Control and Cybernetics

vol. **34** (2005) No. 1

Fast level set based algorithms using shape and topological sensitivity information

by

M. Hintermüller

Department of Computational and Applied Mathematics,
Rice University
Houston, Texas, USA

Abstract: A framework for descent algorithms using shape as well as topological sensitivity information is introduced. The concept of gradient-related descent velocities in shape optimization is defined, a corresponding algorithmic approach is developed, and a convergence analysis is provided. It is shown that for a particular choice of the bilinear form involved in the definition of gradient-related directions a shape Newton method can be obtain. The level set methodology is used for representing and updating the geometry during the iterations. In order to include topological changes in addition to merging and splitting of existing geometries, a descent algorithm based on topological sensitivity is proposed. The overall method utilizes the shape sensitivity and topological sensitivity based methods in a serial fashion. Finally, numerical results are presented.

Keywords: descent algorithm, level set method, Newton method, sensitivity analysis, shape optimization, topology optimization.

1. Introduction

In this paper we are interested in devising efficient numerical algorithms for solving the *shape optimization problem*

$$\text{minimize } J(\omega) \quad \text{over } \omega \in \mathcal{D}, \tag{1}$$

where $\mathcal{D}$ denotes the set of admissible shapes in $\mathbf{R}^m$, $m \in \mathbf{N}$. Whenever ω denotes a domain in $\mathbf{R}^m$, we write $\omega = \Omega$. If ω represents a lower dimensional, typically $m-1$ dimensional, manifold we write $\omega = \Gamma$. In the latter case we refer to J as a *boundary shape functional*, while in the former case we call it a *domain shape functional.* In some applications, like for instance in edge detector based image segmentation, as in Caselles et al. (1993), or Yezzi et al. (1997), J is written as the sum of a domain and a boundary shape functional.

In subsequent sections we present shape gradient, shape gradient-related and shape Newton-type algorithms. All these algorithms require sensitivity information of J with respect to perturbations of ω. By now, there is a considerable number of papers and monographs available that deal with shape sensitivity issues related to functionals such as J. Here we only cite the two recent monographs by Delfour and Zolésio (2001), and Sokolowski and Zolésio (1992), and the many references therein. Modern concepts rely on the *speed method*; see, e.g., Section 2.9 in Sokolowski and Zolésio (1992). We give a brief account of the method in Section 2, which is then used for computing the *shape gradient* $j(\omega)$ and the *shape Hessian* $H(\omega)$ of J at ω. The famous structure theorem (Theorem 2.27 in Sokolowski and Zolésio, 1992, or Theorem 3.5 of Chapter 8 in Delfour and Zolésio, 2001) states that $j(\Omega)$ has its support in $\Gamma = \partial\Omega$. An analogous assertion is true for $H(\Omega)$ as well. For the shape Hessian based on the speed method it is well-known that it is in general a non-symmetric expression involving a Lie bracket; see Delfour and Zolésio (2001), p. 373. Symmetry is achieved by imposing additional conditions on the perturbations of ω. Positive-definite symmetric forms representing the shape Hessian or approximations thereof are essential in devising descent algorithms for minimizing J. Further, interesting relations between second and first variations of shape functionals and an outlook to shape gradient-descent and shape Newton-type iterations can be found in Simon (1989).

Descent algorithms are iterative methods which, based on a current approximation ω_k of an optimal solution of (1), compute a search direction $d(\omega_k)$ such that

$$\langle d(\omega_k), j(\omega_k)\rangle_\Gamma < 0, \tag{2}$$

where $\langle\cdot,\cdot\rangle_\Gamma$ denotes a suitable pairing. The subscript Γ relates to the fact that, by the structure theorem, the shape gradient has its support in an $(m-1)$-dimensional set. Then, this direction is used to update ω_k such that $J(\omega_{k+1}) < J(\omega_k)$. The reduction in J has to be sufficiently large such that the method achieves in the limit a *stationary point* ω^*, i.e., a shape satisfying $j(\omega^*) = 0$.

One of the key aspects of numerical methods which iteratively update the geometry is related to the transport of geometry from one iteration to the next. Marker or particle techniques discretize the boundary of the geometry and evolve these boundary points with a prescribed velocity in the direction normal to the boundary. As the iterations proceed these boundary points (markers) may cluster which eventually degrades the numerical performance. This problem is typically circumvented by redistributing the points, if necessary, from time to time. Other techniques based on, e.g., parameterizations or particular discretizations of the boundary may suffer from similar effects. Also, changes of the topology require expensive reparameterizations. Another common feature is given by the Lagrangian nature of these approaches. Rather than operating on a fixed reference frame, these techniques follow the particles along their trajectories.

In Osher and Sethian (1988) a concept was introduced which operates on a fixed reference frame, or Cartesian grid in the discretized setting, and represents the geometry of interest by means of the zero-level set of a so-called *geometrical implicit function* ϕ. This fact coined the name *level set method*. It is well-known that the level set method is a numerically robust technique which can easily handle topological changes of the domain, such as merging or splitting. To describe some of its key features, let us assume that $\omega(t) = \Omega(t)$, or otherwise it represents a closed curve $\Gamma(t) = \partial\Omega(t)$. A typical requirement in the definition of the continuous function $\phi = \phi(x,t)$ is

$$\Omega(t) = \{x : \phi(x,t) < 0\} \quad \text{and} \quad \mathbf{R}^m \setminus \overline{\Omega(t)} = \{x : \phi(x,t) > 0\}.$$

The transport of the geometry is achieved by advecting the *level-set function* $\phi = \phi(x,t)$ by

$$\frac{\partial \phi}{\partial t}(x,t) + F(x,t)|\nabla\phi(x,t)|_2 = 0, \quad \phi(x,0) = \phi_0(x), \tag{3}$$

where $|\cdot|_2$ denotes the Euclidian norm of a vector in $\mathbf{R}^m$. The equation in (3) is called the *level set equation*. By applying the chain rule, it can be easily derived from the requirement that $\Gamma(t)$ remains the zero-level set of ϕ for all times t. The scalar-valued function F in (3) denotes the velocity along the outward unit normal $n(x,t) = \nabla\phi(x,t)/|\nabla\phi(x,t)|_2$. In our case, it is related to the shape gradient or Newton-type direction.

The combination of level set methods and shape sensitivity was considered in a variety of applications; see Allaire, Jouve and Toader (2004), Burger (2004), Dorn, Miller and Rappaport (2000), Hintermüller and Ring (2004), Litman, Lesselier and Santosa (1998), Osher and Santosa (2001), and Santosa (1996). For a more general functional-analytic setting we refer to Burger (2003). Our goal in this paper is to develop a unified framework based on descent methods.

Though robust and flexible, one major drawback of level-set-type algorithms based on shape sensitivity information is the lack of creating topological changes other than merging or splitting of existing components of ω. Note that in a level set based approach using shape sensitivity, components of ω may even vanish, but new ones cannot be created (except for splitting). As a remedy, we propose to consider topological sensitivity information in addition to shape sensitivity information.

In a pioneering work, Eschenauer and Schumacher (1994) introduced the so-called *bubble method* for topology and shape optimization of structures. In Garreau, Guillaume and Masmoudi (2001), or Sokolowski and Żochowski (1999) the *topological asymptotic* and the *topological gradient* are introduced, respectively. Roughly, these results describe the sensitivity of J with respect to creating a hole in Ω. In an algorithmic framework one may now use velocities based on topological sensitivity for updating Ω. This is done again in a time-evolution scheme slightly different to (3). We refer to Burger, Hackl and Ring (2004) for some preliminary numerical experience. In the present paper we combine shape and topological sensitivities in a serial fashion. Hence, our algorithm starts with a topology optimization phase which is then followed by a phase using solely shape sensitivities. If necessary, this cycle is repeated until some suitable stopping rule is satisfied.

