BODY-FITTED TRACKING WITHIN A SURFACE VIA A LEVEL SET BASED MESH EVOLUTION METHOD

C. BRITO-PACHECO [1] AND C. DAPOGNY[1]

[1] *Univ. Grenoble Alpes, CNRS, Grenoble INP*[1]*, LJK, 38000 Grenoble, France.*

Abstract. This article introduces a robust numerical strategy for tracking the arbitrarily large motion of a region $G(t)$ within a three-dimensional surface S under the effect of a complex velocity field $V(t,x)$. Following our earlier work about evolving domains in the Euclidean space $\mathbb{R}^d$, two complementary representations of the region $G(t) \subset S$ are combined at each stage of the iterative process. On the one hand, $G(t)$ is meshed exactly, which allows for precise geometric and finite element computations, such as those required by the evaluation of $V(t,x)$. On the other hand, $G(t)$ is represented implicity, via the level set method – a format under which dramatic deformations of this region can be captured, including changes in its topology. Efficient numerical algorithms make it possible to switch consistently from one of these representations to the other, depending on its relevance with respect to the ongoing operation. After numerical validation, this strategy is applied to address two concrete physical problems, namely the simulation of the evolution of a fire front within a complex landscape, and the optimization of the shape of regions supporting the boundary conditions of a mechanical boundary value problem.

Contents

1. Introduction

The broad task of representing the evolution of a domain $G(t) \subset \mathbb{R}^d$ ($d = 2$ or 3 in practice) has sparked extensive mathematical and numerical investigations. This topic plays a central role in various applied disciplines, ranging from computer graphics and vision [31, 109] to the numerical simulation of physical phenomena such as fracture propagation [25] or fluid interface dynamics [38, 40], including inverse problems and shape optimization [8, 11, 29].

Multiple numerical strategies have been proposed to address this task, with competing assets and drawbacks. However, all implementations face a major challenge: in complex, realistic situations, the velocity

[1] Institute of Engineering Univ. Grenoble Alpes

field $V(t,x)$ driving the motion of $G(t)$ has a physical origin and depends on high-order geometric features of this region (notably, the curvature of $\partial G(t)$), or on the solution to a boundary value problem posed on the latter. It is then notoriously difficult to find a framework reconciling an accurate description of $G(t)$ allowing for precise calculations of these quantities at any time t – which ideally demands an exact, high-quality mesh – with the robust treatment of its evolution.

In this regard, Lagrangian strategies, tracking the evolution of an exact mesh of $G(t)$ by displacing its vertices according to $V(t,x)$ between successive iterations of the process, are usually undermined by a severe degradation of the quality of the mesh, which rapidly becomes invalid and incompatible with computations, see e.g. [54, 55]. Admittedly, several heuristics enhance the robustness of this practice. For instance, one may alternate deformations of the mesh with occasional remeshing steps aimed to improve its quality. Going further, one may detect and remove ill-shaped elements before they lead to complete degeneracy, or even modify the velocity of the internal vertices of the mesh to reduce the onset of overlapping patterns, see e.g. [14, 19, 30, 50, 106] about these ideas.

In this spirit, the recent Deformable Simplicial Complex technique has demonstrated the ability to cope with impressively large motions by using the formation of nearly degenerate mesh elements near the boundary of the domain as a trigger for topological changes, see [37, 36, 77]. Even more recently, the X-mesh method [78] proceeds by displacing the vertices of the mesh of $G(t)$ according to $V(t,x)$ up the point where the measure of some elements equals zero. The motion is then relayed between neighboring nodes while preserving the connectivity of the mesh. This strategy rests on the assumption that boundary value problems can be effectively solved on meshes with degenerate elements, under appropriate assumptions on their aspect ratios. Despite these noteworthy achievements, let us emphasize that such Lagrangian mesh deformation strategies are usually reserved for the description of "relatively small" motions of the set $G(t)$.

To overcome the weaknesses of Lagrangian strategies, Eulerian interface capturing techniques are based on an implicit description of the moving shape $G(t)$. Among these, the level set method, introduced in [86], features a description of $G(t)$ as the negative subdomain of an auxiliary "level set function" $\phi(t,\cdot) : \mathbb{R}^d \to \mathbb{R}$ defined on the whole ambient space $\mathbb{R}^d$, see [85, 93]. The domain $G(t)$ is never meshed explicitly, being rediscovered at each iteration of the process from the values of $\phi(t,\cdot)$. Although it allows to describe arbitrarily large motions, such an implicit representation is unfortunately less amenable to the accurate solution of partial differential equations defined on $G(t)$.

These central and popular questions in the numerical analysis of the motion of a domain $G(t)$ have received surprisingly little attention when the ambient medium is a manifold – notably a surface S in $\mathbb{R}^3$; yet, this alternative context embraces multiple applications of interest:

- Geometric flows, such as the mean curvature flow or the Willmore flow, where the velocity field $V(t,x)$ depends on high-order geometric features of $G(t)$, can be adapted to the case of a region within a surface [100].
- Diverse operations of interest in the field of computer graphics are conveniently formulated in terms of the evolution of a region within a surface. For instance, one popular model for the generation of textures on a surface relies on the resolution of a reaction-diffusion equation, see e.g. [102, 103]; also, image segmentation on surfaces can be addressed thanks to a suitable adaptation of the Chan-Vese algorithm [100].
- Various physical evolution problems occur within a surface, see for instance [83] about the solidification of a thin fluid film front on a surface substrate, and [99] about the dynamics of phase changes on surfaces in material science.
- The wish to optimize the shape of regions within a given ambient surface arises for instance in the optimal design of shells [101], of curvilinear electronic devices [72], in the context of surface flows [48], or in the identification of optimal fixation systems for mechanical structures, see e.g. [45, 104].

To the best of our knowledge, the first numerical simulations of the evolution of a region $G(t)$ within a surface S, proposed in [35] and [70], were concerned with the geodesic curvature flow. These leverage a version of the level set method tailored to the datum of S as a parametrized patch, or as the graph of a function defined on the 2d space, respectively. A more general setting is proposed in [23] and [34], where the closed surface $S := \left\{x \in \mathbb{R}^d, \ \psi(x) = 0\right\}$ is represented as the 0 level set of a fixed function $\psi : \mathbb{R}^d \to \mathbb{R}$, and $G(t) = \{x \in S, \ \phi(t,x) < 0\}$ is the negative subregion of S induced by another (time-dependent) level set function $\phi(t,\cdot) : \mathbb{R}^d \to \mathbb{R}$. Here and in the subsequent investigations in this framework [22, 65, 90], the equation governing the evolution of the level set function $\phi(t,\cdot)$ is formulated in the whole space $\mathbb{R}^d$ with the help of projection operators. Another series of contributions [75, 89] leverages the so-called closest point method of [74], devoted to the solution of partial differential equations on surfaces. As the latter solely requires the datum of a mapping associating to any point $x \in \mathbb{R}^d$ one closest point (in terms of Euclidean distance) to the surface S, this latter framework leaves the room for S to be open.

The present article aims to introduce a robust numerical methodology for tracking arbitrarily large motions of a region $G(t)$ within an ambient surface $S \subset \mathbb{R}^3$ – including changes of its topology – while maintaining an exact meshed representation of the latter throughout the process. The proposed strategy is a natural extension of our earlier contributions [4, 5, 6] – which were devoted to evolving domains of the Euclidean space $\mathbb{R}^d$ – to the present context where the ambient medium is a surface S in $\mathbb{R}^d$. It combines two complementary representations of $G(t)$ at each stage of the evolution: on the one hand, $G(t)$ is explicitly discretized, as a submesh of a high-quality surface triangulation $\mathcal{T}$ of the ambient surface S, which allows to accurately calculate its geometric features or to solve related boundary value problems via the finite element method – and thereby to evaluate the velocity field $V(t,x)$ precisely. On the other hand, $G(t)$ is described implicitly via the level set method, as the negative subdomain of a scalar function $\phi(t,\cdot) : S \to \mathbb{R}$, so that arbitrarily large motions of $G(t)$ can be realized. The cornerstone of this strategy is a set of efficient meshing algorithms and numerical schemes for passing from one representation to the other.

This article is organized as follows. In the next Section 2, we present in more details the issue of tracking the evolution of a region within a surface. Then, in Section 3, we describe the proposed numerical strategy to realize this task and we detail its main ingredients; we notably discuss the calculation of the signed distance function to a region on a surface, the resolution of the level set evolution equation in this context, and our main remeshing operations dedicated to surface triangulations. Interestingly, these numerical methods are implemented in open-source codes which can easily be used in a black-box fashion. A few numerical applications of our framework are presented in Section 4: after appraising its efficiency on an analytical test-case, we consider the motion of a complex interface accounting for a fire front, whose expansion within a fixed landscape is driven by geometric quantities attached to the fire and those of the landscape. We then deal with the optimization of the shape of the regions supporting the boundary conditions of a boundary value problem – an iterative process which requires an accurate solution of the latter at each state of the process. Eventually, we draw the main conclusions of this work and outline a few perspectives in Section 5.

2. Evolution of a region on a surface in the level set framework

This section introduces the issue of evolving regions within a surface and sets the main notations used throughout.

Let S be a smooth hypersurface, with or without boundary, in the d-dimensional space $\mathbb{R}^d$. Here, d equals 2 or 3, and although we shall focus on the three-dimensional case $d = 3$, which is on any point more challenging than its 2d counterpart, we retain the generic notation d whenever possible. For simplicity of the discussion, the hypersurface S is assumed to be oriented, but our developments and numerical methods extend readily to the case of a non-orientable surface. For any point $x \in S$, $n(x)$ is the unit normal vector to S at x and the tangent plane to S at x – that is, the vector plane orthogonal to $n(x)$ – is denoted by $T_xS := \left\{ v \in \mathbb{R}^d, \; v \cdot n(x) = 0 \right\}$.

Let $G \subset S$ be a smooth open subdomain of S with boundary $\Sigma := \partial G$. For $x \in \Sigma$, we denote by $n_\Sigma(x) \in T_xS$ the conormal vector to Σ at x, pointing outward G, see Fig. 1. Let $V : (0,T) \times S \to \mathbb{R}^3$ be a smooth vector field defined over the considered time period $(0,T)$, which is tangential to S, that is:

$$\forall t > 0 \text{ and } x \in S, \quad V(t,x) \in T_xS. \tag{2.1}$$

We wish to track the region $G(t)$ evolving from $G(0) \equiv G$ under the effect of $V(t,x)$ over $[0,T]$. This notion of evolution is intuitively defined as follows: for any $x \in S$, let $t \mapsto \chi(t,0,x)$ be the characteristic curve of V emerging from x at time 0, i.e. the solution to the ordinary differential equation

$$\left\{ \begin{array}{l} \frac{\mathrm{d}\chi}{\mathrm{d}t}(t,0,x) = V(t,\chi(t,0,x)), \quad \text{for } t \in (0,T), \\ \chi(0,0,x) = x. \end{array} \right. \tag{2.2}$$

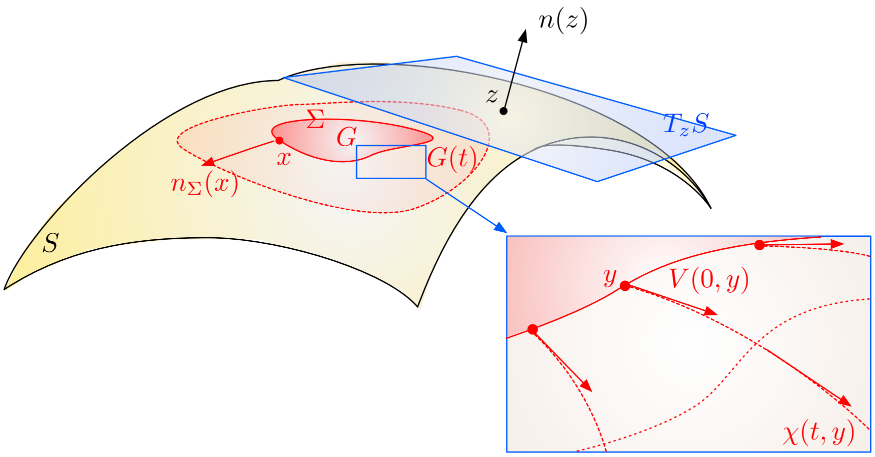

FIGURE 1. *Example of a region $G(t)$ of a three-dimensional surface S evolving according to a tangential velocity field $V(t,x)$.*

The region $G(t)$ is then defined as the set of the positions occupied at time t by the points lying in G at time 0:

$$G(t) = \Big\{\chi(t,0,x),\ \ x \in G\Big\}. \tag{2.3}$$

As we have mentioned in Section 1, the numerical simulation of the evolution of a domain has long been a thorny issue. Among the numerous frameworks implemented to achieve this goal, the level set method has proved to be particularly convenient since its introduction in [86]; we refer to [85, 93] for a presentation of various aspects of the level set method and its countless applications in scientific computing, see also [64] about its mathematical aspects and its impact on the theory of moving domains.

In the context of the present article, where the ambient medium is a surface $S \subset \mathbb{R}^d$ and $G \subset S$ is an open region with boundary $\Sigma := \partial G$, an auxiliary "level set" function $\phi : S \to \mathbb{R}$ is introduced, whose negative subset coincides with G, that is:

$$\forall x \in S, \quad \left\{\begin{array}{ll} \phi(x) < 0 & \text{if } x \in G, \\ \phi(x) = 0 & \text{if } x \in \Sigma, \\ \phi(x) > 0 & \text{otherwise.} \end{array}\right. \tag{2.4}$$

A formal use of the chain rule reveals that if $G(t)$ is a smooth region of S evolving according to a smooth tangential velocity field $V(t,x)$ over a time period $[0,T]$, any associated level set function $\phi(t,x)$ satisfies the so-called level set advection equation:

$$\frac{\partial \phi}{\partial t}(t,x) + V(t,x) \cdot \nabla_S \phi(t,x) = 0,\ \ t \in (0,T),\ \ x \in S, \tag{2.5}$$

where $\nabla_S \phi := \nabla \phi - (\nabla \phi \cdot n)\, n$ denotes the tangential gradient of the function ϕ (with respect to the spatial variable). Alternatively, introducing the component $v(t,x)$ of $V(t,x)$ in the direction of the conormal vector $n_{\Sigma(t)}(x) = \frac{\nabla_S \phi(t,x)}{|\nabla_S \phi(t,x)|}$ to $G(t)$, that is

$$v(t,x) := V(t,x) \cdot n_{\Sigma(t)}(x), \tag{2.6}$$

the equation (2.5) rewrites as a Hamilton-Jacobi equation:

$$\frac{\partial \phi}{\partial t}(t,x) + v(t,x)|\nabla_S \phi(t,x)| = 0, ,\ \ t \in (0,T),\ \ x \in S. \tag{2.7}$$

In our applications, notably those targeting the description of the evolution of physical interfaces, the velocity field $V(t,x)$ may depend on the moving region $G(t)$ in a very complicated way, often not only through geometric quantities but also via the solution to partial differential equations involving $G(t)$. Hence, the velocity field $V(t,x)$ and its normal component $v(t,x)$ in (2.6) depend in an implicit way on $G(t)$, and thus on ϕ itself. The only realistic means to address the numerical resolution of (2.5) is a fully explicit procedure: the total time interval $(0,T)$ is decomposed into a series of subintervals of the form (t^n, t^{n+1}), which are "small enough" so that $V(t,x)$ can be frozen in time, that is

$$\forall t \in (t^n, t^{n+1}), \quad V(t,x) \approx V(t^n, x);$$

this practice leads to the solution of a series of "true" advection equations of the form (2.5) with time-independent fields $V(t^n, \cdot)$ computed from the region $G(t^n)$ (or the level set function $\phi(t^n, \cdot)$). An alternative approach consists in freezing only the normal component $v(t,x)$ over each subinterval (t^n, t^{n+1}), i.e.

$$\forall t \in (t^n, t^{n+1}), \quad v(t,x) \approx v(t^n, x),$$

which leads to the resolution of a series of "true" non linear Hamilton-Jacobi equations of the form (2.7), with time-independent normal velocities $v(t^n, \cdot)$. This second possibility retains more information from the original evolution equation (2.5) (namely, the fact that the motion is consistently oriented in the direction of the normal vector $n_{\Sigma(t)}$), but it requires the solution of more complex, non linear evolution equations.

Remark 2.1. *In physical applications, the velocity $V(t,x)$ of the region $G(t)$ often makes sense only on the boundary $\Sigma(t)$, while the formalism of the level set method requires that it should extended to the whole surface S, see (2.5) and (2.7). Actually, it is a classical feature of the level set method that under "mild assumptions", the 0 level set of the solution to (2.5) or (2.7) does not depend on the choice of such an extension for $V(t,\cdot)$ outside $\Sigma(t)$, see [64]. In practice, however, the choice of a particular extension may have a great impact on the numerical realization of the motion of $G(t)$, and it may be conducted differently depending on the application, see Section 4 for several examples.*

Remark 2.2. *From the mathematical viewpoint, the intuitive definition (2.3) of the evolving region $G(t)$ makes sense as long as $G(t)$ and $V(t,x)$ are "smooth enough", which is usually the case when the time t is "small enough". Unfortunately, even in the case of a "simple" flow, featuring a "smooth" initial state $G(0)$, the region $G(t)$ or the velocity $V(t,x)$ will inevitably become singular in finite time [9, 64]. One possibility to define a generalized motion (2.3) beyond this point rests on the level set method: considering one level set function $\phi(0,\cdot)$ for the initial region $G(0)$, the evolution equation (2.5) is solved in the generalized sense of viscosity, see [41], or again [9, 64] about this notion. It turns out that, under reasonable assumptions, (2.5) has a unique viscosity solution $\phi(t,\cdot)$, from which $G(t)$ is then defined by*

$$G(t) := \Big\{ x \in S, \ \ \phi(t,x) < 0 \Big\}.$$

Such theoretical questions have been extensively studied, particularly in situations where $V(t,x)$ is composed of geometric quantities attached to $G(t)$, see for instance [68] about the study of the mean curvature flow within a surface. Without entering into details, let us mention a few concurrent attempts to the level set method aimed at generalizing this motion of $G(t)$ past the onset of singularities.

