Mathematical Modelling and Numerical Analysis Will be set by the publisher
Modélisation Mathématique et Analyse Numérique

GEOMETRIC CONSTRAINTS IN DESCENT METHODS FOR SHAPE OPTIMISATION

Daniel Kraft[1]

Abstract. Many shape-optimisation problems arising from practical applications feature geometric constraints. We are particularly interested in the situation that certain regions of a hold-all domain should always or never be part of the optimal shape. This can be used, for instance, to model design constraints or to include additional information into the optimisation process. In the context of descent methods, there are two fundamental ways to account for constraints by means of projections: One can project the descent direction before taking a step to stay feasible, or one can project the resulting point back to the feasible region after taking a step. The latter is usually called projected-gradient method and is more commonly used. For shape optimisation, both approaches create additional difficulties that are not present in the classical context of optimisation in vector spaces. In this paper, we analyse these issues. We are able to show that certain conditions ensure that both approaches behave in a very similar way, although one or the other can be better suited to special situations. Our theoretical results are confirmed by numerical experiments based on a Chan-Vese-like model for image segmentation.

2010 Mathematics Subject Classification. 49Q10, 65D18, 65J22, 65K10.

November 16th, 2015, revised April 7th, 2017.

The original publication is available at `https://www.esaim-m2an.org/`.

Introduction

Many applications for shape optimisation are based on a practical problem where the desired shape cannot be chosen entirely freely. In such situations, it may be necessary to impose *geometric constraints* on the admissible shapes. A problem with this characteristic has been tackled recently with the help of penalisation in [9]. In this paper, we want to discuss ways to enforce constraints directly, resulting in *feasible* gradient-descent methods. For this, let us consider *forbidden regions*, i. e., for some set B that is a subset of the hold-all domain D, we require that $\Omega \cap B = \emptyset$ for all considered shapes Ω. It is straight-forward to modify our results to constraints of the form $B \subset \Omega$ instead. See [8] for a practical problem that gives rise to both types of constraints in a natural way. Section 5 of [8] also briefly discusses methods to deal with the constraints, as does Section 2.13 of [7]. In this work, we want to expand this discussion and give more details, particularly about the underlying theoretical considerations.

The methods we discuss are based on gradient descents in the framework of a scalar speed method for shape optimisation. In order to handle the constraints and ensure feasibility of the descent iterates, we have to introduce *projections*. As we will see in more detail later in Section 2, there are two basic possibilities for that: After taking an unconstrained step, one can project the resulting (possibly infeasible) shape back onto the admissible set, leading to a projected-gradient method. Alternatively, one can also adapt the descent direction *before* evolving the current shape in such a way that the whole shape evolution based on this direction stays feasible. The latter approach is similar to the method suggested in [16], and it fits quite naturally to the speed method that we use. In contrast to any existing work we are aware of, however, we focus particularly on the geometrical analysis of the shape evolutions

Keywords and phrases: Shape Optimisation, Geometric Constraints, Gradient-Descent Method, Projected Gradient, Shape Projection, Image Segmentation

[1] University of Graz, Institute of Mathematics, NAWI Graz, Universitätsplatz 3, 8010 Graz, Austria; e-mail: d@domob.eu

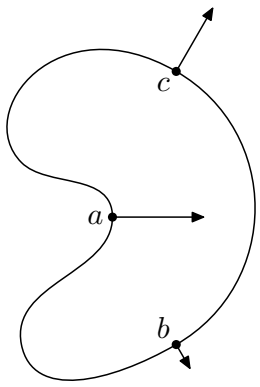


FIGURE 1. The scalar speed method in normal direction. In the shown situation, the signs relate as $F(a) < 0 < F(b) < F(c)$.

resulting from the methods we discuss. Let us also mention that our projection methods are not the only possible strategies for dealing with constraints. It may, for instance, be possible to develop methods based on a generalisation of Lagrange multipliers or other concepts. Such approaches are out of scope for the current paper, but they could be interesting topics for further research. A good overview of the general theory for constrained optimisation is given in Chapter 12 of [15].

The paper is structured as follows: We give a brief introduction into the basic optimisation framework we use as well as some other background material from the literature in Section 1. Our gradient-descent method including the suggested ways for handling constraints is discussed in Section 2. We then define and analyse a method for projecting shapes (instead of speed fields) in Section 3 and give a theoretical analysis of the relation between projections of the iterates and the speed fields in Section 4. It turns out that both methods behave in the same way for infinitesimally small step lengths if the forbidden geometry is smooth. Finally, we analyse our methods also with numerical experiments in Section 5. Note that our code implementing the speed method as well as some aspects of the methods described here (including the shape projections) has been released as free software in the `level-set` package [11] for GNU Octave [4].

1. Background

Before we can start to analyse our methods for handling geometric constraints, we first have to prepare the stage. In this section, we build up the necessary background for the later discussions by collecting various concepts and ideas from the literature. This is done only to the extent necessary for the remainder of this paper, and the reader is referred to the cited publications for more details. In particular, we introduce our framework for describing and evolving shapes in Subsection 1.1, and we briefly recall two well-known concepts for defining distances between shapes in Subsection 1.2. They will be used later to compare the geometries that result from different approaches for handling our geometric constraints.

1.1. Shape Evolutions According to a Speed Field

Of course, a suitable description for shapes *and changes to them* is an integral component in any scheme for shape optimisation. We use the level-set method introduced by Osher and Sethian in [17] to describe an open domain $\Omega \subset \mathbb{R}^n$ and its boundary $\Gamma = \partial\Omega$. More important for our purposes, however, is the description of *changes*: A particular "direction" for evolving an initial shape Ω_0 is encoded by means of a *scalar speed field* $F\colon \mathbb{R}^n \to \mathbb{R}$. For a point $x \in \Gamma_0$ on the boundary, the value $F(x)$ gives the speed with which the boundary should move in normal direction. Positive speed corresponds to a growing domain, while negative values lead to local shrinking. This is illustrated in Figure 1. Following this line of thought, we can define evolved shapes Ω_t and Γ_t for any time $t \geq 0$. Classically, this process is described by solving a level-set equation in the sense of viscosity solutions; let us refer to [5] for a thorough treatment.

In the present work, however, we will not do this. Instead, let us follow the framework described in Chapter 3 of [12] and in [13]. For this, we always assume that the speed field F is Lipschitz continuous and has compact support. Consequently, it is also bounded. Let $|F| \leq \overline{F}$ be the case throughout $\mathbb{R}^n$. Furthermore, we will, without loss of generality, only consider non-negative speed fields below in this subsection. The cases of arbitrary signs can

be reduced to this situation following Theorem 2 in [13]. For convenience, let us quickly recall the most important foundations of the framework. The crucial ingredients are (shortest) paths and the *F-induced distance* to the initial geometry Ω_0:

Definition 1.1. Let $F\colon \mathbb{R}^n \to \mathbb{R}^+$ be a speed field as described above. For $x, y \in \mathbb{R}^n$ and a continuous path $\xi \in \mathrm{W}^{1,\infty}([0,1], \mathbb{R}^n)$ with $\xi(0) = x$ and $\xi(1) = y$, we define its length as

$$l(\xi) = \int_0^1 \frac{|\xi'(t)|}{F(\xi(t))}\, dt.$$

The set of all such paths is denoted by $X_{\mathrm{ad}}(x, y)$. Furthermore, we define the *F-induced distance* by

$$d(x, y) = \inf_{\xi \in X_{\mathrm{ad}}(x,y)} l(\xi).$$

The distance to the initial domain Ω_0 is then given by

$$d_0(x) = \inf_{y \in \Gamma_0 \cup \Omega_0} d(x, y).$$

Note that the distance $d(x, y)$ may be infinite if all paths connecting x and y have to cross through the set $\Omega^z = \{x \in \mathbb{R}^n \mid F(x) = 0\}$. Furthermore, a basic estimate using Lipschitz continuity of F implies that $l(\xi)$ blows up whenever a path ξ comes close to the boundary of $\Omega^+ = \{x \in \mathbb{R}^n \mid F(x) > 0\}$. See Lemma 5 in [13] for more details.

The distance d_0 can be related to a control problem, whose Hamilton-Jacobi-Bellman equation is precisely the classical level-set equation. In this way, it is possible to show (see Theorem 5 in [13]) that the following *Hopf-Lax representation formulas* can be used to characterise the evolving sets:

$$\begin{aligned} \Gamma_t \cup \Omega_t &= \{x \in \mathbb{R}^n \mid d_0(x) \le t\}, \\ \Omega_t &= \{x \in \mathbb{R}^n \mid d_0(x) < t\}, \\ \Gamma_t &= \{x \in \mathbb{R}^n \mid d_0(x) = t\} \cup (\Gamma_0 \cap \Omega^z) \end{aligned}$$

They hold for all $t > 0$. Let us emphasise, in particular, that the sets are stationary in time wherever $F = 0$. Inside of Ω^+, on the other hand, the set Ω_t grows monotonically with increasing t.

Finally, one can also use this framework for *shape-sensitivity analysis*: For a domain functional

$$J(\Omega_t) = \int_{\Omega_t} f(x, \Omega_t)\, dx, \tag{1}$$

the shape derivative in direction F is given by the boundary integral

$$J'(\Omega_t) = dJ(\Omega_t; F) = \int_{\Gamma_t} Ff\, d\sigma + \int_{\Omega_t} f'\, dx. \tag{2}$$

Here, f' is the shape derivative of the shape-dependent integrand f in direction F. This formula can be shown rigorously at least in a weak sense; see Subsection 5.2 of [13]. Note that the shape derivative is supported on the *boundary* Γ_t, following the well-known *Hadamard-Zolésio structure theorem* (see Theorem 3.6 on page 479 of [3]). Based on (2), we will develop a gradient-descent method for shape optimisation in Section 2.