The remaining sections are organized as follows. In Section 2 we collect some basic results from shape and topological sensitivity which are needed for our algorithms. The definition of shape gradient-related search directions and a corresponding descent algorithm involving an Armijo-type line search procedure are the subject of Section 3. We also provide a convergence result and discuss modifications for obtaining a shape Newton method. The subsequent Section 4 is concerned with a practical procedure for updating geometry from one iteration to the next within a gradient-related descent method. We propose a technique based on the level set method. In Section 5 we devise a descent algorithm utilizing the topological gradient. Finally, in Section 6 we report on an excerpt of numerical test runs.

2. Basics in shape and topological sensitivity

2.1. Shape sensitivity

Let $V = V(x)$ denote a sufficiently smooth admissible vector field with values in $\mathbf{R}^m$. For the conditions of admissibility we refer to Sokolowski and Zolésio (1992). Also note that we focus here on the autonomous case, i.e., the case where V does not depend on t. This vector field is used for defining suitable perturbations of $\omega \subset D$, where the bounded set $D \subset \mathbf{R}^m$ defines the "hold-all" domain. In fact, let $T_t(V)(X) = x(t)$ denote the solution to the differential equation

$$\begin{cases} \frac{dx}{dt}(t) = V(x(t)), \quad 0 < t < \tau, \\ x(0) = X \in \mathbf{R}^m \end{cases}$$

for $\tau > 0$ sufficiently small. Then the mapping T_t allows to define

$$\omega(t)(V) = T_t(V)(\omega) = \{T_t(V)(X) : \forall X \in \omega\} \subset D$$

with $\omega(0)(V) = \omega$. The *Eulerian semiderivative* of J at ω in direction V is defined as

$$dJ(\omega; V) := \lim_{t \downarrow 0} \frac{J(\omega(t)(V)) - J(\omega)}{t} \tag{4}$$

if the limit in (4) exists and is finite. The shape function J is said to be *shape differentiable* at ω if $dJ(\omega; V)$ exists for all V, and the mapping $V \mapsto dJ(\omega; V)$ is linear and continuous. If $\omega = \Omega$ with $\Gamma = \partial\Omega$ and V are sufficiently smooth, then, by the structure theorem, there exist $k \geq 0$, and a scalar distribution $j(\Omega) \in C^k(\Gamma)'$ such that

$$dJ(\Omega; V) = \langle j(\Omega), \gamma_\Gamma(V) \cdot n \rangle_{C^k(\Gamma)', C^k(\Gamma)} \quad \text{for all } V \in \mathcal{D}^k(\mathbf{R}^m, \mathbf{R}^m). \tag{5}$$

In the above, $\mathcal{D}^k(\mathbf{R}^m, \mathbf{R}^m)$ denotes the k-times continuously differentiable functions with compact support in Ω, γ_Γ is the trace operator on Γ, and n is the outward unit normal to Γ. The distribution $j(\omega)$ is referred to as the *shape gradient* of J at ω. We remark that (5) may also be related to the Lebesgue respectively Sobolev space settings.

EXAMPLE 2.1 *In Hintermüller and Ring (2003) the shape gradient and the shape Hessian for the shape functional*

$$J(\Gamma) = \int_\Gamma g dS + \nu \int_\Omega g dx \tag{6}$$

were considered. Here, $\nu > 0$ is fixed and g denotes a sufficiently smooth and non-negative function which is independent of $\Gamma = \partial\Omega$. The Eulerian semiderivative of J at Γ is given by

$$dJ(\Gamma; V) = \int_\Gamma \left\langle \left(\frac{\partial g}{\partial n} + g\,(\kappa + \nu) \right), V|_\Gamma \cdot n \right\rangle dS,$$

where κ denotes the mean curvature of Γ; see Sokolowski and Zolésio (1992). From this we can identify the shape gradient by

$$j(\Gamma) = \left(\frac{\partial g}{\partial n} + g\,(\kappa + \nu) \right).$$

In analogy to (4) one defines the *second-order Eulerian semiderivative* of J at ω in direction (V, W). In fact, let V and W be admissible vector fields and assume that for all $t \in [0, \tau]$, $\tau > 0$ sufficiently small, $dJ(\omega(t)(W); V)$ exists. Then J is said to have a second-order Eulerian semiderivative at ω in direction (V, W), if

$$d^2 J(\omega; V; W) = \lim_{t \downarrow 0} \frac{dJ(\omega(t)(W); V) - dJ(\omega; V)}{t} \tag{7}$$

exists. Note that (4) and (7) together imply that the following expansion of $J(\omega(t)(V))$ makes sense:

$$J(\omega(t)(V)) = J(\omega) + t dJ(\omega; V) + \frac{t^2}{2} d^2 J(\omega; V; V) + \mathcal{O}(t^2). \tag{8}$$

The function J is said to be *twice shape differentiable* at ω, if $d^2 J(\omega; V; W)$ exists for all admissible V and W, and the map $(V; W) \mapsto d^2 J(\omega; V; W)$ is bilinear and continuous. The latter bilinear form is denoted by h. Let $H(\omega) \in (\mathcal{D}(\mathbf{R}^m, \mathbf{R}^m) \otimes \mathcal{D}(\mathbf{R}^m, \mathbf{R}^m))'$ denote the vector distribution associated with h, i.e.,

$$d^2 J(\omega; V; W) = \langle H(\omega), V \otimes W \rangle = h(V, W),$$

where $(V \otimes W)_{ij}(x,y) = V_i(x)W_j(y)$ for all $1 \leq i, j, \leq m$. Then $H(\omega)$ is called the *shape Hessian* of J at ω. It has its support in $\Gamma \times \Gamma$, i.e., there exists a vector distribution $h_{\Gamma \oplus \Gamma}(\Omega)$ such that for all V, W there holds

$$\langle h_{\Gamma\oplus\Gamma}(\Omega), (\gamma_\Gamma(V)) \oplus (\gamma_\Gamma(W) \cdot n)\rangle_\Gamma = d^2 J(\Omega; V; W).$$

Here $(\gamma_\Gamma(V)) \oplus (\gamma_\Gamma(W) \cdot n)$ is defined as the tensor product

$$\left((\gamma_\Gamma(V)) \oplus (\gamma_\Gamma(W) \cdot n)\right)_i (x,y) = (\gamma_\Gamma(V_i))(x) \left((\gamma_\Gamma W) \cdot n\right)(y) \text{ for } x, y \in \Gamma.$$

For more details we refer to chapter 8.6 in Delfour and Zolésio (2001).