- *Parametric methods insist on the description (2.3) of the evolving region $G(t)$, relying on a (tedious) classification of the various types of possible singularities and on an appropriate selection of what is the "correct" evolution of $G(t)$ in each case, see for instance [56].*
- *Varifold solutions were initially proposed in [26] to deal with the problem of domain evolution. These are measure-theoretic solutions which unfortunately lack uniqueness in their characterization of the evolving set $G(t)$.*
- *Phase field methods were introduced in this mathematical context in [27, 32, 46]. They encode the evolution problem of $G(t)$ into a a scalar "phase field" function, taking values -1 "well inside" $G(t)$ and 1 "well outside" $G(t)$; the thin transition region between both zones is sought as the solution to an energy minimization problem, see [52] for a recent overview.*

3. Presentation of the numerical strategy

In this section again, $G(t)$ denotes a region of a fixed surface $S \subset \mathbb{R}^d$, evolving over a time period $(0,T)$ according to a tangential velocity field $V(t,x)$, see (2.1). We do not specify the nature of V for the moment, but we assume that the calculation of $V(t,\cdot) : S \to \mathbb{R}^d$ at one particular time is difficult and costly, as it involves either geometric quantities, or the solution to a boundary value problem attached to $G(t)$.

The proposed numerical strategy for tracking the evolution of $G(t)$ is summarized in Section 3.1. The pivotal ingredients involved in its implementation are fairly classical when the ambient medium is the 2d

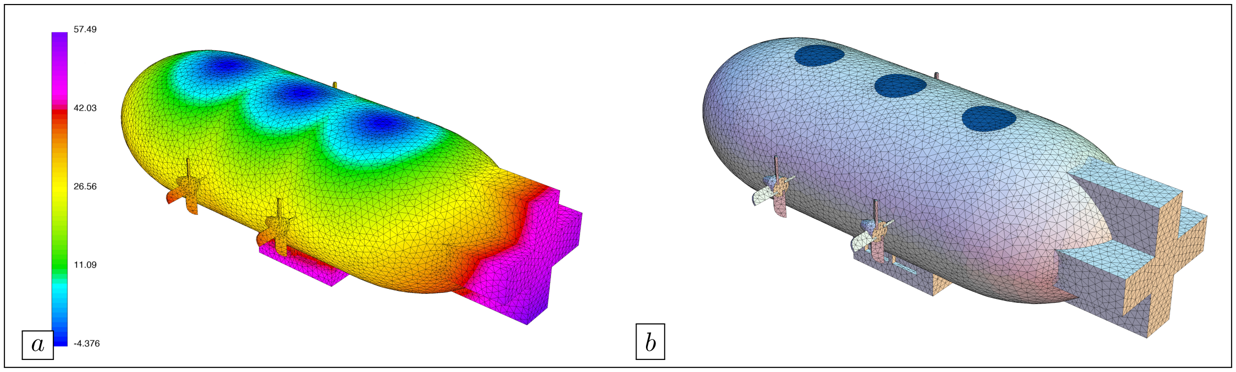

FIGURE 2. *(a) Level set function $\phi : S \to \mathbb{R}$ for a region $G \subset S$ defined at the vertices of the triangulation of a surface $S \subset \mathbb{R}^3$; (b) High-quality mesh $\mathcal{T}$ of S, enclosing a submesh $\mathcal{T}_{\text{int}}$ of G (in dark blue).*

or 3d space, but much less so when it is a surface S in $\mathbb{R}^3$. Hence, we present them in this context with somedetails in the next Sections 3.2 to 3.4.

3.1. Outline of the numerical algorithm

The time interval $(0, T)$ is split into a series of subintervals of the form (t^n, t^{n+1}), $n = 0, \dots, N-1$, where $t^n = n\Delta t$ and Δt is a "small" time step; we indicate with a superscript ^{n} the value of a time-dependent object at t^n: for instance, G^n stands for $G(t^n)$, $V^n(\cdot)$ denotes the velocity field $V(t^n, \cdot)$, etc.

The proposed algorithm for tracking the motion of $G(t)$ is based on two complementary representations of each intermediate configuration G^n:

- A *level set representation.* On the one hand, G^n is known as the negative subset of a level set function $\phi^n : S \to \mathbb{R}$, i.e. (2.4) holds. In practice, ϕ^n is supplied at the vertices of a triangular mesh of S, see Fig. 2 (a).
- A *meshed representation.* On the other hand, G^n is meshed exactly. More precisely, a triangular mesh $\mathcal{T}^n$ of S is available, with the following properties:
  (i) $\mathcal{T}^n$ is valid: the intersection between any two different open triangles T_i, $T_j \in \mathcal{T}^n$ is empty;
  (ii) $\mathcal{T}^n$ is conforming: the intersection $\overline{T_i} \cap \overline{T_j}$ between the closures of any two different triangles T_i, $T_j \in \mathcal{T}^n$ is either a vertex or an edge of $\mathcal{T}^n$;
  (iii) $\mathcal{T}^n$ has high quality: the triangles $T \in \mathcal{T}^n$ are close to being equilateral;
  (iv) $\mathcal{T}^n$ is made of two submeshes $\mathcal{T}^n_{\text{int}}$, $\mathcal{T}^n_{\text{ext}}$ – that is, collections of subsets of its triangles – associated to the respective regions G^n and $S \setminus \overline{G^n}$.

  The requirements (i-iii) are ubiquitous in the scientific computing literature, notably when it comes to guaranteeing the accuracy of finite element computations on S [39]; they are illustrated on Fig. 3. The property (iv) is more specific to our meshed representation, see Fig. 2 (b).

The meshed representation of G^n is particularly useful when it comes to calculating some of its geometric features (e.g. the normal vector field n_{Σ^n} to Σ^n, its curvature, etc.), or to solve related "physical" boundary value problems – operations which are involved in the definition of the velocity field $V^n(x)$ or its normal component $v^n(x)$. In turn, the level set representation ϕ^n allows for a robust description of the motion of $G(t)$ between the times t^n and t^{n+1} via the solution of the advection equation (2.5) with velocity field V^n, or that of the Hamilton-Jacobi equation (2.7) with normal velocity v^n. Efficient numerical algorithms make it possible to switch between these representations so that every operation involving G^n can be carried out within the most appropriate framework.

Our numerical algorithm for the simulation of the motion of $G(t)$ proceeds as follows, see Algorithm 1 for a summarizing sketch. Every iteration $n = 0, \dots$ starts with the datum of the region G^n under meshed representation: a valid, conforming and high-quality mesh $\mathcal{T}^n$ of S is available, a submesh $\mathcal{T}^n_{\text{int}}$ of which is an explicit mesh for the region G^n. By performing geometric or mechanical calculations related to G^n on this mesh, the velocity field $V^n : S \to \mathbb{R}^d$ is calculated at the vertices of $\mathcal{T}^n$. A particular level set

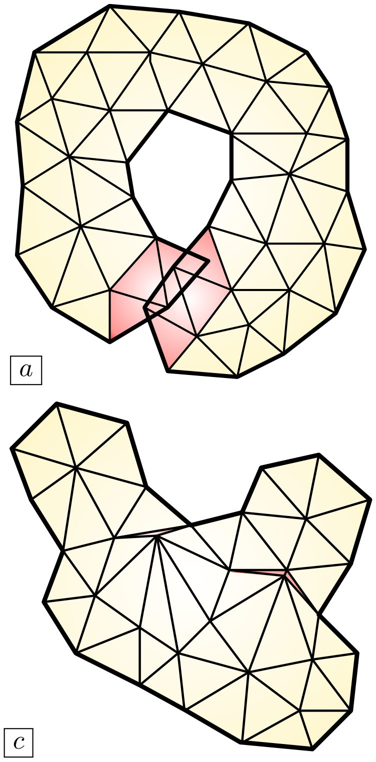

FIGURE 3. *(a) Invalid two-dimensional triangulation, presenting overlapping triangles (in red); (b) Valid, yet non conforming mesh (the red node lies strictly inside an edge); (c) Valid, conforming, but ill-shaped mesh (the red triangles are nearly degenerate); (d) High-quality computational mesh.*

representation $\phi^n : S \to \mathbb{R}$ for G^n is then calculated at the vertices of $\mathcal{T}^n$, as the signed distance function d_{G^n} to G^n. The evolution of $G(t)$ between times t^n and t^{n+1} is carried out by solving the evolution equation (2.5) over the time period (t^n, t^{n+1}) with the velocity field V^n and the initial datum ϕ^n. This yields a level set representation $\phi^{n+1} : S \to \mathbb{R}$ for G^{n+1}, at the vertices of the mesh $\mathcal{T}^n$. Finally, a meshed representation for G^{n+1} is obtained from these data, with the help of suitable remeshing algorithms; this produces a new high-quality mesh $\mathcal{T}^{n+1}$ of S is produced, a submesh $\mathcal{T}^{n+1}_{\text{int}}$ that explicitly discretizes G^{n+1}.

The main stages of this method are described in more detail in the next sections: in Section 3.2, we discuss the numerical computation of the signed distance function to a subregion G of a surface S, which allows to pass from a meshed description of G to a level set description. In Section 3.3, we describe the numerical solution of the level set advection equation (2.5) on the surface S, accounting for the update of $G(t)$ between successive iterations. Finally, in Section 3.4, we outline the remeshing operations involved in the construction of a meshed representation of a region $G \subset S$ from a level set representation $\phi : S \to \mathbb{R}$.

3.2. **Computation of the signed distance function on a surface**

Let $\mathcal{T}$ be a triangulation of a surface $S \subset \mathbb{R}^d$, and let $\mathcal{T}_{\text{int}}$ denote a submes for a region $G \subset S$; we wish to generate a level set function $\phi : S \to \mathbb{R}$ for G at the vertices of $\mathcal{T}$, i.e. a function ϕ satisfying (2.4).

Although many choices are possible, stability issues in the numerical practice of the level set method raise the need to select one which presents "neither too steep, nor too flat" variations, see e.g. [35]. To comply with this requirement, and due to its desirable properties in connection with the geometry of G (see e.g.

```
Algorithm 1: Body-fitted tracking of the evolution of a region $G(t) \subset S$.
Input: Mesh $\mathcal{T}^0$ of $S$ featuring an explicit discretization of the initial region $G^0$.
for $n = 0, \dots, N-1$ do
    (1) Compute the velocity field $V^n(x)$ at the vertices $x$ of $\mathcal{T}^n$ with the help of the meshes $\mathcal{T}^n_{\text{int}}$, $\mathcal{T}^n_{\text{ext}}$ of
        $G^n$ and $S \setminus \overline{G^n}$, respectively.
    (2) Compute the signed distance function $d_{G^n}$ to $G^n$ at the vertices of the mesh $\mathcal{T}^n$ of $S$.
    (3) Solve the advection equation
        $\left\{ \begin{array}{rl} \frac{\partial \phi}{\partial t}(t,x) + V^n(x) \cdot \nabla_S \phi(t,x) = 0 & \text{for } \ (t,x) \in (0,\Delta t) \times S, \\ \phi(0,x) = d_{G^n}(x) & \text{for } x \in S, \end{array} \right.$
        on the total mesh $\mathcal{T}^n$ of $S$. A new level set function $\phi^{n+1} = \phi(\Delta t, \cdot)$ is obtained for
        $G^{n+1} = \left\{ x \in S, \ \phi^{n+1}(x) < 0 \right\}.$
    (4) From the datum of $\phi^{n+1}$ at the vertices of $\mathcal{T}^n$, create a new, high-quality mesh $\mathcal{T}^{n+1}$ of $S$ made of
        two submeshes $\mathcal{T}^{n+1}_{\text{int}}$ and $\mathcal{T}^{n+1}_{\text{ext}}$ for $G^{n+1}$ and $S \setminus \overline{G^{n+1}}$, respectively.
end
Output: Mesh $\mathcal{T}^N$ of $S$ featuring an explicit discretization $\mathcal{T}^N_{\text{int}}$ of $G^N$.
```

[47]), we calculate the signed distance function d_G to G, defined by

$$\text{(3.1)} \qquad \forall x \in S, \quad d_G(x) = \left\{ \begin{array}{cl} -d^S(x,\Sigma) & \text{if } \ x \in G, \\ 0 & \text{if } \ x \in \Sigma, \\ d^S(x,\Sigma) & \text{if } x \in S \setminus \overline{G}, \end{array} \right.$$

where the distance $d^S(x,\Sigma) = \inf\limits_{p \in \Sigma} d^S(x,p)$ from a point $x \in S$ to Σ is defined from the geodesic distance

$$\forall x, y \in S, \quad d^S(x,y) = \inf \left\{ \int_0^1 |\gamma'(u)| \, \mathrm{d}u, \ \ \gamma \in \mathcal{C}^1([0,1], S), \ \ \gamma(0) = x, \ \gamma(1) = y \right\}.$$

Multiple numerical algorithms allow to calculate the signed distance function to a subdomain of $\mathbb{R}^2$ or $\mathbb{R}^3$: the fast marching method [92], the fast sweeping method [108], among others. Much fewer allow to deal with the case where the ambient medium is a surface of $\mathbb{R}^d$ equipped with a surface triangulation, as is our concern in the present work; in our implementation, we rely on the non trivial extension of the fast marching algorithm proposed in [71].

3.3. Resolution of the level set advection equation on the surface S

In this section, we discuss the numerical realization of the evolution of $G(t)$ over a generic time period $(0, T_{\text{g}})$ (which stands for any of the intervals (t^n, t^{n+1}) featured in Algorithm 1) according to a stationary tangential vector field $V(x)$ (accounting for $V^n(x)$), or normal velocity $v(x)$, starting from an initial datum ϕ_0 (representing ϕ^n). We consider the solution of the level set evolution equation under advection form:

$$\text{(3.2)} \qquad \left\{ \begin{array}{rl} \dfrac{\partial \phi}{\partial t}(t,x) + V(x) \cdot \nabla_S \phi(t,x) = 0, & \text{for } t \in (0, T_{\text{g}}), \ x \in S, \\ \phi(0,x) = \phi_0(x), & \text{for } x \in S, \end{array} \right.$$

or that of its Hamilton-Jacobi counterpart:

$$\text{(3.3)} \qquad \left\{ \begin{array}{rl} \dfrac{\partial \phi}{\partial t}(t,x) + v(x) \, |\nabla_S \phi(t,x)| = 0, & \text{for } t \in (0, T_{\text{g}}), \ x \in S, \\ \phi(0,x) = \phi_0(x), & \text{for } x \in S. \end{array} \right.$$

This topic is quite classical in the literature when the ambient medium is the Euclidean space $\mathbb{R}^2$ or $\mathbb{R}^3$. Efficient numerical schemes are available if the latter is discretized with a Cartesian grid, see notably [94, 96, 69] about (weighted) Essentially Non Oscillatory finite difference methods, and [107] for a survey. In the case when the computational support is a simplicial mesh, we refer to [1, 20, 53, 84] for adapted numerical schemes for the Hamilton-Jacobi equation (3.3), and more recently to [13, 49] about discontinuous Galerkin methods for the advection equation (3.2).

By contrast, the resolution of (3.2) or (3.3) has been seldom considered in the present context where the ambient medium is a surface S in $\mathbb{R}^3$. The aforementioned articles [22, 65, 74, 89, 90], which are based on a level set or closest point description of S, rely on the construction of reformulations of the equations (3.2) or (3.3) on the whole ambient space $\mathbb{R}^3$. In [54], a finite element method on a triangulation of S is proposed for the conservative counterpart of the advection equation (3.2).

In our implementation, following [28], we solve the advection equation (3.2) thanks to the *method of characteristics* [87], a procedure which can be understood as a semi-Lagrangian scheme for the original evolution equation (2.5), see [97]. This method relies on the explicit expression of the solution to (3.2) in terms of the characteristic curves $t \mapsto \chi(t, t_0, x)$ of the velocity field V. Like in Section 2, for $t_0 \in (0, T_{\text{g}})$ and $x \in S$, $\chi(\cdot, t_0, x)$ is characterized by the following ordinary differential equation:

$$\left\{ \begin{array}{l} \frac{\mathrm{d}\chi}{\mathrm{d}t}(t, t_0, x) = V(t, \chi(t, t_0, x)), \quad \text{for } t \in (0, T_{\text{g}}), \\ \qquad\qquad \chi(t_0, t_0, x) = x. \end{array} \right. \tag{3.4}$$

The solution to (3.2) then reads:

$$\phi(t, x) = \phi_0(\chi(0, t, x)), \;\; t \in [0, T_g], \;\; x \in S, \tag{3.5}$$

that is, the value of ϕ at time t and point x is the value taken by the initial function ϕ_0 at the position initially occupied by the particle lying in x at time t.

