



Flip procedure in geometric approximation of multiple-components shapes – Application to multiple-inclusions detection

PIERRE BONNELIE ¹
LOÏC BOURDIN ²
FABIEN CAUBET ³
OLIVIER RUATTA ⁴

¹ Institut de recherche XLIM. Département de Mathématiques et d’Informatique. UMR CNRS 7252. Université de Limoges, France

E-mail address: pierre.bonnelie@etu.unilim.fr

² Institut de recherche XLIM. Département de Mathématiques et d’Informatique. UMR CNRS 7252. Université de Limoges, France

E-mail address: loic.bourdin@unilim.fr

³ Institut de Mathématiques de Toulouse. UMR CNRS 5219. Université de Toulouse, France

E-mail address: fabien.caubet@math.univ-toulouse.fr

⁴ Institut de recherche XLIM. Département de Mathématiques et d’Informatique. UMR CNRS 7252. Université de Limoges, France

E-mail address: olivier.ruatta@unilim.fr.

ABSTRACT. We are interested in geometric approximation by parameterization of two-dimensional multiple-components shapes, in particular when the number of components is *a priori* unknown. Starting a standard method based on successive shape deformations with a one-component initial shape in order to approximate a multiple-components target shape usually leads the deformation flow to make the boundary evolve until it surrounds all the components of the target shape. This classical phenomenon tends to create double points on the boundary of the approximated shape.

In order to improve the approximation of multiple-components shapes (without any knowledge on the number of components in advance), we use in this paper a piecewise Bézier parameterization and we consider two procedures called *intersecting control polygons detection* and *flip procedure*. The first one allows to detect collisions between two parts of the boundary of the approximated shape, and the second one permits to change its topology by dividing a one-component shape into a two-components shape.

For an experimental purpose, we include these two processes in a basic geometrical shape optimization algorithm and test it on the classical inverse obstacle problem. This new approach allows to reconstruct numerically the unknown inclusion, detecting both the topology (*i.e.* the number of connected components) and the shape of the obstacle. Several numerical simulations are performed and underline the good behavior of this method.

Keywords. Shape approximation; free-form shapes; multiple-components shapes; Bézier curves; intersecting control polygons detection; flip procedure; inverse obstacle problem; shape optimization.

Math. classification. 68U05; 68W25; 49Q10; 65N21.

1. Introduction

Geometric shape approximation methods are frequently based on successive shape deformations, where the boundary of the approximated shape is parameterized and evolves at each step in a direction given by the deformation flow. This technique is widely used for example in shape optimization problems where the flow is given by the so-called *shape gradient* (see, e.g., Chapter 5 of the book [17] of Henrot *et al.*), or in image segmentation (see, e.g., [18]). Numerous parameterizations of the boundary have been considered in the literature, such as polygons, Fourier series, etc. Each of these parameterizations has its own advantages and drawbacks, that depend on the nature of the problem studied.

In this paper we are especially interested in the geometric approximation of multiple-components shapes, in particular when the number of components is *a priori* unknown. Starting a parameterization method with a one-component initial shape in order to approximate a multiple-components target shape usually leads the deformation flow to make the boundary evolve until it surrounds all the components of the target shape (see Figure 1 for illustrations). This classical phenomenon tends to create double points on the boundary of the approximated shape.

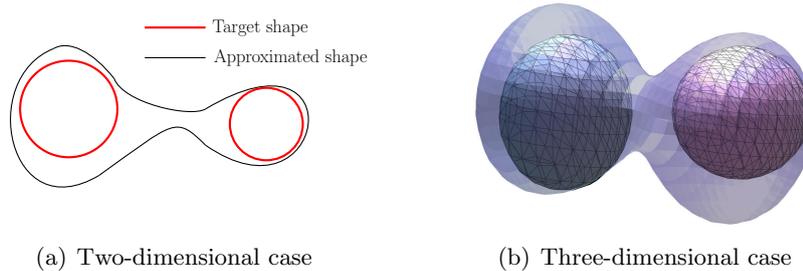


FIGURE 1. Geometric shape approximation of a two-components target shape starting from a one-component initial shape.

In order to improve the approximation of multiple-components shapes, our idea is to look for an appropriate parameterization that allows to achieve two numerical tasks. Firstly the parameterization has to be well-suited in order to detect the formation of double points, *i.e.* to locate the parts of the boundary that are close to each other. Secondly it has to be adapted in order to introduce a *flip procedure* that allows to change the topology of the approximated shape, precisely to divide a one-component shape into a two-components shape. Moreover, for practical uses, we look in this paper for a complete method that is easily implementable with a relatively low numerical cost.

We present here a method based on a Bézier parameterization. The main idea is that this polynomial parameterization can be approximated by its control polygon. As a consequence, one can easily detect the formation of double points by looking for intersecting axis-aligned bounding boxes of the control points of the Bézier parameterization. We refer to Section 3.3 for details on the so-called *intersecting control polygons detection*. Once this first step is achieved, one can easily reorganize the previously detected control points of the Bézier parameterization in order to modify the topology of the shape, precisely in order to divide one component into two. We refer to Section 3.4 for details on the so-called *flip procedure*.¹

In this work we detail the above method in the two-dimensional case, using piecewise Bézier curves. It is worth to mention that this method can be extended to the three-dimensional case, using piecewise Bézier surfaces.

In order to test the flip procedure introduced in this paper, we perform numerical simulations on a particular problem. Precisely we consider the inverse problem of detecting inclusions in a bounded domain from boundary measurements. In this paragraph we briefly recall the three major techniques used in the literature in order to study this problem. The first technique is the so-called *level set* approach (see, e.g., the survey [7] of Burger *et al.* and references therein). In order to detect several inclusions, this method does not need any *a priori* knowledge on the number of inclusions. However, the level set methods are based on an implicit representation of the approximated shape and require relatively complex techniques to be implemented, particularly in the case of inverse problems which usually need some regularization methods. The second major technique is based on *shape derivatives*

¹A similar procedure can be developed in order to merge two components into one. We refer to Section 4.3.6 for details.

(see, e.g., the work [1] of Afraites *et al.*). However, the standard algorithm based on shape derivatives does not provide the opportunity to change the topology of the shape and consequently the number of inclusions has to be known in advance. The third major method is the so-called *topological gradient* introduced by Schumacher in [23] and Sokolowski *et al.* in [25]. However, this method is based on asymptotic expansions and consequently is essentially adapted for relatively small inclusions. Moreover, even if the topological optimization is useful in order to find the number of inclusions, it may be not well-suited in order to find the actual shape of the inclusions. To conclude this paragraph, let us mention that combinations of several shape optimization methods have recently been tested by several authors. For instance we refer to the works of Allaire *et al.* in [2] and Burger *et al.* in [6] that combine the classical geometric shape optimization through the level set method and the topological gradient, to the work of Pantz *et al.* in [22] which develops an algorithm using boundary variations, topological derivatives and homogenization methods and to the work of Caubet *et al.* in [9] which couples topological and shape derivatives approaches.

The parameterization by piecewise Bézier curves and the flip procedure introduced in this paper have several advantages in order to study the above inverse obstacle problem, in particular in the case where the number of inclusions is *a priori* unknown. Firstly, the explicit Bézier representation of the boundary is very simple to implement. Secondly, we use in this paper the shape derivatives approach which allows to approximate the shape of the inclusions. Finally, the flip procedure permits to dynamically change the topology of the shape in order to find the number of inclusions.

Organization of the paper. The paper is organized as follows. Section 2 recalls some basics and notations about piecewise Bézier curves. Section 3 is concerned with the two main features of this paper, that is, the intersecting control polygons detection and the flip procedure. Section 4 is dedicated to several numerical simulations in the context of the inverse obstacle problem studied through a shape optimization approach.