Example 2.2 *Here we continue Example 2.1 and present the second-order Eulerian semiderivative of J. Since our goal is to use the shape Hessian in a shape Newton method, we are interested in a symmetric expression for $d^2 J$. As noted earlier, this requires a restriction on the admissible vector fields. For this purpose let $F : \Gamma \to \mathbf{R}$ and let F_{ext} denote an extension of F to D such that*

$$\nabla F_{ext} \cdot n = 0, \quad \textit{and} \quad F_{ext}|_\Gamma = F. \tag{9}$$

Here n denotes a suitable extension of the unit normal on Γ. Then V_F is given by

$$V_F = F_{ext}\, n. \tag{10}$$

Analogously we define V_G. Using these vector fields, the second-order Eulerian semiderivative is found to be

$$d^2 J(\Gamma; V_F; V_G) = \int_\Gamma \Big[\Big(\frac{\partial^2 g}{\partial n^2} + (2\kappa + \nu)\frac{\partial g}{\partial n} + \nu\kappa g\Big) FG + g\langle \nabla_\Gamma F, \nabla_\Gamma G\rangle\Big] dS \tag{11}$$

Here $\nabla_\Gamma F$ denotes the tangential gradient of F which is given by

$$\nabla_\Gamma F = \nabla F|_\Gamma - \frac{\partial F}{\partial n} n.$$

The second term under the integral is the weak form of the Laplace-Beltrami *operator on Γ. If Γ is a closed curve it is considered together with periodic boundary conditions.*

2.2. Topological sensitivity

The concept of sensitivity of a domain shape functional with respect to the nucleation of holes is considered, e.g., in Garreau, Guillaume and Masmoudi (2001), Sokolowski and Żochowski (1999). Here, to some extent, we follow Sokolowski and Żochowski (1999). For $x \in \Omega$ let $B_r(x) := \{y \in \mathbf{R}^m : |y - x|_2 < r\}$ with $r > 0$. By $\overline{B_r(x)}$ we denote the closure of $B_r(x)$ and by $|\overline{B_r(x)}|$ the measure of $\overline{B_r(x)}$. Assume that the limit

$$T(x) = \lim_{r \downarrow 0} \frac{J(\Omega \setminus \overline{B_r(x)}) - J(\Omega)}{|\overline{B_r(x)}|} \tag{12}$$

exists. Then $T(x)$ is called the *topological derivative* of J at x.

Example 2.3 *Consider the problem*

$$\text{minimize} \quad \bar{J}(u,\Omega) = \frac{1}{2}\int_\Omega (u-u_d)^2 dx + \mu \int_\Omega 1 dx \tag{13}$$

$$\text{subject to} \quad -\Delta u = f_\Omega \quad \text{in } D, \quad u = 0 \text{ on } \partial D, \tag{14}$$

where $u_d \in L^2(D)$ *is given data,* $\mu > 0$ *fixed and*

$$f_\Omega(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ 0 & \text{else.} \end{cases}$$

Since the equation (14) admits a unique solution $u(\Omega)$*, we can reduce problem (13)–(14) to*

$$\text{minimize} \quad J(\Omega) = \frac{1}{2}\int_\Omega (u(\Omega)-u_d)^2 dx + \mu \int_\Omega 1 dx.$$

Then the topological gradient of J *at* $x \in \overline{\Omega}$ *is given by*

$$T(x) = -p(x) - \mu,$$

where $p \in H_0^1(D)$ *solves the* adjoint equation

$$-\Delta p = u(\Omega) - u_d \quad \text{in } D, \quad p = 0 \text{ on } \partial D.$$

Let us point out that due to the lack of requiring additional boundary conditions on the boundary of the hole, i.e., $\partial B_r(x)$, our notion of topological sensitivity differs from the ones in Garreau, Guillaume and Masmoudi (2001) and Sokolowski and Żochowski (1999). In this sense, the hole is not cut into the computational domain D which would support additional boundary conditions as in the above references. Rather it is cut into the support of Ω, which, in the context of our example, relates to the right hand side of the partial differential equation in (14).

3. Shape gradient-related descent methods

In this section we introduce the notion of shape gradient-related descent directions. Based on this concept we then devise descent methods for solving (1). In the sequel we work under the assumption that the domains of interest are sufficiently regular.

We are interested in search directions or, equivalently, descent flows for $J(\omega)$, i.e., we are interested in velocity fields V yielding $J(\omega(t)(V)) < J(\omega)$ for all sufficiently small t. To this end, let $\mathcal{H}$ denote a suitable Hilbert space of admissible vector fields, which might depend on the current shape ω, and observe that $V \in \mathcal{H}$ with

$$\langle V, W\rangle_{\mathcal{H}} = -dJ(\omega; W) \quad \text{for all } W \in \mathcal{H} \tag{15}$$

satisfies

$$dJ(\omega;V) = -\|V\|_{\mathcal{H}}^2. \tag{16}$$

Further, $\|V\|_{\mathcal{H}} = 0$ if and only if $dJ(\omega;W) \geq 0$ for all $W \in \mathcal{H}$.

DEFINITION 3.1 *Let $dJ(\omega;W)$ denote the Eulerian semiderivative of J at ω in direction $W \in \mathcal{H}$. If*

$$dJ(\omega;W) \geq 0 \quad \textit{for all } W \in \mathcal{H}$$

then ω is stationary.

If $j(\omega)$ exists and ω is stationary, then, by the structure theorem, $j(\omega) = 0$.

Let $b : \mathcal{H} \times \mathcal{H} \to \mathbf{R}$ denote a bounded and uniformly positive bilinear form, i.e., there exist constants $M > 0$ and $\epsilon > 0$ independently of ω such that $b(V,W) \leq M\|V\|_{\mathcal{H}}\|W\|_{\mathcal{H}}$ and $b(W,W) \geq \epsilon\|W\|_{\mathcal{H}}^2$. Now we can generalize (15). If $V \in \mathcal{H}$ satisfies

$$b(V,W) = -dJ(\omega;W) \quad \text{for all } W \in \mathcal{H}, \tag{17}$$

then

$$dJ(\omega;V) \leq -\epsilon\|V\|_{\mathcal{H}}^2. \tag{18}$$

Again we have $\|V\|_{\mathcal{H}} = 0$ if and only if $dJ(\omega;W) \geq 0$ for all $W \in \mathcal{H}$.

DEFINITION 3.2 *Let $dJ(\omega;W)$ denote the Eulerian semiderivative of J at ω in direction $W \in \mathcal{H}$. We call $V \in \mathcal{H}$* shape gradient-related *if it satisfies (17) for some bounded and uniformly positive bilinear form $b : \mathcal{H} \times \mathcal{H} \to \mathbf{R}$.*

We point out that in an iterative process for finding a stationary shape, the bilinear form b may vary with the iterations. Then, one has to require that the bounds ϵ and M are also uniform with respect to the iteration index. This is in the spirit of variable metric methods in classical finite dimensional optimization as in Nocedal and Wright (1999).

Let $0 < \underline{\epsilon} < \epsilon$ be fixed. We say that $V \in \mathcal{H}$ yields a *sufficient reduction* (or *sufficient decrease*) of J at ω in direction V, if there exists some $\tau > 0$ such that

$$J(\omega(t)(V)) \leq J(\omega) - \underline{\epsilon}t\|V\|_{\mathcal{H}}^2 \quad \text{for all } 0 \leq t < \tau. \tag{19}$$

This sufficient decrease definition is useful because it allows to argue convergence of descent methods which satisfy (19) in every iteration. For a gradient-related direction we have the following result:

LEMMA 3.1 *Let J be twice shape differentiable at ω, and let $V \in \mathcal{H}$ be a gradient-related vector field for J at ω, then there exists $\tau > 0$ such that (19) is satisfied.*

Proof. Since J is twice shape differentiable we have

$$J(\omega(t)(V)) = J(\omega) + t dJ(\omega; V) + o(t).$$

The direction V is gradient-related for J at ω. Thus, (18) is satisfied which yields for some $\underline{\epsilon}$ with $0 < \underline{\epsilon} < \epsilon$

$$\begin{aligned} J(\omega(t)(V)) &\leq J(\omega) - t\epsilon \|V\|^2_{\mathcal{H}} + o(t) \\ &= J(\omega) - t\underline{\epsilon}\|V\|^2_{\mathcal{H}} + t(\underline{\epsilon} - \epsilon)\|V\|^2_{\mathcal{H}} + o(t) \\ &\leq J(\omega) - t\underline{\epsilon}\|V\|^2_{\mathcal{H}}, \end{aligned}$$

where the last inequality holds true for all sufficiently small t. ■

Our results so far can be used for defining a general descent framework algorithm and analyze its convergence properties.

