



# Quasi-Newton methods for topology optimization using a level-set method

Sebastian Blauth<sup>1</sup> · Kevin Sturm<sup>2</sup>

Received: 29 March 2023 / Revised: 5 July 2023 / Accepted: 10 August 2023 / Published online: 2 September 2023  
© The Author(s) 2023

## Abstract

The ability to efficiently solve topology optimization problems is of great importance for many practical applications. Hence, there is a demand for efficient solution algorithms. In this paper, we propose novel quasi-Newton methods for solving PDE-constrained topology optimization problems. Our approach is based on and extends the popular solution algorithm of Amstutz and André (J Comput Phys 216: 573–588, 2006). To do so, we introduce a new perspective on the commonly used evolution equation for the level-set method, which allows us to derive our quasi-Newton methods for topology optimization. We investigate the performance of the proposed methods numerically for the following examples: Inverse topology optimization problems constrained by linear and semilinear elliptic Poisson problems, compliance minimization in linear elasticity, and the optimization of fluids in Navier–Stokes flow, where we compare them to current state-of-the-art methods. Our results show that the proposed solution algorithms significantly outperform the other considered methods: They require substantially less iterations to find a optimizer while demanding only slightly more resources per iteration. This shows that our proposed methods are highly attractive solution methods in the field of topology optimization.

**Keywords** Topology optimization · Topological sensitivity · Level-set method · PDE constrained optimization · Numerical optimization

**Mathematical Subject Classification** 65K05 · 74P15 · 49Q10 · 49M41 · 35Q93

## 1 Introduction

Topology optimization is concerned with the optimization of a domain by altering its geometrical features. Whereas in shape optimization only the boundary of a domain is variable, topology optimization considers the addition or removal of material to the geometry. Originally introduced in the context of solid mechanics, topology optimization has been considered for many practical applications, e.g., compliance minimization in elasticity (Eschenauer et al. (1994); Allaire et al. (2005); Amstutz and André (2006)), design optimization in the context of fluid mechanics (Borrvall

and Petersson (2003); Sá et al. (2016)), electrical machines (Gangl and Langer (2012)), as well as the solution of inverse problems (Canelas et al. (2015); Hintermüller and Laurain (2008); Laurain et al. (2013); Beretta et al. (2017)) and the modeling and simulation of fracture evolution (Xavier et al. (2018, 2017)).

The goal of topology optimization is to optimize a cost functional  $J$  depending on the set  $\Omega$ , which plays the role of the design variable. To achieve this goal, usually all possible designs are assumed to belong to a fixed domain  $D \subset \mathbb{R}^d$ ,  $d \in \mathbb{N}_{>0}$ , where  $\mathbb{N}$  denotes the set of positive integers, which is referred to as hold-all domain. The understanding how a functional varies under perturbations of  $\Omega$  is crucial for the development of numerical methods. Of particular importance are perturbations obtained by removing or adding a small inclusion  $\omega_\varepsilon(x_0)$  at  $x_0 \in \Omega$  or  $x_0 \in D \setminus \Omega$  of size  $\varepsilon$ . The first order variation of the functional under this perturbation is called the “topological derivative”. This leads to the definition of the perturbed domains

---

Responsible editor: Jun Wu

✉ Sebastian Blauth  
sebastian.blauth@itwm.fraunhofer.de

<sup>1</sup> Fraunhofer ITWM, Kaiserslautern, Germany

<sup>2</sup> Institute of Analysis and Scientific Computing, TU Wien, Vienna, Austria

$$\Omega_\varepsilon := \Omega_\varepsilon(x_0, \omega) := \begin{cases} \Omega \cup \omega_\varepsilon(x_0) & \text{for } x_0 \in D \setminus \overline{\Omega}, \\ \Omega \setminus \omega_\varepsilon(x_0) & \text{for } x_0 \in \Omega, \end{cases} \quad (1.1)$$

and  $\omega_\varepsilon(x_0) := x_0 + \varepsilon\omega$  with  $\omega \subset \mathbb{R}^d$  being a simply connected domain with  $0 \in \omega$ . Then, the topological derivative can be defined by

$$DJ(\Omega)(x_0, \omega) = \lim_{\varepsilon \searrow 0} \frac{J(\Omega_\varepsilon(x_0, \omega)) - J(\Omega)}{f(\varepsilon)}, \quad (1.2)$$

where  $f$  satisfies  $\lim_{\varepsilon \searrow 0} f(\varepsilon) = 0$ . In fact, the topological derivative can also depend on the inclusion  $\omega$ , which we omit in our notation for simplicity and just write  $DJ(\Omega)$ . For many shape functionals in practice the function  $f$  is given by the volume of  $\omega_\varepsilon$ , however, there are problems where this is not the case, e.g., when Dirichlet boundary conditions are imposed on the inclusion boundary  $\partial\omega_\varepsilon$  (Amstutz (2022)).