2. Notations and basics on piecewise Bézier curves

In this section we fix our notations and recall some basics about Bézier curves (see, e.g., [15, 24] for more details). Let $d \in \mathbb{N}^*$ and a set of $d + 1$ points P_0, \dots, P_d of \mathbb{R}^2 . The associated Bézier curve, denoted by $B([P_0, \dots, P_d])$, is defined by

$$\forall t \in [0, 1], B([P_0, \dots, P_d], t) := \sum_{j=0}^d P_j b_{j,d}(t),$$

where $b_{j,d}$ are the classical Bernstein polynomials given by

$$b_{j,d}(t) := \binom{d}{j} t^j (1-t)^{d-j}.$$

The integer d is the *degree* of the curve and the points P_0, \dots, P_d are its *control points* (or its *control polygon*). Note that a Bézier curve does not go through its control points in general. However it starts at P_0 and finishes at P_d . If $P_0 = P_d$, the Bézier curve is said to be *closed*. Each point of a Bézier curve is a convex combination of its control points. As a consequence, a Bézier curve lies in the convex hull of its control polygon (see Figure 2).

Remark 2.1. As Bézier curves are widely used in Computer Aided Geometric Design (see [15, 24]), they are commonly defined as parametric curves lying in the euclidean space \mathbb{R}^2 (or \mathbb{R}^3). However this definition can be extended to \mathbb{R}^n for any $n \in \mathbb{N}^*$. In this paper, we are only interested in the two-dimensional case $n = 2$.

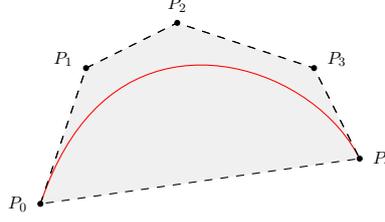


FIGURE 2. A non-closed Bézier curve of degree 4 lying in the convex hull of its control polygon.

In this paper we focus on the geometric approximation of continuous boundaries of two-dimensional bounded shapes with the help of Bézier curves. In the sequel no distinction will be done between a two-dimensional bounded shape and its continuous boundary.

Using a single closed Bézier curve in order to approximate a two-dimensional bounded shape is not an efficient method for several reasons. Indeed, in order to approximate a shape with a lot of geometric features, one would need to increase the degrees of freedom, *i.e.* the number of control points. However, as very well-known, increasing the degree of an approximating polynomial curve leads to a classical oscillations phenomenon and, in the particular case of a Bézier polynomial curve, it leads to numerical instability (due to the ill-conditionness of the Bernstein-Vandermonde matrices). Moreover, since each control point has a global influence on the curve, one could not handle local complexities of a shape with a single Bézier curve. The classical idea is then to divide the curve in several Bézier curves of small degrees. This leads us to recall the following definition of piecewise Bézier curves.

Let $N \in \mathbb{N}^*$, $d \in \mathbb{N}^*$ and a set of $N(d+1)$ control points $P_{1,0}, \dots, P_{1,d}, \dots, P_{N,d}$ of \mathbb{R}^2 satisfying the *continuity relations* $P_{i,d} = P_{i+1,0}$ for every $i = 1, \dots, N-1$.² The associated piecewise Bézier curve, denoted by $B([P_{1,0}, \dots, P_{N,d}])$ ³, is defined by

$$\forall t \in [0, 1], B([P_{1,0}, \dots, P_{N,d}], t) := B([P_{i,0}, \dots, P_{i,d}], Nt - i + 1), \text{ if } t \in \left[\frac{i-1}{N}, \frac{i}{N} \right].$$

The global curve is then composed by N Bézier curves called *patches*. Note that a piecewise Bézier curve goes through $P_{i,0}$ and $P_{i,d}$ for all $i = 1, \dots, N$. If $P_{1,0} = P_{N,d}$, the piecewise Bézier curve is said to be *closed*.

Remark 2.2. In practice we use cubic patches ($d = 3$) because they are sufficient in order to recover many geometrical situations, such as inflexion points (see Figure 3).

Remark 2.3. In this paper, since each Bézier patch has the same degree d , the curve is said to be *uniform in degree*. Nevertheless one can easily build piecewise Bézier curves with patches of different degrees.

Adapting the proof of the classical Stone-Weierstrass theorem, one can easily prove the following result (which corresponds to a particular case of the classical Bishop theorem, see [4]).

Theorem 2.4. *Let $f \in \mathcal{C}([0, 1], \mathbb{R}^2)$. For all $\varepsilon > 0$ and all $d \in \mathbb{N}^*$, there exist $N \in \mathbb{N}^*$ and a set of $N(d+1)$ control points $P_{1,0}, \dots, P_{1,d}, \dots, P_{N,d}$, satisfying the continuity relations, such that $\|f(t) - B([P_{1,0}, \dots, P_{N,d}], t)\|_{\mathbb{R}^2} \leq \varepsilon$ for all $t \in [0, 1]$.*

This result fully justifies the use of piecewise Bézier curves in order to approximate two-dimensional bounded shapes.

²The continuity relations guarantee the well-definedness and the continuity of the piecewise Bézier curve.

³One would note here a conflict in notations of a Bézier curve and of a piecewise Bézier curve. In the sequel no confusion is possible since we will only consider piecewise Bézier curves.

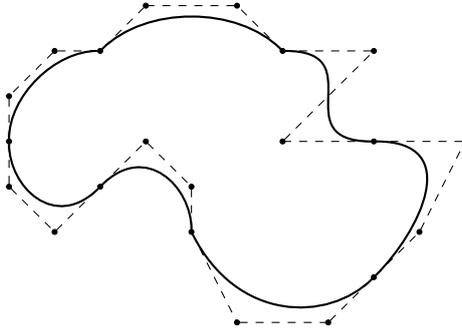


FIGURE 3. A closed piecewise Bézier curve composed by seven cubic patches.

Remark 2.5. Recall that the use of polar coordinates, where the radius is expanded in a truncated Fourier series, is another common and efficient strategy in order to approximate two-dimensional bounded shapes (see, e.g., [1] in the context of inclusions detection). However it has two main drawbacks. Firstly it allows to represent only star-shaped domains and secondly, due to a classical oscillations phenomenon, it cannot represent rigorously straight lines (see, e.g., [11, Figure 5 p.140] in the context of inclusions detection). The use of piecewise Bézier curves is then an alternative in order to approximate non star-shaped domains and straight lines (see Section 4.3 for some numerical simulations in the context of inclusions detection). To conclude this remark, let us mention that the *flip procedure*, which is the main feature of this paper, is based on the detection of collisions between two parts of the boundary of the shape (see Section 3 for more details). Thus, it is worth to precise that a parameterization based on polar coordinates, where the radius is expanded in a truncated Fourier series, is not adapted to detect such collisions, in contrary to a piecewise Bézier parameterization (as explained in Section 3.3).

3. Intersecting control polygons detection and flip procedure

In this paper we are interested in geometric two-dimensional shape approximation problems in which the target shape can have multiple connected components but the number of components is unknown. In such a case, starting a classical geometric approximation with a one-component initial shape may lead to the situation depicted in Figure 4, that is, the deformation flow makes the boundary evolve until it surrounds all the components of the target shape. This classical phenomenon tends to create a collision between two parts of the boundary of the approximated shape.

In this paper our major aim is to provide a simple and new concept (called *flip procedure*) that can be added to any shape approximation algorithm based on piecewise Bézier curves, and which allows to change the topology of the approximated shape. Precisely, the flip procedure allows to divide a one-component shape into a two-components shape.

Remark 3.1. In this paper, we focus on piecewise cubic Bézier curve ($d = 3$, see Remark 2.2). However, this method can be extended to any $d \geq 2$.