1.2. Distance Measures between Shapes

Since we want to compare shapes resulting from different approaches later on, we have to introduce a notion of distance between geometries in the first place. Let us refer to [3] for more details, which defines a wealth of such distance measures and discusses their properties. For our purposes, we concentrate only on two particular concepts. The first is based on the symmetric set difference:

Definition 1.2. Let $A, B \subset \mathbb{R}^n$ be measurable. We define the *symmetric set difference* as

$$A \,\Delta\, B = (A \setminus B) \cup (B \setminus A)$$

and the *distance in measure* between A and B as the Lebesgue measure $\mathrm{vol}(A \,\Delta\, B)$ of this set.

The distance in measure corresponds to the L^1-distance between the characteristic functions of the sets, as discussed in Section 5.3 of [3]. It is straight-forward to define and will be quite simple to analyse. A more complicated but also very important classical concept is the *Hausdorff distance*, whose definition we also recall for convenience:

Definition 1.3. For $A, B \subset \mathbb{R}^n$, we define the *one-sided distance* as

$$d_H(A \to B) = \sup_{x \in A} \mathrm{dist}(x, B) = \sup_{x \in A} \inf_{y \in B} |x - y|.$$

With it, the *Hausdorff distance* is the symmetrised expression

$$d_H(A, B) = \max\left(d_H(A \to B),\, d_H(B \to A)\right).$$

Note that the additional symmetrisation step is really necessary, since $d_H(A \to B) \neq d_H(B \to A)$ in general. See Subsection 6.2.2 of [3] for more details about the Hausdorff distance. For open sets, it may be interesting to consider either the Hausdorff distance between the sets themselves or the distance between their boundaries—depending on whether the problem at hand is tied more closely to open domains or evolving fronts. In the following, however, we will only consider the distance $d_H(A, B)$ between the sets themselves.

2. Steepest Descent for Shape Optimisation

Let us now turn our attention to the implementation of a simple gradient-descent method for a cost of the form (1). For this, we would like to use the (negative) derivative (2) to define a steepest-descent direction. Note, however, that the functional $dJ(\Omega; \cdot)$ is supported on the *boundary* Γ, while a speed field (corresponding to a direction) must be defined on the *domain* D. Thus, one needs a suitable method of extending the shape derivative from the boundary onto the hold-all domain. For a general discussion of this topic, let us refer to Chapter 6 of [12] and Section 3 of [14]. In the context of this work, we always apply the following idea: Note that the shape derivative (2) can be interpreted as a continuous linear functional operating on the speed field F, at least if we assume that the shape derivative f' of the integrand is of the same form. Thus, if we choose some Hilbert space $\mathcal{H}$ for the speed fields, we can find the *Riesz representative* of the shape derivative and use it as speed field. See [1] for a general functional-analytic treatment of various spaces in this context, and [8] for a numerical comparison of a few selected spaces. For our purposes, the choice $\mathcal{H} = \mathrm{H}^1(D)$ seems to perform quite well in practice. Thus, for a given current shape, we first solve the variational problem

$$\int_D \left(GH + \beta \left\langle \nabla G, \nabla H \right\rangle\right) dx = dJ(\Omega; H), \tag{3}$$

which must be satisfied for all $H \in \mathcal{H} = \mathrm{H}^1(D)$. The solution $G \in \mathrm{H}^1(D)$ is the Riesz representative of the shape derivative, which we call *shape gradient*. The value $\beta > 0$ in (3) is a parameter, which can be used to control the smoothing properties of the inverse Laplacian appearing in the solution of (3). In practice, it is often beneficial to choose $\beta \ll 1$. See, for instance, [10] for a numerical demonstration. Then, $-G$ is a valid speed field that is, in some sense, a steepest-descent direction. We can evolve the current domain Ω into this direction based on the method discussed above in Subsection 1.1. The evolution time t can be chosen according to some line-search algorithm, for instance, the Armijo rule (see Section 3.1 of [15]). Putting all of these ideas together, we arrive at a basic gradient-descent algorithm for shape optimisation of (1).

Based on this foundation, we are now mainly interested in the question of *how to incorporate geometric constraints into this method*. In the face of constraints, there are, in general, two basic projection strategies to ensure that the gradient descent stays feasible: One can either modify the shape after each step to satisfy the constraints, or one can adapt the speed field before a step to ensure that it only leads to feasible evolved shapes. The former approach is usually called *projected-gradient method* (see, for instance, Algorithm 2.3 on page 105 of [6]). In the context of shapes, however, there is no obvious way to "project" them to ensure feasibility. (It is, nevertheless, possible to define a "reasonable" shape projection. This will be done in Section 3.) Thus, it also makes sense to consider the second idea: Based on the speed method presented in Subsection 1.1, it is clear that requiring $F = 0$ on a forbidden region B for the speed field F ensures that $B \cap \Omega_t$ is stationary throughout the time evolution. In other words, feasible initial shapes always lead to evolved shapes that also satisfy the constraints. A similar idea of projecting the speed field was already used in [16], although for different types of shape constraints and without any deeper analysis. In our case, we want to ensure that the speed field vanishes on the forbidden region. To do so, there are, again, two possible strategies we want to discuss: The most natural way is to project the speed field in the same Hilbert space $\mathcal{H}$ that is also used for the computation of the shape gradient with (3). This is done in [8], based, in particular, on Theorem 3 there. Projection in $\mathcal{H}$ ensures that the constraints are incorporated into the descent method in a consistent way. It also has the feature that the speed field *gets drawn to zero continuously* towards the forbidden region. While this leads to better regularity properties, it also has a potential drawback: Due to the blow up of path lengths near the boundary of Ω^+ discussed in Subsection 1.1, this enforces small movements of the boundary close to the forbidden region. In particular, the boundary will never fully reach B, leading to a kind of "interior-point method". Once the boundary gets close to B, it also takes a relatively long time to get away again. Thus, let us also consider cutting off the speed in a way that avoids these potential issues:

$$F^p(x) = \begin{cases} 0 & \text{if } x \in B \text{ and } F(x) > 0, \\ F(x) & \text{else} \end{cases} \tag{4}$$

The speed field F^p can be seen as projection in L^2 instead of the Hilbert space $\mathcal{H}$ with higher regularity. The particular form of (4), however, also ensures that we force the speed to zero *only when growing towards* B. On the other hand, a negative speed is allowed inside of B. This ensures that the evolving boundary of Ω is able to get away

again after touching B. This projection is implemented (for more general shape constraints) as `ls_enforce_speed` in [11].

Furthermore, one can also apply the following strategy: Instead of minimising $J(\Omega)$ with constraints, one can optimise the reduced cost $J^p(\Omega) = J(\pi(\Omega))$ without constraints. Here, $\pi(\Omega)$ is a suitable projection, such as the one introduced in Section 3 below. For a projection as simple as (5), it is actually easy to compute the shape derivative of J^p and use it to implement an unconstrained descent method. In some sense, this method can be interpreted as a combination of the other projection methods: We project already the *shape derivative* before computing the speed field by solving (3), allow infeasible shapes for the evolution steps, and project the domains whenever we actually need to evaluate the cost. Similar to a speed projection in $\mathcal{H}$, this leads to a descent algorithm that is consistent with respect to the constraints.

To summarise, we have identified four methods for handling shape constraints in our descent method:

- Projection of the speed in $\mathcal{H}$, resulting in a continuous speed field,
- projection of the speed in L^2 according to (4), resulting in a *discontinuous* speed field F^p,
- projection of the shape after evolving it, and
- projection of the shape derivative in combination with possibly infeasible descent steps.

These methods are formulated more explicitly in Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. We will see in Section 4 and, in particular, Theorem 4.4, that the second and third strategy lead to very similar results under certain regularity assumptions on B. Section 5 compares all four methods to each other in numerical experiments. Let us, however, stress once again that only Algorithm 1 and Algorithm 4 are consistent with respect to the gradient descent. Only those ensure that we end up with a descent direction at every step of the algorithm. The other methods tamper with the speed field and geometry in a way that invalidates the directional derivative computed from (3), which may lead to problems with the line search and descent in general. This effect will be seen in the numerics, but we will also see that all methods have their justification in certain circumstances.

Algorithm 1 Descent step with projection of the speed field in $\mathcal{H}$.

Require: Ω_0 is the current, feasible shape
Ensure: Ω_1 is the update shape and also feasible
1: compute the shape derivative (linear functional) $dJ(\Omega_0;\cdot)$
2: $G \leftarrow$ solution of (3) in the space $\mathrm{H}^1(D) \cap \mathrm{H}^1_0(\mathbb{R}^n \setminus B)$ ▷ This ensures that G vanishes on ∂B.
3: $F \leftarrow -G$, extend $F = 0$ on B ▷ F is the steepest-descent speed field.
4: evolve Ω_0 in direction F to get Ω_1 ▷ See Subsection 1.1.

Algorithm 2 Descent step with discontinuous projection of the speed field in L^2.

Require: Ω_0 is the current, feasible shape
Ensure: Ω_1 is the update shape and also feasible
1: compute the shape derivative (linear functional) $dJ(\Omega_0;\cdot)$
2: $G \leftarrow$ solution of (3) in $\mathrm{H}^1(D)$
3: $F \leftarrow -G$ and F^p is defined according to (4) ▷ F will be continuous, F^p in general not.
4: evolve Ω_0 in direction F^p to get Ω_1

Algorithm 3 Descent step with projection of the shape. This method corresponds to the classical projected-gradient method.