The topological derivative was first formally introduced as the bubble method in Eschenauer et al. (1994) and later mathematically justified in Sokolowski and Zochowski (1999); Garreau et al. (2001). Topological derivatives have been established for a variety of PDE constrained functionals and we refer to the monographs Novotny and Sokolowski (2013, 2020) for further information. For the computation of the topological derivative there exist several methods, for instance a direct approach (Novotny and Sokolowski (2013)), where first the expansion of the state variable is computed and afterwards the topological expansion of the cost function is derived via Taylor's formula. Lagrangian approaches provide another way to compute topological derivatives, e.g., a Lagrangian method using an averaged adjoint equation is presented in Sturm (2020), a method using a perturbed adjoint equation in conjunction with an unperturbed state equation is found in Amstutz and Andrä (2006), and another Lagrangian method using an unperturbed adjoint equation in Gangl and Sturm (2020). We refer to Baumann and Sturm (2022) for a comparison of Lagrangian methods and their advantages and disadvantages.

From a numerical perspective, one can either use the topological derivative directly to find the location of an optimal design (see, e.g., Hintermüller and Laurain (2008); Novotny and Sokolowski (2020)) or one can use a level-set approach (Amstutz and Andrä (2006)) in an iterative fashion to find the optimal topology. However, it should be noted that only the topological derivative is capable to create new holes while the level-set approach only allows to close already present holes. Additionally, Newton-type algorithms to find circular inclusions have been introduced using higher order topological expansions (Laurain et al. (2013); Canelas et al. (2015) or (Novotny et al. 2019, Chapter 10)). This leads to rapidly converging

algorithms, however, it requires a combinatorial search, which makes it numerically expensive and also the study of nonlinear problems remains an open problem.

In the related field of shape optimization, the development, modification, and analysis of efficient solution algorithms has received lots of attention in recent years, e.g., in Blauth (2021b, 2022) and Schulz et al. (2016), where nonlinear conjugate gradient and quasi-Newton methods for shape optimization have been proposed, in Blauth (2022), where a space mapping technique for shape optimization is presented, and in Deckelnick et al. (2022), where a  $W^{1,\infty}$  approach for shape optimization is introduced.

In this paper, we follow these developments and propose novel quasi-Newton methods for solving PDE constrained topology optimization problems based on the topological derivative. Our approach is based on and extends the popular level-set approach of Amstutz and Andrä (2006). In particular, we first provide a new perspective on the evolution equation for the level-set function given in Amstutz and Andrä (2006) to derive a gradient descent-type algorithm. This algorithm is the foundation for the quasi-Newton methods we propose afterwards. In particular, we derive a limited-memory BFGS method for topology optimization. We investigate the novel methods numerically for several problem classes: Inverse topology optimization problems constrained by linear and semilinear Poisson problems, compliance minimization in linear elasticity, and the optimization of fluids in Navier–Stokes flow. We compare our methods behavior to the widely popular method of Amstutz and Andrä (2006) as well as a simpler convex combination method from Gangl and Sturm (2021). The results show that the novel quasi-Newton methods usually have a significantly better convergence behavior than the other solution algorithms. Particularly, they require substantially less iterations to find an optimizer of the problems while demanding only slightly more numerical resources per iteration. This makes our proposed quasi-Newton methods highly attractive for solving topology optimization problems.

This paper is structured as follows. In Sect. 2, we briefly recall basic results from topology optimization as well as the level-set method for topology optimization and the solution algorithms from Amstutz and Andrä (2006) and Gangl and Sturm (2021), which are widely used in the literature. In Sect. 3, we present a new perspective on the evolution equation for the level-set function. This allows us to derive a gradient descent algorithm for topology optimization and, afterwards, propose novel quasi-Newton methods for topology optimization. Finally, we investigate the methods numerically and compare our quasi-Newton methods to the current state-of-the-art methods in Sect. 4.