3.1. Overview

Let us consider a general geometric shape approximation algorithm in which the boundary of the approximated shape is parameterized by a piecewise cubic Bézier curve. It starts from a one-component initial shape ω_0 and produces a sequence of one-component shapes $(\omega_k)_{k \geq 0}$ by deforming the boundary at each step. Our idea consists in two phases (that are summarized in Figure 5):

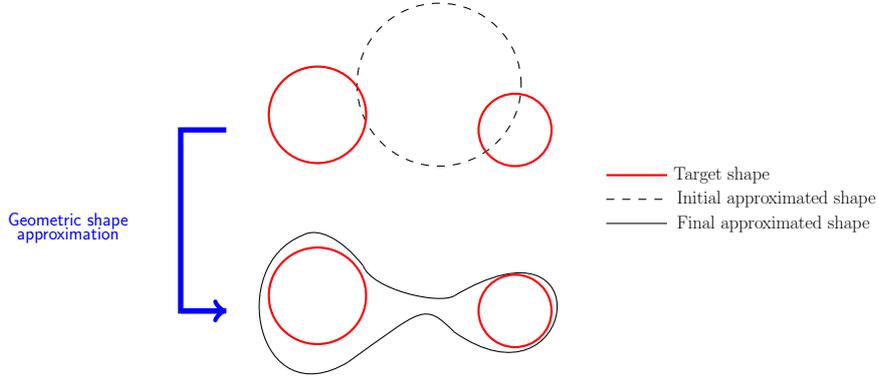


FIGURE 4. A geometric shape approximation of a two-components target shape starting from a one-component initial approximated shape. The final approximated shape surrounds the two components.

- (1) check, at each step of the approximation algorithm, if the current shape ω_k is in the situation depicted in Figure 4, that is, if two parts of the boundary come closer and closer to each other. The parameterization by piecewise Bézier curves allows us to detect such a situation by looking for intersecting control polygons. This procedure will be called *intersecting control polygons detection* and will be detailed in Section 3.3;
- (2) if some control polygons intersect each other, we apply the flip procedure in order to obtain a two-components shape by keeping unchanged all other control polygons. The flip procedure is detailed in Section 3.4.

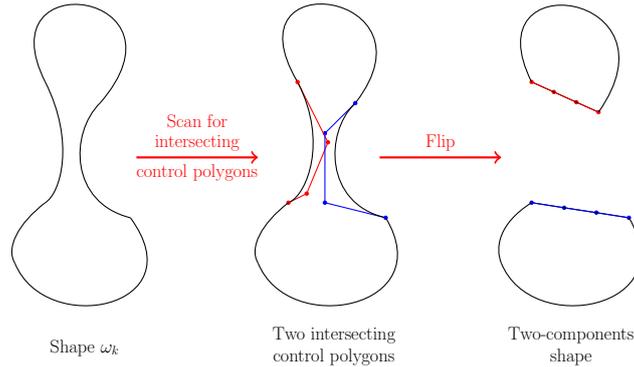


FIGURE 5. Overview of the complete procedure.

3.2. Two assumptions

In the sequel we will assume that the following hypotheses are both satisfied:

- (1) the size of the control polygons (that is, the diameter of their convex hull) evolves in a fixed interval $[S_{\min}, S_{\max}]$, with $0 < S_{\min} < S_{\max}$;

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

- (2) the deformation step size of the shape approximation algorithm is chosen very small with respect to S_{\min} .

The first hypothesis allows to maintain numerical stability, avoiding to deal with very large patches and/or very small ones.⁴ The second hypothesis allows to avoid the situation depicted in Figure 6.

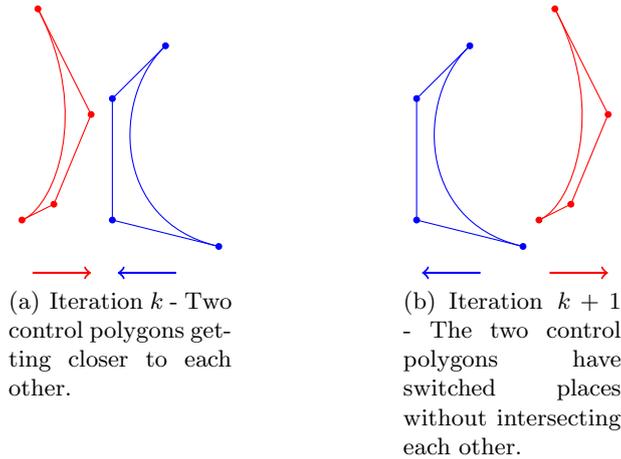


FIGURE 6. Deformation step too large.

Under these two assumptions, only two natural situations of intersection occur in practice. Either one control polygon intersects exactly another one, or one control polygon intersects exactly two consecutive ones (see Figure 7).⁵ The flip procedure handle these two situations, see Section 3.4.

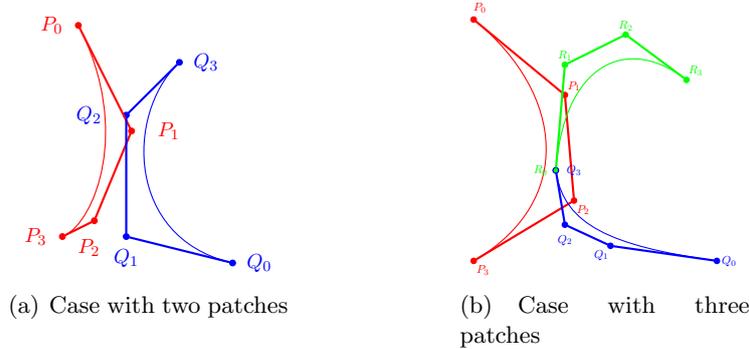


FIGURE 7. Two natural situations of intersecting control polygons.

Remark 3.2 (Control of the size of the patches using split and merge functions). In order to keep the sizes of the control polygons in the range $[S_{\min}, S_{\max}]$, the diameter of the convex hull of each control polygon is computed at each iteration. If a patch does not satisfy the size condition, it is either split into two patches or merged with a neighbor patch. The split and merge functions (see [20] for more details) are inverse operations and both use interpolation in order to compute the new control

⁴The first hypothesis is actually not restrictive (see details in Remark 3.2).

⁵Of course other situations of intersection can theoretically occur. However they are exceptional and very unlikely in practice. In this paper, since we are interested in a practical use of the flip procedure, we will only consider the two mentioned situations.

polygons (see Figure 8). The split function divides a control polygon into two. Precisely, it interpolates the first half of the patch and, in a second time, interpolates the other half. Since each half of the patch is a Bézier curve, the shape is not modified after a split. The merge function is the reverse operation. From two consecutive control polygons \mathbf{Q} and \mathbf{R} , it computes one patch that interpolates the four points $B(\mathbf{Q}, 0)$, $B(\mathbf{Q}, \frac{2}{3})$, $B(\mathbf{R}, \frac{1}{3})$ and $B(\mathbf{R}, 1)$. Then one starts from eight control points and ends with four. Note that merging polygons modifies slightly the boundary.

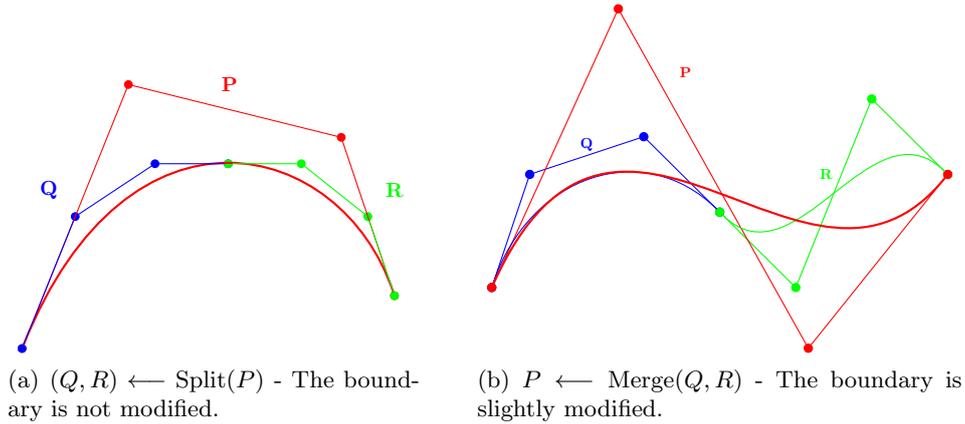


FIGURE 8. Examples of the split and merge functions.