We leverage this property by discretizing explicitly the formula (3.5). In our implementation, the surface S is equipped with a triangulation $\mathcal{T}$; the velocity field $V(x)$ and the initial datum ϕ_0 are Lagrange $\mathbb{P}_1$ finite element functions on $\mathcal{T}$: they are defined by their values at the vertices of $\mathcal{T}$, and their evaluation at other points on S is achieved by piecewise linear interpolation. For every vertex x of $\mathcal{T}$, we solve the ordinary differential equation (3.4) for the position $\chi(0, T_g, x)$ thanks to a classical Euler scheme, or by a more involved Runge-Kutta strategy. This task brings into play (yet another) subdivision of the time interval $(0, T_{\text{g}})$; it relies on efficient data structures for locating the neighbors of the triangles in the mesh $\mathcal{T}$, and some care is needed when realizing linear combinations of the various velocity vectors attached to different points $x \in S$, which belong to different tangent planes. We refer to [81] for the implementation of Runge-Kutta methods for the solution of ordinary differential equations on surfaces.

Remark 3.1. *In practical situations, some of the characteristic lines $u \mapsto \chi(u, T_g, x)$ may not be defined over the whole interval $[0, T_g]$. This happens when S is open and $V(x) \cdot n_S(x) < 0$ at some points $x \in \partial S$, where $n_S(x) \in T_x S$ is the conormal vector to the surface S. Physically, the velocity field enters the surface at such points, and the equation (3.2) has to be complemented with adequate boundary conditions at such "entrant" regions of the boundary ∂S. When this situation occurs, we simply linearly interpolate the values of V and ϕ outside the surface ∂S from their values on ∂S to complete the integration of (3.4) with consistent values.*

3.4. Meshing of the negative subdomain of a level set function

Let $\mathcal{T}$ be a triangulation of a surface S in $\mathbb{R}^3$, and let $\phi : S \to \mathbb{R}$ be a level set function for a region $G \subset S$, which is supplied by its values at the vertices of $\mathcal{T}$. We aim to construct a new, high-quality computational mesh $\widetilde{\mathcal{T}}$ of S which comprises two submesh $\widetilde{\mathcal{T}_{\text{int}}}$ and $\widetilde{\mathcal{T}_{\text{ext}}}$ for the regions G and $S \setminus \overline{G}$. This operation can be achieved by various strategies, see e.g. [62, 63], and we adopt here that of our previous work [42].

The latter proceeds in two steps:

(1) The triangles $T \in \mathcal{T}$ crossing the 0 level set $\Sigma = \partial G$ of ϕ are identified from the values of this function at the vertices of $\mathcal{T}$, and Σ is discretized explicitly into $\mathcal{T}$. This pretty simple operation is based on the so-called marching tetrahedra algorithm [51] – a variant of the famous marching cubes method [73]: pre-defined patterns are used to split each triangle $T \in \mathcal{T}$ into a valid, conforming configuration where the line segments $\Sigma \cap T$ appears explicitly. This step results in a surface triangulation $\mathcal{T}_{\text{temp}}$ of S featuring explicit submeshes $\mathcal{T}_{\text{temp,int}}$ and $\mathcal{T}_{\text{temp,ext}}$ of G and $S \setminus \overline{G}$. Unfortunately, $\mathcal{T}_{\text{temp}}$ is ill-shaped: it inevitably features thin, nearly flat elements, which makes it unsuitable for the practice of accurate geometric and finite element calculations, see again [39] about this classical issue.

(2) The intermediate mesh $\mathcal{T}_{\text{temp}}$ is iteratively modified to improve the quality of its elements, i.e. to make them close to equilateral, insofar as possible. A new, high-quality mesh $\widetilde{\mathcal{T}}$ of S is obtained, which provides explicit discretizations of G and $S \setminus \overline{G}$.

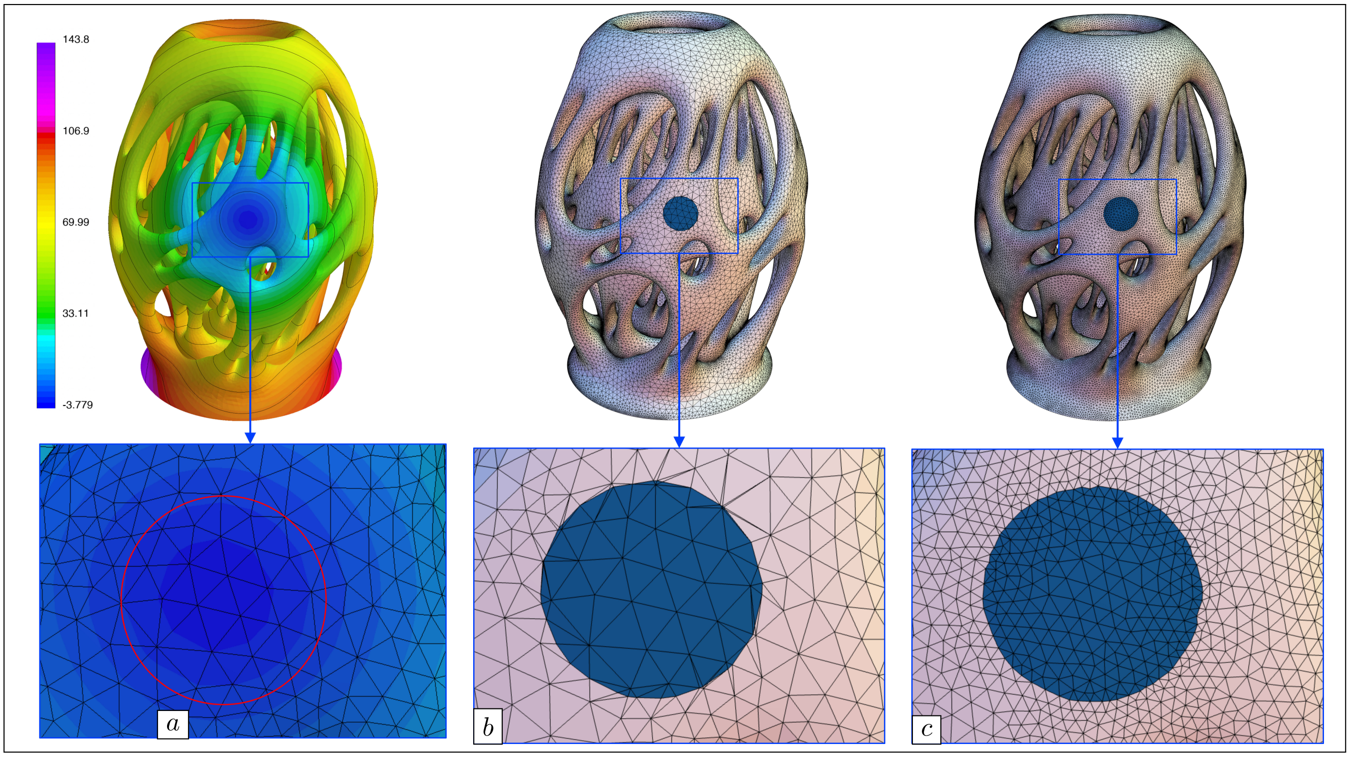

FIGURE 4. *(a) Values of a level set function $\phi : S \to \mathbb{R}$ for a region $G \subset S$; (b) Ill-shaped mesh $\mathcal{T}_{\text{temp}}$ obtained after explicit discretization of G into $\mathcal{T}$; (c) High-quality mesh $\widetilde{\mathcal{T}}$ obtained after remeshing $\mathcal{T}_{\text{temp}}$.*

The latter step is by far the most complicated of the process, and it deserves a few comments. It starts with a series of geometric computations, such as the normal vector n to S, the conormal vector n_Σ to Σ and the deviation of their values at neighboring vertices of $\mathcal{T}$. This allows to identify the suitable local size of the elements of the mesh ensuring an accurate approximation of S and G.

Then, guided by this information, four local remeshing operations are intertwined, provided they improve the global quality of the mesh, see e.g. [61].

- *Edge split.* An edge pq in $\mathcal{T}$ which is "too long" is split by introducing a new point m and reconnecting the triangles sharing pq as an edge accordingly, see Fig. 5 (a).
- *Edge collapse.* The endpoints of an edge pq which is "too short" are merged, see Fig. 5 (b).
- *Edge swap.* The edge pq between two adjacent triangle pqr and pqs is suppressed and the alternate configuration, featuring the edge rs and the triangles rsp and rsq, is retained, see Fig. 5 (c).
- *Vertex relocation.* A vertex p of $\mathcal{T}$ is slightly moved on the continuous surface, see Fig. 5 (d).

Importantly, this remeshing stage leaves room for an adaptation of the computational mesh $\mathcal{T}$ with respect to geometric quantities of S or G, or to a priori or a posteriori error estimates attached to the resolution of partial differential equations on surfaces.

Remark 3.2. *In some applications, such as those of Section 4.4, it happens that the considered surface S is the boundary $\partial\Omega$ of a three-dimensional domain Ω equipped with a tetrahedral mesh $\mathcal{K}$, and that the considered surface triangulation $\mathcal{T}$ is the boundary part of $\mathcal{K}$. In such a situation, the above operations can be applied to the whole tetrahedral mesh $\mathcal{K}$, producing a new mesh $\widetilde{\mathcal{K}}$ of Ω, whose surface part $\widetilde{\mathcal{T}}$ is a triangulation of $\partial\Omega$ enclosing submeshes $\widetilde{\mathcal{T}}_{\text{int}}$ and $\widetilde{\mathcal{T}}_{\text{ext}}$ of G and $S \setminus \overline{G}$, respectively.*

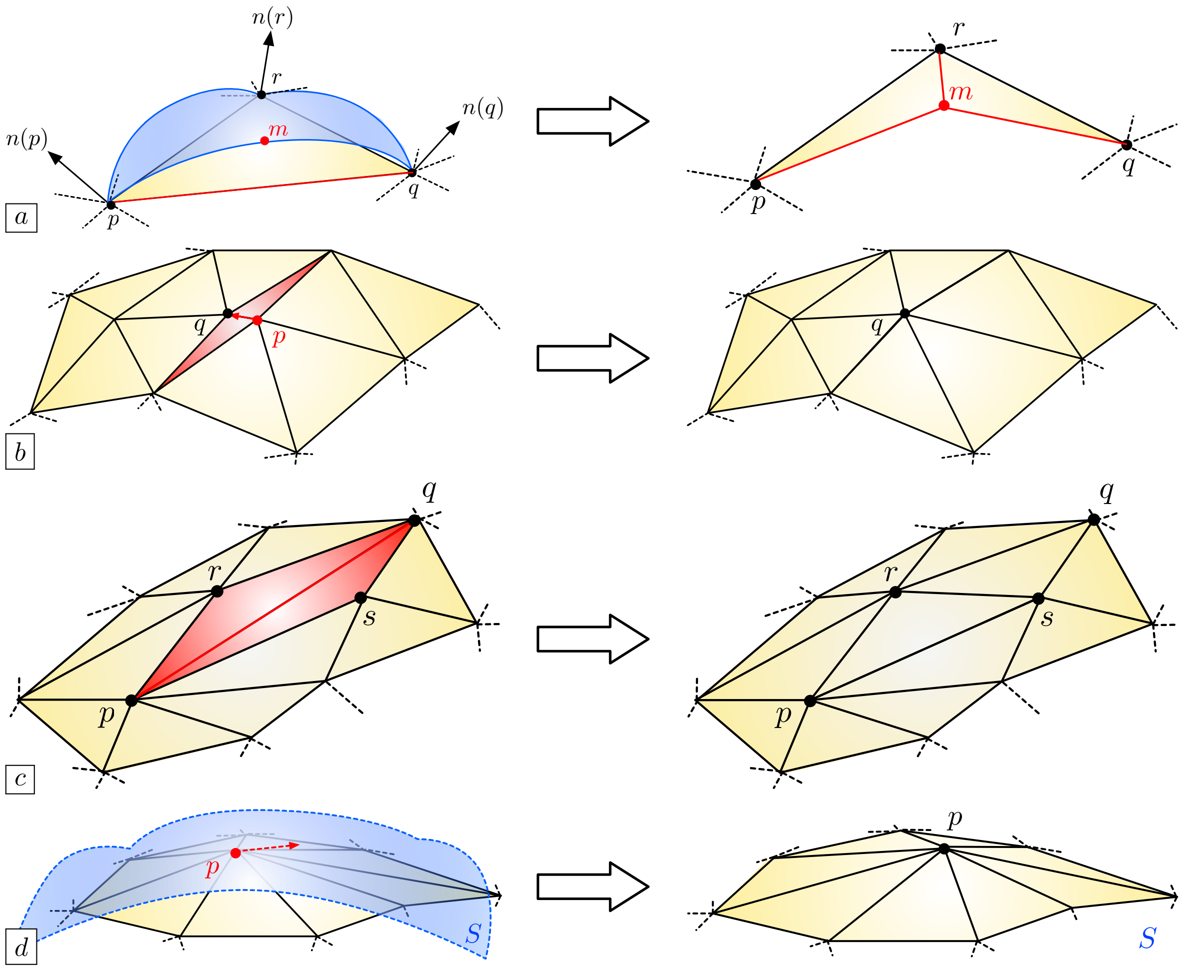

FIGURE 5. *Illustration of the four operations involved in the remeshing strategy of Section 3.4; (a) Split of the "long" edge pq into two edges pm and mp, where the new point m is introduced on S; (b) Collapse of the "short" edge pq; (c) Swap of the edge pq between the triangles pqr and pqs for the alternate configuration, featuring the edge rs and the triangles rsp and rsq; (d) Relocation of the vertex p which slides along S.*

4. Numerical examples

In this section, we present numerical examples illustrating the main features of the proposed evolution Algorithm 1. After a short description of our practical implementation in Section 4.1, the first example in Section 4.2 deals with a simple situation meant to appraise its accuracy, that of a region $G(t)$ on the unit sphere $\mathbb{S}^2$ in $\mathbb{R}^3$ evolving according to its conormal vector field. In Section 4.3 we turn to the numerical simulation of the physical evolution of a fire front, which is driven by the geometric features of the front and those of the underlying landscape S. Finally in Section 4.4, we leverage our methodology to optimize the shape of the regions supporting the boundary conditions of a boundary value problem with respect to a physical performance criterion. The gradient flow associated to the minimization yields an evolution problem of the form considered in this article, where the velocity field is the shape gradient of the objective criterion, depending on the solution to several versions of the considered boundary value problem.

4.1. Numerical framework

As presented in Section 3, the numerical implementation of the examples proposed in the subsequent sections relies on a surface triangulation $\mathcal{T}$ of the ambient surface $S \subset \mathbb{R}^d$, which is modified between the consecutive steps $n = 0, \ldots$ of the evolution process. Typically, various scalar and vector fields are defined and handled on S, such as level set functions $\phi : S \to \mathbb{R}$ for regions $G \subset S$ or velocity fields $V : S \to \mathbb{R}^d$. They are discretized as Lagrange $\mathbb{P}_1$ finite element functions on $\mathcal{T}$, i.e. they are continuous, and their restriction to each triangle $T \in \mathcal{T}$ is affine. As such, these quantities are characterized by their values at the vertices of $\mathcal{T}$ and their evaluation at other points $x \in S$ is realized by linear interpolation from these data.

We rely on various computational tools to conduct the main steps of our evolution Algorithm 1:

- Computations involving finite element functions on the triangulation $\mathcal{T}$, such as their evaluation or the solution of boundary value problems, are handled by the finite element library `MFEM`, described in [12] and available at:

  https://github.com/mfem/mfem/ .
- The computation of the signed distance function d_G to a region $G \subset S$ is realized thanks to the `ISCD Mshdist` software [44], which can be downloaded at:

  https://github.com/ISCDtoolbox/Mshdist ,

  see again Section 3.2.
- The solution of the level set evolution equation (2.5) on the triangulation $\mathcal{T}$ of S relies on the `ISCD Advection` software [28], available at:

  https://github.com/ISCDtoolbox/Advection .

  This tool implements the method described in Section 3.3.
- The operations of our framework related to mesh modification leverage the `MMG` library, found at:

  https://github.com/MmgTools/mmg ,

  see the companion articles [15, 42].

These resources are freely available in the form of open-source libraries and programs, which can be used in a black-box fashion; they have been combined into an educational implementation devoted to 2d and 3d shape and topology optimization problems which is described in [43].

The numerical experiments discussed in the next Sections 4.2 to 4.4 are executed on a regular laptop Apple MacBookPro 18,3 (M1 Pro chip) with 10 cores and 16 GB of memory. The exact code used in their treatment is open source and can be downloaded at the following address:

https://github.com/cbritopacheco/rodin .