## 2 Preliminaries

### 2.1 Topological sensitivity analysis

In this section we recall basics on the topological derivative. We start by recalling the definition of the topological derivative (Amstutz (2022)).

**Definition 2.1** Let  $\omega$  be a simply connected, bounded, and open subset of  $\mathbb{R}^d$ ,  $d \in \mathbb{N}_{>0}$ , with  $0 \in \omega$ , let  $D$  be a bounded hold-all domain, and let  $\mathcal{P}(D)$  be the power set of  $D$ , i.e.,  $\mathcal{P}(D) = \{\Omega \subset \mathbb{R}^d : \Omega \subset D\}$ . Let  $J : \mathcal{P}(D) \rightarrow \mathbb{R}$  be a shape functional. We say that  $J$  has a topological derivative at  $\Omega \in \mathcal{P}(D)$  and at the point  $x_0 \in D$  w.r.t.  $\omega$  if there exists some function  $f : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$ , where  $\mathbb{R}_{>0}$  denotes positive real numbers, with  $\lim_{\varepsilon \searrow 0} f(\varepsilon) = 0$  so that the following limit exists

$$DJ(\Omega)(x_0) := \lim_{\varepsilon \searrow 0} \frac{J(\Omega_\varepsilon) - J(\Omega)}{f(\varepsilon)},$$

where the perturbed domain  $\Omega_\varepsilon$  is defined by

$$\Omega_\varepsilon = \Omega_\varepsilon(x_0, \omega) = \begin{cases} \Omega \setminus \overline{\omega_\varepsilon(x_0)} & \text{for } x_0 \in \Omega, \\ \Omega \cup \omega_\varepsilon(x_0) & \text{for } x_0 \in D \setminus \overline{\Omega}, \end{cases}$$

and  $\omega_\varepsilon(x_0) := x_0 + \varepsilon\omega$ .

We follow the approach of Amstutz and Andrä (2006) and represent a set  $\Omega \subset D$  with the help of a continuous level set function  $\psi : D \rightarrow \mathbb{R}$  as follows

$$\begin{aligned} \psi(x) < 0 &\Leftrightarrow x \in \Omega, \\ \psi(x) > 0 &\Leftrightarrow x \in D \setminus \overline{\Omega}, \\ \psi(x) = 0 &\Leftrightarrow x \in \partial\Omega \setminus \partial D. \end{aligned}$$

We write  $\Omega_\psi := \Omega$  for a domain  $\Omega \subset D$  which is represented by the level-set function  $\psi$ .

**Definition 2.2** Let  $J : \mathcal{P}(D) \rightarrow \mathbb{R}$  be a shape functional, let  $\Omega \in \mathcal{P}(D)$  be an open set, and let  $\Gamma = \partial\Omega \setminus \partial D$ . Assume that the topological derivative  $DJ(\Omega)(x)$  exists for all  $x \in D \setminus \Gamma$ . Then, we define the generalized topological derivative by

$$DJ(\Omega)(x) := \begin{cases} -DJ(\Omega)(x) & \text{for } x \in \Omega, \\ DJ(\Omega)(x) & \text{for } x \in D \setminus \overline{\Omega}. \end{cases} \quad (2.1)$$

The idea behind the generalized topological derivative originates from the following observation. Let  $\psi : D \rightarrow \mathbb{R}$  be a level set function representing  $\Omega$ . If there is a constant  $c > 0$  such that

$$DJ(\Omega_\psi)(x) = c\psi(x) \quad \text{for } x \in D, \quad (2.2)$$

then

$$DJ(\Omega)(x) \geq 0 \quad \text{for all } x \in D \setminus \partial\Omega, \quad (2.3)$$

which is the necessary condition for  $\Omega$  to be optimal (cf. Amstutz and Andrä (2006)). This observation is the starting point of the popular solution algorithm from Amstutz and Andrä (2006), which we present in the next section.