Require: Ω_0 is the current, feasible shape
Ensure: Ω_1 is the update shape and also feasible
1: compute the shape derivative (linear functional) $dJ(\Omega_0;\cdot)$
2: $G \leftarrow$ solution of (3) in $\mathrm{H}^1(D)$
3: $F \leftarrow -G$
4: evolve Ω_0 in direction F to get Ω ▷ Ω may be infeasible at this point.
5: $\Omega_1 \leftarrow \pi(\Omega)$ ▷ See Section 3 below for $\pi(\cdot)$.

Let us also point out that since speed fields F^p obtained through (4) are not continuous, our formalism for shape evolutions described in Subsection 1.1 is not fully justified for them. There are, however, ways to extend the framework to support these cases, but this is beyond the scope of the current paper and not related to shape constraints. For the practical examples that we considered, this did not appear to be an issue.

Algorithm 4 Infeasible descent step with projection of the shape derivative.

Require: Ω_0 is the current, possibly infeasible shape
Ensure: Ω_1 is the update shape, $\pi(\Omega_1)$ is feasible
1: compute the shape derivative (linear functional) of $J^p(\Omega) = J(\pi(\Omega))$ for $\Omega = \Omega_0$
2: $G \leftarrow$ solution of (3) in $\mathrm{H}^1(D)$
3: $F \leftarrow -G$
4: evolve Ω_0 in direction F to get Ω_1 ▷ Project the shape when evaluating J^p for a line search.

3. Shape Projections

Assume now that $\Omega \subset D$ is some open domain, but that the constraint $\Omega \cap B = \emptyset$ is not fulfilled. The straightforward way to fix this is, of course, to define the *shape projection*

$$\pi(\Omega) = \Omega \setminus B. \tag{5}$$

If we assume B to be closed, then $\pi(\Omega)$ is again an open domain. However, the usual, more systematic approach to define projections (in general) is to consider a minimisation of the form

$$\min_{\Omega'} d^s\left(\Omega, \Omega'\right), \tag{6}$$

where $d^s(\cdot,\cdot)$ is some kind of distance measure and Ω' runs through all admissible shapes, i. e., all open sets $\Omega' \subset D$ with $\Omega' \cap B = \emptyset$. Let us consider (6) now with respect to both metrics for shapes introduced in Subsection 1.2. The distance in measure (recall Definition 1.2) is easy to do:

Lemma 3.1. *The set $\pi(\Omega)$ from (5) is, up to changes of measure zero, the unique minimiser of (6) if $d^s(\cdot,\cdot)$ is the distance in measure. Consequently, it is also the unique minimiser among open sets.*

Proof. Let Ω' be feasible, i. e., $\Omega' \cap B = \emptyset$. Then necessarily

$$\Omega \cap B \subset \Omega \setminus \Omega' \subset \Omega \,\Delta\, \Omega',$$

so that also

$$\mathrm{vol}\left(\Omega \,\Delta\, \Omega'\right) \geq \mathrm{vol}\left(\Omega \cap B\right) = \mathrm{vol}\left(\Omega \,\Delta\, \pi(\Omega)\right).$$

Hence, $\pi(\Omega)$ is indeed a minimiser of (6).

To show uniqueness, assume now that Ω' is also a minimiser of (6), which means

$$\mathrm{vol}\left(\Omega \,\Delta\, \Omega'\right) = \mathrm{vol}\left(\Omega \cap B\right) = \mathrm{vol}\left(\Omega \,\Delta\, \pi(\Omega)\right). \tag{7}$$

Note that the disjoint decomposition

$$\Omega \,\Delta\, \Omega' = (\Omega \cap B) \cup (\Omega' \setminus \Omega) \cup (\Omega \setminus (\Omega' \cup B))$$

holds. Taking the measure and observing (7) implies

$$\mathrm{vol}\left(\Omega' \setminus \Omega\right) = \mathrm{vol}\left(\Omega \setminus (\Omega' \cup B)\right) = 0.$$

Since by (5) also

$$\mathrm{vol}\left(\Omega' \,\Delta\, \pi(\Omega)\right) = \mathrm{vol}\left(\Omega' \setminus (\Omega \setminus B)\right) + \mathrm{vol}\left((\Omega \setminus B) \setminus \Omega'\right) = \mathrm{vol}\left(\Omega' \setminus \Omega\right) + \mathrm{vol}\left(\Omega \setminus (\Omega' \cup B)\right)$$

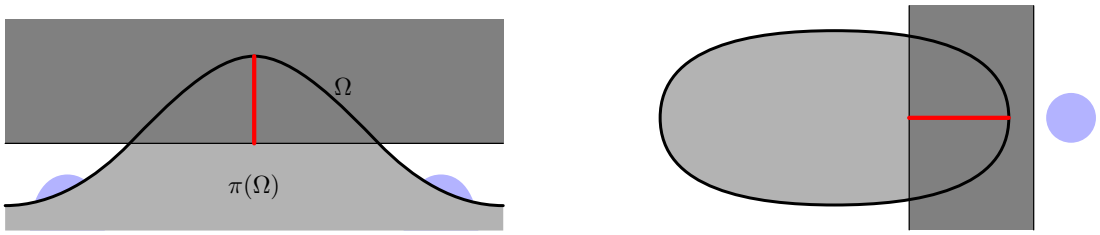
holds, this implies $\mathrm{vol}\left(\Omega' \,\Delta\, \pi(\Omega)\right) = 0$ and thus we get uniqueness up to changes of measure zero. □

Let us now consider the Hausdorff distance (see Definition 1.3) in (6) instead. This makes things more complicated. First, we can immediately observe that uniqueness does not hold in general for a minimiser of (6) with respect to the Hausdorff distance. It is possible to add arbitrary disturbances to any minimiser, as long as they are smaller in amplitude than the Hausdorff distance to Ω. This is illustrated with the blue parts in Figure 2a. Second, the minimiser of (6) may not be a subset of Ω. Adding the blue part in Figure 2b actually *reduces* the Hausdorff distance with respect to using $\pi(\Omega)$ alone (the light grey area). However, it makes, of course, sense to require that the projected shape should be a subset of Ω if we only want to *forbid* certain regions. Thus, let us make the following assumption:

Assumption 3.2. Let B be bounded. Since ∂B is compact, for arbitrary $x \in \Omega \cap B$ there exists $y_x \in \partial B$ as minimiser of the distance, i. e., such that

$$|x - y_x| = \mathrm{dist}\left(x, \partial B\right) = \inf_{y \in \partial B} |x - y|. \tag{8}$$

We assume that $y_x \in \Omega$ for all $x \in \Omega \cap B$.



(A) (B)

FIGURE 2. Shape projection according to (6) with $d^s(\cdot,\cdot)$ chosen as the Hausdorff distance. The dark strip is the forbidden region B, the light grey area the projected shape $\pi(\Omega)$ of (5). The red line indicates $d_H(\Omega \to \pi(\Omega))$.

This assumption excludes a situation like Figure 2b. On the other hand, it is usually satisfied if Ω is produced from some feasible initial shape Ω_0 by slightly moving its boundary. It allows us to show that $\pi(\Omega)$ is also a minimiser of (6) for the Hausdorff distance:

Lemma 3.3. *Let Assumption 3.2 hold or restrict the considered Ω' in (6) to subsets of Ω. Assume that $\Omega \setminus B$ is not empty. Then $\pi(\Omega)$ is a minimiser of (6) with respect to the Hausdorff distance $d_H(\cdot,\cdot)$.*

Proof. If $\Omega \cap B = \emptyset$ already, then $\pi(\Omega) = \Omega$ and $d_H(\Omega, \pi(\Omega)) = 0$. In this case, the result is clear. Thus assume $\Omega \cap B \neq \emptyset$. Since $\pi(\Omega) \subset \Omega$, we know that $d_H(\pi(\Omega) \to \Omega) = 0$. On the other hand,

$$d_H(\Omega, \pi(\Omega)) = d_H(\Omega \to \pi(\Omega)) = \sup_{x\in\Omega} \inf_{y\in\Omega\setminus B} |x-y| = \sup_{x\in\Omega\cap B} \inf_{y\in\Omega\setminus B} |x-y|.$$

Now assume that Ω' is admissible in (6), i. e., $\Omega' \cap B = \emptyset$. Again, we get

$$d_H(\Omega, \Omega') \geq d_H(\Omega \to \Omega') = \sup_{x\in\Omega} \inf_{y\in\Omega'} |x-y| \geq \sup_{x\in\Omega\cap B} \inf_{y\in\Omega'} |x-y|.$$

If now $\Omega' \subset \Omega$, then also $\Omega' \subset \Omega \setminus B$. Thus

$$d_H(\Omega, \Omega') \geq \sup_{x\in\Omega\cap B} \inf_{y\in\Omega'} |x-y| \geq \sup_{x\in\Omega\cap B} \inf_{y\in\Omega\setminus B} |x-y| = d_H(\Omega, \pi(\Omega)).$$

This finishes the proof for this case. So let Assumption 3.2 hold and note that

$$\sup_{x\in\Omega\cap B} \inf_{y\in\Omega'} |x-y| \geq \sup_{x\in\Omega\cap B} \min_{y\in\partial B} |x-y|$$

must be true since $\Omega' \subset \mathbb{R}^n \setminus B$. Let $x \in \Omega \cap B$ be arbitrary. By the assumption, there exists $y_x \in \partial B$ such that (8) holds. Since $y_x \in \Omega$ and Ω is open, there exists a sequence $(y_k)_{k\in\mathbb{N}} \subset \Omega \setminus B$ converging to y_x as $k \to \infty$. This sequence also realises the minimum in the limit since the Euclidean distance is continuous. Hence

$$\min_{y\in\partial B} |x-y| = |x-y_x| = \lim_{k\to\infty} |x-y_k| \geq \inf_{y\in\Omega\setminus B} |x-y|.$$