4.2. Numerical validation: motion in the direction of the conormal vector field

This first example aims to evaluate the efficiency of our numerical Algorithm 1 in the context of a simple motion where an analytical solution is available. The ambient surface S is the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$; it is equipped with the spherical coordinates (α, β) centered at the origin 0, which induce the following (non injective) parametrization:

$$\begin{array}{rccl} \sigma : & [0, 2\pi] \times [0, \pi] & \to & \mathbb{S}^2 \\ & (\alpha, \beta) & \mapsto & (\cos\alpha \sin\beta, \sin\alpha\sin\beta, \cos\beta). \end{array}$$

In this setting, we wish to track the region $G(t) \subset S$ evolving from the upper spherical cap $\mathcal{C}_a$ with azimuth angle $a \in (0, \frac{\pi}{2})$,

$$G(0) = \mathcal{C}_a := \sigma([0, 2\pi] \times [0, a)),$$

according to the conormal vector field to the interface $\Sigma(t) = \partial G(t)$:

$$\forall t \in (0, T), \;\; x \in \Sigma(t), \quad V(t, x) = n_{\Sigma(t)}(x),$$

over the time period $(0, T)$, with $T = \frac{\pi}{2} - a$, see Fig. 6. Note that $V(t, x)$ can be conveniently extended to the whole surface S as:

$$V(t, x) = \nabla_S d_{G(t)}(x), \quad \text{for a.e. } x \in S, \tag{4.1}$$

where $d_{G(t)}$ is the signed distance function (3.1) to $G(t)$, see e.g. [47] about this property and Remark 2.1.

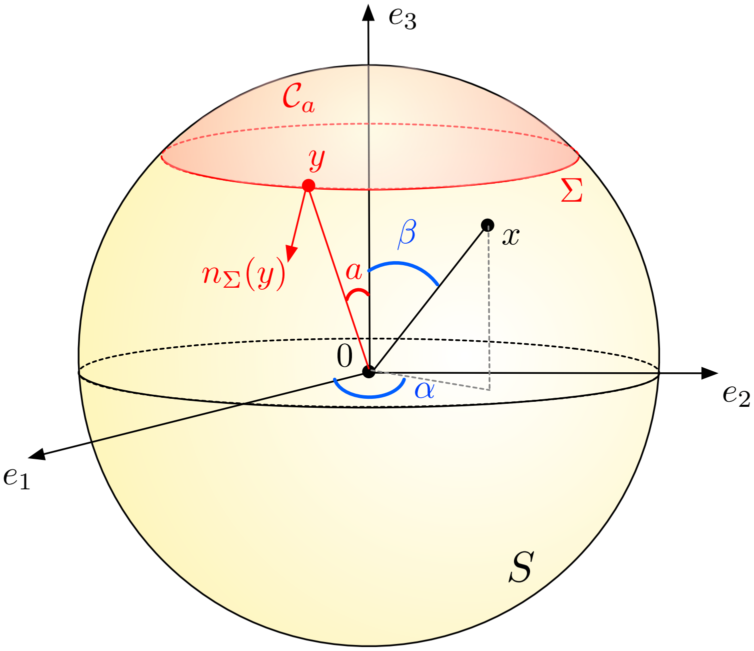

FIGURE 6. *Setting of the motion of a region $G(t)$ within the unit sphere $\mathbb{S}^2$ along the conormal vector field $n_{\Sigma(t)}$ considered in Section 4.2*

As we have mentioned, in this simple setting, all the features attached to the evolution of $G(t)$ can be expressed analytically. Indeed, for $t \in (0,T)$, the deformed domain $G(t)$ is the upper spherical cap with azimuth angle $a+t$:

$$G(t) = \mathcal{C}_{a+t} = \sigma\Big([0,2\pi] \times [0, a+t)\Big). \tag{4.2}$$

In particular, since $T = \frac{\pi}{2} - a$, the final region $G(T)$ coincides with the upper half-sphere. Besides, the geodesic signed distance function $d_{G(t)}$ to $G(t)$ has the following analytical expression:

$$\forall (\alpha,\beta) \in [0,2\pi] \times [0,\pi], \quad d_{\mathcal{C}_a(t)}(\sigma(\alpha,\beta)) = \beta - t - a. \tag{4.3}$$

Finally, let us observe that, in this very specific case where the normal component of the velocity field $V(t,x)$ identically equals 1, the exact solution $\phi(t,x)$ to the evolution equation (2.5) when initialized with the signed distance function $\phi(0,\cdot) = d_{G(0)} = d_{\mathcal{C}_a}$ is given by:

$$\phi(t,x) = d_{\mathcal{C}_a(t)}(x), \tag{4.4}$$

in particular, $\phi(t,\cdot)$ is the signed distance function to $G(t)$ for all $t > 0$.

In our numerical simulation of the motion of $G(t)$ by means of Algorithm 1, the time interval $(0,T)$ is discretized into N subintervals of the form (t^n, t^{n+1}), where $0 \leq n \leq N-1$, $t^n = n\Delta t$, and Δt is a "small" time step. In order to appraise the convergence of our strategy, we measure the least-square difference $\mathcal{E}$ between the numerical level set function ϕ^N for the tracked region G^N (that is, the outcome of the calculation of the signed distance function d_{G^N} with the help of the final mesh $\mathcal{T}^N$ of S) at the final iteration N, and the exact value $\phi(T,\cdot)$ of the solution to (3.2), given by (4.3) and (4.4):

$$\mathcal{E} := \left(\int_S \left|\phi^N(x) - \phi(T,x)\right|^2 \, \mathrm{d}s(x) \right)^{1/2}; \tag{4.5}$$

thus, $\mathcal{E}$ measures the potential accumulation of errors incurred by the various stages of Algorithm 1 throughout the iterations $n = 0, \ldots$, namely the inexact calculation of the signed distance function to the regions G^n, the error in the numerical resolution of the advection equation (3.2) between consecutive times t^n and t^{n+1} and the approximation of the geometry of G^n caused by its explicit discretization into the mesh $\mathcal{T}^n$.

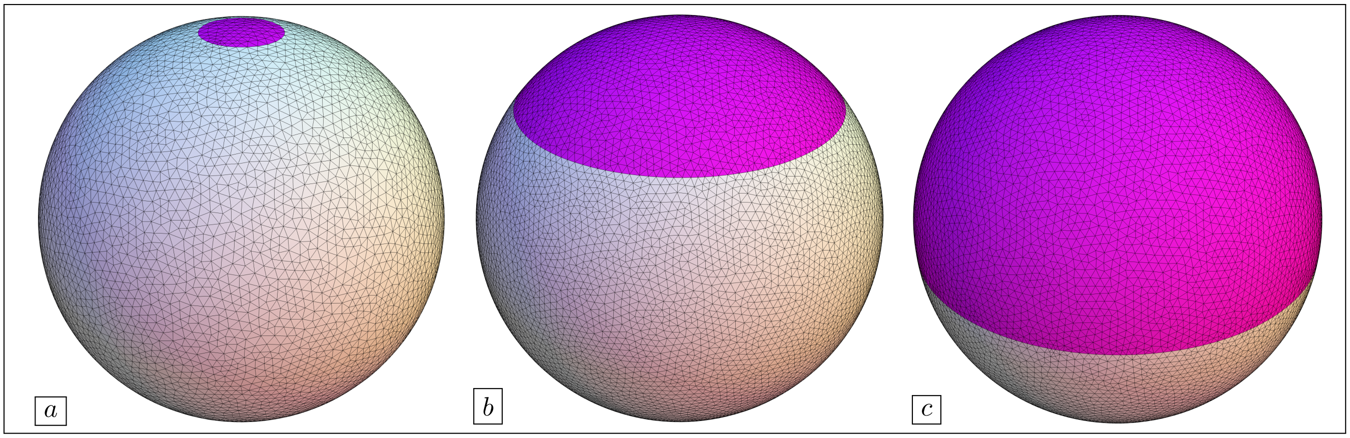

Figure 7. *Evolution $G(t)$ of the spherical cap $\mathcal{C}_a$ under unit conormal velocity considered in Section 4.2; (a) Initial configuration $G(0) = \mathcal{C}_a$; (b) Intermediate configuration $G(t)$at time $t = \pi/4$; (c) Final region $G(T)$ corresponding to the upper half-sphere.*

We wish to study the behavior of the error $\mathcal{E}$ in the limit where the time step Δt vanishes and the mesh is refined, i.e. the average size h of the edges of the mesh tends to 0. Since a fully explicit procedure is used for approximating the solution to (2.5) by that of a series of linear evolution equations, featuring a frozen in time velocity field (see Section 2), we anticipate that it is relevant to relate these two parameters as $\Delta t = ch$, where the parameter$c$ plays the role of a CFL number, accounting for the number of mesh elements crossed by the interface $\Sigma(t)$ during a single time step.

Several numerical simulations of the motion of $G(t)$ are performed, associated with different values of c and h; a few iterations of the evolution process in one of these instances are depicted on Fig. 7, and the behavior of the error $\mathcal{E}$ in the various considered situations is shown in Fig. 8. As expected, for a fixed, “not too small” value of the ratio c, $\mathcal{E}$ tends to 0 as the mesh is refined (i.e. as h tends to 0). On the contrary, this convergence is not observed when c is “small”, which can be attributed to the large accumulation of numerical errors during the various stages of Algorithm 1 when the algorithm has to perform a “too large” number of iterations with respect to the limited precision guaranteed by a fixed mesh size. The results of Fig. 8 suggest to select a value of c within the range $(\frac{1}{3}, 1)$, so that at least $\frac{1}{3}$ element is crossed at each iteration of the process. Note that identical (unreported) experiments taking place in the context of a planar ambient surface S show similar trends as regards the behavior of our strategy with respect to the values of c and h.

4.3. **Wildland fire propagation**

With the increasing global warming and the intensification of human activity, among other factors, wildfires have emerged as a significant concern in recent decades, and much endeavour has been made towards understanding, and ultimately predicting, their dynamics. The obvious limitations of practical experiments raise the need to develop accurate models and efficient numerical simulation algorithms, see e.g. [17, 18, 76, 91, 10, 88] and [80]. In this section, we show how the strategy of Section 3.1 for tracking the evolution of a region $G(t)$ within a surface $S \subset \mathbb{R}^3$ is a valuable tool in the numerical realization of a model for the simulation of such a phenomenon, which is driven by the geometric characteristics of the evolving front and of the landscape surface. We rely on the physical model of [16], which has been validated in e.g. [18, 58, 59] and is used in softwares such as `ForeFire` [58]. For the convenience of the reader, we provide a short, intuitive presentation of this model, referring to [16] for the details.

The ambient surface $S \subset \mathbb{R}^3$ represents the topography of the landscape. It is defined as the graph of a function over a domain in the 2d horizontal plane, or, equivalently, in terms of the height function $s : S \to \mathbb{R}$:

$$\forall x = (x_1, x_2, x_3) \in S, \quad s(x_1, x_2, x_3) = x_3.$$

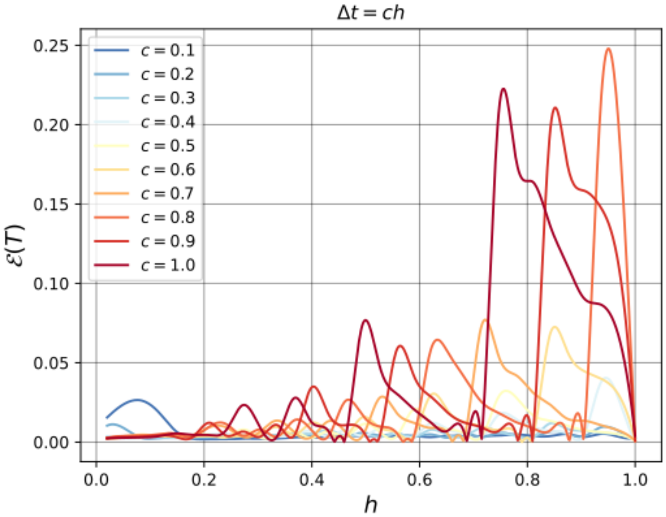

FIGURE 8. *History of the least-square error $\mathcal{E}$ in (4.5) for different values of the mesh size h and "CFL number" c in the conormal advection example of Section 4.2.*

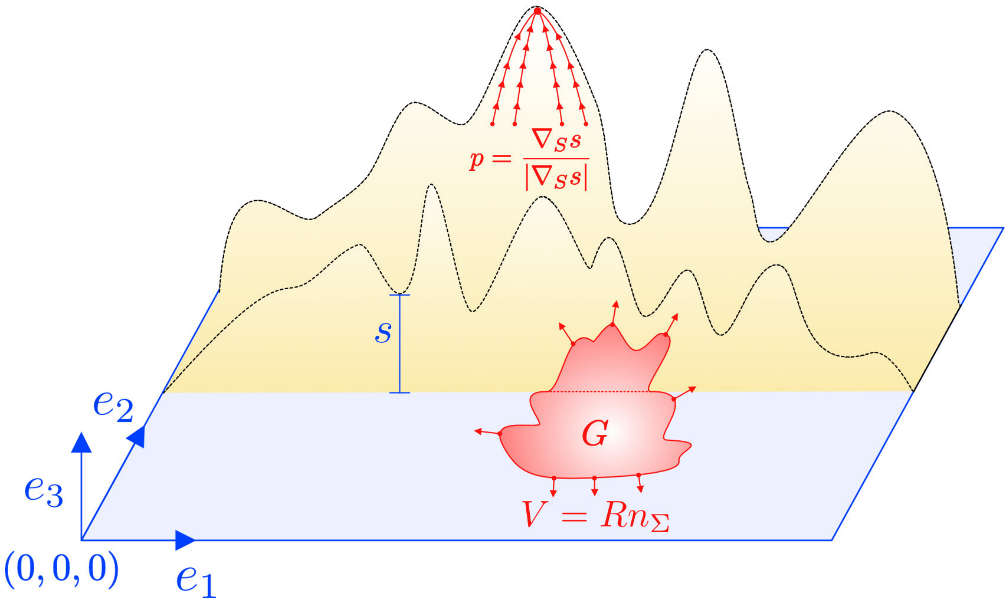

FIGURE 9. *Illustration of the evolution of a fire on a topography. The slope vector p indicates the direction of greatest ascent as visualized by the arrows pointing towards the peaks. The burning region G is indicated by the yellow-red zone whose motion is prescribed by the arrows representing V.*

The ground slope vector p is the unit tangential vector field pointing in the direction of the largest variation of altitude:

$$\forall x \in S, \quad p(x) = \frac{\nabla_S s(x)}{|\nabla_S s(x)|},$$

and the local slope of the land is the angle between this vector and the tangential directions:

$$\alpha : S \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \alpha = \arcsin(p \cdot e_3),$$

see Fig. 9. The evolving subdomain $G(t) \subset S$ and its boundary $\Sigma(t) = \partial G(t)$ represent the burnt region and the fire front, respectively, while the complement $S \setminus \overline{G(t)}$ stands for the vegetal stratum, i.e., the part of the land covered with vegetation that has not yet been ignited. The motion of $\Sigma(t)$ is oriented along its conormal vector field:

$$\forall t \in (0,T), \quad x \in \Sigma(t), \quad V(t,x) = R(t,x) n_{\Sigma(t)}(x), \tag{4.6}$$

where the scalar component $R(t,x) > 0$ is the rate of spread of the front.

The quantity $R(t,x)$ depends on the geometric features of the landscape S, the fire front $\Sigma(t)$, and on the physical characteristics of the actual situation. These data, whose values are determined through measurements are the following:

- The rate of spread $R_0 > 0$ of the burnt region in the absence of slope and wind (expressed in ms^{-1});
- The velocity u_0 of the combustion gas in the absence of slope (in ms^{-1});
- The (dimensionless) ratio $A > 0$ between the incident radiant energy and the ignition energy of the (wet) vegetal fuel;
- The velocity of the wind $U : S \to \mathbb{R}^d$ (in ms^{-1}).

The behavior of the rate of spread $R(t,x)$ at some point x in the fire front $\Sigma(t)$ depends on whether the flame is directed towards the burnt region or the vegetal stratum. This feature is measured by the so-called tilt angle $\gamma(t,\cdot) : \Sigma(t) \to \mathbb{R}$, whose values depend on the velocity of the wind $U(x)$, on the local slope $\alpha(x)$, and on the vector $n_{\Sigma(t)}(x)$ via the following relation:

$$\forall x \in \Sigma(t), \quad \tan \gamma(t,x) = \tan \alpha(x) \cos \beta_p(t,x) + \frac{|U(x)|}{u_0} \cos \beta_U(t,x), \tag{4.7}$$

where $\beta_p(t,x)$ is the angle between the ground slope vector $p(x)$ and $n_{\Sigma(t)}(x)$, and $\beta_U(t,x)$ is the angle between the wind $U(x)$ and $n_{\Sigma(t)}(x)$. Roughly speaking, at some point $x \in \Sigma(t)$, the flame is tilted towards the burnt region if $\gamma(t,x) \leq 0$ and towards the vegetal stratum if $\gamma(t,x) \geq 0$, see Fig. 10. The rate of spread $R(t,x)$ then takes "small" values in the former situation, and large values in the latter one; its definition of $R(t,x)$ brings into play the following two regimes:

- *Case 1 (Slow backing fire spread)* $\gamma(t,x) \leq 0$. The flame axis at x is tilted towards the burnt region and $R(t,x) = R_0$.
- *Case 2 (Fast fire spread)* $\gamma > 0$. The flame axis at x is tilted towards the unburnt vegetation, thus accelerating the ignition of the latter and the expansion of the front; $R(t,x)$ is given by:

$$R(t,x) = 0.5 \left(R_a(t,x) + \left(R_a(t,x)^2 + \frac{48 R_0^2}{\cos \gamma(t,x)} \right)^{\frac{1}{2}} \right), \text{ where} \tag{4.8}$$

$$R_a(t,x) = R_0 + 12 A R_0 \frac{1 + \sin \gamma(t,x) - \cos \gamma(t,x)}{\cos \gamma(t,x)} - 12 R_0 \frac{1}{\cos \gamma(t,x)}.$$

We conduct three experiments of the evolution of a burning region within a landscape S with complex topography, associated to as many different scenarios. The surfaces S used in these examples are fictitious: they are generated as graphs of random non negative functions defined over the horizontal base $[0,50] \times [0,50]$ – where the retained unit for spatial coordinates is the km; see Fig. 11 (a,c) for an illustration. For simplicity, we assume that a homogeneous vegetal stratum covers the entire terrain, so that the physical parameters R_0, u_0 and A are constant; their values are taken from [16] and are reported in Table 1. We also report in there the common values of the numerical parameters used in all three situations.