Algorithm 3.1

(i) Initialize $\omega_0 := \omega(0) \in D$; set $k := 0$ and $t_0 := 0$. Choose $\zeta \in (0,1)$, $0 < \underline{\epsilon} < \epsilon$, and $\alpha > 0$.

(ii) Unless a stopping rule is satisfied, compute the Eulerian semiderivative $dJ(\omega_k; V)$ for arbitrary $V \in \mathcal{H}_k$.

(iii) Compute a gradient-related descent direction by solving
$$b(V, W) = -dJ(\omega_k; W) \quad \text{for all } W \in \mathcal{H}_k.$$
Let the solution be denoted by V_k.

(iv) Find the first element $\Delta t_{k,\ell}$ of the sequence $\{\Delta t_{k,l}\} = \{\zeta^l \alpha\}_{l=0}^{\infty}$ such that
$$J(\omega_k(\Delta t_{k,\ell})(V_k)) - J(\omega_k(0)) \leq -\Delta t_{k,\ell}\underline{\epsilon}\|V_k\|^2_{\mathcal{H}_k}. \tag{20}$$

(v) Set $t_{k+1} = t_k + \Delta t_{k,\ell}$, $\omega_{k+1} = \omega_{k+1}(0) = \omega_k(\Delta t_{k,\ell})$, and $k := k+1$. Go to (ii).

Concerning the convergence of the above algorithm we first establish an auxiliary result which, under suitable assumptions, shows that there exists $\underline{t} > 0$ such that $\Delta t_{k,\ell} \geq \underline{t}$ for all $k \in \mathbf{N}$. In Algorithm 3.1 and below we use the notation $\mathcal{H}_k$ to indicate the possible dependence of the space of admissible velocity fields on the actual shape ω_k.

Lemma 3.2 *Assume that J is twice shape differentiable, that the shape Hessian is uniformly bounded, the remainder term in (8) for $\omega = \omega_k$ is uniform in k, and $\|V_k\|_{\mathcal{H}_k} \leq M_V$ for all k. Then there exists $\underline{t} > 0$ such that $\Delta t_{k,\ell}$ of step (iii) in Algorithm 3.1 satisfies*

$$\Delta t_{k,\ell} \geq \underline{t} \quad \textit{for all } k \in \mathbf{N}.$$

Proof. Since J is twice shape differentiable we have

$$J(\omega_k(\Delta t_{k,\ell})) - J(\omega_k(0)) = \Delta t_{k,\ell} dJ(\omega_k(0); V_k) + \frac{(\Delta t_{k,\ell})^2}{2} d^2 J(\omega_k(0); V_k; V_k) + o((\Delta t_{k,\ell})^2)\|V_k\|^2_{\mathcal{H}_k},$$

where $\mathcal{O}(\cdot)$ is uniform with respect to k. Let C denote the bound to the second-order Eulerian semiderivative. Then

$$\begin{aligned} J(\omega_k(\Delta t_{k,\ell})) - J(\omega_k(0)) &\leq (-\epsilon \Delta t_{k,\ell} + \frac{C(\Delta t_{k,\ell})^2}{2} + \mathcal{O}((\Delta t_{k,\ell})^2)) \|V_k\|^2_{\mathcal{H}_k} \\ &\leq (-\epsilon \Delta t_{k,\ell} + \frac{C(\Delta t_{k,\ell})^2}{2} + (\Delta t_{k,\ell})^2 \frac{\epsilon - \underline{\epsilon}}{2}) \|V_k\|^2_{\mathcal{H}_k}, \end{aligned}$$

where we used the existence of $\bar{t} > 0$ such that

$$\mathcal{O}((\Delta t_{k,\ell})^2) \leq \frac{\epsilon - \underline{\epsilon}}{2} (\Delta t_{k,\ell})^2$$

for all $\Delta t_{k,\ell} \leq \bar{t}$. Thus, if

$$\left(-\epsilon \Delta t_{k,\ell} + \frac{C(\Delta t_{k,\ell})^2}{2} + (\Delta t_{k,\ell})^2 \frac{\epsilon - \underline{\epsilon}}{2} \right) \|V_k\|^2_{\mathcal{H}_k} \leq -\underline{\epsilon} \Delta t_{k,\ell} \|V_k\|^2_{\mathcal{H}_k}, \quad (21)$$

then (20) is satisfied. A simple calculation shows that for

$$\Delta t_{k,\ell} \leq \frac{2(\epsilon - \underline{\epsilon})}{C + \epsilon - \underline{\epsilon}}$$

(21) is fulfilled. The update rule for $\Delta t_{k,l}$ yields

$$\Delta t_{k,\ell} \geq \zeta \frac{2(\epsilon - \underline{\epsilon})}{C + \epsilon - \underline{\epsilon}} =: \underline{t} > 0.$$

■

Next we address convergence of Algorithm 3.1.

THEOREM 3.1 *Let the assumptions of Lemma 3.2 be satisfied. If $J(\omega)$ is bounded from below, then* $\lim_{k\to\infty} J(\omega_k) = J^*$ *for some* $J^* \in \mathbf{R}$ *and* $\lim_{k\to\infty} \|V_k\|_{\mathcal{H}_k} = 0$.

Proof. Note that by Lemma 3.2 there exists $\underline{t} > 0$ such that

$$J(\omega_{k+1}(0)) \leq J(\omega_k(0)) - \underline{\epsilon}\underline{t} \|V_k\|^2_{\mathcal{H}_k} \quad \text{for all } k \in \mathbf{N}_0. \quad (22)$$

Since $\{J(\omega_k)\}$ is bounded from below and monotonically decreasing, it is convergent to some $J^* \in \mathbf{R}$. Further, from (22) and the convergence of $\{J(\omega_k)\}$ we infer $\lim \|V_k\|_{\mathcal{H}_k}$=0. ■

Theorem 3.1 shows that if there is a limit shape ω^* of $\{\omega_k\}$ and $\{\mathcal{H}_k\}$ is well-behaved, then it is stationary in the sense of Definition 3.1.

REMARK 3.1 *We point out that the existence and boundedness assumptions on the first and second-order shape derivatives are implied by sufficient smoothness of the iterates ω_k. This is also true for the uniformity assumptions with respect to (8) in Lemma 3.2.*

The smoothness requirements for $\{\omega_k\}$ alluded to in Remark 3.1 are related to regularity properties of the velocity fields utilized by the speed method; see, e.g., chapter 7 in Delfour and Zolésio (2001). By choosing ω_0 sufficiently regular and close to a solution ω_*, and requiring that the velocity fields be sufficiently regular as well, one can argue that the iterates ω_k remain regular (at least for small $\Delta t_{k,\ell} > 0$).