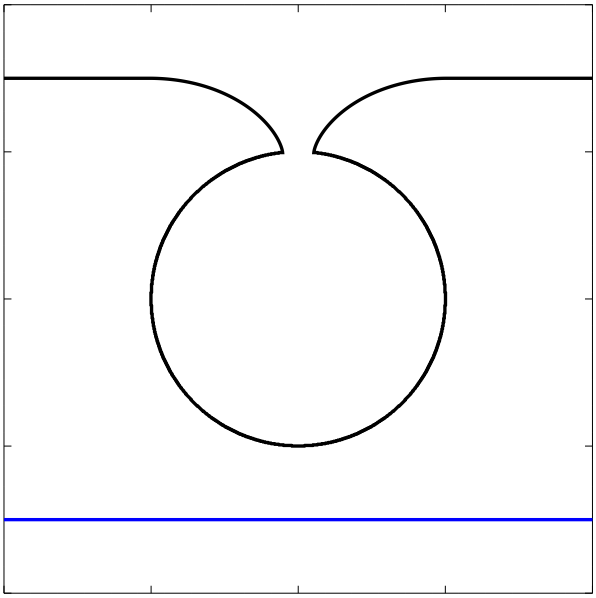
This inequality remains intact if we take the supremum over all $x \in \Omega \cap B$, yielding the desired

$$d_H(\Omega, \Omega') \geq \sup_{x\in\Omega\cap B} \min_{y\in\partial B} |x-y| \geq \sup_{x\in\Omega\cap B} \inf_{y\in\Omega\setminus B} |x-y| = d_H(\Omega, \pi(\Omega)).$$

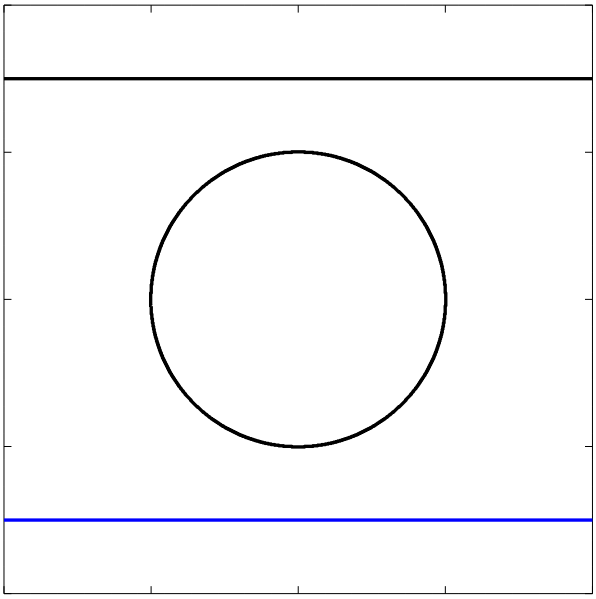
□

Initially, we defined the projection (5) mainly based on intuition. With Lemma 3.1 and Lemma 3.3, we are now able to justify it based on the abstract shape projection (6):

Corollary 3.4. *Let Assumption 3.2 hold and let $\Omega \setminus B \neq \emptyset$. Then the shape projection $\pi(\Omega)$ from (5) is the unique open set that minimises (6) for the distance in measure. It is, at the same time, a minimiser for the Hausdorff distance.*

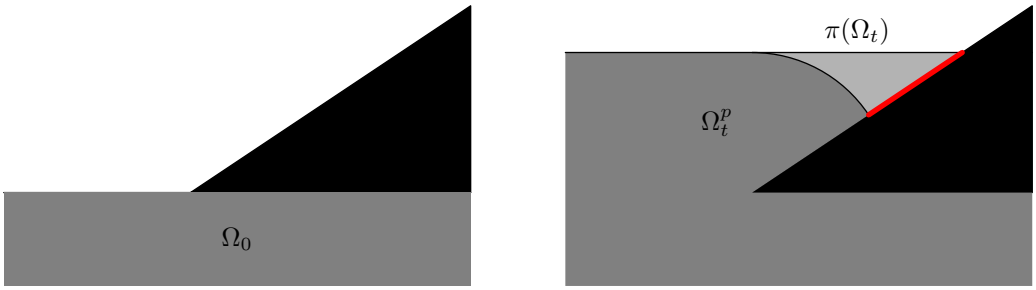


(A) Ω_t^p corresponding to a speed projection in L^2.



(B) Projected shape $\pi(\Omega_t)$.

FIGURE 3. Comparison of the evolved domains Ω_t^p and $\pi(\Omega_t)$ (in black). The initial geometry is the front shown in blue, which evolves with $F = 1$ around a circular forbidden region.



(A) The initial situation. (B) Evolution after some time t.

FIGURE 4. The shadowing effect for a corner. The forbidden region is black, the evolved shape Ω_t^p is dark, and the additional set part of $\pi(\Omega_t)$ is light grey. The red line indicates the Hausdorff distance between the two evolved sets.

4. The Relation between Shape and Speed Projections

Following up on Section 3, let us now compare the result of a shape projection with (5) to a projection of the speed field in L^2 according to (4). Both methods are in contrast to a speed projection in $\mathcal{H}$, since they allow the constraint to become fully active. As before, let us assume $F \geq 0$ here. This assumption is necessary for our analysis below, but it is also not a strong restriction with respect to the situation that is relevant in practice: Typically, one evolves a given current shape with a fixed speed field only for a relatively short time, namely for a single gradient-descent step. Thus, *locally* around the interesting points where the active set of the shape constraints changes, the speed field is either positive or negative. The former case is covered by our assumption of $F \geq 0$ and our analysis, and the latter case is (for a forbidden region) equivalent to just ignoring the shape constraints at all.

Let us further denote the evolved shape according to the projected speed F^p by Ω_t^p, where $t \geq 0$ is, as usual, the propagation time. Since $F^p \leq F$, it follows from the Hopf-Lax formula introduced in Subsection 1.1 that the inclusion $\Omega_t^p \subset \Omega_t$ holds. Noting that $\pi(\Omega_t)$ is the largest admissible subset of Ω_t, we can also conclude $\Omega_t^p \subset \pi(\Omega_t)$. This inclusion is, in general, strict: With a projected speed, the forbidden region creates a kind of "shadow" for the shape evolution. This effect is not present if the shape is allowed to propagate unhindered at first and only projected later according to (5). See Figure 3 for an illustration.

The worst-case situation for this shadowing effect is a sharp corner, as shown in Figure 4. The Hausdorff distance between Ω_t^p and $\pi(\Omega_t)$ for this situation is indicated with the red line in Figure 4b. A basic geometric argument quickly reveals that it is proportional to the evolution time t. If we exclude this case, however, and assume that the boundary of B is smooth, it turns out that both approaches yield approximately the same shapes for small step sizes, meaning that $d_H(\Omega_t^p, \pi(\Omega_t)) = o(t)$ for $t \to 0^+$. This will be our main result of the current section, formulated and shown below in Theorem 4.4. The kind of smoothness required is a well-known geometric condition, which we recall briefly for convenience (see also, for instance, Definition 3.1 on page 68 of [3]):

Definition 4.1. Let $\Omega \subset \mathbb{R}^n$ be open. Assume that for every point $p \in \partial\Omega$ there exist $r > 0$ and a bijection $h\colon Q \to B_r(p)$ such that both h and h^{-1} are in C^1. Furthermore, we require that

$$h(Q_0) = \partial\Omega \cap B_r(p) \quad \text{and} \quad h(Q_-) = \Omega \cap B_r(p).$$

Here, $Q = B_1(0)$ is the open unit ball and

$$Q_0 = \{x \in Q \mid x_n = 0\}, \quad Q_- = \{x \in Q \mid x_n < 0\}.$$

If this is the case, then Ω is said to be a C^1-*domain*.

The main ingredient for the estimation of $d_H(\Omega_t^p, \pi(\Omega_t))$ is the ability to construct paths between points $x, y \notin B$ that do not cross the forbidden region B and are not too long. This can be achieved with a C^1-domain as illustrated in Figure 5: If we "zoom in" far enough towards a piece of the boundary, it becomes almost flat due to the required smoothness. This makes it easy to construct a path $\xi \in X_{\mathrm{ad}}(x, y)$ that avoids B entirely. This idea can be formalised, which yields the main technical construction for the proof of Theorem 4.4:

Lemma 4.2. *Let B be compact and such that the interior B° is a C^1-domain. Then for every $\epsilon > 0$ there exists a neighbourhood U of ∂B and a $\delta > 0$ such that for every $x, y \in U$ with $|x - y| < \delta$ there is a path $\xi \in X_{\mathrm{ad}}(x, y)$ with*

$$|\xi| \leq (1 + \epsilon)\,|x - y|.$$

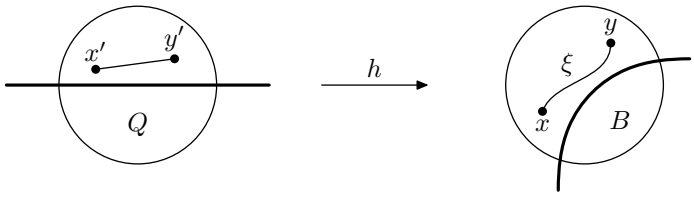


FIGURE 5. Sketch of the transformation h from Definition 4.1 mapping the reference domain Q (left) to a neighbourhood of the boundary of B (right). We also show a possible path connecting x and y without going through B.

If $x, y \notin B$, then also ξ is entirely outside of B. Here, $|\xi|$ denotes the arc length (not based on a speed field as in Definition 1.1 or alternatively based on the speed $F = 1$) of the path ξ.