3.3. Intersecting control polygons detection

Checking if each control polygon intersects another one may be very expensive in terms of computations. Axis-Aligned Bounding Boxes (AABBs) are a very common tool in Computer Graphics and Computational Geometry in order to detect the collision of two objects (see, e.g., [14]), with a relatively low computational cost. AABB is defined as the smallest rectangle, whose sides are aligned with the axes, containing the control polygon (see Figure 9).

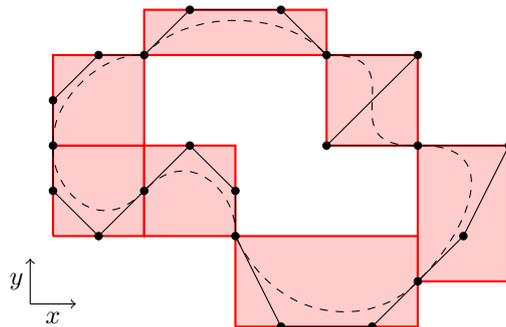


FIGURE 9. AABBs of control polygons.

A necessary condition for two intersecting control polygons is clearly the intersection of their respective AABBs. As a consequence, instead of looking directly for intersecting control polygons, we first look for intersecting AABBs. Thus, the intersecting control polygons detection consists in two steps:

- (1) we first list all the pairs of intersecting AABBs;

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

- (2) in a second time, we check these pairs in order to see if the associated control polygons intersect. To do so, we directly check the 9 segment-segment intersections of the polygons (see, e.g., [21]).

Finally, each pair of intersecting control polygons will be given as input to the flip procedure detailed in the following section.

3.4. The flip procedure

As above mentioned in Section 3.2, only two cases of intersecting control polygons are considered (see Figure 7). The flip procedure described in this section is a simple tool that can be easily implemented.

First case: two intersecting polygons. From $\mathbf{P} = \{P_0, P_1, P_2, P_3\}$ and $\mathbf{Q} = \{Q_0, Q_1, Q_2, Q_3\}$ being two intersecting polygons of a same connected component, the flip procedure builds two new polygons as follows, (see Figure 10):

$$\left\{ P_0, P_0 + \frac{1}{3}\overrightarrow{P_0Q_3}, P_0 + \frac{2}{3}\overrightarrow{P_0Q_3}, Q_3 \right\} \quad \text{and} \quad \left\{ Q_0, Q_0 + \frac{1}{3}\overrightarrow{Q_0P_3}, Q_0 + \frac{2}{3}\overrightarrow{Q_0P_3}, P_3 \right\}.$$

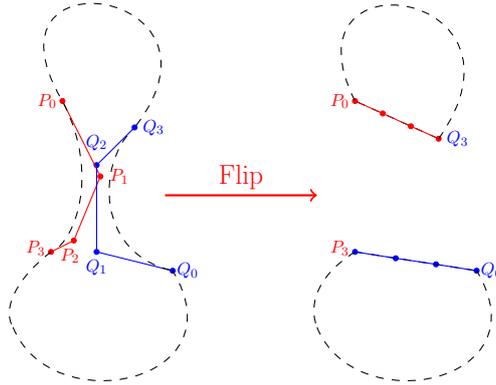


FIGURE 10. Flip procedure - Case of two control polygons.

Second case: three intersecting polygons. The case with three control polygons is very similar. From $\mathbf{P} = \{P_0, P_1, P_2, P_3\}$ being a control polygon intersecting two consecutive ones $\mathbf{Q} = \{Q_0, Q_1, Q_2, Q_3\}$ and $\mathbf{R} = \{R_0, R_1, R_2, R_3\}$, the flip procedure builds two new polygons as follows, (see Figure 11):

$$\left\{ P_0, P_0 + \frac{1}{3}\overrightarrow{P_0R_3}, P_0 + \frac{2}{3}\overrightarrow{P_0R_3}, R_3 \right\} \quad \text{and} \quad \left\{ Q_0, Q_0 + \frac{1}{3}\overrightarrow{Q_0P_3}, Q_0 + \frac{2}{3}\overrightarrow{Q_0P_3}, P_3 \right\}.$$

4. Application to multiple-inclusions detection

This section focuses on the problem of reconstructing numerically an obstacle ω_{ex} living in a larger bounded domain Ω of \mathbb{R}^2 from boundary measurements. Our aim is in particular to test the flip procedure introduced in this paper in the case where ω_{ex} is a two-components obstacle (see Section 4.3.4).

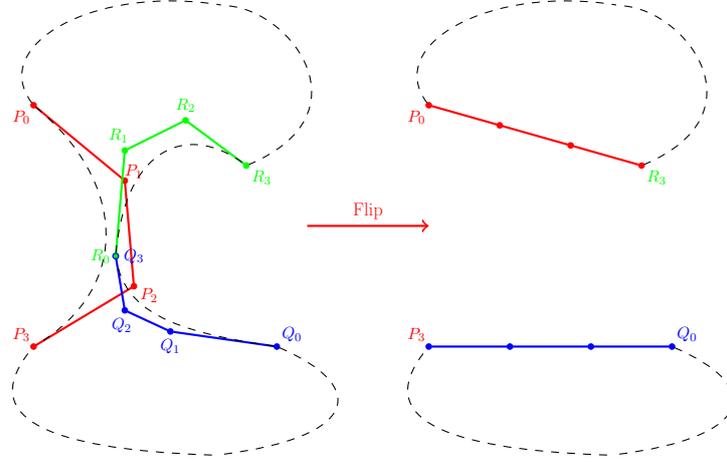


FIGURE 11. Flip procedure - Case of three control polygons.

In order to solve numerically the above inverse obstacle problem, we will actually consider a shape optimization problem, by minimizing a shape cost functional. In this paper we use the classical geometrical shape optimization approach, based on shape derivatives and on a shape gradient descent method. We refer to the classical books of Henrot *et al.* [17] and of Sokolowski *et al.* [26] for more details on the techniques of shape differentiability.

Let us fix some notations that will be used in this section. We denote by L^p , $W^{m,p}$ and H^s the usual Lebesgue and Sobolev spaces. We note in bold the vectorial functions and spaces, such as $\mathbf{W}^{m,p}$. Let Ω be a nonempty bounded connected and Lipschitz open set of \mathbb{R}^2 and let $g \in H^{1/2}(\partial\Omega)$ such that $g \neq 0$. We denote by \mathbf{n} the external unit normal to $\partial\Omega$, and for a smooth enough function u , we denote by $\partial_{\mathbf{n}}u$ the normal derivative of u .

Let $d_0 > 0$ be fixed (small). In the sequel \mathcal{O}_{d_0} stands for the set of all open subsets ω strictly included in Ω , with a $C^{1,1}$ boundary, such that the distance $d(x, \partial\Omega)$ from x to the compact $\partial\Omega$ is strictly greater than d_0 for all $x \in \omega$, and such that $\Omega \setminus \overline{\omega}$ is connected. Finally we also introduce Ω_{d_0} an open set with a C^∞ boundary such that

$$\{x \in \Omega; d(x, \partial\Omega) > d_0/2\} \subset \Omega_{d_0} \subset \{x \in \Omega; d(x, \partial\Omega) > d_0/3\}.$$

4.1. Problem setting

We focus on the following inverse problem. Assume that an unknown obstacle $\omega_{\text{ex}} \in \mathcal{O}_{d_0}$ is located inside Ω . We consider hereafter the Laplace equation in $\Omega \setminus \overline{\omega_{\text{ex}}}$ with homogeneous Dirichlet boundary condition on $\partial\omega_{\text{ex}}$ and non-homogeneous Dirichlet boundary condition on $\partial\Omega$. Precisely we denote by $u_{\text{ex}} \in H^1(\Omega \setminus \overline{\omega_{\text{ex}}})$ the unique solution of the problem

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \setminus \overline{\omega_{\text{ex}}}, \\ u = g & \text{on } \partial\Omega, \\ u = 0 & \text{on } \partial\omega_{\text{ex}}. \end{cases} \quad (4.1)$$

Our main purpose is to reconstruct the unknown shape ω_{ex} , assuming that a measurement is done on the exterior boundary $\partial\Omega$. Precisely we assume in this paper that we know exactly the value of the measure $f_b := \partial_{\mathbf{n}}u_{\text{ex}} \in H^{-1/2}(\partial\Omega)$ on $\partial\Omega$. Thus we are interested in the following geometric inverse

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

problem:

find $\omega \in \mathcal{O}_{d_0}$ and $u \in H^1(\Omega \setminus \bar{\omega})$ which satisfies the overdetermined system

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \setminus \bar{\omega}, \\ u = g & \text{on } \partial\Omega, \\ \partial_{\mathbf{n}} u = f_b & \text{on } \partial\Omega, \\ u = 0 & \text{on } \partial\omega. \end{cases} \quad (4.2)$$

The existence of a solution is trivial since we assume that the measurement f_b is exact. From the classical Holmgren theorem one can easily obtain an identifiability result for this inverse problem which claims that the solution is unique. This fundamental question about uniqueness of a solution to the overdetermined problem (4.2) was deeply studied, see for example [5, Theorem 1.1], [12, Theorem 5.1] or also [13, Prop. 4.4 p. 87].