### 2.2 Topology optimization algorithms using a level-set method

Based on the discussion in the previous section, a solution algorithm for topology optimization has been derived in Amstutz and Andrä (2006), which we briefly recall in the following. For the derivation, we introduce a fictitious time  $t$  and consider a family of domains  $\Omega(t)$  represented by a level-set function  $\psi : [0, T] \times D \rightarrow \mathbb{R}$ . To derive a solution algorithm, the idea is to establish an evolution equation for the level-set function  $\psi$  which ensures that  $\Omega(t)$  converges to a minimizer for  $t \rightarrow \infty$  and to discretize this evolution equation. A natural idea would be to evolve the level-set function according to the generalized topological derivative  $DJ(\Omega(t))$ , so that

$$\frac{\partial\psi(t, \cdot)}{\partial t} = DJ(\Omega(t)), \quad t \geq 0.$$

For the sake of better readability, we drop the dependence on the fictitious time  $t$  in the domain  $\Omega(t)$  throughout the rest of this paper.

In Amstutz and Andrä (2006), the authors note that, in general, using this formulation does not guarantee convergence since the generalized topological derivative  $DJ(\Omega)$  does not vanish for an optimal geometry (2.3). Instead, the main idea of Amstutz and Andrä (2006) is to use a modified version of this equation, where the topological derivative is projected to the orthogonal complement of  $\psi$  in  $L^2(D)$ , which is written as

$$\frac{\partial\psi}{\partial t} = P_{\psi^\perp}(DJ(\Omega)), \quad (2.4)$$

where the operator  $P_{\psi^\perp}$  is defined as

$$P_{\psi^\perp}(a) = a - \frac{(a, \psi)}{\|\psi\|_{L^2(D)}^2} \psi.$$

Here,  $(a, b) = (a, b)_{L^2(D)}$  denotes the  $L^2(D)$  scalar product between  $a, b \in L^2(D)$ . Now, if the right-hand side of (2.4) vanishes, i.e., if  $P_{\psi^\perp}(DJ(\Omega)) = 0$ , then it holds that there exists some  $\alpha \in \mathbb{R}$  so that  $DJ(\Omega) = \alpha\psi$ . If  $\alpha > 0$ , then the necessary optimality conditions (2.3) for the optimization problem are satisfied and the geometry  $\Omega$  described by the corresponding level-set function  $\psi$  is a local minimizer. For

this reason, we also consider the norm of the projected topological derivative, i.e.,

$$\left\| P_{\psi}^{\perp}(\mathcal{D}J(\Omega)) \right\|_{L^2(D)} \tag{2.5}$$

as a second convergence criterion for our numerical experiments in Sect. 4.

$$\theta_k = \arccos \left( \frac{(\psi_k, \mathcal{D}J(\Omega_k))}{\|\psi_k\|_{L^2(D)} \|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} \right).$$

Note that the iteration is terminated if the angle between generalized topological derivative and level-set function is

---

**Algorithm 1:** Euler’s method on the sphere algorithm for topology optimization.

---

**Input:** Initial level-set function  $\psi_0$  and corresponding geometry  $\Omega_0$ , initial step size  $\lambda_0 = 1$ , stopping tolerance  $\tau > 0$ , maximum number of iterations  $k_{\max} \in \mathbb{N}$

```

1 for  $k = 0, 1, 2, \dots, k_{\max}$  do
2   Compute the generalized topological derivative  $\mathcal{D}J(\Omega_k)$ 
3   Compute  $\theta_k = \arccos \left( \frac{(\psi_k, \mathcal{D}J(\Omega_k))}{\|\psi_k\|_{L^2(D)} \|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} \right)$ 
4   if  $\theta_k < \tau$  then
5     Stop with approximate minimizer  $\Omega_k$ 
6   Set  $\lambda_k = \min(1, 1.5\lambda_{k-1})$ 
7   while  $J \left( \frac{1}{\sin(\theta_k)} \sin((1 - \lambda_k)\theta_k) \psi_k + \sin(\lambda_k\theta_k) \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} \right) > J(\psi_k)$  do
8     Decrease the step size  $\lambda_k = \frac{\lambda_k}{2}$ 
9   Update the level-set function:  $\psi_{k+1} = \frac{1}{\sin(\theta_k)} \sin((1 - \lambda_k)\theta_k) \psi_k + \sin(\lambda_k\theta_k) \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(D)}}$ 

```

---

For the numerical solution, it is proposed in Amstutz and Andrä (2006) to discretize (2.4) with Euler’s scheme on the sphere using a step size  $\lambda$ , leading to

$$\psi_{k+1} = \frac{1}{\sin(\theta_k)} \sin((1 - \lambda_k)\theta_k) \psi_k + \sin(\lambda_k\theta_k) \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(D)}}, \tag{2.6}$$

where  $\theta_k$  is the angle between  $\psi_k$  and  $\mathcal{D}J(\Omega_k)$ , i.e.

sufficiently small, so that  $P_{\psi^{\perp}}(\mathcal{D}J(\Omega)) \approx 0$  and the necessary optimality conditions are satisfied approximately.