Proof. Let us start with a standard compactness argument applied to ∂B: For every $x \in \partial B$, let r_x denote the radius of the neighbourhood $B_r(x)$ given by Definition 4.1. Then we can choose a finite subset $Z \subset \partial B$ such that

$$\partial B \subset \bigcup_{z \in Z} B_{r_z/3}(z) = U.$$

Choose $\delta_1 = \min_{z \in Z} r_z/3$. Now let $x, y \in U$ and $|x - y| < \delta_1$. Then there is some $z \in Z$ with $x \in B_{r_z/3}(z)$. Since $|z - y| \leq |z - x| + |x - y| < 2/3 \cdot r_z$, it follows that $x, y \in B_{2/3 r_z}(z)$. Also note the following: If we denote the bijection of Definition 4.1 for each $x \in \partial B$ by h_x, then all functions h_z, ${h_z}^{-1}$, Dh_z and $D{h_z}^{-1}$ are uniformly bounded and uniformly continuous on their respective domains for $z \in Z$. We can, furthermore, find bounds that are actually independent of z.

Consider now some fixed $z \in Z$ and $x, y \in B_{r_z}(z)$. Let us write $h = h_z$ for the bijection on $B_{r_z}(z)$. We define the path $\xi \in X_{\mathrm{ad}}(x, y)$ by

$$\xi(t) = (h \circ S_{x'y'})(t) = h(x' + t(y' - x')) = h(h^{-1}(x) + t(h^{-1}(y) - h^{-1}(x))),$$

i. e., we use a straight line in the transformed picture (on Q), as shown also in Figure 5. Obviously, if $x, y \notin B$, then also ξ will never be inside B. The fundamental theorem of calculus states that

$$y' - x' = h^{-1}(y) - h^{-1}(x) = \int_0^1 Dh^{-1}(x + s(y - x))\, ds.$$

(Note that $B_{r_z}(z)$ is convex, so that all intermediate values are in the domain of h^{-1}.) This implies also the following estimate for the derivative of ξ:

$$\begin{aligned}
\left|\xi'(t)\right| &= \left|Dh(x' + t(y' - x'))(y' - x')\right| = \left|\int_0^1 Dh(x' + t(y' - x'))Dh^{-1}(x + s(y - x))(y - x)\, ds\right| \\
&\leq |y - x| \int_0^1 \left\|Dh(x' + t(y' - x'))Dh^{-1}(x + s(y - x))\right\| ds \\
&= |y - x| \int_0^1 \left\|1 + Dh(x' + t(y' - x'))\left(Dh^{-1}(x + s(y - x)) - Dh^{-1}(h(x' + t(y' - x')))\right)\right\| ds
\end{aligned}$$

Here we have used the well-known relation between the derivatives of a function and its inverse,

$$\left(Dh(x' + t(y' - x'))\right)^{-1} = Dh^{-1}(h(x' + t(y' - x'))).$$

Further estimation yields

$$\left|\xi'(t)\right| \leq |y - x| \left(1 + \|Dh\|_\infty \int_0^1 \left\|Dh^{-1}(x + s(y - x)) - Dh^{-1}(h(x' + t(y' - x')))\right\| ds\right). \tag{9}$$

Also note that $|\xi|$ can be bounded by the same quantity, since $|\xi| = \int_0^1 |\xi'(t)|\, dt$ by definition.

Thus, it remains to find a uniform bound for the norms in the right-hand side of (9). The term $\|Dh_z\|_\infty$ can be bounded uniformly in z by compactness as discussed above. Furthermore, note that for every $\delta' > 0$ there is $\delta_2 > 0$ such that $|x - y| < \delta_2$ implies

$$\left|x + s(y - x) - h_z(x' + t(y' - x'))\right| < \delta'$$

for all $t, s \in [0,1]$, $z \in Z$ and $x, y \in B_{r_z}(z)$. To see this, note $h_z(x') = x$ and that h_z and ${h_z}^{-1}$ are uniformly continuous. Thus, uniform continuity of $D{h_z}^{-1}$ implies that we can have

$$\|Dh_z\|_\infty \int_0^1 \left\| D{h_z}^{-1}(x + s(y-x)) - D{h_z}^{-1}(h_z(x' + t(y'-x'))) \right\| ds \leq \epsilon$$

if only $|x-y| < \delta = \min(\delta_1, \delta_2)$. This finishes the proof. □

We can even get rid of the neighbourhood U in Lemma 4.2 with a simple further construction:

Lemma 4.3. *Lemma 4.2 holds also for all $x, y \in \mathbb{R}^n$ with $|x-y| < \delta$, dropping the condition $x, y \in U$.*

Proof. Choose δ for ϵ according to Lemma 4.2 and let $x, y \in \mathbb{R}^n$ be given with $|x-y| < \delta$. We consider the straight line S_{xy}. If it does not intersect B, then we can simply use $\xi = S_{xy}$ with $|\xi| = |x-y|$. So assume that $x, y \notin B$ but that S_{xy} intersects B at some point. Then there exist times $0 \leq t_1 < t_2 \leq 1$ such that $x_1, x_2 \in U \setminus B$, where we have introduced the notation $x_1 = S_{xy}(t_1)$ and $x_2 = S_{xy}(t_2)$. We can, in addition, choose t_1 and t_2 such that S_{xy} does neither intersect B on $[0, t_1]$ nor on $[t_2, 1]$. Apply now Lemma 4.2 to get a path $\xi \in X_{\mathrm{ad}}(x_1, x_2)$ that connects the two points without intersecting B. It is clear that this is possible since we have $|x_1 - x_2| \leq |x-y| < \delta$. Then set ξ_c to be the concatenation of S_{xy} on $[0, t_1]$, ξ and S_{xy} on $[t_2, 1]$. This yields

$$|\xi_c| = |x_1 - x| + |\xi| + |y - x_2| \leq |x_1 - x| + (1+\epsilon)|x_2 - x_1| + |y - x_2| \leq (1+\epsilon)|y - x|,$$

where we have used that the points x, x_1, x_2 and y are collinear. □

Based on Lemma 4.3, we can now show the main estimate of $d_H(\Omega_t^p, \pi(\Omega_t))$. For this, it mostly remains to handle the speed field F. Due to Lipschitz continuity, this is not a big problem.

Theorem 4.4. *Let Ω_0 be bounded and assume that B is compact and such that B° is a C^1-domain. Let F be Lipschitz continuous, $F \geq 0$ and $\inf_{x \in B} F(x) > 0$. Then $d_H(\Omega_t^p, \pi(\Omega_t)) = o(t)$ as $t \to 0^+$.*

Proof. Note that our assumptions imply that there exists an open neighbourhood $U \supset B$ and $\underline{F} > 0$ such that $F \geq \underline{F}$ on U. Furthermore, since Ω_0 is bounded and F continuous, we may assume, without loss of generality, that $F \leq \overline{F}$. Since $\Omega_t^p \subset \pi(\Omega_t)$, we only have to consider $d_H(\pi(\Omega_t) \to \Omega_t^p)$. Let $t > 0$ be small and $y \in \pi(\Omega_t) \setminus \Omega_t^p$. Since $y \in \Omega_t$, there exist $x \in \Omega_0$ and $\xi_0 \in X_{\mathrm{ad}}(x, y)$ with $l(\xi_0) < t$. Note that $|x-y| \leq \overline{F} \cdot l(\xi_0) < \overline{F} t$ by Definition 1.1, so that $|x-y|$ gets arbitrarily small in the limit $t \to 0^+$. Because $y \notin \Omega_t^p$, we can conclude that the path ξ_0 must intersect B. By making t and thus $|x-y|$ small enough, we can assume that $x, y \in U$.

For any given $\epsilon > 0$, we can now apply Lemma 4.3 to get a corresponding δ. Assume that t is chosen small enough to yield $|x-y| < \delta$. By reducing δ further, we can also guarantee $(1+\epsilon)L|x-y| < \underline{F}$, where L denotes the Lipschitz constant of F. For our x and y, Lemma 4.3 gives a path $\xi \in X_{\mathrm{ad}}(x, y)$ with $|\xi| \leq (1+\epsilon)|x-y|$ and such that $\xi(\tau) \notin B$ for all $\tau \in [0,1]$. Using Lipschitz continuity of F, one can derive the following estimates (see Lemma 9 in [13] for the details):

$$l(\xi) \leq \frac{1}{L} \log\left(\frac{F(y)}{F(y) - (1+\epsilon)L|x-y|} \right), \quad t > d(x,y) \geq \frac{1}{L} \log\left(\frac{F(y) + L|x-y|}{F(y)} \right)$$

Setting $\tilde{d} = l(\xi) - t$ and $\lambda = L|x-y|/F(y)$, these estimates imply further

$$e^{L\tilde{d}} - 1 \leq \frac{F(y)}{F(y) - (1+\epsilon)L|x-y|} \cdot \frac{F(y)}{F(y) + L|x-y|} - 1 = \frac{\epsilon\lambda + (1+\epsilon)\lambda^2}{1 - \epsilon\lambda - (1+\epsilon)\lambda^2}.$$