The use of Algorithm 1 in this context is straightforward: at each iteration $n = 0, \ldots,$ corresponding to the time $t^n = n\Delta t$, the computation of the velocity field $V^n = V(t^n, \cdot)$ in (4.6) and (4.8) depends on that of geometric quantities attached to S and G^n, that is conveniently realized with the help of the mesh $\mathcal{T}^n$ of S, where G^n is explicitly discretized. Note that, in keeping with Remark 2.1, in this example again, the velocity field $V(t,x)$ can be extended to the whole surface S by leveraging the same extension formula (4.1) for the conormal vector field $n_{\Sigma(t)}$ as in the previous Section 4.2.

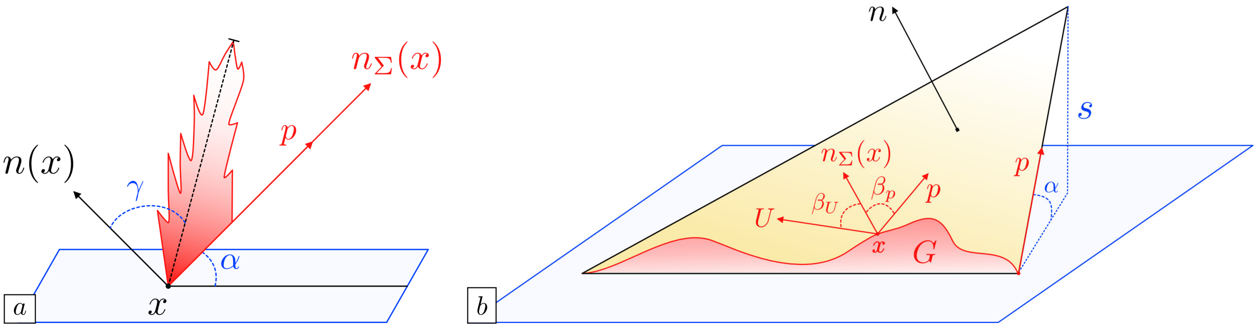

FIGURE 10. *Illustration of the geometric relations involved in the computation of the rate of spread R at the level of one triangle $T \in \mathcal{T}$, in the fire propagation model of Section 4.3.*

Parameter	R_0	u_0	A	Δt	$h_{\max}$
Value	2.7 ms^{-1}	39.8 ms^{-1}	1.25	600 s	400 m

TABLE 1. *Values of the parameters used in the experiments concerning the propagation of a fire front in Section 4.3.*

The landscape S considered in our first experiment is represented in Fig. 11 (b,d), and the initially burnt region $G(0)$ is a surface disk with radius 1 km, centered at the point $x = (x_1, x_2, x_3) \in S$ with coordinates $(x_1, x_2) = (19.5, 19.5)$. In this situation, the effect of the wind is neglected, i.e. $U \equiv 0$. We simulate the evolution of $G(t)$ thanks to Algorithm 1 over the time period $[0, T]$, where $T = 400$ mn, using the parameters reported in Table 1. A few intermediate meshes $\mathcal{T}^n$ obtained in the course of the evolution are displayed in Fig. 12. In this experiment, the values of the rate of spread $R(t, x)$ range between 0.4 and 27.35 ms^{-1}, with an average value of 2.26 ms^{-1}. The values presented here align with findings in [16] as well as in other studies like [98, 33], which also provide data on the mean, minimum, and maximum rates of spread. As expected from the formulas in (4.8), the fire spreads rapidly towards regions where the flame is tilted towards the unburnt region (typically in mountains).

From the technical vantage, let us point out that, at each iteration of the process, the complex landscape surface S and burnt region G^n are equipped with exact, high-quality meshes, which are refined in the vicinity of their sharp features. This allows for accurate calculations of quantities such as the conormal vector $n_{\Sigma(t)}$ to the fire front as opposed to "classical" simulation methods and implementations in the fire dynamics literature, that use marker methods on Cartesian grids and projections or reconstructions of the fire front, see [60, 59, 58, 10].

We next turn to the study of a second scenario, aimed to appraise the behavior of our method in dealing with regions with multiple connected components. The landscape S is the same as that used in the first experiment, see again Fig. 11 (b,d); the initial region $G(0)$ is the reunion of four disjoint surface disks with radius 1 km, centered at the points of S whose whose horizontal coordinates equal $(12.5, 12.5)$, $(12.5, 37.5)$, $(37.5, 12.5)$, $(37.5, 37.5)$, respectively. Still, wind is omitted.

The evolution of $G(t)$ is tracked until the final time $T = 500$ mn. A few snapshots of the evolution process are depicted on Fig. 13. In the course of the evolution, the four initially burnt regions expand and eventually merge – the description of such complex topological changes being considerable eased by the use of our mesh evolution Algorithm 1, which takes advantage of the level set method to deal with the update of the moving region.

We finally turn to a third experiment, where the motion of the burning region $G(t)$ is influenced by the presence of a rotating wind. The landscape S is represented on Fig. 14 (a), and the wind velocity $U : S \to \mathbb{R}^d$ accounts for a rotation around the center of the landscape:

$$\forall x \in S, \quad U(x_1, x_2, x_3) = (x_2 - 25, -(x_1 - 25), 0).$$

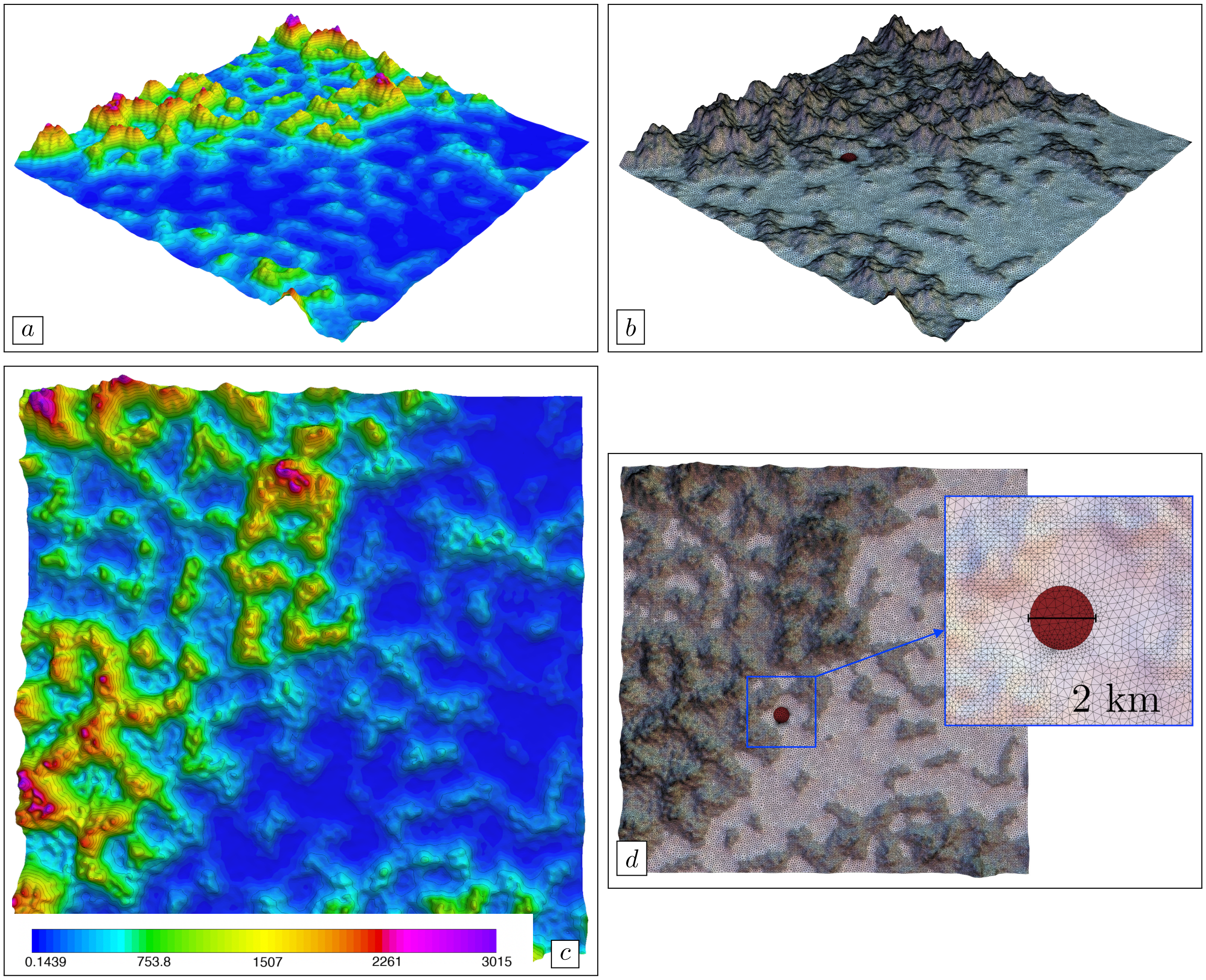

Figure 11. *Landscape surface S used in the first experiment of Section 4.3; the initially burnt region $G(0)$, visualized in (b) and (d), is represented in red, and the color scale in (a) and (c) accounts for the height function s. The lower left corner of S is set to $(0,0,0) \in \mathbb{R}^3$.*

A few snapshots of the simulation of the evolution of $G(t)$ are presented in Fig. 14; here the final time is $T = 160$ mn. Notably, the algorithm successfully accounts for the rotational characteristics of the wind vector field and effectively managed the merging of burning regions resulting from such complex motions. Understandably enough, the burning region tends to expand more rapidly when the fire is tilted towards the unburnt vegetation and the latter is located in the general direction of the wind. Moreover, the motion exhibits a faster rate of spread $R(t,x)$ compared to that in the previous experiments, which is, again, caused by the wind. Again, the algorithm successfully manages to account for the merging of the various components of the burnt region.

Let us conclude this section with two general comments. Firstly, the meshes $\mathcal{T}^n$ involved in each of the above three experiments contain on average about 240,000 triangles, and each iteration of our algorithm takes approximately 25 seconds, which suggests that our implementation proves quite efficient at simulating fire propagation in real time. Secondly, as far as the sensitivity of the simulation with respect to the choice of the parameters h and Δt is concerned, the same trends as in Section 4.2 are observed: unreported numerical tests indicate that the general aspect of the burnt region does not depend very much on the choice of these parameters, as long as the time step Δt is of the order of the mesh size h.

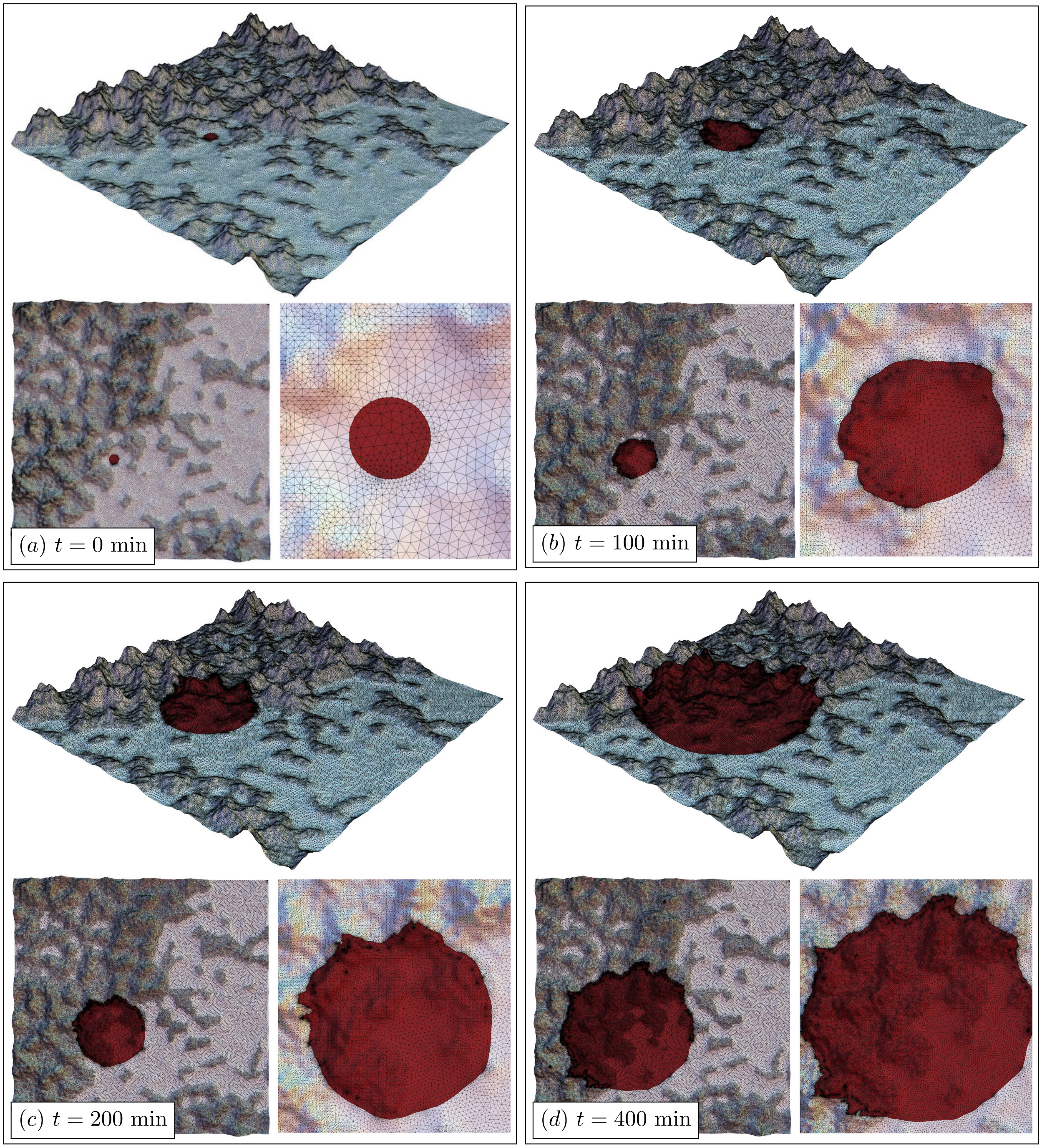

FIGURE 12. *Snapshots of the first experiment of propagation of a fire front conducted in Section 4.3.*

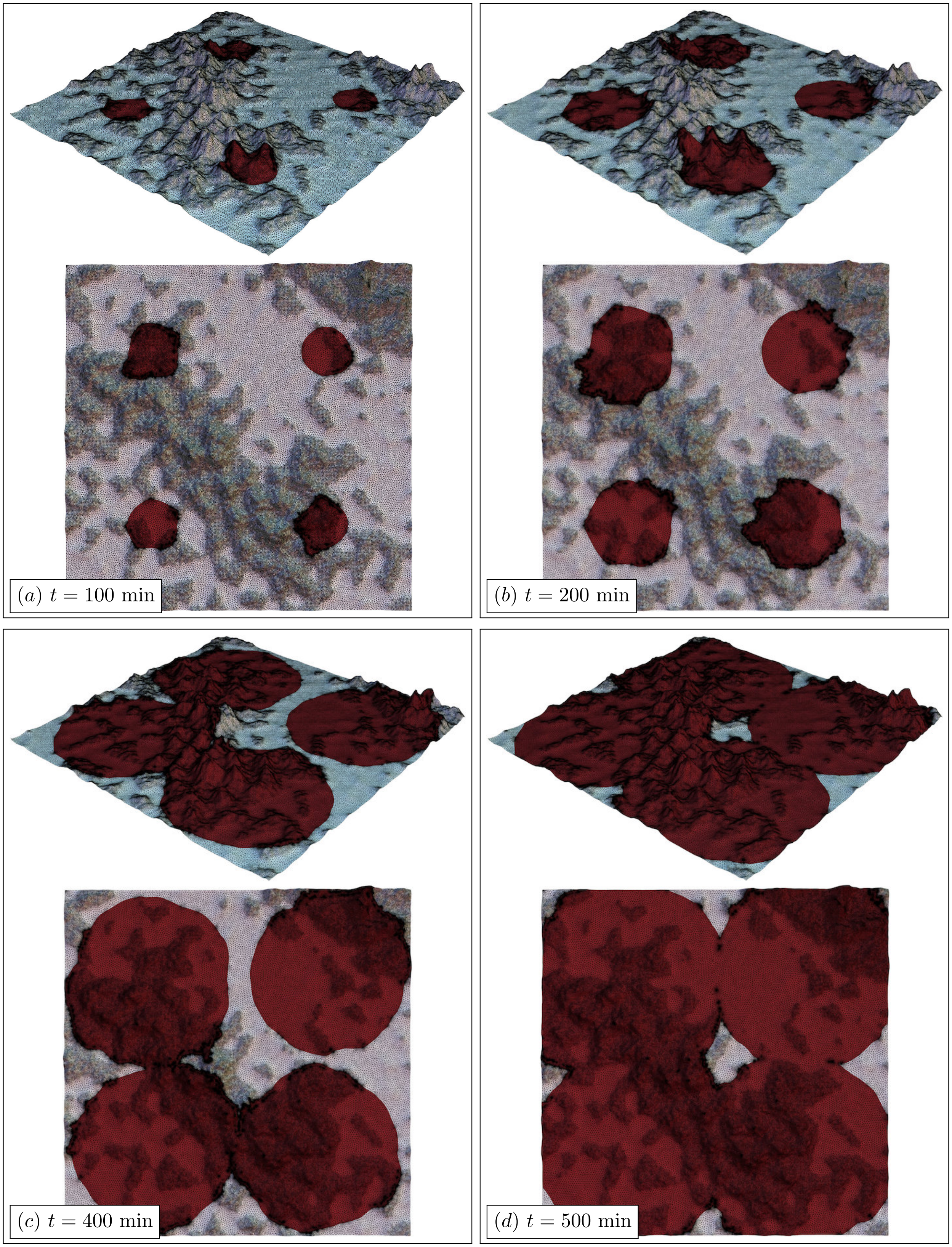

FIGURE 13. *Snapshots of the second experiment of propagation of a fire front conducted in Section 4.3.*

