs on evolving spaces, Port. Math. 72 (1) (2015) 1–46.
[2] A. Alphonse, C.M. Elliott, B. Stinner, On some linear parabolic PDEs on moving hypersurfaces, Interfaces Free Bound. 17 (2) (2015) 157–187.
[3] B. Andrews, Volume-preserving anisotropic mean curvature flow, Indiana Univ. Math. J. 50 (2) (2001) 783–827.
[4] S. Angenent, Parabolic equations for curves on surfaces. I. Curves with p-integrable curvature, Ann. Math. (2) 132 (3) (1990) 451–483.
[5] S. Angenent, Parabolic equations for curves on surfaces. II. Intersections, blow-up and generalized solutions, Ann. Math. (2) 133 (1) (1991) 171–215.
[6] M. Astorino, J.-F. Gerbeau, O. Pantz, K.-F. Traoré, Fluid–structure interaction and multi-body contact: application to aortic valves, Comput. Methods Appl. Mech. Eng. 198 (45) (2009) 3603–3612, Models and Methods in Computational Vascular and Cardiovascular Mechanics.
[7] G. Barles, H.M. Soner, P.E. Souganidis, Front propagation and phase field theory, SIAM J. Control Optim. 31 (2) (1993) 439–469.
[8] J.W. Barrett, H. Garcke, R. Nürnberg, A parametric finite element method for fourth order geometric evolution equations, J. Comput. Phys. 222 (1) (2007) 441–467.
[9] J.W. Barrett, H. Garcke, R. Nürnberg, A variational formulation of anisotropic geometric evolution equations in higher dimensions, Numer. Math. 109 (1) (2008) 1–44.
[10] K. Blazakis, A. Madzvamuse, C. Reyes-Aldasoro, V. Styles, C. Venkataraman, Whole cell tracking through the optimal control of geometric evolution laws, J. Comput. Phys. 297 (Sep 2015) 495–514.
[11] A. Chambolle, M. Novaga, Implicit time discretization of the mean curvature flow with a discontinuous forcing term, Interfaces Free Bound. (2008) 283–300.
[12] A. Chicco-Ruiz, P. Morin, M.S. Pauletti, The shape derivative of the Gauss curvature, Rev. Unión Mat. Argent. (May 2018) 311–337.
[13] W. Croft, C.M. Elliott, G. Ladds, B. Stinner, C. Venkataraman, C. Weston, Parameter identification problems in the modelling of cell motility, J. Math. Biol. 71 (2) (Sep 2014) 399–436.
[14] C.B. Davis, S.W. Walker, A mixed formulation of the Stefan problem with surface tension, Interfaces and Free Boundaries 17 (4) (2015) 427–464.
[15] M.C. Delfour, J.-P. Zolésio, Shapes and Geometries: Analysis, Differential Calculus, and Optimization, 2nd edition, Advances in Design and Control, vol. 4, SIAM, 2011.
[16] R. Dziri, J.-P. Zolésio, Shape optimization and optimal design, chapter Eulerian derivative for non-cylindrical functionals, in: Lecture Notes in Pure and Applied Mathematics, Dekker, New York, 2001, pp. 87–107.
[17] R.S. Falk, S.W. Walker, A mixed finite element method for EWOD that directly computes the position of the moving interface, SIAM J. Numer. Anal. 51 (2) (2013) 1016–1040.
[18] Y. Giga, Surface Evolution Equations, Monographs in Mathematics, vol. 99, Birkhäuser Verlag, Basel, 2006.
[19] A. Henrot, M. Pierre, A geometrical analysis, in: Shape Variation and Optimization, in: EMS Tracts in Mathematics. European Mathematical, vol. 28, Society (EMS), Zürich, 2018, English version of the French publication [MR2512810] with additions and updates.
[20] M. Hintermüller, A. Laurain, Optimal shape design subject to elliptic variational inequalities, SIAM J. Control Optim. 49 (3) (2011) 1015–1047.