Because $y \notin \Omega_t^p$, we know that $l(\xi) \geq t$ must hold. Hence, the intermediate-value theorem implies that there exists some $\tau \in (0,1]$ such that ξ split into ξ_1 on $[0, \tau]$ and ξ_2 on $[\tau, 1]$ implies $l(\xi_1) = t$. Set $\tilde{x} = \xi_1(1) = \xi_2(0) = \xi(\tau)$. It is clear that $\tilde{x} \in \overline{\Omega_t^p}$ since $l(\xi_1) = t$. Because x is in the *interior* of the open set Ω_0, we actually also get $\tilde{x} \in \Omega_t^p$. Since $\xi_2 \in X_{\mathrm{ad}}(\tilde{x}, y)$ and $l(\xi_2) = l(\xi) - t = \tilde{d}$, another estimate based on Lipschitz continuity of F and Lemma 9 in [13] yields

$$|\tilde{x} - y| \leq \frac{F(y)}{L}\left(e^{L\tilde{d}} - 1\right) \leq \frac{F(y)}{L} \cdot \frac{\epsilon\lambda + (1+\epsilon)\lambda^2}{1 - \epsilon\lambda - (1+\epsilon)\lambda^2}. \tag{10}$$

By a similar argument, we can also derive

$$|x-y| \leq \frac{F(y)}{L}\left(e^{Lt} - 1\right) \quad \Leftrightarrow \quad \lambda \leq e^{Lt} - 1.$$

Note that the right-hand side of (10) is increasing in λ if $|x-y|$ and thus λ are small enough. Thus, we can combine everything and use a series expansion to finally find

$$|\tilde{x} - y| \leq \frac{F(y)}{L}\left(L\epsilon t + O\left(t^2\right)\right) \leq \overline{F}\epsilon \cdot t + O\left(t^2\right).$$

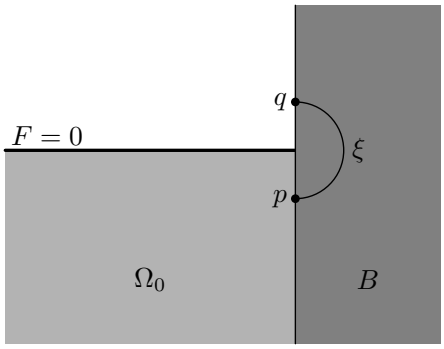


FIGURE 6. Sketch of a possible situation where Ω_t^p and $\pi(\Omega_t)$ do not match qualitatively in their time evolution. The dark grey area is the forbidden region B. We choose $F = 0$ on the horizontal line and $F > 0$ everywhere else.

Taking the supremum over y in this inequality yields an estimate for $d_H(\pi(\Omega_t) \to \Omega_t^p)$. Since ϵ can be chosen arbitrarily small for the limit $t \to 0^+$, this implies the claim. □

We require that $\inf_{x \in B} F(x)$ must be strictly positive to avoid a situation like Figure 6. If a line with $F = 0$ touches B, then we may get a qualitative difference between the time evolution of Ω_t^p and that of $\pi(\Omega_t)$: In the shown situation, $\Omega_t^p = \Omega_0$ for all $t \geq 0$ since there is no possibility for the initial region to grow anywhere with speed F^p. On the other hand, Ω_t will grow inside B and, at some point, also reach the upper area outside of B. When this happens, $\pi(\Omega_t)$ contains parts of the quarter plane in the north west. Let us briefly analyse this effect also more explicitly. For this, choose the speed field as

$$F(x,y) = \begin{cases} |y| & \text{if } x \leq 0, \\ \sqrt{x^2 + y^2} & \text{if } x \geq 0. \end{cases}$$

The forbidden region is the half-plane $B = \{(x,y) \in \mathbb{R}^2 \mid x \geq 0\}$. This matches the schematic illustration of Figure 6. For Ω_0 to reach the north-west region, we have to consider paths like $\xi \in X_{\text{ad}}(p,q)$ shown in the sketch. Note that F is constant along such a path, so that

$$l(\xi) = \frac{\pi r}{F} = \pi,$$

where r is the radius of the circular arc. In particular, *the path length is independent of* r. This implies that it is not possible to reach the north west after arbitrarily small times, even tough p and q can be as close together as we wish. Hence, the statement of Theorem 4.4 is actually still true for this situation (since we take the limit $t \to 0^+$). We believe that it is possible to exploit Lipschitz continuity of F to show that this must always be the case and that Theorem 4.4 holds generally without the strict positivity requirement.

5. Numerical Experiments

To conclude our discussion, let us consider some numerical results. We have implemented all strategies for handling shape constraints discussed above, as per Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. They are applied for the minimisation of the cost

$$J(\Omega) = \int_\Omega (u(x) - \overline{u})^2 \, dx - 2\gamma \cdot \sigma \cdot \operatorname{vol}(\Omega) \tag{11}$$

with various kinds of forbidden regions. We use $D \subset \mathbb{R}^2$ as hold-all domain, and assume that a grey-scale image $u \colon D \to I \subset \mathbb{R}$ is given. The quantities $\overline{u}$ and σ,

$$\overline{u} = \frac{1}{\operatorname{vol}(\Omega)} \int_\Omega u(x)\, dx, \quad \sigma^2 = \frac{1}{\operatorname{vol}(\Omega)} \int_\Omega (u(x) - \overline{u})^2 \, dx,$$

are the mean intensity and standard deviation of the image u over Ω, respectively, and $\gamma > 0$ is a constant parameter.

Finding an optimal shape Ω for (11) can be interpreted as image segmentation based on the Chan-Vese model [2]. This is illustrated in Figure 7. For a more thorough interpretation and discussion of the particular segmentation model, let us refer to [14] and Chapter 6 of [12]. Note that we are neither aware of any other work about image segmentation with geometric constraints nor of an actual application of such a problem. It is, however, a very instructive model problem for an analysis of the behaviour of the descent methods. Furthermore, we believe that geometric constraints in a segmentation model could be used to incorporate "outside information" into the segmentation process. This

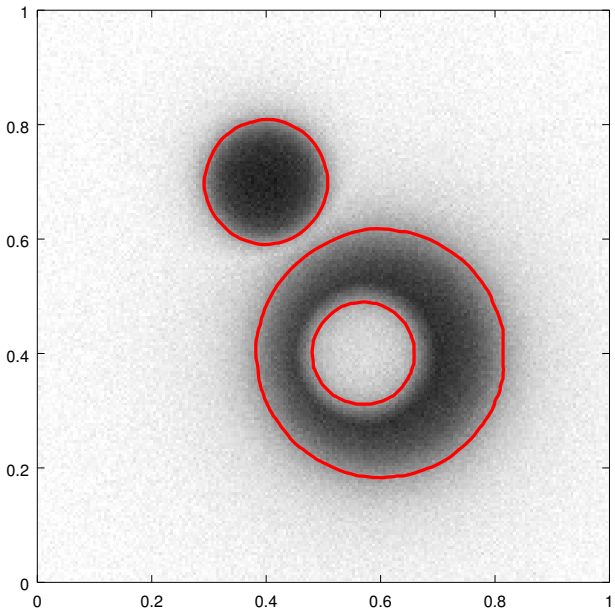
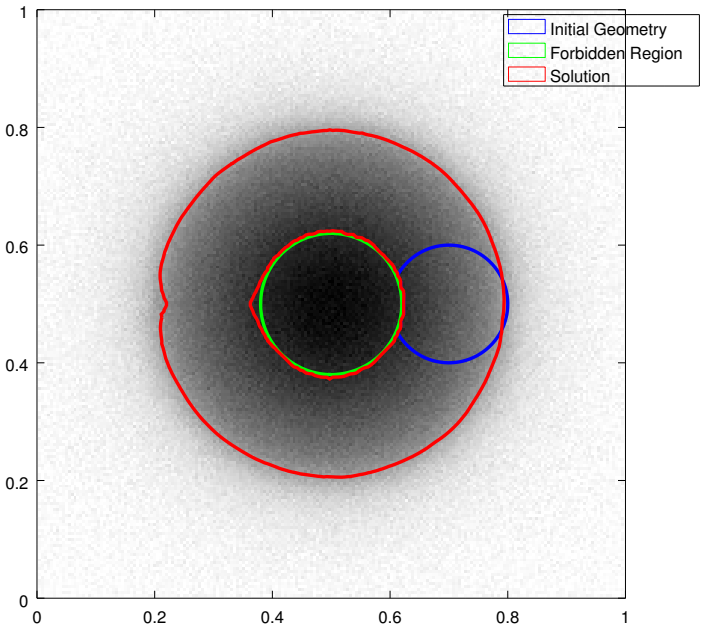


FIGURE 7. Our goal of the example optimisation problem considered for the numerical experiments in Section 5 is to find a shape matching one segment of a given grey-scale image. Here, we have not yet applied any constraints so far.

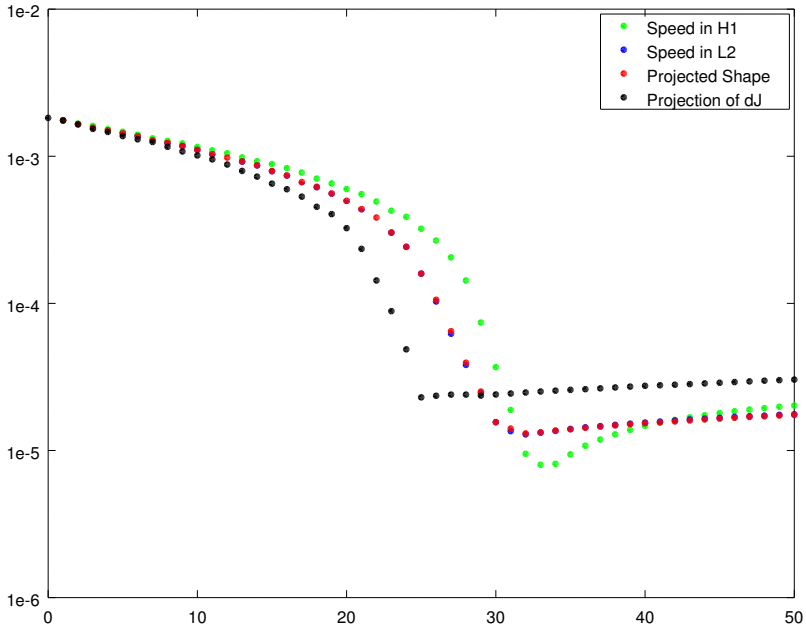
additional information could either be provided by a human operator in a semi-automatic method or it could come from a second channel of measurement data.