Remark 4.1. Actually we could assume that the measurement f_b is done only on a nonempty subset O of $\partial\Omega$. All the presented result can be adapted to this case (see, e.g., [8]).

In order to solve the inverse problem (4.2) we will actually focus on the shape optimization problem

$$\omega^* \in \underset{\omega \in \mathcal{O}_{d_0}}{\operatorname{argmin}} J(\omega), \quad (4.3)$$

where J is the nonnegative least-squares functional defined by

$$J(\omega) := \int_{\partial\Omega} |\partial_{\mathbf{n}} u_\omega - f_b|^2,$$

where $u_\omega \in H^1(\Omega \setminus \bar{\omega})$ is the unique solution of the problem

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \setminus \bar{\omega}, \\ u = g & \text{on } \partial\Omega, \\ u = 0 & \text{on } \partial\omega. \end{cases} \quad (4.4)$$

Indeed, the identifiability result ensures that $J(\omega) = 0$ if and only if $\omega = \omega_{\text{ex}}$. Finally, in order to solve numerically the shape optimization problem (4.3), we will now compute the shape gradient of the cost functional J and apply a classical gradient descent method.

4.2. Computation of the shape gradient.

In order to define shape derivatives, we will use the *velocity method* introduced by Murat *et al.* in [19]. We first introduce the space of admissible deformations given by

$$\mathbf{U} := \{\mathbf{V} \in \mathbf{W}^{2,\infty}; \operatorname{Supp} \mathbf{V} \subset \overline{\Omega_{d_0}}\}. \quad (4.5)$$

In particular we are interested in the shape gradient of J defined by

$$DJ(\omega) \cdot \mathbf{V} := \lim_{t \rightarrow 0} \frac{J((\mathbf{I} + t\mathbf{V})(\omega)) - J(\omega)}{t},$$

for every $\omega \in \mathcal{O}_{d_0}$ and every $\mathbf{V} \in \mathbf{U}$. For sake of completeness, we recall the proof of the following result in Appendix A.

Proposition 4.2. *The least-squares functional J is differentiable at $\omega \in \mathcal{O}_{d_0}$ in the direction $\mathbf{V} \in \mathbf{U}$ with*

$$DJ(\omega) \cdot \mathbf{V} = - \int_{\partial\omega} \partial_{\mathbf{n}} u_\omega \partial_{\mathbf{n}} w_\omega (\mathbf{V} \cdot \mathbf{n}), \quad (4.6)$$

where $w_\omega \in H^1(\Omega \setminus \bar{\omega})$ is the unique solution of the adjoint problem given by

$$\begin{cases} -\Delta w = 0 & \text{in } \Omega \setminus \bar{\omega}, \\ w = 2(\partial_{\mathbf{n}} u_\omega - f_b) & \text{on } \partial\Omega, \\ w = 0 & \text{on } \partial\omega. \end{cases} \quad (4.7)$$

From the above explicit formulation of the shape gradient of J , we are now in a position to implement some numerical simulations based on a classical gradient descent method, including the flip procedure introduced in this paper in order to detect in particular a multiple-components obstacle.

4.3. Numerical simulations

Before coming to numerical simulations, let us recall that many difficulties can be encountered in order to solve numerically Problem (4.3), as explained in [1, Theorem 1] (see also [3, Proposition 2.4]). Indeed, the gradient has not a uniform sensitivity with respect to the deformation directions. However, we use in this paper a parametric model of shape variations using piecewise Bézier curves which corresponds to a regularization method (as the truncated Fourier series used in [1]) allowing to overcome the ill-posedness of the inverse problem and then to solve it numerically.

4.3.1. Framework for the numerical simulations

The numerical simulations presented hereafter are performed in the two-dimensional case using the finite element library *FreeFem++* (see [16]). The exterior boundary $\partial\Omega$ is assumed to be the circle centered in the origin and of radius 10 and we consider the exterior Dirichlet boundary condition $g = 100$. In order to get a suitable measure f_b , we use a *synthetic data*, that is, we fix a shape ω_{ex} and solve Problem (4.1) using a finite element method (here P2 finite element discretization) and extract the measurement f_b by computing $\partial_{\mathbf{n}} u_{\text{ex}}$ on $\partial\Omega$.

Then we use a P1 finite element discretization to solve Problems (4.4) and (4.7) with 50 discretization points for both the exterior boundary and each cubic Bézier patch describing the shape ω . In order to numerically solve the optimization problem (4.3), we use the following classical gradient descent algorithm and we include the flip procedure at Step (3).

Algorithm A

- (1) Fix $k = 0$, fix an initial shape ω_0 , fix a maximal number $M \in \mathbb{N}^*$ of iterations and fix $\lambda \geq 1$ a given tolerance coefficient for the flip procedure (see Step (3), λ should be chosen close to 1).
- (2) Control the size of the patches of ω_k (see Remark 3.2).
- (3) Scan ω_k looking for intersecting control polygons (see Section 3.3):
 - (a) in the case of no intersecting control polygons, go to Step (4);
 - (b) in the case of intersecting control polygons:
 - (i) apply the flip procedure and obtain a multiple-components shape $\omega_k^1 \cup \omega_k^2$ (see Section 3.4);
 - (ii) compute $J(\omega_k)$ and $J(\omega_k^1 \cup \omega_k^2)$:
 - (A) if $J(\omega_k^1 \cup \omega_k^2) \geq \lambda J(\omega_k)$, then go to Step (4);
 - (B) else, do $\omega_k \leftarrow \omega_k^1 \cup \omega_k^2$.
- (4) Solve Problems (4.4) and (4.7) with $\omega = \omega_k$.
- (5) Compute the shape gradient $DJ(\omega_k)$ from Formula (4.6).

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

- (6) Move the control points of the shape, that is, do $\omega_{k+1} \leftarrow \omega_k - \alpha DJ(\omega_k)$, where α is a positive coefficient chosen by a classical line search.
- (7) Do $k \leftarrow k + 1$ and get back to Step (2) while $k < M$.

4.3.2. First simulations: detection of smooth and convex shapes

We first test Algorithm \mathcal{A} on the problem of detecting one smooth convex object. Precisely, we begin by detecting the circle centered at the origin and of radius 6 and the ellipse $\{(8 \cos \theta, 5 \sin \theta), \theta \in [0, 2\pi]\}$ using four cubic Bézier patches. Figure 12 shows the effectiveness of our method for these objects.