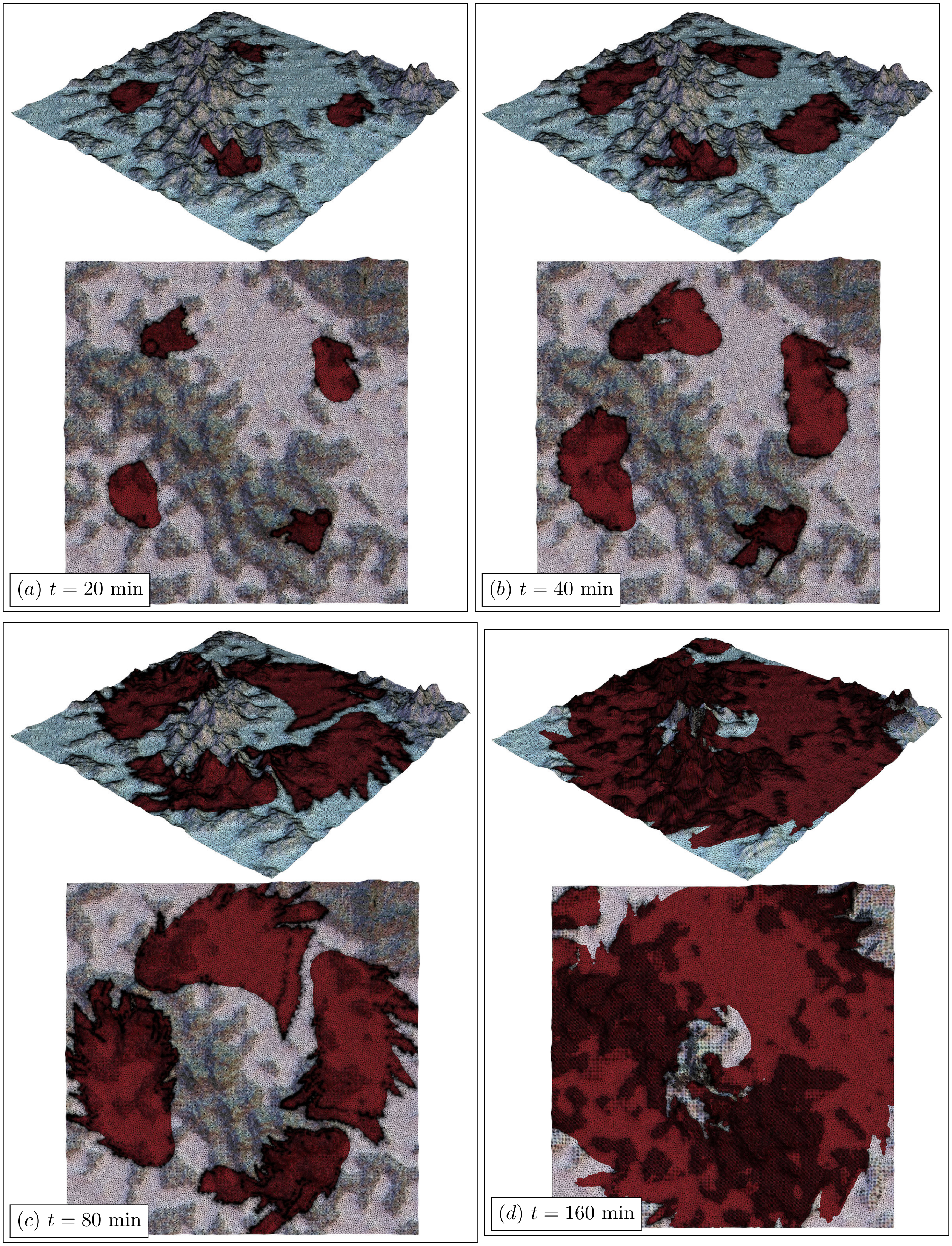

Figure 14. *Snapshots of the third experiment of propagation of a fire front conducted in Section 4.3.*

4.4. Shape optimization of the regions supporting boundary conditions

This section exemplifies the efficiency of our algorithmic strategy in a context where the need for an exact, meshed representation of the evolving region $G(t) \subset S$ stems from the desire to accurately solve physical boundary value problems involving $G(t)$ with the finite element method.

More precisely, we consider a situation where the ambient surface S is the boundary of a fixed, smooth and bounded domain Ω in $\mathbb{R}^d$, and the region G of interest is the support of a particular boundary condition accompanying a physical partial differential equation posed on Ω. The shape of G is optimized with respect to a measure $J(G)$ of the performance of the configuration: we track the region $G(t)$ evolving in (pseudo-) time t, starting from an initial design G^0, via a velocity field $V(t,x)$ induced by the shape derivative of $J(G)$.

After recalling a few basic notions about shape optimization in the first Section 4.4.1, which are adapted to the present context where the ambient medium is a surface, we formulate in Section 4.4.2 a model problem in thermal mechanics where the region of the boundary of the computational domain bearing homogeneous Dirichlet boundary conditions is optimized. A 3d numerical example is presented and discussed in Section 4.4.3.

4.4.1. *Optimization of the shape of a region on a surface by the method of Hadamard*

Shape and topology optimization aims to find the optimal design of a device with respect to a criterion measuring its physical performance. This general ambition has received a tremendous amount of attention lately, due to the dramatic increase in the cost of raw materials and the pressing need for energy savings. Spurred by the development of efficient mathematical programming techniques and the rise in computational power, shape and topology optimization is by now a fairly mature discipline, which is able to handle increasingly realistic situations; it is nowadays employed in a whole gammut of physical and industrial applications, in structural mechanics, fluid mechanics, electromagnetism, etc. We refer to [7, 8, 21, 79] for overviews of such applications and for expositions of the main numerical frameworks.

The present investigations arise in a context which slightly differs from the prevalent studies in shape and topology optimization, while leveraging the main concepts: we aim to optimize the shape of a region $G \subset S$ of a fixed surface $S \subset \mathbb{R}^d$. More precisely, we consider the problem

$$\min_{G \subset S} J(G), \tag{4.9}$$

where $J(G)$ is the considered objective function; for the sake of simplicity, we omit constraints in this formulation, although they could be brought into play without much additional difficulty. We refer to e.g. [45, 101, 105] for various concrete applications of this setting.

The numerical solution of a problem such as (4.9) hinges on the derivative of the objective $J(G)$. This notion can be given various meanings, and in the present study, we use an adapted version of the boundary variation method of Hadamard, see [8, 47, 67, 82, 95]. In a nutshell, we consider variations of the region G of the form

$$G_\theta := (\mathrm{Id} + \theta)(G), \text{ where } \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \text{ is a “small” tangential vector field:} \\ ||\theta||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1 \text{ and } \theta \cdot n = 0, \tag{4.10}$$

see Fig. 15 for an illustration. The function $J(G)$ is then said to be shape differentiable at G if the underlying mapping $\theta \mapsto J(G_\theta)$, from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $\mathbb{R}$, is Fréchet differentiable at $\theta = 0$. Denoting by $J'(G)(\theta)$ the corresponding “shape derivative”, the following expansion holds:

$$J(G_\theta) = J(G) + J'(G)(\theta) + \mathrm{o}(\theta), \text{ where } \frac{|\mathrm{o}(\theta)|}{||\theta||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)}} \xrightarrow{\theta \to 0} 0.$$

Remark 4.1. *The considered variations G_θ of a region $G \subset S$ in (4.10) are defined from tangential vector fields θ, accounting for a sliding displacement of G while the ambient surface S remains unaltered. Note that this operation does not exactly leave S invariant, which is only observed at “first order” in terms of θ; a definition of the variation G_θ involving the flow of the tangential vector field θ would enforce strictly this property, as in the so-called velocity method discussed in [95]. For simplicity, and since both approaches yield the same notion of first-order derivative of a function $J(G)$, we ignore this technicality in the following.*

On a different note, variations (4.10) of a region $G \subset S$ can be defined from non tangential vector fields θ, resulting in an additional "normal" motion of the surface S itself. This allows to optimize jointly the shape of the underlying surface and that of the region G, as in our previous work [45].

Under mild assumptions, the shape derivative $J'(G)(\theta)$ of a function $J(G)$ of the region $G \subset S$ turns out to be of the form

$$J'(G)(\theta) = \int_\Sigma v_G \, \theta \cdot n_\Sigma \, \mathrm{d}\ell, \tag{4.11}$$

where the scalar field $v_G : \Sigma \to \mathbb{R}$ depends on the region G and the considered objective function $J(G)$. Depending on the nature of the latter, the expression of v_G may involve the solution to one or several boundary value problems attached to S and G, see for instance (4.11) below. The structure (4.11) readily yields a descent direction for $J(G)$ as $\theta = -v_G n_\Sigma$. In practice, more involved strategies are often used to extract a suitable descent direction, see e.g. [7] and the references therein.

The numerical implementation of these concepts fits in the general context of the evolution of a region $G(t)$ within a surface $S \subset \mathbb{R}^d$ discussed in Section 2. Introducing a pseudo-time t, one considers the evolution $G(t)$ of the region starting from an initial guess $G^0 \subset S$, under the effect of the velocity field

$$V(t,x) = -v_{G(t)} n_{\Sigma(t)},$$

where v_G is defined in (4.11). This resulting sequence of regions $G(t)$ smoothly decreases the value of the criterion $J(G)$ until a (local) minimizer of (4.9) is attained. The numerical realization of this program can be conducted by applying Algorithm 1.

Remark 4.2.

- *Shape optimization problems of the form (4.9) are usually ill-posed, due to the so-called homogenization effect: briefly, to get optimal with respect to a rather arbitrary criterion $J(G)$ the shape G tends to develop very thin patterns, up to the "microscopic" level, see e.g. [2, 8] about this phenomenon. This effect also explains the existence of multiple local minima.*
- *The method of Hadamard, relying on variations of a region G of the form (4.10) does not allow for all types of topological changes: separate parts of the boundary Σ of the optimized region G may collide and merge, but no hole can emerge inside G. Since the problem (4.9) has multiple local minimizers, this makes the optimization process very sensitive to the initial design. To remedy this, the above setting is usually complemented with topological derivatives – a notion of sensitivity of $J(G)$ with respect to the nucleation of small holes. The development of such techniques in the present context is the subject of future work, see Section 5.*

4.4.2. *A model optimization problem of the regions supporting boundary conditions of a thermal mechanics problem*

In order to illustrate the general setting of Section 4.4.1, we consider a situation in thermal mechanics which elaborates on our previous work [45]. Let Ω be a fixed, smooth domain in $\mathbb{R}^d$, made of a material with smooth thermal conductivity satisfying

$$\text{There exist } 0 < \alpha \leq \beta < \infty \quad \text{s.t.} \quad \forall x \in \Omega, \quad \alpha \leq \gamma(x) \leq \beta.$$

As depicted on Fig. 15, the boundary $S = \partial\Omega$ is divided into three disjoint regions:

$$S = \overline{G} \cup \overline{\Gamma_N} \cup \overline{\Gamma}, \quad \overline{G} \cap \overline{\Gamma_N} = \emptyset,$$

where

- A heat flux $g \in L^2(\Gamma_N)$ is imposed on Γ_N;
- The temperature is maintained at fixed temperature 0 on G;
- The domain Ω is insulated from the outside on Γ.

Assuming that a heat source $f \in L^2(\Omega)$ is acting within the medium, the temperature u_G inside Ω is then the unique solution in $H^1(\Omega)$ to the following boundary value problem:

$$\left\{ \begin{array}{cl} -\text{div}(\gamma \nabla u_G) = f & \text{in } \Omega, \\ u_G = 0 & \text{on } G, \\ \gamma \frac{\partial u_G}{\partial n} = g & \text{on } \Gamma_N, \\ \gamma \frac{\partial u_G}{\partial n} = 0 & \text{on } \Gamma. \end{array} \right. \tag{4.12}$$

In this situation, we aim to identify the geometry of the region G which minimizes the temperature within the domain Ω, namely, we consider the shape optimization problem (4.9), involving the functional $J(G)$

$$J(G) = T(G) + \ell \text{Per}(G),$$

where $T(G)$ is the mean temperature inside Ω:

$$T(G) = \frac{1}{|\Omega|} \int_\Omega u_G \, \text{d}x,$$

and $\text{Per}(G) = \int_G \text{d}s$ is the surface area of G.

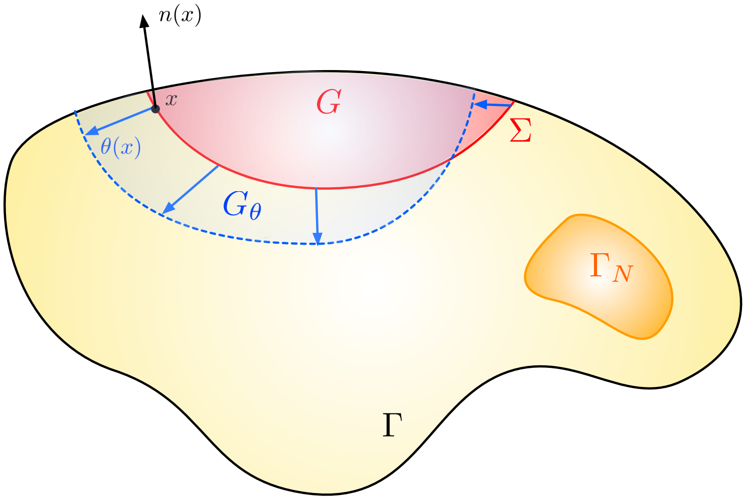

Figure 15. *Optimization of the region G of the boundary S of a shape with respect to a physical performance criterion by the method of Hadamard in Sections 4.4.1 and 4.4.2.*

Beyond the classical issues associated to the computation of shape derivatives, the present situation raises a number of issues, which are notably due to the weak singularity of the solution u_G to (4.12) near Σ, see [66]. To alleviate these, following [45], we consider an approximate version $T_\varepsilon(G)$ of the mean temperature functional $T(G)$, which leads to the approximate objective $J_\varepsilon(G)$ defined by:

$$J_\varepsilon(G) = T_\varepsilon(G) + \ell \text{Per}(G), \text{ where } T_\varepsilon(G) = \frac{1}{|\Omega|} \int_\Omega u_{G,\varepsilon} \, \text{d}x, \tag{4.13}$$

and $u_{G,\varepsilon}$ is the solution to an approximate version of (4.12) where the transition $\Sigma := \partial G$ between G and Γ is smoothed:

$$\left\{ \begin{array}{cl} -\text{div}(\gamma \nabla u_{G,\varepsilon}) = f & \text{in } \Omega, \\ \gamma \frac{\partial u_{G,\varepsilon}}{\partial n} + h_\varepsilon u_{G,\varepsilon} = 0 & \text{on } G \cup \Gamma, \\ \gamma \frac{\partial u_{G,\varepsilon}}{\partial n} = g & \text{on } \Gamma_N. \end{array} \right. \tag{4.14}$$

In this formulation, $h_\varepsilon : S \to \mathbb{R}$ is defined by:

$$h_\varepsilon(x) = \frac{1}{\varepsilon} h\left(\frac{d_G(x)}{\varepsilon} \right), \tag{4.15}$$

bringing into play the signed distance function d_G to G on $\partial\Omega$, and $h : \mathbb{R} \to \mathbb{R}$ satisfies:

$$0 \leq h \leq 1, \;\; h \equiv 1 \text{ on } (-\infty, -1), \;\; h(0) > 0, \;\; h \equiv 0 \text{ on } [1, \infty). \tag{4.16}$$

Omitting details for simplicity, the shape derivative of the resulting functional $J_\varepsilon(G)$ can be approximated by the following formula:

$$\text{For any tangential deformation } \theta, \quad J'_\varepsilon(G)(\theta) \approx -\frac{1}{\varepsilon^2}\int_\Sigma u_{G,\varepsilon} \; p_{G,\varepsilon} \; \theta \cdot n_\Sigma \, \mathrm{d}\ell + \ell \int_\Sigma \theta \cdot n_\Sigma \, \mathrm{d}\ell, \tag{4.17}$$

where $u_{G,\varepsilon} \in H^1(\Omega)$ is the solution to (4.14), and $p_{G,\varepsilon} \in H^1(\Omega)$ is the solution to the adjoint problem:

$$\begin{cases} -\mathrm{div}(\gamma \nabla p_{G,\varepsilon}) = -1 & \text{in } \Omega, \\ \gamma \frac{\partial p_{G,\varepsilon}}{\partial n} + h_\varepsilon p_{G,\varepsilon} = 0 & \text{on } G \cup \Gamma, \\ \gamma \frac{\partial p_{G,\varepsilon}}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases} \tag{4.18}$$

see [3, 45] for the details.