All computations are done with 2% of Gaussian noise added and for $\beta = 10^{-2}$. (Recall the parameter β from (3), which is used to reduce smoothing effects of the inverse Laplacian when solving the variational problem.) We perform the descent run for 50 different random configurations in each problem setting to make the qualitative result more representative. The cost evolutions are averaged over all runs, for which we employ the *geometric mean* since it fits better to the logarithmic scaling of the y-axis. We chose a small minimum step length of $t_{\min} = 10^{-6}$ to ensure that no effects of the constraints (which are our main object of interest) are obscured by artefacts introduced by enforced longer steps. Since we do not know the true optimal cost, we use the smallest achieved value for a particular configuration (over all iteration steps and with all methods) as base value. Note that the numerical examples exhibit both positive and negative speeds in the movement of the boundary (as illustrated in Figure 11b). This shows that the condition $F \geq 0$, which we assumed for some of our theoretical results above, can be lifted in practice.

Figure 8, Figure 9 and Figure 10 show the results for three different configurations of the forbidden region. For each case, the top plot shows the basic situation (initial geometry Ω_0, forbidden region B and the approximate final solution). The bottom plot compares the averaged decrease of the cost value for the four methods we consider. The first situation in Figure 8 has the "easiest" constraint. The region B in this case is, in particular, smooth. Consequently, Theorem 4.4 tells us that the results of using the speed projection in L^2 and projecting the shape should be very similar. This statement is, indeed, confirmed by Figure 8b: The blue and red dots match quite well. The green dots, corresponding to a speed projection in H^1 instead, show a different behaviour: The decrease is more slowly here in the beginning, which can be explained by the fact that the speed field gets drawn to zero where the constraint is almost active and thus the whole descent is done in a more cautious fashion. This, however, also leads to a (slightly) better minimal cost in the end. In Figure 9, the forbidden region is a triangle with sharp corners. This set B *does not* fulfil the requirements of Theorem 4.4, and we can see that also the corresponding cost descents in Figure 9b are not as similar as before. The blue and red dots clearly diverge from each other around iteration eight, which is when the growing domain starts to wrap around the corners of the triangle. This is in nice correspondence to our theory discussed in Section 4. The green curve shows the same behaviour as before, but much more pronounced

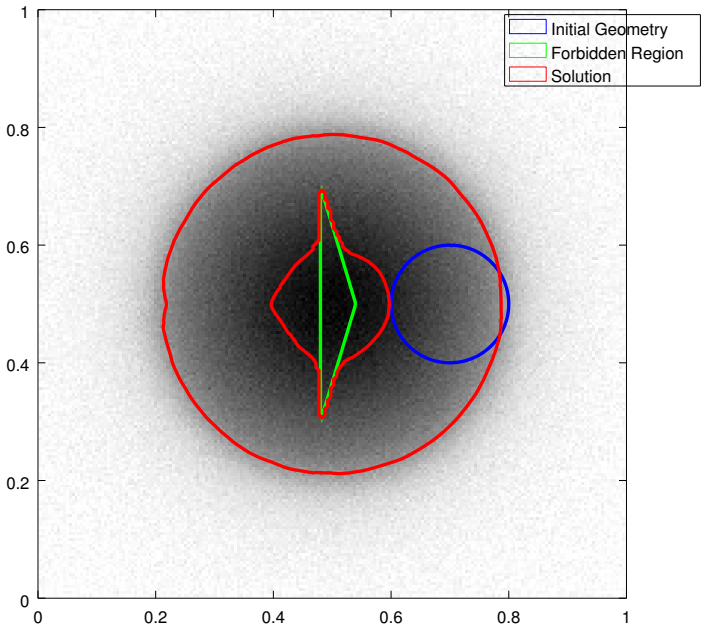


(A) The basic configuration in this example.

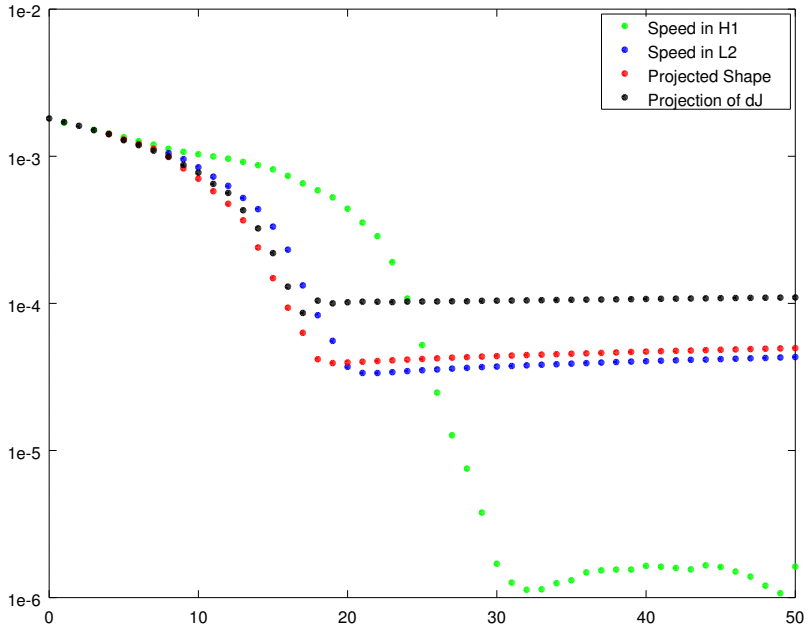


(B) Average evolution of the cost during the iteration for all methods.

FIGURE 8. Comparison of the suggested methods for handling constraints. These are projection of the speed in H^1, in L^2 with (4), projection of the shape with (5), and infeasible steps with projection of the shape derivative.

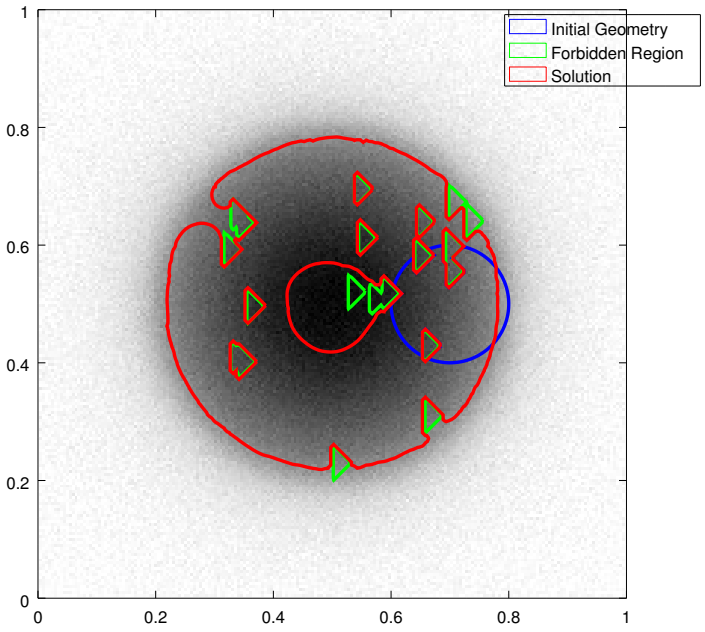


(A)

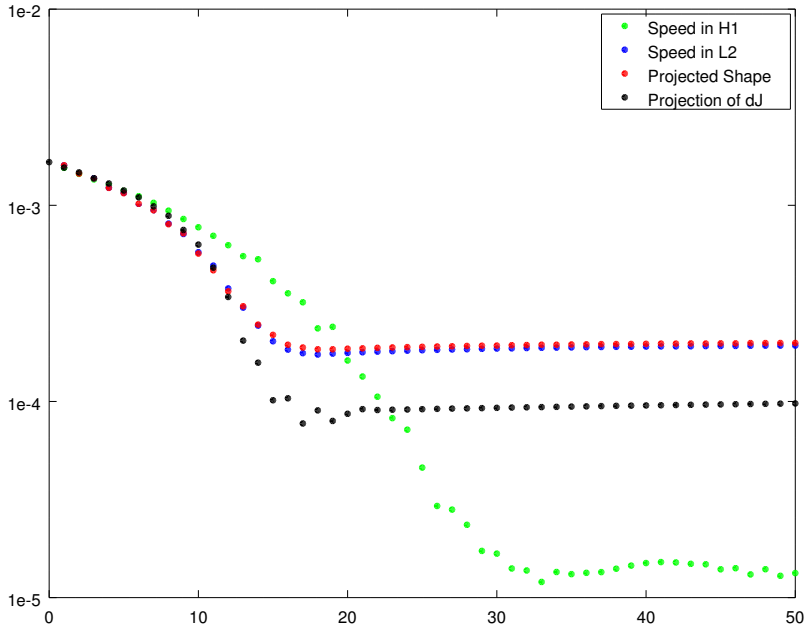


(B)

FIGURE 9. Continuation of Figure 8 for a different configuration of the forbidden region.



(A)



(B)

FIGURE 10. Continuation of Figure 8 with a forbidden region that consists of multiple, randomly placed triangles. Each run has not only different noise, but also a different arrangement of the triangles.

this time. The decrease is more slowly at the beginning, but the descent is more robust and leads to a much better result.