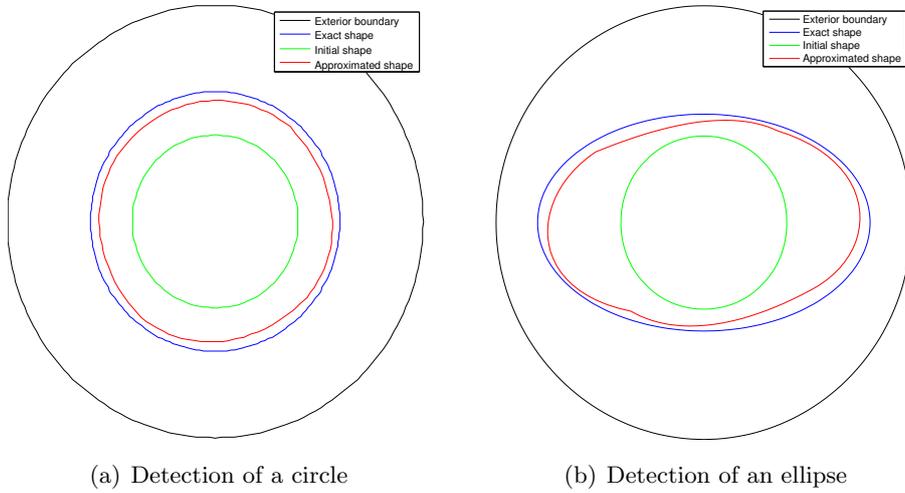


FIGURE 12. Detection of simple, convex and smooth obstacles.

4.3.3. Detection of a non-smooth shape and of a non-convex shape

We test now the effectiveness of Algorithm \mathcal{A} on the problem of detecting a non-smooth shape and of detecting a non-convex shape (see Figure 13). Precisely we first consider the square of side 10 and centered at the origin and we use four cubic Bézier patches. As one can see in Figure 13(a), each Bézier patch detects a side of the square. Secondly, in Figure 13(b), we consider the non-convex shape parameterized by $\{(2.8(1.6 + \cos(3\theta)) \cos(\theta), 2.8(1.6 + \cos(3\theta)) \cos(\theta)), \theta \in [0, 2\pi]\}$, using six cubic Bézier patches.⁶ One can see that the reconstruction is here effective.

4.3.4. Detection of two obstacles starting from a one-component shape

In this section we test the flip procedure introduced in Section 3 in order to detect a two-components shape starting from a one-component initial shape. We consider two circles of radius 2 centered at $(-4, -4)$ and $(4, 4)$. We present different states of the algorithm in Figure 14. The initial Bézier shape consists in a single component with four cubic Bézier patches, located at the center (Figure 14(a)).

⁶This shape is also considered in [10, Figure 4] where authors obtained the convex hull of the shape. However, note that the authors used a different method where the descent direction is obtained by solving a boundary value problem involving the kernel of the shape gradient.

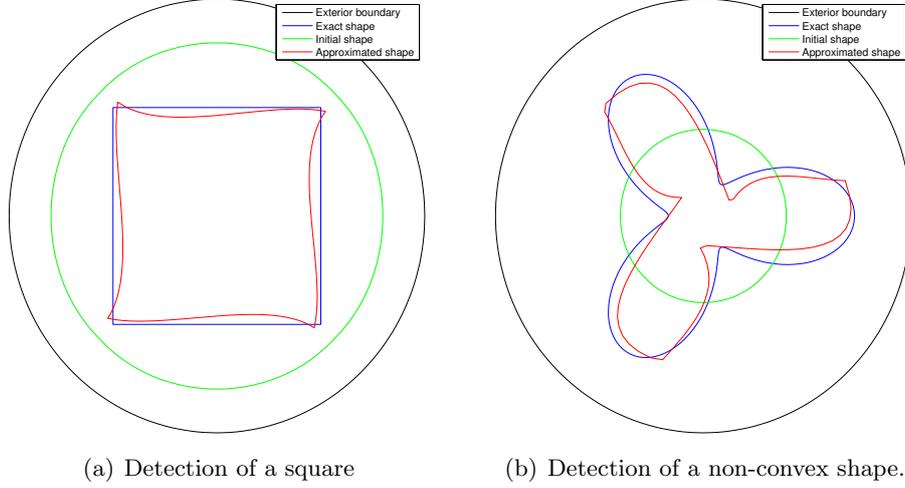


FIGURE 13. Detection of a non-smooth obstacle and of a non-convex obstacle.

The shape grows and surrounds the two objects until two control polygons intersect each other (Figure 14(b)). The flip procedure is performed and the shape is divided in two connected components (Figure 14(c)). At the end, the algorithm has detected the two obstacles (Figure 14(d)).

4.3.5. Checking the objective function value after a flip procedure

In Algorithm \mathcal{A} , Step (3(b)iiA) makes sure that, whenever a flip procedure is performed, the objective function value does not significantly increase. If $J(\omega_k^1 \cup \omega_k^2) \geq \lambda J(\omega_k)$ (for instance $\lambda = 1.1$), then we consider that adding another component to the shape is not a wise choice and we cancel the flip procedure. This situation can emerge when the target shape has a single component with two parts of its boundary very close to each other. In such a case Algorithm \mathcal{A} probably leads to two control polygons intersecting each other and to a flip performance, while the target shape has a single component. We present an example of such a situation in Figure 15. One can see in blue the obstacle composed by one component that has two parts of its boundary very close to each other. The current shape ω_k of the algorithm in red has two control polygons intersecting each other. The objective function value before the flip procedure is $J(\omega_k) = 1061$ and after the flip procedure, it has increased to $J(\omega_k^1 \cup \omega_k^2) = 1574$. Since the ratio is greater than λ , the algorithm cancels the flip procedure and go to Step (4).

4.3.6. Detection of one obstacle starting from a two-components shape

In this paper we have introduced the flip procedure as a method that enables to divide a one-component shape into a two-components shape. Actually the flip procedure can be easily adapted in order to perform the reverse operation, that is, to merge a two-components shape into a one-component shape (see Figure 16).

We focus now on the detection of the one-component shape $\{(4 \cos \theta, 6 + 2.5 \sin \theta), \theta \in [0, 2\pi]\}$ and we start Algorithm \mathcal{A} with a two-components shape. We present different states of the algorithm in Figure 17. At the end, the algorithm has detected the one-component obstacle.