4.4.3. *Numerical example*

In order to appraise the efficiency of the proposed framework, we consider the mechanical part Ω depicted on Fig. 16. The latter is equipped with a tetrahedral mesh $\mathcal{K}$. Here, the conductivity of the material is set to $\gamma \equiv 1$, the volumetric source f identically equals 1 and the region Γ_N where inhomogeneous Neumann boundary conditions are imposed is empty.

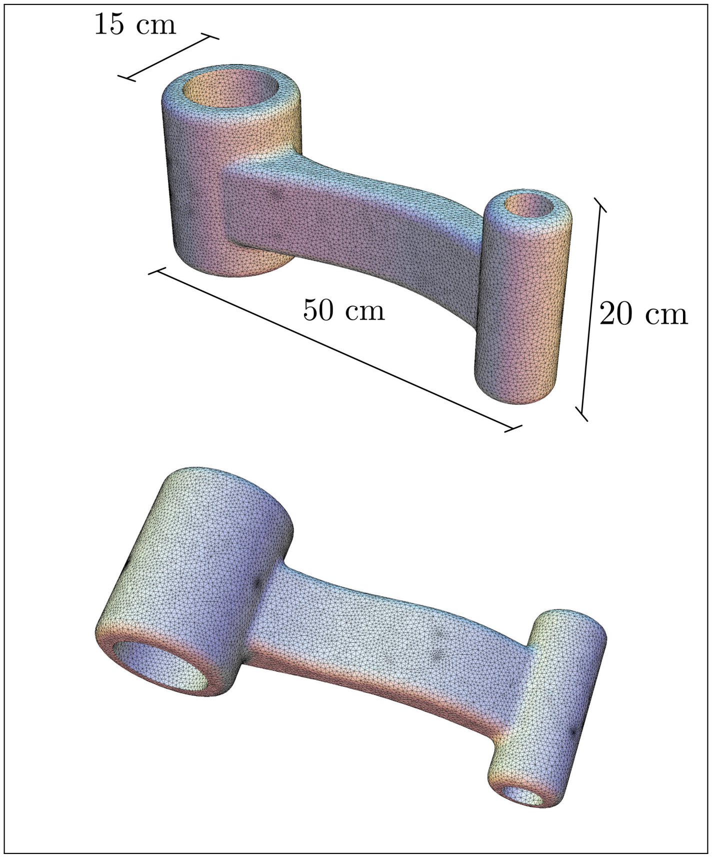

Figure 16. *Illustration of the mechanical part Ω where boundary conditions are optimized in Section 4.4.3.*

In this setting, we solve the following unconstrained shape optimization problem associated to the functional $J_\varepsilon(G)$ in (4.13):

$$\min_{G\subset S} J_\varepsilon(G), \text{ where } J_\varepsilon(G) = T_\varepsilon(G) + \ell\text{Per}(G), \tag{4.19}$$

and the Lagrange multiplier ℓ is set to 0.01. The average size of the mesh edges equals $h = 0.07$. A few intermediate designs obtained in the course of the optimization process are displayed in Fig. 17; the final design and the associated convergence history are reported on Fig. 18. The values of the objective function $J_\varepsilon(G)$ smoothly decrease to a local minimum. Interestingly, the optimization process starts by expanding the region G endowed with homogeneous Dirichlet boundary conditions in an attempt to minimize the mean temperature $T_\varepsilon(G)$ within Ω. Then, the algorithm attempts to distribute this region everywhere in S while avoiding to create patterns with large area, which results in the creation of tree-like branches. This branching phenomenon agrees with typical results in the optimal design of thermal structures, see for instance [57], and it is rooted in the homogenization theory [2].

The very same problem (4.19) is considered with the same initialization, but the different numerical parameters $\ell = 0.05$ and $h = 0.05$; the optimized region G resulting from this experiment and the convergence history are reported on Fig. 19.

5. Conclusions and perspectives

In this article, we have proposed an efficient numerical framework for tracking arbitrarily large evolutions of a region $G(t)$ embedded within a fixed (two- or) three-dimensional surface S. Our method critically hinges on the combination of an explicit representation of $G(t)$ (i.e. with a high-quality mesh) allowing for precise geometric and mechanical computations, with the level set method for capturing its (possibly dramatic) motion. The cornerstone of this strategy is a set of efficient numerical algorithms for passing from one representation to the other. Beyond its numerical validation, we have illustrated the efficiency of this strategy with two physical applications: the evolution of a fire front under a velocity field depending on its geometric features and on those of the landscape surface S, and the optimization of the region of the boundary $S = \partial\Omega$ of a 3d domain Ω bearing the homogeneous Dirichlet boundary condition of a related problem in thermal mechanics.

These developments pave the way to multiple interesting applications that we intend to consider in the near future. Notably, elaborating upon our previous work [45], we wish to extend the range of the model example of Section 4.4 to different mechanical contexts, including that of linearly elastic mechanical structures, where the optimization of regions supporting boundary conditions encompasses the design of fixture systems. Furthermore, leveraging the results of the recent contribution [24], we wish to design and incorporate a notion of topological derivative dedicated to the optimization of such regions. On a different note, and on a perhaps longer term, we wish to investigate the reinforcement and design of openings in shell structures, whose delicate mechanical equations could benefit from the datum of a high-quality, body-fitted computational meshes.

Acknowledgements. The authors are very grateful to E. Bonnetier and R. Estevez for the multiple discussions which initially motivated this work and consistently oriented its realization. This work was completed while C. D. was visiting the Laboratoire Jacques-Louis Lions from Université Paris-Sorbonne, whose hospitality is thankfully acknowledged.

References

[1] R. Abgrall, *Numerical discretization of the first-order hamilton-jacobi equation on triangular meshes*, Communications on pure and applied mathematics, 49 (1996), pp. 1339–1373.

[2] G. Allaire, *Shape optimization by the homogenization method*, vol. 146, Springer Science & Business Media, 2002.

[3] G. Allaire, C. Dapogny, G. Delgado, and G. Michailidis, *Multi-phase structural optimization via a level set method*, ESAIM: Control, Optimisation and Calculus of Variations, 20 (2014), pp. 576–611.

[4] G. Allaire, C. Dapogny, and P. Frey, *Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh*, Comptes Rendus Mathematique, 349 (2011), pp. 999–1003.

[5] ———, *A mesh evolution algorithm based on the level set method for geometry and topology optimization*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 711–715.

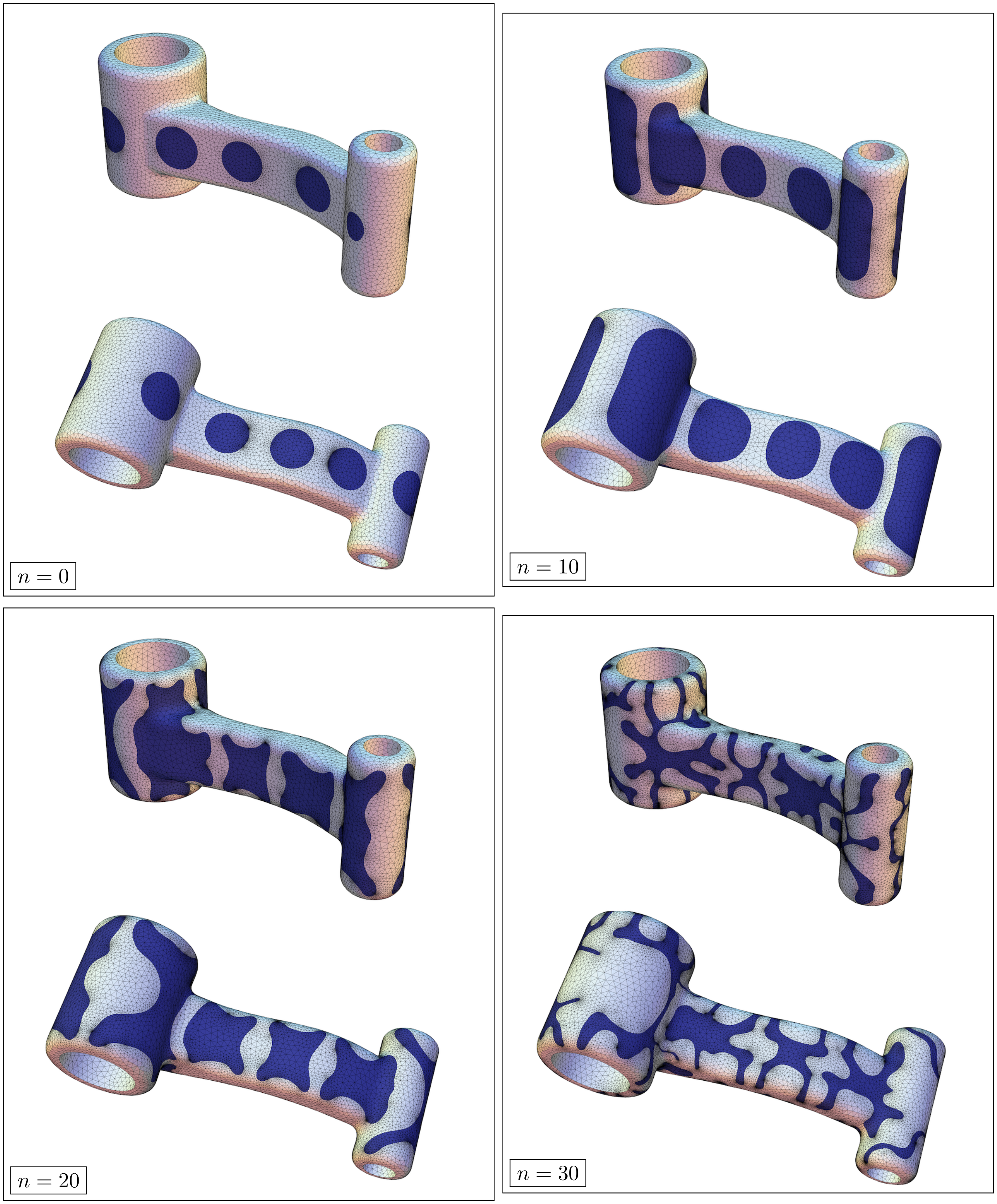

FIGURE 17. *Boundary optimization process of the mechanical device considered in Section 4.4 with penalization parameter* $\ell = 0.01$ *and mesh size* $h_{\max} = 0.08$. *The optimized region* G *bearing homogeneous Dirichlet boundary conditions is represented in blue.*

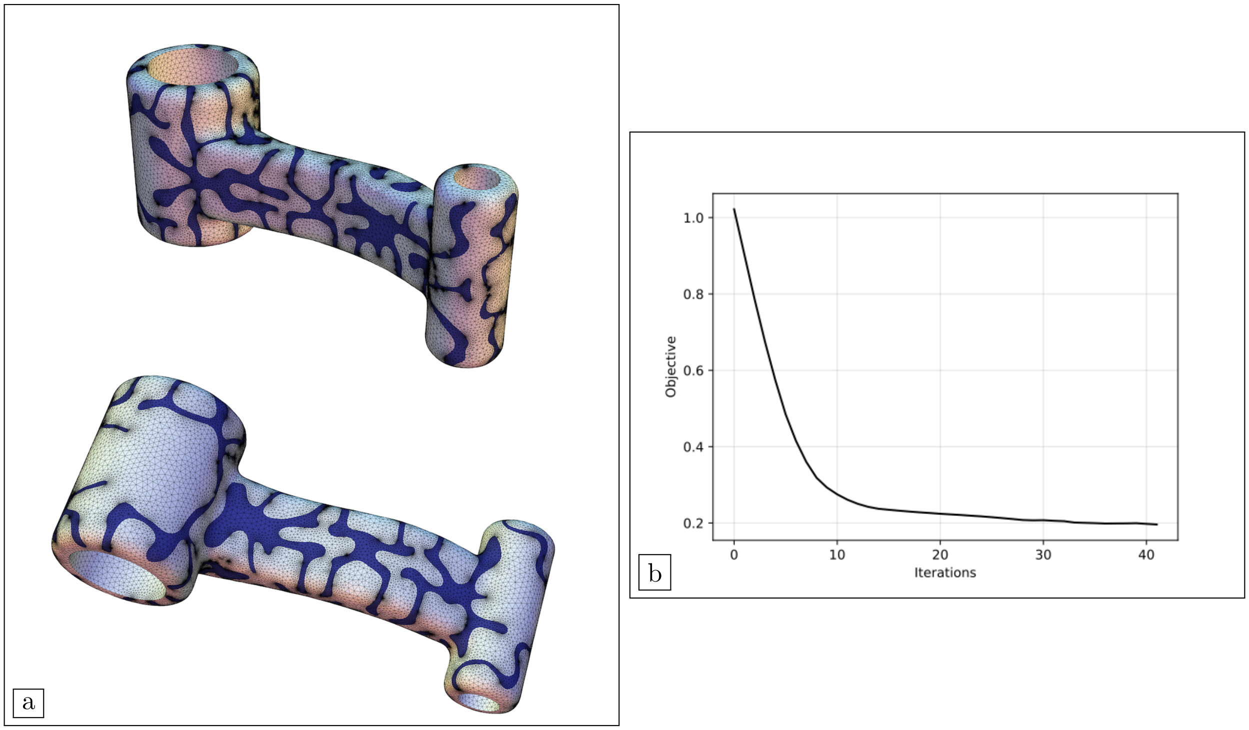

Figure 18. *(a) Optimized design ($n = 40$) in the boundary optimization process of the mechanical device considered in Section 4.4 with penalization parameter $\ell = 0.01$ and mesh size $h_{\max} = 0.08$ and (b) Convergence history.*

[6] ———, *Shape optimization with a level set based mesh evolution method*, Computer Methods in Applied Mechanics and Engineering, 282 (2014), pp. 22–53.

[7] G. Allaire, C. Dapogny, and F. Jouve, *Shape and topology optimization*, in Geometric partial differential equations, part II, A. Bonito and R. Nochetto eds., Handbook of Numerical Analysis, 22 (2021), pp. 1–132.

[8] G. Allaire and M. Schoenauer, *Conception optimale de structures*, vol. 58, Springer, 2007.

[9] L. Ambrosio, N. Dancer, and L. Ambrosio, *Geometric evolution problems, distance function and viscosity solutions*, Springer, 2000.

[10] M. Ambroz, M. Balažovjech, M. Medl'a, and K. Mikula, *Numerical modeling of wildland surface fire propagation by evolving surface curves*, Advances in Computational Mathematics, 45 (2019), pp. 1067–1103.

[11] H. B. Ameur, M. Burger, and B. Hackl, *Level set methods for geometric inverse problems in linear elasticity*, Inverse Problems, 20 (2004), p. 673.

[12] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, et al., *Mfem: A modular finite element methods library*, Computers & Mathematics with Applications, 81 (2021), pp. 42–74.

[13] B. Ayuso and L. D. Marini, *Discontinuous galerkin methods for advection-diffusion-reaction problems*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 1391–1420.

[14] T. J. Baker, *Mesh movement and metamorphosis*, Engineering with Computers, 18 (2002), pp. 188–198.

[15] G. Balarac, F. Basile, P. Bénard, F. Bordeu, J.-B. Chapelier, L. Cirrottola, G. Caumon, C. Dapogny, P. Frey, A. Froehly, et al., *Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing*, to appear in Maths in Action, (2021).

[16] J. H. Balbi, F. Morandini, X. Silvani, J. B. Filippi, and F. Rinieri, *A physical model for wildland fires*, Combustion and Flame, 156 (2009), pp. 2217–2230.

[17] J.-H. Balbi, J.-L. Rossi, T. Marcelli, and P.-A. Santoni, *A 3d physical real-time model of surface fires across fuel beds*, Combustion Science and Technology, 179 (2007), pp. 2511–2537.

[18] J. Baptiste Filippi, F. Bosseur, C. Mari, C. Lac, P. Le Moigne, B. Cuenot, D. Veynante, D. Cariolle, and J.-H. Balbi, *Coupled atmosphere-wildland fire modelling*, Journal of Advances in Modeling Earth Systems, 1 (2009).

[19] N. Barral and F. Alauzet, *Three-dimensional cfd simulations with large displacement of the geometries using a connectivity-change moving mesh approach*, Engineering with Computers, 35 (2019), pp. 397–422.

[20] T. J. Barth and J. A. Sethian, *Numerical schemes for the hamilton–jacobi and level set equations on triangulated domains*, Journal of Computational Physics, 145 (1998), pp. 1–40.