The resulting numerical algorithm, which is analyzed in Amstutz (2011), can be seen in Algorithm 1. By slight misuse of notation, we write  $J(\psi)$  instead of  $J(\Omega)$ , where  $\psi$  is the level-set function representing the domain  $\Omega$ , in line 7 of Algorithm 1 for the sake of better readability.

---

**Algorithm 2:** Convex combination algorithm for topology optimization.

---

**Input:** Initial level-set function  $\psi_0$  and corresponding geometry  $\Omega_0$ , initial step size  $\lambda_0 = 1$ , stopping tolerance  $\tau > 0$ , maximum number of iterations  $k_{\max} \in \mathbb{N}$

```

1 for  $k = 0, 1, 2, \dots, k_{\max}$  do
2   Compute the generalized topological derivative  $\mathcal{D}J(\Omega_k)$ 
3   Compute  $\theta_k = \arccos \left( \frac{(\psi_k, \mathcal{D}J(\Omega_k))}{\|\psi_k\|_{L^2(D)} \|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} \right)$ 
4   if  $\theta_k < \tau$  then
5     Stop with approximate minimizer  $\Omega_k$ 
6   Set  $\lambda_k = \min(1, 2\lambda_{k-1})$ 
7   while  $J \left( \lambda_k \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} + (1 - \lambda_k)\psi_k \right) > J(\psi_k)$  do
8     Decrease the step size  $\lambda_k = \frac{\lambda_k}{2}$ 
9   Update the level-set function:  $\psi_{k+1} = \lambda_k \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} + (1 - \lambda_k)\psi_k$ 

```

---

The main idea of Algorithm 1 is to iteratively use a convex combination of generalized topological derivative and level-set function to reach a minimum of the optimization problem, where the weights for the convex combination are chosen according to Euler’s method on the sphere (cf. Amstutz and Andrä (2006)). A simpler idea was used in Gangl and Sturm (2021), where the authors consider the following convex combination to evolve the level-set function

$$\psi_{k+1} = \lambda \frac{\mathcal{D}J(\Omega_k)}{\|\mathcal{D}J(\Omega_k)\|_{L^2(\mathcal{D})}} + (1 - \lambda)\psi_k, \tag{2.7}$$

where the parameter  $\lambda$  plays the role of a step size. For a fixed step size  $\lambda > 0$ , it is easy to see that if the method becomes stationary, i.e., if  $\psi_{k+1} = \psi_k$ , then we have that  $\psi_k = \mathcal{D}J(\Omega_k)/\|\mathcal{D}J(\Omega_k)\|_{L^2(\mathcal{D})}$  and, in particular, the necessary optimality conditions (2.2) are satisfied. This idea gives rise to Algorithm 2.

Note that particularly the solution method presented in Algorithm 1 is widely popular and represents the state-of-the-art algorithm for solving topology optimization problems with a level-set function.

### 3 Quasi-Newton methods for topology optimization

In this section, we present novel quasi-Newton methods for topology optimization. To do so, we first take a different perspective on equation (2.4) and formulate a new gradient descent method for topology optimization. This enables us to define (limited memory) BFGS methods for topology optimization afterwards.

#### 3.1 A novel perspective on the level-set evolution equation

Before we can introduce quasi-Newton methods for topology optimization, we note that the algorithmic frameworks

presented in Algorithms 1 and 2 are not suitable for defining such methods. The reason for this is that the update rules for the level-set function given in (2.6) and (2.7) do consider a convex combination of the level-set function and the generalized topological derivative, which is different to the classical form of descent methods, where the update of the design variables (the level-set function) is performed by subtracting the gradient (topological derivative), scaled by an appropriate step size, from the current iterate.