[21] H. Kasumba, K. Kunisch, A. Laurain, A bilevel shape optimization problem for the exterior Bernoulli free boundary value problem, Interfaces Free Bound. 16 (4) (2014) 459–487.
[22] I. Kim, D. Kwon, On mean curvature flow with forcing, Commun. Partial Differ. Equ. 45 (5) (Dec. 2019) 414–455.

[23] I. Kim, D. Kwon, Volume preserving mean curvature flow for star-shaped sets, Calc. Var. Partial Differ. Equ. 59 (2) (2020) 81.
[24] B. Kovács, C. Lubich, Linearly implicit full discretization of surface evolution, Numer. Math. 140 (1) (Mar. 2018) 121–152.
[25] A. Laurain, S.W. Walker, Droplet footprint control, SIAM J. Control Optim. 53 (2) (2015) 771–799.
[26] P. Le Tallec, J.-F. Gerbeau, P. Hauret, M. Vidrascu, Fluid structure interaction problems in large deformation, C. R., Méc. 333 (12) (2005) 910–922, Fluid-solid interactions: modeling, simulation, bio-mechanical applications.
[27] C. Mantegazza, Lecture Notes on Mean Curvature Flow, Springer Basel, 2011.
[28] U.F. Mayer, A singular example for the averaged mean curvature flow, Exp. Math. 10 (1) (2001) 103–107.
[29] M. Moubachir, J.-P. Zolésio, Moving Shape Analysis and Control: Applications to Fluid Structure Interactions, Pure and Applied Mathematics, vol. 277, Chapman and Hall/CRC, 2006.
[30] A. Napov, Y. Notay, Algebraic analysis of aggregation-based multigrid, Numer. Linear Algebra Appl. 18 (3) (2011) 539–564.
[31] A. Napov, Y. Notay, An algebraic multigrid method with guaranteed convergence rate, SIAM J. Sci. Comput. 34 (2) (2012) A1079–A1109.
[32] Y. Notay, An aggregation-based algebraic multigrid method, Electron. Trans. Numer. Anal. 37 (2010) 123–146.
[33] Y. Notay, Aggregation-based algebraic multigrid for convection-diffusion equations, SIAM J. Sci. Comput. 34 (4) (2012) A2288–A2316.
[34] Y. Notay, A new algebraic multigrid approach for Stokes problems, Numer. Math. 132 (2016) 51–84.
[35] Y. Notay, Algebraic multigrid for Stokes equations, SIAM J. Sci. Comput. 39 (5) (2017) S88–S111.
[36] O. Pironneau, D.F. Katz, Optimal swimming of flagellated microorganisms, J. Fluid Mech. 66 (2) (1974) 391–415.
[37] M. Rumpf, O. Vantzos, Numerical gradient flow discretization of viscous thin films on curved geometries, Math. Models Methods Appl. Sci. 23 (05) (Feb. 2013).
[38] J. Sokolowski, J.-P. Zolésio, Introduction to Shape Optimization, Springer Series in Computational Mathematics, Springer-Verlag, 1992.
[39] W. Velte, P. Villaggio, On the detachment of an elastic body bonded to a rigid support, J. Elast. 27 (1992) 133–142.
[40] S.W. Walker, A mixed formulation of a sharp interface model of Stokes flow with moving contact lines, ESAIM: Math. Model. Numer. Anal. 48 (2014) 969.
[41] S.W. Walker, The Shapes of Things: A Practical Guide to Differential Geometry and the Shape Derivative, 1st edition, Advances in Design and Control, vol. 28, SIAM, 2015.
[42] S.W. Walker FELICITY, A Matlab/C++ toolbox for developing finite element methods and simulation modeling, SIAM J. Sci. Comput. 40 (2) (2018) C234–C257.
[43] O.-Y. Zhong-can, W. Helfrich, Bending energy of vesicle membranes: general expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders, Phys. Rev. A 39 (10) (May 1989) 5280–5288.