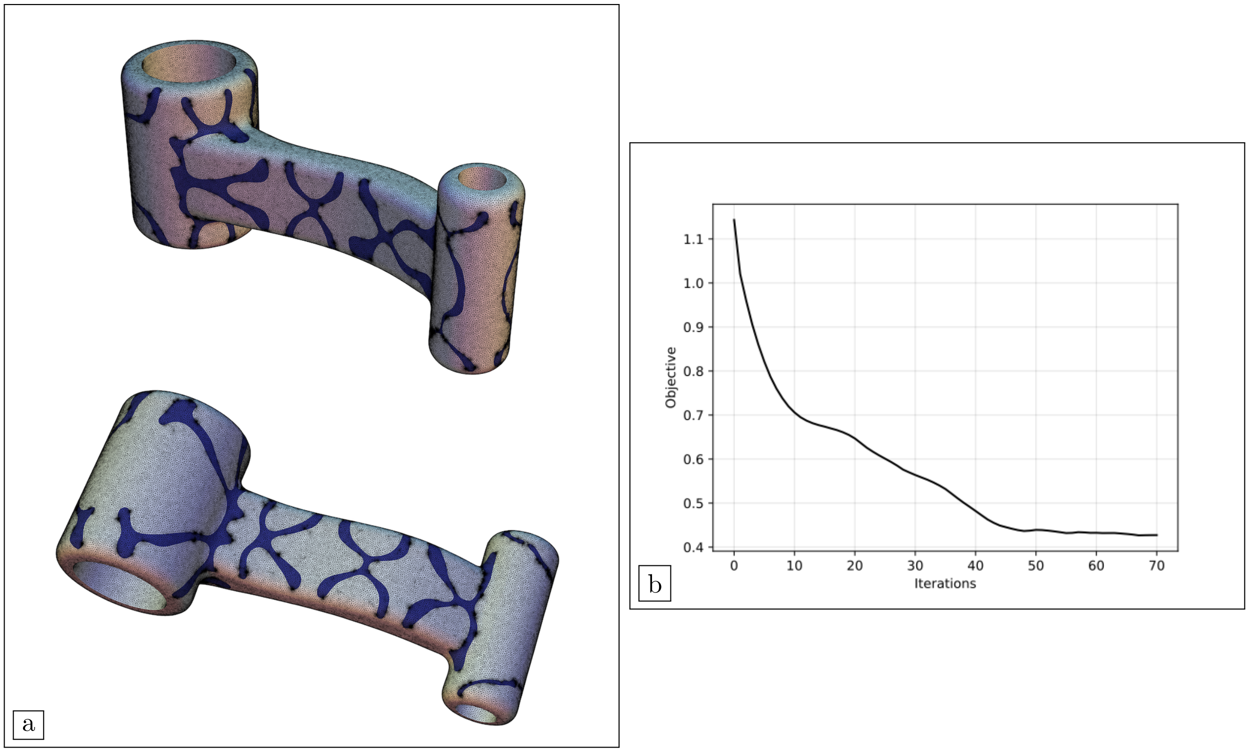

Figure 19. *(a) Optimized design ($n = 60$) in the boundary optimization process of the mechanical device considered in Section 4.4 with penalization parameter $\ell = 0.05$ and meshsize $h_{\max} = 0.05$; (b) Convergence history.*

[21] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.

[22] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro, *Variational problems and partial differential equations on implicit surfaces*, Journal of Computational Physics, 174 (2001), pp. 759–780.

[23] M. Bertalmio, G. Sapiro, and G. Randall, *Region tracking on level-sets methods*, IEEE transactions on Medical Imaging, 18 (1999), pp. 448–451.

[24] E. Bonnetier, C. Dapogny, and M. S. Vogelius, *Small perturbations in the type of boundary conditions for an elliptic operator*, Journal de Mathématiques Pures et Appliquées, (2022).

[25] B. Bourdin, G. A. Francfort, and J.-J. Marigo, *The variational approach to fracture*, Journal of elasticity, 91 (2008), pp. 5–148.

[26] K. Brakke, *The motion of a surface by its mean curvature*, Princeton University.

[27] L. Bronsard and R. V. Kohn, *Motion by mean curvature as the singular limit of ginzburg-landau dynamics*, Journal of differential equations, 90 (1991), pp. 211–237.

[28] C. Bui, C. Dapogny, and P. Frey, *An accurate anisotropic adaptation method for solving the level set advection equation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 899–922.

[29] M. Burger and S. J. Osher, *A survey on level set methods for inverse problems and optimal design*, European journal of applied mathematics, 16 (2005), pp. 263–301.

[30] M. Campen and L. Kobbelt, *Exact and robust (self-) intersections for polygonal meshes*, in Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 397–406.

[31] T. F. Chan and L. A. Vese, *Active contours without edges*, IEEE Transactions on image processing, 10 (2001), pp. 266–277.

[32] X. Chen, *Generation and propagation of interfaces for reaction-diffusion equations*, Journal of Differential equations, 96 (1992), pp. 116–141.

[33] N. Cheney, J. Gould, and W. Catchpole, *The influence of fuel, weather and fire shape variables on fire-spread in grasslands*, International Journal of Wildland Fire, 3 (1993), pp. 31–44.

[34] L.-T. Cheng, P. Burchard, B. Merriman, and S. Osher, *Motion of curves constrained on surfaces using a level-set approach*, Journal of Computational Physics, 175 (2002), pp. 604–644.

[35] D. L. Chopp and J. A. Sethian, *Flow under curvature: singularity formation, minimal surfaces, and geodesics*, Experimental Mathematics, 2 (1993), pp. 235–255.

[36] A. N. Christiansen, J. A. Bærentzen, M. Nobel-Jorgensen, N. Aage, and O. Sigmund, *Combined shape and topology optimization of 3d structures*, Computers & Graphics, 46 (2015), pp. 25–35.

[37] A. N. Christiansen, M. Nobel-Jorgensen, N. Aage, O. Sigmund, and J. A. Bærentzen, *Topology optimization using an explicit interface representation*, Structural and Multidisciplinary Optimization, 49 (2014), pp. 387–399.

[38] T. J. Chung et al., *Computational fluid dynamics*, Cambridge university press, 2002.

[39] P. G. Ciarlet, *The finite element method for elliptic problems*, vol. 40, Siam, 2002.

[40] G.-H. Cottet and E. Maitre, *A level set method for fluid-structure interactions with immersed surfaces*, Mathematical models and methods in applied sciences, 16 (2006), pp. 415–438.

[41] M. G. Crandall, H. Ishii, and P.-L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bulletin of the American mathematical society, 27 (1992), pp. 1–67.

[42] C. Dapogny, C. Dobrzynski, and P. Frey, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, Journal of computational physics, 262 (2014), pp. 358–378.

[43] C. Dapogny and F. Feppon, *Shape optimization using a level set based mesh evolution method: an overview and tutorial*, Comptes Rendus. Mathématique, 361 (2023), pp. 1267–1332.

[44] C. Dapogny and P. Frey, *Computation of the signed distance function to a discrete contour on adapted triangulation*, Calcolo, 49 (2012), pp. 193–219.

[45] C. Dapogny, N. Lebbe, and E. Oudet, *Optimization of the shape of regions supporting boundary conditions*, Numerische Mathematik, 146 (2020), pp. 51–104.

[46] P. De Mottoni and M. Schatzman, *Geometrical evolution of developed interfaces*, Transactions of the American Mathematical Society, 347 (1995), pp. 1533–1589.

[47] M. C. Delfour and J.-P. Zolésio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, SIAM, 2011.

[48] Y. Deng, W. Zhang, Z. Liu, J. Zhu, and J. G. Korvink, *Topology optimization for surface flows*, Journal of Computational Physics, 467 (2022), p. 111415.

[49] D. A. Di Pietro and A. Ern, *Mathematical aspects of discontinuous Galerkin methods*, vol. 69, Springer Science & Business Media, 2011.

[50] C. Dobrzynski and P. Frey, *Anisotropic delaunay mesh adaptation for unsteady simulations*, in Proceedings of the 17th international Meshing Roundtable, Springer, 2008, pp. 177–194.

[51] A. Doi and A. Koide, *An efficient method of triangulating equi-valued surfaces by using tetrahedral cells*, IEICE TRANSACTIONS on Information and Systems, 74 (1991), pp. 214–224.

[52] Q. Du and X. Feng, *The phase field method for geometric moving interfaces and their numerical approximations*, Handbook of numerical analysis, 21 (2020), pp. 425–508.

[53] L. J. Durlofsky, B. Engquist, and S. Osher, *Triangle based adaptive stencils for the solution of hyperbolic conservation laws*, Journal of Computational Physics, 98 (1992), pp. 64–73.

[54] G. Dziuk and C. M. Elliott, *Finite elements on evolving surfaces*, IMA journal of numerical analysis, 27 (2007), pp. 262–292.

[55] ———, *Finite element methods for surface pdes*, Acta Numerica, 22 (2013), pp. 289–396.

[56] K. Ecker, *Regularity theory for mean curvature flow*, vol. 57, Springer Science & Business Media, 2012.

[57] F. Feppon, G. Allaire, C. Dapogny, and P. Jolivet, *Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing*, Journal of Computational Physics, (2020), p. 109574.

[58] J.-B. Filippi, F. Bosseur, and D. Grandi, *Forefire: open-source code for wildland fire spread models*, Advances in forest fire research'.(Ed. DX Viegas)(Imprensa da Universidade de Coimbra: Coimbra, Portugal), (2014).

[59] J.-B. Filippi, X. Pialat, and C. B. Clements, *Assessment of forefire/meso-nh for wildland fire/atmosphere coupled simulation of the fireflux experiment*, Proceedings of the Combustion Institute, 34 (2013), pp. 2633–2640.

[60] M. A. Finney, *Calculation of fire spread rates across random landscapes*, International Journal of Wildland Fire, 12 (2003), pp. 167–174.

[61] P. J. Frey and P.-L. George, *Mesh generation: application to finite elements*, ISTE, 2007.

[62] T.-P. Fries, S. Omerović, D. Schöllhammer, and J. Steidl, *Higher-order meshing of implicit geometries—part i: Integration and interpolation in cut elements*, Computer Methods in Applied Mechanics and Engineering, 313 (2017), pp. 759–784.

[63] T.-P. Fries and D. Schöllhammer, *Higher-order meshing of implicit geometries, part ii: Approximations on manifolds*, Computer Methods in Applied Mechanics and Engineering, 326 (2017), pp. 270–297.

[64] Y. Giga, *Surface evolution equations*, Springer, 2006.

[65] J. B. Greer, *An improvement of a recent eulerian method for solving pdes on general geometries*, Journal of Scientific Computing, 29 (2006), pp. 321–352.

[66] P. Grisvard, *Elliptic problems in nonsmooth domains*, SIAM, 2011.

[67] A. Henrot and M. Pierre, *Shape Variation and Optimization*, EMS Tracts in Mathematics Vol. 28, 2018.

[68] T. Ilmanen, *Generalized flow of sets by mean curvature on a manifold*, Indiana University mathematics journal, (1992), pp. 671–705.

[69] G.-S. Jiang and D. Peng, *Weighted eno schemes for hamilton–jacobi equations*, SIAM Journal on Scientific computing, 21 (2000), pp. 2126–2143.

[70] R. Kimmel, *Intrinsic scale space for images on surfaces: The geodesic curvature flow*, Graphical models and image processing, 59 (1997), pp. 365–372.

[71] R. Kimmel and J. A. Sethian, *Computing geodesic paths on manifolds*, Proceedings of the national academy of Sciences, 95 (1998), pp. 8431–8435.

[72] H. C. Ko, G. Shin, S. Wang, M. P. Stoykovich, J. W. Lee, D.-H. Kim, J. S. Ha, Y. Huang, K.-C. Hwang, and J. A. Rogers, *Curvilinear electronics formed using silicon membrane circuits and elastomeric transfer elements*, small, 5 (2009), pp. 2703–2709.

[73] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, ACM siggraph computer graphics, 21 (1987), pp. 163–169.

[74] C. B. Macdonald and S. J. Ruuth, *Level set equations on surfaces via the closest point method*, Journal of Scientific Computing, 35 (2008), pp. 219–240.

[75] ———, *The implicit closest point method for the numerical solution of partial differential equations on surfaces*, SIAM Journal on Scientific Computing, 31 (2010), pp. 4330–4350.

[76] J. Margerit and O. Séro-Guillaume, *Modelling forest fires. part ii: reduction to two-dimensional models and simulation of propagation*, International Journal of Heat and Mass Transfer, 45 (2002), pp. 1723–1737.

[77] M. K. Misztal and J. A. Bærentzen, *Topology-adaptive interface tracking using the deformable simplicial complex*, ACM Transactions on Graphics (TOG), 31 (2012), pp. 1–12.

[78] N. Moes, J.-F. Remacle, J. Lambrechts, B. Le, and N. Chevaugeon, *The extreme mesh deformation approach (x-mesh) for the stefan phase change model*, Journal of Computational Physics, (2023), p. 111878.

[79] B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Oxford university press, 2010.

[80] D. Morvan, G. Accary, S. Meradji, and N. Frangieh, *Fifty years of progress in wildland fire modelling: from empirical to fully physical cfd models*, Comptes Rendus. Mécanique, 350 (2022), pp. 1–9.

[81] H. Munthe-Kaas, *High order runge-kutta methods on manifolds*, Applied Numerical Mathematics, 29 (1999), pp. 115–127.

[82] F. Murat and J. Simon, *Sur le contrôle par un domaine géométrique*, Pré-publication du Laboratoire d'Analyse Numérique,(76015), (1976).

[83] T. G. Myers, J. P. Charpin, and S. J. Chapman, *The flow and solidification of a thin fluid film on an arbitrary three-dimensional surface*, Physics of Fluids, 14 (2002), pp. 2788–2803.

[84] C. F. Ollivier-Gooch, *Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction*, Journal of Computational Physics, 133 (1997), pp. 6–17.

[85] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Science & Business Media, 2006.

[86] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.

[87] O. Pironneau, *Finite element methods for fluids*, Wiley Chichester, 1989.

[88] R. C. Rothermel, *A mathematical model for predicting fire spread in wildland fuels*, vol. 115, Intermountain Forest & Range Experiment Station, Forest Service, US ..., 1972.

[89] S. J. Ruuth and B. Merriman, *A simple embedding method for solving partial differential equations on surfaces*, Journal of Computational Physics, 227 (2008), pp. 1943–1961.

[90] G. Sapiro, *Geometric partial differential equations and image analysis*, Cambridge university press, 2006.

[91] O. Séro-Guillaume and J. Margerit, *Modelling forest fires. part i: a complete set of equations derived by extended irreversible thermodynamics*, International Journal of Heat and Mass Transfer, 45 (2002), pp. 1705–1722.

[92] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts.*, Proceedings of the National Academy of Sciences, 93 (1996), pp. 1591–1595.

[93] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cambridge university press, 1999.

[94] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of computational physics, 77 (1988), pp. 439–471.

[95] J. Sokolowski and J.-P. Zolésio, *Introduction to shape optimization*, Springer, 1992.

[96] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 469–491.

[97] J. Strain, *Semi-lagrangian methods for level set equations*, Journal of Computational Physics, 151 (1999), pp. 498–533.

[98] A. L. Sullivan, *Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models*, International Journal of Wildland Fire, 18 (2009), pp. 369–386.

[99] P. Tang, F. Qiu, H. Zhang, and Y. Yang, *Phase separation patterns for diblock copolymers on spherical surfaces: A finite volume method*, Physical Review E, 72 (2005), p. 016710.

[100] L. Tian, C. B. Macdonald, and S. J. Ruuth, *Segmentation on surfaces with the closest point method*, in 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 3009–3012.

[101] E. A. Träff, O. Sigmund, and N. Aage, *Topology optimization of ultra high resolution shell structures*, Thin-Walled Structures, 160 (2021), p. 107349.

[102] G. Turk, *Generating textures on arbitrary surfaces using reaction-diffusion*, Acm Siggraph Computer Graphics, 25 (1991), pp. 289–298.

[103] A. Witkin and M. Kass, *Reaction-diffusion textures*, in Proceedings of the 18th annual conference on computer graphics and interactive techniques, 1991, pp. 299–308.

[104] Q. Xia, M. Y. Wang, and T. Shi, *A level set method for shape and topology optimization of both structure and support of continuum structures*, Computer Methods in Applied Mechanics and Engineering, 272 (2014), pp. 340–353.

[105] Q. Ye, Y. Guo, S. Chen, N. Lei, and X. D. Gu, *Topology optimization of conformal structures on manifolds using extended level set methods (x-lsm) and conformal geometry theory*, Computer Methods in Applied Mechanics and Engineering, 344 (2019), pp. 164–185.

[106] A. Zaharescu, E. Boyer, and R. Horaud, *Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2010), pp. 823–837.

[107] Y.-T. Zhang and C.-W. Shu, *Eno and weno schemes*, in Handbook of numerical analysis, vol. 17, Elsevier, 2016, pp. 103–122.

[108] H. Zhao, *A fast sweeping method for eikonal equations*, Mathematics of computation, 74 (2005), pp. 603–627.

[109] H.-K. Zhao, S. Osher, and R. Fedkiw, *Fast surface reconstruction using the level set method*, in Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on, IEEE, 2001, pp. 194–201.