To remedy this problem, we start by considering the continuous equation (2.4) for evolving the level-set function from Amstutz and Andrä (2006). Instead of using the elaborate approach of Amstutz and Andrä (2006), we discretize (2.4) by an explicit Euler method, which yields the discretized equation

$$\frac{\psi_{k+1} - \psi_k}{\Delta t} = P_{\psi_k^\perp}(DJ(\Omega_k)) \iff \psi_{k+1} = \psi_k + \Delta t P_{\psi_k^\perp}(DJ(\Omega_k)) \tag{3.1}$$

where  $k$  denotes the current time step. For discretizing equation (2.4), one usually would consider small time steps  $\Delta t$  converging to 0, which would yield a so-called gradient flow method. However, we change our viewpoint and interpret (3.1) as gradient descent method, where the time step  $\Delta t$  now plays the role of a step size. The benefit of this interpretation is that we can (potentially) make use of the convergence behavior of the gradient descent method and use large step sizes when appropriate, reducing the computational cost of our algorithm. Therefore, we can now interpret  $g_k = -P_{\psi_k^\perp}(DJ(\Omega_k))$  as the “gradient” associated to our topology optimization problem. The resulting optimization algorithm is presented in Algorithm 3. Note, that the main benefit of this method is that it follows the “standard” form of a gradient descent method, which makes it amenable to define quasi-Newton methods, which we do in the next section. Additionally, our numerical experiments in Sect. 4 show, that Algorithm 3 can also yield faster convergence compared to Algorithms 1 and 2.

---

**Algorithm 3:** Gradient descent method for topology optimization.

---

**Input:** Initial level-set function  $\psi_0$ , initial step size  $\lambda_0 = 1$ , stopping tolerance  $\tau > 0$ , maximum number of iterations  $k_{\max} \in \mathbb{N}$

- 1 **for**  $k = 0, 1, 2, \dots, k_{\max}$  **do**
- 2     Compute the generalized topological derivative  $\mathcal{D}J(\Omega_k)$
- 3     Set  $g_k = -P_{\psi_k^\perp}(\mathcal{D}J(\Omega_k)) = -\left(\mathcal{D}J(\Omega_k) - \frac{(\mathcal{D}J(\Omega_k), \psi_k)}{\|\psi_k\|_{L^2(\mathcal{D})}^2} \psi_k\right)$
- 4     Compute  $\theta_k = \arccos\left(\frac{(\psi_k, \mathcal{D}J(\Omega_k))}{\|\psi_k\|_{L^2(\mathcal{D})} \|\mathcal{D}J(\Omega_k)\|_{L^2(\mathcal{D})}}\right)$
- 5     **if**  $\theta_k < \tau$  **then**
- 6         Stop with approximate minimizer  $\Omega_k$
- 7     Set  $\lambda_k = 2\lambda_{k-1}$
- 8     **while**  $J(\psi_k - \lambda_k g_k) > J(\psi_k)$  **do**
- 9         Decrease the step size  $\lambda_k = \frac{\lambda_k}{2}$
- 10     Update the level-set function:  $\psi_{k+1} = \psi_k - \lambda_k g_k$

---



### 3.2 A Limited Memory BFGS Method for Topology Optimization

With the gradient descent method described in the previous section, we now focus our attention to quasi-Newton methods for topology optimization, which can now be derived analogously to the finite-dimensional case (see, e.g., Nocedal and Wright (2006); Kelley (1999)). To do so, we introduce the functions  $s_k = \psi_{k+1} - \psi_k$  and  $y_k = g_{k+1} - g_k$ . The quasi-Newton methods rely on the so-called secant equation, which in our setting can be written as

$$B_{k+1}s_k = y_k,$$

where  $B_{k+1}$  is an isomorphism from  $L^2(D)$  to  $L^2(D)$  which can be seen as approximation of the Hessian, and we denote its inverse by  $H_{k+1}$ . In the following, we will describe a BFGS method for topology optimization based on  $H_k$ , the inverse of the Hessian approximation, which makes it easier to derive the limited-memory version of the method which we have implemented in the software package cashocs (Blauth (2021a, 2023)). In particular, the search direction for the BFGS method is given by

$$p_k = -H_k g_k \tag{3.2}$$

and the update formula for  $H_k$  is given by

$$H_{k+1} = \left( \text{Id}_{L^2(D)} - \frac{s_k \otimes y_k}{(y_k, s_k)_{L^2(D)}} \right) H_k \left( \text{Id}_{L^2(D)} - \frac{y_k \otimes s_k}{(s_k, y_k)_{L^2(D)}} \right) + \frac{s_k \otimes s_k}{(y_k, s_k)_{L^2(D)}}, \tag{3.3}$$

where  $\otimes$  denotes the outer product of  $L^2(D)$ , i.e.,  $(a \otimes b)c = (b, c)_{L^2(D)}a$ .