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

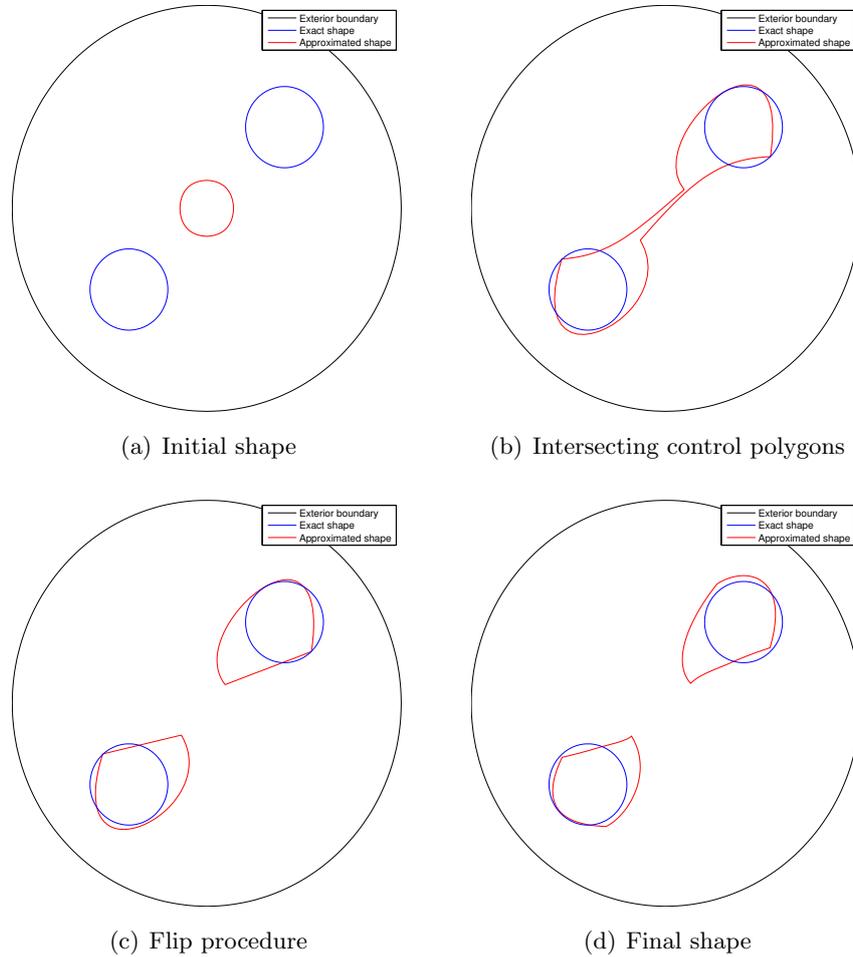


FIGURE 14. Detection of two obstacles starting from a one-component shape

5. Conclusion and perspectives

In this paper we studied the use of piecewise Bézier parameterizations for the representation of shapes in geometric approximation based on successive shape deformations. We proposed procedures and algorithms in order to manipulate these parameterizations and showed how to manage changes of topology and so multiple-components shape approximation. We applied this approach to a problem of multiple-inclusions detection and showed numerically its efficiency. We described results obtained from an experimental implementation using *FreeFem++* and showed the good behavior of the algorithm. The computational efficiency of the method arises from the simplicity and the flexibility of the proposed parameterizations. We considered here a two-dimensional problem, but extension to the higher dimensional case may be interesting and the algorithmic contents can be generalized. To conclude, the implementation described here was made for an experimental purpose and a complete and optimized implementation may be proposed. Applications to other shape detection problems may be studied also.

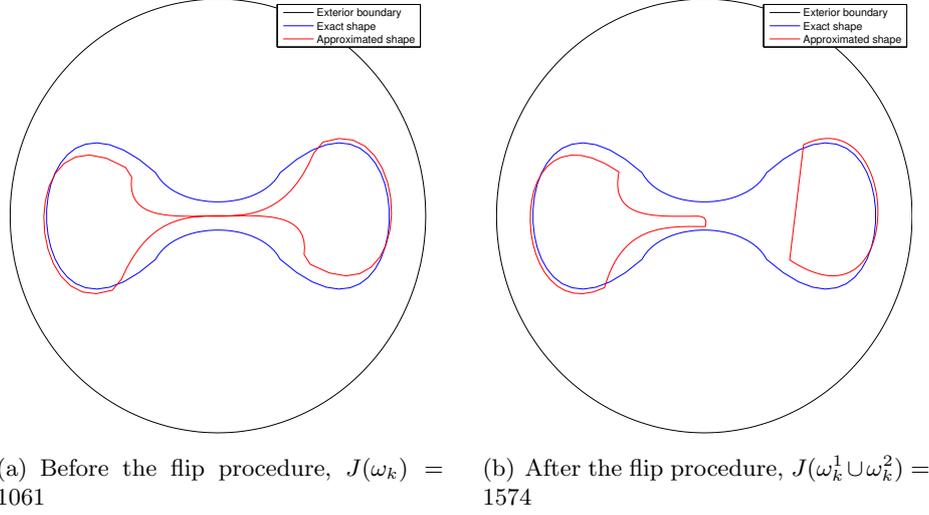


FIGURE 15. The objective function value significantly increases whenever the flip procedure is not needed.

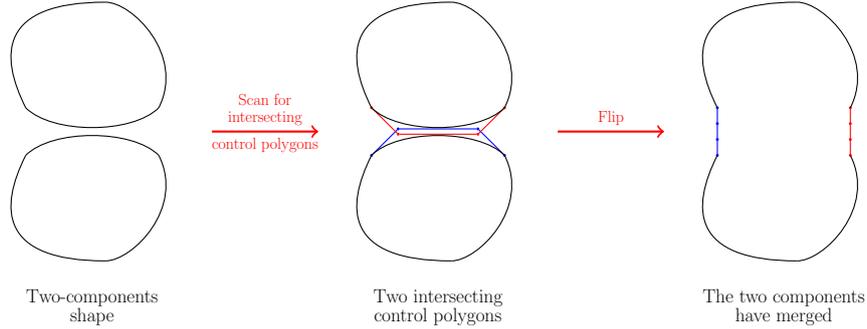


FIGURE 16. The flip procedure merges a two-components shape into a one-component shape.

Appendix A. Proof of Proposition 4.2

We detail here the classical proof of Proposition 4.2 for the reader's convenience. For any $\mathbf{V} \in \mathbf{U}$ (where \mathbf{U} is defined by (4.5)), we introduce the perturbed domain $\omega_t := (\mathbf{I} + t\mathbf{V})(\omega)$ and the functional j defined for all $t \in [0, T)$ by $j(t) := J(\omega_t)$ and we consider the unique solution $u_t \in H^1(\Omega \setminus \overline{\omega_t})$ of the perturbed problem

$$\begin{cases} -\Delta u_t = 0 & \text{in } \Omega \setminus \overline{\omega_t}, \\ u_t = f & \text{on } \partial\Omega, \\ u_t = 0 & \text{on } \partial\omega_t. \end{cases}$$

Let us recall the definition of the shape derivative in our situation (see [17] for details). We introduce

$$\mathbf{u} := \left\{ \boldsymbol{\theta} \in \mathbf{U}; \|\boldsymbol{\theta}\|_{2,\infty} < \min\left(\frac{d_0}{3}, 1\right) \right\}.$$