3.1. Shape Newton methods

In Section 2 we defined the shape Hessian $H(\omega)$ and its associated bilinear form $h(V, W)$. If h is uniformly positive (with constant $\epsilon > 0$) and continuous on $\mathcal{H}$, then it may serve in (17) instead of b. Then our descent algorithm becomes a *shape Newton method* for minimizing the shape functional J. The actual descent direction is computed as the solution $V_k \in \mathcal{H}_k$ to the Newton-type equation

$$d^2 J(\omega_k; W; V) = -dJ(\omega_k; W) \quad \text{for all } W \in \mathcal{H}_k. \tag{23}$$

From classical optimization – see Nocedal and Wright (1999) – it is known that, under suitable assumptions, Newton's method converges locally at a fast rate. We will address this aspect in our numerics Section 6 in connection with level set based approaches. However, let us mention here another aspect. In many situations the shape Hessian acts as a smoothing operator. For instance, in our Example 2.2 the shape Hessian contains the Laplace-Beltrami operator which is a second-order elliptic differential operator. In this case (23) amounts to solving a Helmholtz-type problem with periodic boundary conditions on Γ_k. As a consequence, V_k enjoys typically more regularity properties than the shape gradient. Besides the analytical aspect, this has also an impact on the numerical level.

4. Level set based descent methods

Algorithm 3.1 is conceptual in the sense that in a practical realization we need a method for updating the geometry. A useful tool which can be viewed in function space as well as in a finite dimensional framework after discretization is given by the level set method in Osher and Fedkiw (2002). It relies on the representation of $\Gamma(t)(\cdot)$ as the zero-level set of a sufficiently smooth function $\phi : D \times [0, \tau] \to \mathbf{R}$. Consider

$$\begin{cases} \frac{dx}{dt}(t) = V_F(x(t)), \quad 0 < t < \tau, \\ x(0) = x \in \Gamma(0) = \Gamma, \end{cases} \tag{24}$$

where V_F is according to (9)–(10). This defines $\Gamma(t)(V_F) = \{x \in D : \phi(x,t) = 0\}$. Further we assume that $\Gamma(t)(V_F)$ consists of closed curves for all t and

$$\Omega(t)(V_F) = \{x \in D : \phi(x,t) < 0\}, \tag{25}$$

$$D \setminus \overline{\Omega}(t)(V_F) = \{x \in D : \phi(x,t) > 0\}. \tag{26}$$

Then, for $x \in \Gamma(t)(V_F)$ the gradient $\nabla\phi(x,t)$ points in the direction of the outward normal.

Since $\Gamma(t)(V_F)$ is required to remain the zero-level set of ϕ for all t and assuming that $x(t) \in \Gamma(t)(V_F)$ travels with velocity $F(x(t),t)$ in outward unit normal direction, after differentiating $\phi(x(t),t) = 0$ with respect to t, we obtain the transport equation

$$\frac{\partial \phi}{\partial t}(x,t) + F_{\text{ext}}(x,t)|\nabla\phi(x,t)|_2 = 0, \quad \phi(x,0) = \Gamma(0), \quad x \in D. \tag{27}$$

In (27) F_{ext} denotes a suitable extension of F to D. Given some geometry Γ which is moved along its outward normal direction with velocity F, the level set equation (27) can be used to obtain the updated geometry $\Gamma(t)(V_F)$. Notice that (27) is a partial differential equation of Hamilton-Jacobi type. A popular choice for ϕ satisfying the sign conditions in (25)–(26) is given by signed distance functions. In this case geometrical information can be expressed in a convenient way. For instance, $n(x,t) = \nabla\phi(x,t)$ and $\kappa(x,t) = \Delta\phi(x,t)$. Further, numerically it allows a stable identification of the zero-level set.

If F respectively F_{ext} are gradient-related descent directions, then an algorithmic framework analogous to Algorithm 3.1 can be defined.

Algorithm 4.1

(i) Choose $\Gamma_0(0) = \Gamma$, *and initialize* $\phi_0(x,0)$ *such that*

$$\begin{aligned} \Gamma_0(0) &= \{\phi_0(x,0) = 0\}, \\ \Omega_0(0) &= \{\phi_0(x,0) < 0\}, \\ D \setminus \overline{\Omega_0}(0) &= \{\phi_0(x,0) > 0\}. \end{aligned}$$

Set $k := 0$ *and* $t_0 := 0$, *and choose* $0 < \underline{\epsilon} < \epsilon$, $\zeta \in (0,1)$ *and* $\alpha > 0$.

(ii) Compute a gradient-related descent direction V_k *of* J *at* Γ_k *and its extension* $V_{k,ext}$.

(iii) Update ϕ_k *by solving*

$$\frac{\partial \phi_k}{\partial t}(x,t) - V_{k,ext}(x,t)|\nabla\phi_k(x,t)|_2 = 0, \quad 0 \le t \le \Delta t_{k,\ell},\ x \in D \tag{28}$$

with $\phi_k(x,0) = \phi_{k-1}(x, \Delta t_{k-1,\ell})$, *where the time step* $\Delta t_{k,\ell}$ *is the first element of the sequence* $\{\Delta t_{k,l}\} = \{\zeta^l \alpha\}_{l=0}^{\infty}$ *which satisfies*

$$J(\Gamma_k(\Delta t_{k,\ell})(V_k)) - J(\Gamma_k(0)) \le -\Delta t_{k,\ell}\underline{\epsilon}\|V_k\|^2_{\mathcal{H}_k}. \tag{29}$$

(iv) Set

$$\begin{aligned} t_{k+1} &= t_k + \Delta t_{k,\ell}, \\ \phi_{k+1}(x,0) &= \phi_k(x, \Delta t_{k,\ell}), \\ \Gamma_{k+1}(0) &= \{\phi_{k+1}(x,0) = 0\}, \end{aligned}$$

and $k := k+1$. *Go to (ii).*

The difference between Algorithm 3.1 and Algorithm 4.1 is mainly related to the representation of $\Gamma_k(0)$ as the zero-level set of $\phi_k(x,0)$ and the transport of the geometry by means of the level set equation (28) in step (iii). In detail step (iii) operates as follows: The level set equation is solved in $0 \le t \le \Delta t_{k,l}$, where $\Delta t_{k,l} > 0$ denotes the actual trial time step. Then, $\phi_k(\Delta t_{k,l})$ is determined and $\Gamma_k(\Delta t_{k,l})(V_k) = \{\phi_k(x, \Delta t_{k,l}) = 0\}$ as well as $J(\Gamma_k(\Delta t_{k,\ell})(V_k))$ are computed. If (29) is satisfied, then $\Delta t_{k,\ell} := \Delta t_{k,l}$; else the actual trial time step is reduced and step (iii) is repeated. Since (29) is identical to (20), the convergence of Algorithm 4.1 can be established by similar reasoning as in the case of Algorithm 3.1.

Now the regularity considerations with respect to $\{\omega_k\}$ are related to $\{V_k\}$ via the level set equation. The latter Hamilton-Jacobi-type partial differential equation can be analyzed using the techniques as developed in, e.g., Crandall and Lions (1986) or in Bardi, Crandall, Evans, Soner and Souganidis (1997). These results guarantee that for uniformly continuous initial data $\phi_k(\cdot, 0)$ and sufficiently regular V_k the update $\phi_k(\cdot, \Delta t_{k,\ell})$ remains smooth. Further considerations concerning the consistency of the level-set approach in relation with the original front propagation based on shape gradients can be found in the contribution of Souganidis (pp. 186–242 in Bardi, Crandall, Evans, Soner and Souganidis, 1997).