To avoid storing large dense matrices as discretizations of the operator  $H_k$ , we employ a limited memory BFGS method for our numerical implementation which is shown in Algorithm 4. Particularly, the limited memory BFGS method only requires us to additionally store  $s_k$  and  $y_k$  to compute the application of  $H_k$  to some right-hand side, which is shown in Algorithm 5 (cf. Nocedal and Wright (2006) for a description of the L-BFGS method in a finite-dimensional setting).

**Remark 3.1** Our numerical experiments presented in Sect. 4 showed that the search direction computed with (3.2) may sometimes not yield descent of the cost functional. Therefore, we employ a restarted version of the BFGS method, which replaces the search direction with  $p_k = -g_k$  if the line search procedure in line 9 of Algorithm 4 does not converge.

**Remark 3.2** The idea presented above could also be applied analogously to derive nonlinear conjugate gradient (NCG) methods for topology optimization. As a thorough investigation of the several popular NCG methods is beyond the scope of this paper, we do not consider these methods in the following (cf. Blauth (2021b) for a discussion of NCG methods in the context of shape optimization). However, the NCG methods are already implemented and ready for use in our software package cashocs (Blauth (2021a, 2023)). An investigation of such NCG methods for topology optimization is planned for future research.

---

**Algorithm 4:** Limited-memory BFGS method for topology optimization.

---

**Input:** Initial level-set function  $\psi_0$ , initial step size  $\lambda_0 = 1$ , initial inverse Hessian approximation  $H_0 = \text{Id}_{L^2(D)}$ , stopping tolerance  $\tau > 0$ , maximum number of iterations  $k_{\max} \in \mathbb{N}$ , memory size  $m \geq 0$

- 1 **for**  $k = 0, 1, 2, \dots, k_{\max}$  **do**
- 2     Compute the generalized topological derivative  $\mathcal{D}J(\Omega_k)$
- 3     Set  $g_k = -P_{\psi_k^\perp}(\mathcal{D}J(\Omega_k)) = -\left( \mathcal{D}J(\Omega_k) - \frac{(\mathcal{D}J(\Omega_k), \psi_k)}{\|\psi_k\|_{L^2(D)}^2} \psi_k \right)$
- 4     Compute the search direction  $p_k = -H_k g_k$  via Algorithm 5
- 5     Compute  $\theta_k = \arccos\left( \frac{(\psi_k, \mathcal{D}J(\Omega_k))}{\|\psi_k\|_{L^2(D)} \|\mathcal{D}J(\Omega_k)\|_{L^2(D)}} \right)$
- 6     **if**  $\theta_k < \tau$  **then**
- 7         Stop with approximate minimizer  $\Omega_k$
- 8     Set  $\lambda_k = 1$
- 9     **while**  $J(\psi_k + \lambda_k p_k) > J(\psi_k)$  **do**
- 10         Decrease the step size  $\lambda_k = \frac{\lambda_k}{2}$
- 11     Update the level-set function:  $\psi_{k+1} = \psi_k + \lambda_k p_k$
- 12     Store  $y_k = g_{k+1} - g_k$ ,  $s_k = \lambda_k p_k$ , and  $\rho_k = \frac{1}{(y_k, s_k)}$

---

---

**Algorithm 5:** Limited-memory double loop for computing  $H_k b$ .

---

**Input:** Right-hand side  $b$  and stored elements  $y_k, s_k, \rho_k$  for  $i = k - m, \dots, k - 1$

- 1  $q = b$
- 2 **for**  $i = k - 1, k - 2, \dots, k - m$  **do**
- 3     Store  $\alpha_i = \rho_i(s_i, q)$
- 4      $q = q - \alpha_i y_i$
- 5      $H_0^k = \begin{pmatrix} s_{k-1}, y_{k-1} \\ y_{k-1}, y_{k-1} \end{pmatrix} \text{Id}_{L^2(D)}$
- 6      $r = H_0^k q$
- 7     **for**  $i = k - m, k - m + 1, \dots, k - 1$  **do**
- 8          $\beta_i = \rho_i(y_i, r)$
- 9          $r = r + (\alpha_i - \beta_i) s_i$
- 10 **return**  $H_k b = r$