For both of these examples, infeasible steps with a projection of the shape derivative (black dots, Algorithm 4) show the opposite behaviour to the speed projection in H^1 marked in green: The constraints restrict the descent much less, so that it is faster initially. The results, on the other hand, are worse and less robust, because the constraints are taken much less into account. (While the constraints are used to compute the descent speed field and to find a feasible shape in the end, the individual shapes during the descent (before projection) are allowed to be infeasible.) Note particularly that the optimal shape in Figure 9a contains a hole in the domain around the forbidden triangle. This hole is not generated by descent with infeasible steps, where the descent converges to the local minimum without hole instead. For the feasible methods, the forbidden region forces the resulting domain to the correct topology. While this is, of course, a particular property of the problem we consider here, it still shows that infeasible steps are not suitable for all situations and may lead to less robust descent than a speed projection in H^1.

With the setting of Figure 10, finally, the effect of non-smooth forbidden regions seen in Figure 9 can be interpreted even better: Here, the forbidden region is more complex and difficult than before. Figure 11 shows typical speed fields as they occur in the descent with shape projection (Figure 11a) and the speed projected in H^1 (Figure 11b). In the former case, we get very large speed values around the parts of B that are already surrounded by the growing domain. This is the case since it would be good for decreasing the cost to include them in Ω, and the speed field does not know anything about the constraints that forbid this. With a projection in H^1, on the other hand, the speed field *does* include information about the constraints. Consequently, the speed is small (and continuous) around those regions. Note that our descent algorithm computes the directional shape derivative according to (3) and uses it for the Armijo rule as the expected cost decrease. This, however, only works when the constraints are correctly accounted for in the shape gradient. Consequently, the Armijo rule expects a much larger cost decrease than we can actually achieve for the situation of Figure 11a. This leads to the effect that only the minimal step length is accepted from this iteration onward, and the final shape looks almost the same as the black shape indicated in Figure 11a. This is clearly suboptimal. With the speed projection in H^1, on the other hand, the shape from Figure 11b continues to evolve and results in the correct geometry shown also in Figure 10a. One can confirm this interpretation by lowering the relaxation parameter used in the Armijo rule sharply. This improves the performance of the shape-projection method in relation to speed projection in H^1 considerably, and avoids a stuck situation as in Figure 11a. Infeasible steps with a projection of the shape derivative do not suffer from this problem, either, since Algorithm 4 constructs the speed field in a way that is consistent to the computed shape derivative. As before, however, the final result is not as good as for a speed projection in H^1.

All in all, our numerical results show that all four methods proposed in Section 2 have both their own unique benefits and drawbacks. Projecting the speed field consistently in the original Hilbert space H^1 shows the slowest descent, but leads to the most robust method with the best final results. Infeasible steps with a projection of the shape derivative can speed up the descent, but may (depending on the situation) take the constraints too little into account. The other two methods (shape projection and projection of the speed in L^2) behave similarly, as we expect from Theorem 4.4. They also lead to faster descent, but they introduce inconsistencies between the computed shape derivative and the descent step. This can lead to a stuck descent or other issues for certain configurations of the constraints.

The author would like to thank Wolfgang Ring of the University of Graz for thorough proofreading of the manuscript. He is also particularly grateful for lots of interesting discussions about geometric constraints in shape optimisation with Moritz Keuthen at the Technical University of Munich. The interesting method of Algorithm 4 was suggested during peer review of the paper. This work is supported by the Austrian Science Fund (FWF) and the International Research Training Group IGDK 1754.

References

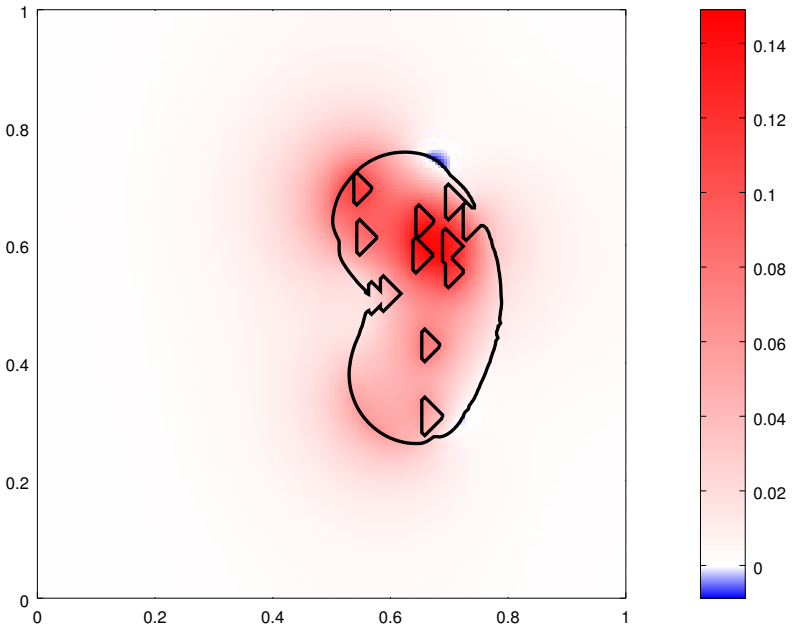
[1] Martin Burger. A Framework for the Construction of Level Set Methods for Shape Optimization and Reconstruction. *Interfaces and Free Boundaries*, 5:301–329, 2003.

[2] Tony F. Chan and Luminita A. Vese. Image Segmentation Using Level Sets and the Piecewise-Constant Mumford-Shah Model. Technical Report 00-14, UCLA CAM, 2000.

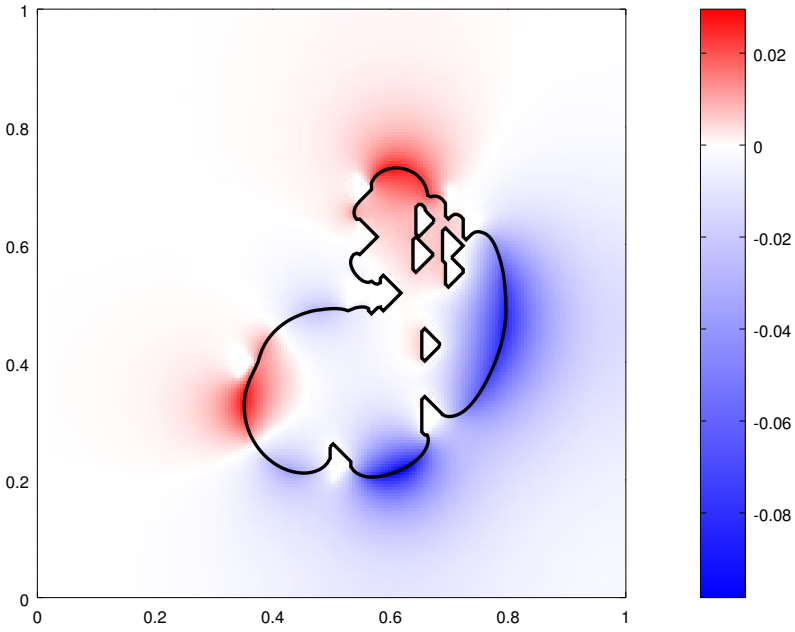
[3] Michel C. Delfour and Jean-Paul Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM, second edition, 2011.

[4] John W. Eaton, David Bateman, Søren Hauberg, and Rik Wehbring. GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations, 2015. `https://www.gnu.org/software/octave/doc/interpreter/`.

[5] Yoshikazu Giga. *Surface Evolution Equations: A Level Set Approach*, volume 99 of *Monographs in Mathematics*. Birkhäuser, 2006.



(A) Projection of the shape with (5).



(B) Projection of the speed field in H^1.

FIGURE 11. Typical speed fields during the descent for the problem from Figure 10.

[6] Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, 2009.

[7] Moritz Keuthen. *Second Order Shape Optimization with Geometric Constraints*. PhD thesis, Technical University of Munich, 2015.

[8] Moritz Keuthen and Daniel Kraft. Shape Optimization of a Breakwater. *Inverse Problems in Science and Engineering*, 24(6):936–956, 2016. `http://dx.doi.org/10.1080/17415977.2015.1077522`.

[9] Moritz Keuthen and Michael Ulbrich. Moreau-Yosida Regularization in Shape Optimisation with Geometric Constraints. *Computational Optimization and Applications*, 62(1):181–216, 2015. `http://dx.doi.org/10.1007/s10589-014-9661-0`.

[10] Daniel Kraft. A Hopf-Lax Formula for the Level-Set Equation and Applications to PDE-Constrained Shape Optimisation. In *Proceedings of the 19th International Conference on Methods and Models in Automation and Robotics*, pages 498–503. IEEE Xplore, 2014.

[11] Daniel Kraft. The `level-set` Package for GNU Octave. Octave Forge, 2014–2015. `http://octave.sourceforge.net/level-set/`.

[12] Daniel Kraft. *A Level-Set Framework for Shape Optimisation*. PhD thesis, University of Graz, 2015. `https://www.domob.eu/research/phd/thesis.pdf`.

[13] Daniel Kraft. A Hopf-Lax Formula for the Time Evolution of the Level-Set Equation and a New Approach to Shape Sensitivity Analysis. *Interfaces and Free Boundaries*, 18(3):317–353, 2016. DOI 10.4171/IFB/366.

[14] Daniel Kraft. Self-Consistent Gradient Flow for Shape Optimization. *Optimization Methods and Software*, 32(4), 2017. `http://dx.doi.org/10.1080/10556788.2016.1171864`.

[15] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operation Research and Financial Engineering. Springer, second edition, 2006.

[16] Stanley J. Osher and Fadil Santosa. Level Set Methods for Optimization Problems Involving Geometry and Constraints: I. Frequencies of a Two-Density Inhomogeneous Drum. *Journal of Computational Physics*, 171(1):272–288, 2001.

[17] Stanley J. Osher and James A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.