Then,

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

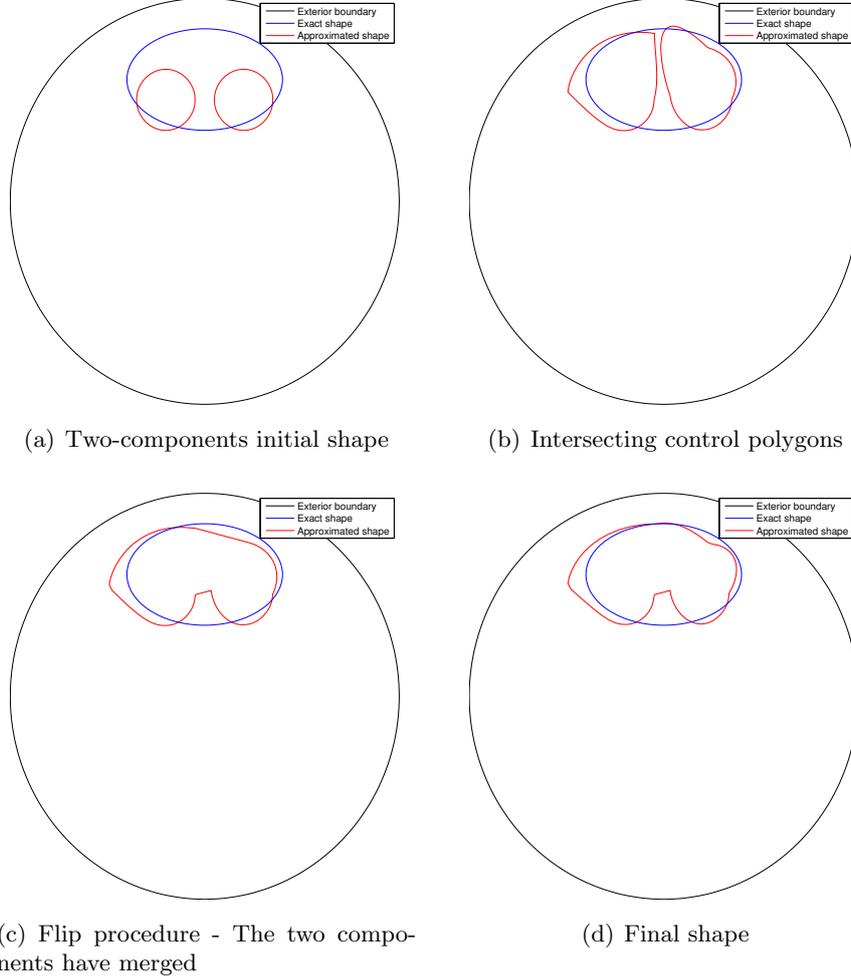


FIGURE 17. Detection of one obstacle starting from a two-components shape

- if the mapping $\boldsymbol{\theta} \in \mathcal{U} \mapsto u_{\boldsymbol{\theta}} \circ (\mathbf{I} + \boldsymbol{\theta}) \in \mathbf{H}^1(\Omega \setminus \bar{\omega})$ is Fréchet differentiable at $\mathbf{0}$, we say that $\boldsymbol{\theta} \mapsto u_{\boldsymbol{\theta}}$ possesses a *total first variation* (or derivative) at $\mathbf{0}$. In such a case, this total first derivative at $\mathbf{0}$ in the direction $\boldsymbol{\theta}$ is denoted by $\dot{u}_{\boldsymbol{\theta}}$;
- if, for every $\mathcal{D} \subset\subset \Omega \setminus \bar{\omega}$, the mapping $\boldsymbol{\theta} \in \mathcal{U} \mapsto u_{\boldsymbol{\theta}|_{\mathcal{D}}} \in \mathbf{H}^1(\mathcal{D})$ is Fréchet differentiable at $\mathbf{0}$, we say that $\boldsymbol{\theta} \mapsto u_{\boldsymbol{\theta}}$ possesses a *local first variation* (or derivative) at $\mathbf{0}$. In such a case, this local first derivative at $\mathbf{0}$ in the direction $\boldsymbol{\theta}$ is denoted by $u'_{\boldsymbol{\theta}}$, is called *shape derivative* and is well defined in the whole domain $\Omega \setminus \bar{\omega}$:

$$u'_{\boldsymbol{\theta}} = \frac{d}{dt}(u_t \boldsymbol{\theta}|_{\mathcal{D}})|_{t=0} \quad \text{in each } \mathcal{D} \subset\subset \Omega \setminus \bar{\omega}.$$

In the sequel, let $\mathbf{V} \in \mathcal{U}$ and let u' be the local first variation $u'_{\mathbf{V}}$ which is referred as the shape derivative of the state.

The differentiability of the cost functional J is directly obtained from the existence of the shape derivative of the state u given for example in [17, Theorem 5.3.1]. Notice that in [17, Theorem 5.3.1], the result claims the differentiability of $t \in [0, T] \mapsto \tilde{u}_t \in \mathbf{L}^2(\Omega)$, where \tilde{u}_t is an extension of u_t in Ω . Since we want to obtain the differentiability of $t \in [0, T] \mapsto \tilde{u}_t \in \mathbf{H}^1(\Omega)$ (in order to differentiate

properly the functional J), we have here to work with the mentioned spaces, that is with a domain ω with a $C^{1,1}$ boundary (and not only Lipschitz) and perturbations \mathbf{V} which belong to $\mathbf{W}^{2,\infty}(\mathbb{R}^2)$ (and not only to $\mathbf{W}^{1,\infty}(\mathbb{R}^2)$).

Moreover we can easily characterize the shape derivative $u' \in H^1(\Omega \setminus \bar{\omega})$ as the solution of the following problem (see again for example [17, Theorem 5.3.1]):

$$\begin{cases} -\Delta u' = 0 & \text{in } \Omega \setminus \bar{\omega}, \\ u' = 0 & \text{on } \partial\Omega, \\ u' = -\partial_{\mathbf{n}}u(\mathbf{V} \cdot \mathbf{n}) & \text{on } \partial\omega. \end{cases} \quad (\text{A.1})$$

Then by differentiation under the sum sign, we obtain

$$j'(0) = 2 \int_{\partial\Omega} \partial_{\mathbf{n}}u'(\partial_{\mathbf{n}}u - f_b).$$

Using the weak formulation of Problem (A.1) solved by u' with w as a test function, we obtain

$$\int_{\Omega \setminus \bar{\omega}} \nabla u' \cdot \nabla w - \int_{\partial(\Omega \setminus \bar{\omega})} w \partial_{\mathbf{n}}u' = 0$$

and using the weak formulation of the adjoint Problem (4.7) solved by w with u' as a test function, we obtain

$$\int_{\Omega \setminus \bar{\omega}} \nabla w \cdot \nabla u' - \int_{\partial(\Omega \setminus \bar{\omega})} u' \partial_{\mathbf{n}}w = 0.$$

Finally, using the boundary conditions, the proof is complete.

Bibliography

- [1] L. Afraites, M. Dambrine, K. Eppler, and D. Kateb. Detecting perfectly insulated obstacles by shape optimization techniques of order two. *Discrete Contin. Dyn. Syst. Ser. B*, 8(2):389–416 (electronic), 2007.
- [2] G. Allaire, F. de Gournay, F. Jouve, and A.-M. Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control Cybernet.*, 34(1):59–80, 2005.
- [3] M. Badra, F. Caubet, and M. Dambrine. Detecting an obstacle immersed in a fluid by shape optimization methods. *Math. Models Methods Appl. Sci.*, 21(10):2069–2101, 2011.
- [4] E. Bishop. A generalization of the Stone-Weierstrass theorem. *Pacific J. Math.*, 11(3):777–783, 1961.
- [5] L. Bourgeois and J. Dardé. A quasi-reversibility approach to solve the inverse obstacle problem. *Inverse Probl. Imaging*, 4(3):351–377, 2010.
- [6] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, 194(1):344–362, 2004.
- [7] M. Burger and S. J. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16(2):263–301, 2005.
- [8] F. Caubet. Instability of an inverse problem for the stationary Navier-Stokes equations. *SIAM J. Control Optim.*, 51(4):2949–2975, 2013.
- [9] F. Caubet, C. Conca, and M. Godoy. On the detection of several obstacles in 2D Stokes flow: topological sensitivity and combination with shape derivatives. Sept. 2015.

FLIP PROCEDURE IN APPROXIMATION OF MULTIPLE-COMPONENTS SHAPES

- [10] F. Caubet, M. Dambrine, and D. Kateb. Shape optimization methods for the inverse obstacle problem with generalized impedance boundary conditions. *Inverse Problems*, 29(11):115011, 26, 2013.
- [11] F. Caubet, M. Dambrine, D. Kateb, and C. Z. Timimoun. A Kohn-Vogelius formulation to detect an obstacle immersed in a fluid. *Inverse Probl. Imaging*, 7(1):123–157, 2013.
- [12] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, second edition, 1998.
- [13] J. Dardé. *Quasi-reversibility and level set methods applied to elliptic inverse problems*. Theses, Université Paris-Diderot - Paris VII, Dec. 2010.
- [14] C. Ericson. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology) (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [15] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.
- [16] F. Hecht. Finite Element Library FREEFEM++. <http://www.freefem.org/ff++/>.
- [17] A. Henrot and M. Pierre. *Variation et optimisation de formes*, volume 48 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer, Berlin, 2005. Une analyse géométrique. [A geometric analysis].
- [18] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Y. Jr. Gradient flows and geometric active contour models, 1994.
- [19] F. Murat and J. Simon. *Sur le contrôle par un domaine géométrique*. Rapport du L.A. 189, 1976. Université de Paris VI, France.
- [20] O. R. O. Labbani-I., P. Merveilleux-O. Free form based active contours for image segmentation and free space perception.
- [21] J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, New York, NY, USA, 2nd edition, 1998.
- [22] O. Pantz and K. Trabelsi. Simultaneous shape, topology, and homogenized properties optimization. *Struct. Multidiscip. Optim.*, 34(4):361–365, 2007.
- [23] A. Schumacher. Topologieoptimisierung von Bauteilstrukturen unter Verwendung von Lopchpositionierungskriterien. *Thesis*, 1995. Universitt-Gesamthochschule-Siegen.
- [24] T. W. Sederberg. Computer aided geometric design, October 2014.
- [25] J. Sokołowski and A. Żochowski. On the topological derivative in shape optimization. *SIAM J. Control Optim.*, 37(4):1251–1272 (electronic), 1999.
- [26] J. Sokołowski and J.-P. Zolésio. *Introduction to shape optimization*, volume 16 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.