Let us assume that $h(V, W)$, the bilinear form associated with the shape Hessian, is uniformly positive and bounded. If we choose $h(V, W)$ as the bilinear form for computing the gradient related direction in step (ii) of Algorithm 4.1, then a level-set based version of the shape Newton method is obtained.

5. A descent method based on topological gradients

It was noted earlier that methods based solely on shape sensitivity information have a limited potential of changing the topology. In this section, we discuss an algorithm which uses the topological derivative (or topological gradient) in a descent framework.

The topological gradient at $x \in \overline{\Omega} \subset D$ is given by

$$T(x) = \lim_{r \downarrow 0} \frac{J(\Omega \setminus \overline{B_r(x)}) - J(\Omega)}{|\overline{B_r(x)}|}.$$

In many applications one has shape functionals of the type

$$J(\Omega) = \int_D g(u(\Omega), \Omega) dx,$$

where $u(\Omega)$ solves a differential equation (on D) with data related to Ω; see Example 2.3. In this case it makes sense to consider the topological gradient in $D \setminus \Omega$ as well:

$$T(x) = \lim_{r \downarrow 0} \frac{J(\Omega \cup \overline{B_r(x)}) - J(\Omega)}{|\overline{B_r(x)}|}.$$

Next we specify how T can be used in a descent framework. For this purpose assume that we have a geometrical implicit function ϕ satisfying (25)–(26) to represent $\Gamma = \partial\Omega$. Let us assume that T is sufficiently smooth. We expect that J can be reduced if $T(x) < 0$ for some $x \in D$. Since we use a geometrical implicit function satisfying (25)–(26) we have to be careful in using T for updating the geometrical variable. The subsequent discussion for computing a topological descent direction D_T is related to Burger, Hackl and Ring (2004), p. 351. Let us assume that $\Omega = \{\phi(x,0) < 0\}$ and $T(x) < 0$.

- If $x \in \Omega$, then $\phi(x,0) < 0$. Since $T(x) < 0$, the nucleation of a hole will reduce J. This implies that a positive quantity has to be added to $\phi(x,0)$ in order to create a hole. One may use $D_T(x) = -T(x)$.
- If $x \in D \setminus \overline{\Omega}$, then $\phi(x,0) > 0$. Now, adding $D_T(x) = T(x) < 0$ reduces $\phi(x,0)$ which is necessary for creating a new topological component which decreases J.

While $j(\omega_*) = 0$ implies that ω_* is a stationary shape, the situation in the case of the topological gradient is different. In fact, if $T(x) \geq 0$ in D, then the topology remains unchanged. However, the quantity

$$\begin{aligned} m_T(x,0) \quad = \quad & \max(\text{sign}\,(\phi(x,0)),0)\min(T(x),0) - \\ & \min(\text{sign}\,(\phi(x,0)),0)\max(-T(x),0). \end{aligned}$$

is zero in such a situation. Hence it can be used as a (numerical) stationarity measure. Note that in general we have to replace $T(x)$ by $T(x,t)$, $\phi(x,0)$ by $\phi(x,t)$ and Ω by $\Omega(t)$. Let $\Omega(t) = \{\phi(x,t) < 0\}$ denote the actual shape. Then the updated shape is given by

$$\Omega(t + \Delta t) = \{\phi(x, t + \Delta t) < 0\}$$

where ϕ satisfies

$$\frac{\partial \phi}{\partial t}(x,t) = D_T(x,t). \tag{30}$$

Definition 5.1 *We call a shape* $\Omega(t)$ stationary in the topological sense*, if* $m_T(x,t) = 0$ *for almost all* $x \in D$.

If $\Omega(t)$ is not stationary in the topological sense, then we can find $\Delta t > 0$ such that

$$J(\Omega(t + \Delta t)) - J(\Omega(t)) \leq -\gamma \Delta t \|D_T(0)\|^2_{\mathcal{W}},$$

where $\mathcal{W}$ denotes a suitable Banach space, and $0 < \gamma < 1$ is fixed.

Next we specify our descent algorithm for optimizing the topology.

Algorithm 5.1

(i) Choose $\Omega_0 = \Omega(0) \subset D$*, and initialize* ϕ_0*. Set* $k := 0$ *and* $t_0 := 0$*. Choose* $\gamma, \zeta \in (0,1)$ *and* $\alpha > 0$*.*

(ii) Compute the topological gradient T_k *of* J *at* Ω_k*.*

(iii) *Unless a stopping rule is satisfied, compute* D_{T_k}*, and determine the first element* $\Delta t_{k,\ell}$ *of the sequence* $\{\Delta t_{k,l}\} = \{\zeta^l \alpha\}_{l=0}^{\infty}$ *such that*
$$J(\Omega_k(\Delta t_{k,\ell})) - J(\Omega_k(0)) \leq -\gamma \Delta t_{k,\ell} \|D_{T_k}\|_{\mathcal{W}}^2.$$
(iv) *Set* $t_{k+1} = t_k + \Delta t_{k,\ell}$*,* $\Omega_{k+1}(0) = \Omega_k(\Delta t_{k,\ell})$*, and* $k := k+1$*. Go to (ii).*

In step (iii) of the above algorithm, for given $\Delta t_{k,l}$ equation (30) is solved in $[0, \Delta t_{k,l}]$, and $\Omega_k(\Delta t_{k,l}) = \{\phi(t_k + \Delta t_{k,l}) < 0\}$ is set. If the line search criterion in step (iii) is satisfied, then $\Delta t_{k,\ell} := \Delta t_{k,l}$; otherwise $\Delta t_{k,l}$ is reduced, and step (iii) is repeated.

6. Numerics

In this section we report on some numerical aspects in connection with Algorithm 4.1 and Algorithm 5.1. Here we focus on applications coming from image segmentation. In Example 2.1 we introduced the shape functional $J(\Gamma)$ which is related to edge detector based image segmentation; see, e.g., Yezzi et al. (1997). The non-negative function g represents an edge detector which is zero at ideal edges and positive elsewhere. Then the first integral in (6) aims at locating Γ at places where g is zero, respectively very small. The second integral acts as a regularization. It is sometimes referred to as a "balloon force". In fact, if $\nu < 0$, then it tends to maximize the area Ω with $\partial\Omega = \Gamma$. Thus, if Algorithm 4.1 is initialized by a closed curve which lies completely inside the feature of interest (homogeneous region which has to be segmented), then the negative ν-value is responsible for the growth of Ω as the iterations proceed, and analogously for $\nu > 0$. The parameter ν has to be chosen appropriately. On the one hand, it has to cope with possible noise in an image and, hence, it has to be sufficiently large; on the other, if it is too large, then the balloon force is too strong and the segmentation fails. In Examples 2.1 and 2.2 the first and second order shape derivatives of J are provided.

In our numerical tests we safeguard the line search in step (iii) of Algorithm 4.1. In fact, we bound the step size in $[t_{\text{cfl}}, \rho t_{\text{cfl}}]$, where t_{cfl} denotes the time step size in the discretized level set equation according to the CFL-condition (see Sethian, 1999) and $\rho > 0$ is fixed. The time step is a first order explicit Euler step. If the line search yields no success after $\ell_{\max}$ cycles, the CFL-time step is used. This is necessary since J, dJ and d^2J are discretized separately and are thus subject to independent approximation and discretization errors. For further numerical details of the discretization and implementation we refer to Hintermüller and Ring (2003).

First we study the convergence rate of Algorithm 4.1. In Section 4 we observed that whenever the bilinear form $h(V, W)$ associated with the shape Hessian is uniformly positive and bounded, then it is a feasible choice when computing the gradient-related direction. The resulting algorithm is a shape Newton-type method. Therefore it is of interest to study the local convergence speed. For both results in Fig. 1 the corresponding test problem is an edge

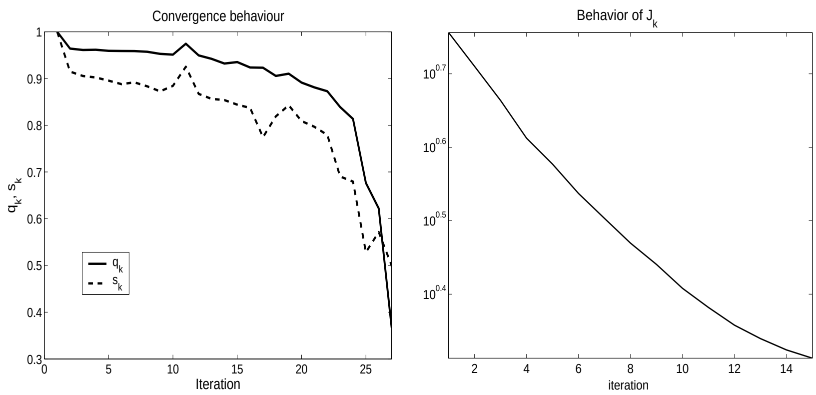

Figure 1. Left plot: Convergence behavior of q_k in $W^{1,\infty}(D)$ (solid) and of s_k (dashed). Right plot: Convergence behavior of $J(\Gamma_k)$.

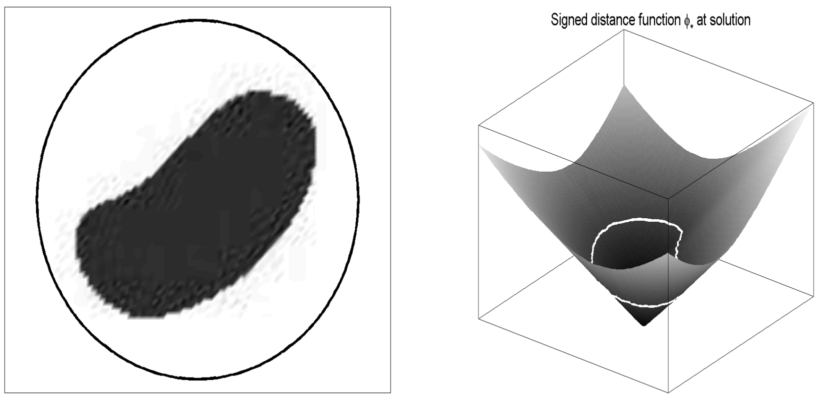

Figure 2. Left plot: Test image with Γ_0. Right plot: Geometrical implicit (here: signed distance) function ϕ_* at the solution with zero-level set Γ_* (white contour).

detector based image segmentation problem for the image shown in Fig. 2 in the left plot. In Fig. 1 (left plot) we show the behavior of the discrete versions of the quotients

$$q_k = \frac{\|\phi_{k+1} - \phi_*\|_{W^{1,\infty}}}{\|\phi_k - \phi_*\|_{W^{1,\infty}}}$$

and

$$s_k = \frac{|(\Omega_{k+1} \setminus \Omega_*) \cup (\Omega_* \setminus \Omega_{k+1})|}{|(\Omega_k \setminus \Omega_*) \cup (\Omega_* \setminus \Omega_k)|}.$$

Here we set $\Omega_k = \{\phi_k < 0\}$ and $\Omega_* = \{\phi_* < 0\}$. The geometrical implicit function ϕ_* at the numerical solution is contained in the right plot of Fig. 2. Note that q_k allows to study the convergence behavior of the signed distance function which can be considered in $W^{1,\infty}(D)$; see Delfour and Zolésio (2001). The second quantity, s_k, provides a rate for the "convergence" of the sets Ω_k. We employed the line search with $\rho = 3$, $\ell_{\max} = 3$, and $\underline{\epsilon} = 0.001$. Since q_k exhibits a decreasing tendency, the results in Fig. 1 indicate that $\{\phi_k\}$ converges eventually at a fast, i.e., superlinear rate to ϕ_*. A similar observation holds true for $\{s_k\}$. Fast local convergence is one of the benefits of Newton-type methods. In the right plot of Fig. 1 $\{J_k\}$ is depicted for a run with a more progressive line search strategy ($\rho = 5$, $\ell_{\max} = 4$, $\underline{\epsilon} = 0.001$).

For the test runs, we used a modified version of h associated to (11) with $g(x) \geq \epsilon > 0$, $\epsilon = 1e - 6$. Further, the first term under the integral in (11) is replaced by

$$s(x) = \max\{\epsilon, (\frac{\partial^2 g}{\partial n^2} + (2\kappa + \nu)\frac{\partial g}{\partial n} + \nu\,\kappa\, g)(x)\}.$$

This implies a uniformly positive bilinear form on $\mathcal{H}(\Gamma) = \{V_F = F_{\text{ext}} n : \nabla F_{\text{ext}} \cdot n = 0, F_{\text{ext}}|\Gamma = F \in H^1(\Gamma)\}$. For details on the problem formulation and the choice of edge detector, we refer to Hintermüller and Ring (2003).

Next we study Algorithm 5.1. It can be used together with Algorithm 4.1 to minimize (6). In our first test example we use Algorithm 5.1 as an initialization procedure for edge detector based image segmentation. This is of interest because of the fact that many state-of-the-art techniques require a particular user specified initial choice; see, e.g., Yezzi et al. (1997). Algorithm 5.1 is applied to solve the following problem

$$\text{minimize } J(\Omega) = \alpha_1 \|u(\Omega) - u_d\|^2_{L^2(D)} + \alpha_2 \|\nabla(u(\Omega) - u_d)\|^2_{L^2(D)} + \mu \int_\Omega 1 dx$$

with $\alpha_1, \alpha_2 \geq 0$, $\alpha_1 + \alpha_2 > 0$, and $\mu \geq 0$. Further, $u(\Omega)$ is the solution of (14). Here $u_d \in H^1_0(D)$ represents given (and possibly preprocessed) image data. Therefore, the first two terms in $J(\Omega)$ above are tracking terms, i.e., the reconstruction $u(\Omega)$ is required to be sufficiently close to u_d. On the other hand, u_d might be contaminated by noise. In the context of topology optimization this may result in a set Ω which consists of many small components, most of them induced by noise patterns. To counteract this behavior we add the term $\mu \int_\Omega 1dx$, which acts as a volume penalty. In fact, for sufficiently large $\mu > 0$ it penalizes the volume of Ω and tends to avoid noise induced components. Note further that an application of the topological derivative to (6) is in general infeasible. Indeed, in regions where $g \geq \delta > 0$ the topological derivative is $+\infty$.

The topological derivative of $J(\Omega)$ is found to be

$$T(x) = \begin{cases} -p(x) - \mu \text{ for a.a. } x \in \overline{\Omega} \\ p(x) + \mu \text{ for a.a. } x \in D \setminus \Omega. \end{cases}$$

This implies the update direction

$$D_T(x) = p(x) + \mu \quad \text{for almost all } x \in D,$$

where $p \in H_0^1(D)$ solves

$$-\Delta p = \alpha_1(u(\Omega) - u_d) + \alpha_2 \Delta(u(\Omega) - u_d) \text{ in } D.$$

This implies that one has $\mathcal{W} = H^1(D)$ in Algorithm 5.1. In our numerics we use the standard five point stencil for the discretization of the Laplacian and $\gamma = 0.001$ in the line search. The left plot of Fig. 3 shows the result obtained from applying Algorithm 5.1 with $\mu = 25$ after 14 iterations. It provides an excellent start-up configuration for a subsequent application of Algorithm 4.1 ($\nu = -1.25$ in (6)). The outcome of this application (after 14 iterations as well) is shown in the right plot of Fig. 3.

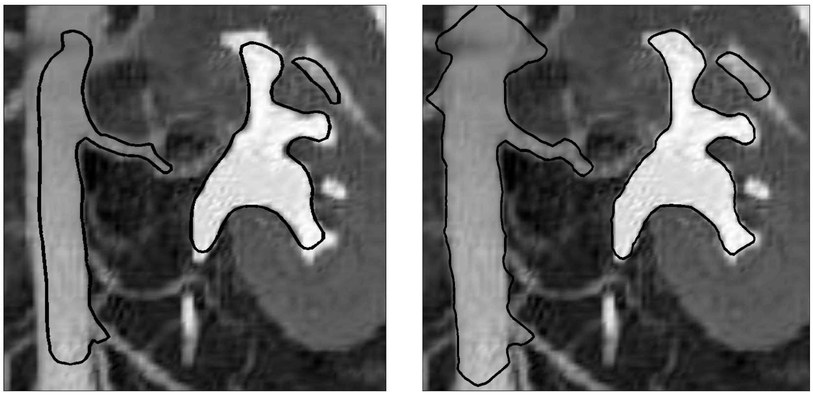

Figure 3. Left plot: Result obtained from Algorithm 5.1. Right plot: Result after applying Algorithm 4.1 initialized by the contour in the left plot.

In Fig. 4 (left plot) we show the result obtained by Algorithm 5.1 for $\mu = 0.01$ after 34 iterations. The initial shape is depicted in the right plot. As we can see, the topological derivative detects the homogeneous features of the image rather well and may itself serve as a segmentation method. This, however, requires a suitable preprocessing stage for producing u_d. In our test runs we use a simple thresholding technique for the given intensity map I of the image. After this step we obtain a function $\tilde{f}$ with values in [0,1]. Then $u_d \in H_0^1(D)$ is computed as the solution to $-\Delta u = \tilde{f}$ in D.

Finally we point out that for all the results discussed here we used $\alpha_1 = 1$ and $\alpha_2 = 1.0e4$.

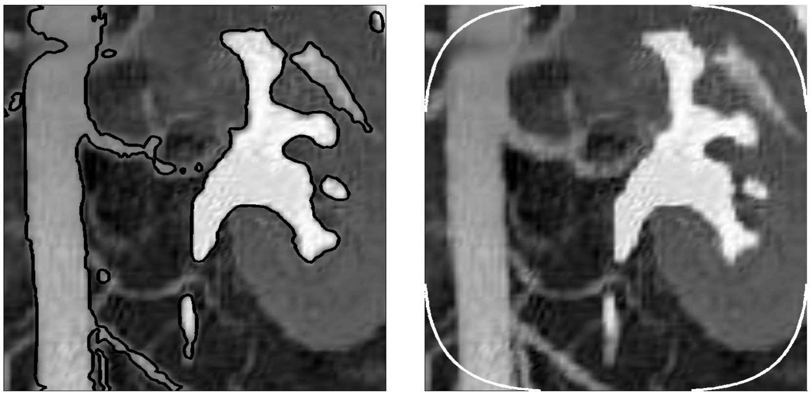

Figure 4. Left plot: Segmentation produced by Algorithm 5.1. Right plot: Initialization.

References

ALLAIRE, G., JOUVE, F. and TOADER, A.-M. (2004) Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, **194** (1), 363–393.

BARDI, M., CRANDALL, M.G., EVANS, L.C., SONER, H.M. and SOUGANIDIS, P.E. (1997) *Viscosity Solutions and Applications.* Lecture Notes in Mathematics, **1660**, Springer-Verlag, Berlin.

BURGER, M. (2003) A framework for the construction of level set methods for shape optimization and reconstruction. *Interfaces Free Bound*, **5** (3), 301-329.

BURGER, M. (2004) Levenberg-Marquardt level set methods for inverse problems. *Inverse Problems*, **20**, 259–282.

BURGER, M., HACKL, B. and RING, W. (2004) Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, **194**, 344–362.

CASELLES, V., CATTÉ, F., COLL, T. and DIBOS, F. (1993) A geometric model for active contours in image processing. *Numer. Math.*, **66** (1), 1–31.

CRANDALL, M.G. and LIONS, P.-L. (1986) On existence and uniqueness of solutions of Hamilton-Jacobi equations. *Nonlinear Analysis*, **10** (1), 353–370.

DELFOUR, M. and ZOLÉSIO, J-P. (2001) *Shapes and Geometries. Analysis, Differential Calculus and Optimization.* SIAM Advances in Design and Control. SIAM, Philadelphia.

DORN, O., MILLER, E.L. and RAPPAPORT, C.M. (2000) A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, **16** (5), 1119–1156.

ESCHENAUER, H. and SCHUMACHER, A. (1994) Bubble method for topology and shape optimization of structures. *Structural Optimization*, **8**, 42–51.

Garreau, S., Guillaume, P. and Masmoudi, M. (2001) The topological asymptotic for pde systems: the elasticity case. *SIAM J. Control Optim.*, **39**, 1756–1778.

Hintermüller, M. and Ring, W. (2003) A second order shape optimization approach for image segmentation. *SIAM J. Appl. Math.*, **64** (2), 442–467.

Hintermüller, M. and Ring, W. (2004) An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *J. Math. Imag. Vision*, **20**, 19–42.

Litman, A., Lesselier, D. and Santosa, F. (1998) Reconstruction of a two-dimensional binary obstacle by controlled evolution of a level-set. *Inverse Problems*, **14** (3), 685–706.

Nocedal, J. and Wright, S.J. (1999) *Numerical Optimization.* Springer Series in Operations Research. Springer-Verlag, New York.

Osher, S.J. and Fedkiw, R.P. (2002) *Level Set Methods and Dynamic Implicit Surfaces.* Springer Verlag, New York.

Osher, S.J. and Santosa, F. (2001) Level set methods for optimization problems involving geometry and constraints. I. Frequencies of a two-density inhomogeneous drum. *J. Comput. Phys.*, **171** (1), 272–288.

Osher, S.J. and Sethian, J.A. (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, **79** (1), 12–49.

Santosa, F. (1996) A level-set approach for inverse problems involving obstacles. *ESAIM Contrôle Optim. Calc. Var*, **1**, 17–33.

Sethian, J.A. (1999) *Level Set Methods and Fast Marching Methods.* Cambridge University Press, Cambridge, second edition.

Simon, J. (1989) Second variations for domain optimization problems. In: *Control and Estimation of Distributed Parameter Systems (Vorau, 1988)*, F. Kappel et al., eds., International Ser. Numer. Math. **91**, Birkhäuser Verlag, Basel, 361–378.

Sokołowski, J. and Żochowski, A. (1999) On the topological derivative in shape optimization. *SIAM J. Control Optim.*, **37** (4), 1251–1272.

Sokołowski, J. and Zolésio, J-P. (1992) *Introduction to Shape Optimization.* Springer-Verlag, Berlin.

Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P. and Tannenbaum, A. (1997) A geometric snake model for segmentation of medical imagery. *IEEE Trans. Med. Imaging*, **16** (3), 199–209.