---

### 4 Numerical investigation of quasi-Newton methods for topology optimization

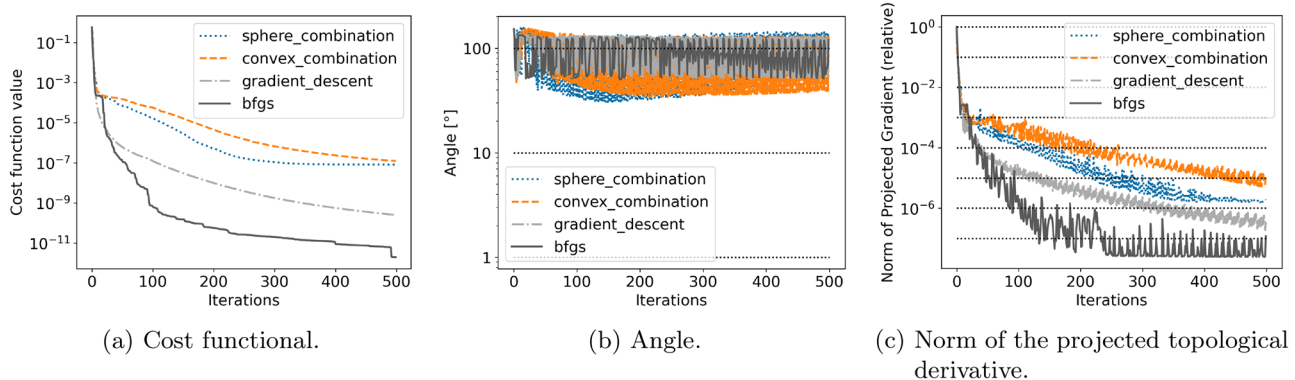
In this section, we consider the practical performance of the proposed quasi-Newton methods from Sect. 3. We consider four problem classes: Inverse topology optimization problems constrained by linear and semilinear Poisson problems, compliance minimization in linear elasticity, and the topological design optimization in Navier–Stokes flow. We solve each of these problems with the four solution algorithms presented in this paper, where we use the following notation for the sake of simplicity. Algorithm 1, originally introduced in Amstutz and Andrä (2006), is called the sphere combination method, Algorithm 2 is called convex combination method, Algorithm 3 is called gradient descent method and, finally, Algorithm 4 is called (limited memory) BFGS method. We remark that, for all numerical test cases considered in this paper, we choose a memory size of  $m = 5$  for the limited memory BFGS methods.

Note that we have implemented all of the optimization algorithms for topology optimization considered in Sect. 2 and the novel gradient descent and BFGS methods from Sect. 3 in our open-source software package cashocs (Blauth (2021a, 2023)), which is a software for solving arbitrary PDE

constrained shape optimization and optimal control problems. Our software is based on the finite element software FEniCS (Alnæs et al. (2015); Logg et al. (2012)) and derives the necessary adjoint systems for the optimization with the help of automatic differentiation. Therefore, for the discretization of the PDE constraints, the finite element method is naturally employed. Moreover, the source code for our numerical experiments is available freely on GitHub (Blauth and Sturm (2023)).




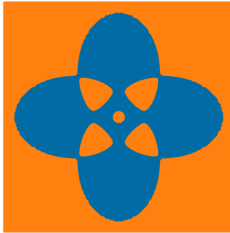



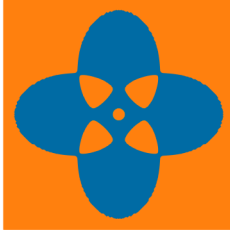








#### 4.1 Linear poisson problem

In this section, we investigate a topology optimization problem constrained by a linear Poisson problem. Let  $D \subset \mathbb{R}^d$  with  $d \in \mathbb{N}_{>0}$  be an open and bounded domain with boundary  $\partial D$  and let  $\Omega \subset D$  be an open subset. We denote by  $\Gamma = \partial\Omega \setminus \partial D$  the interior boundary of  $\Omega$  in  $D$  and by  $\Omega^c = D \setminus \bar{\Omega}$  the complement of  $\Omega$  in  $D$ . We consider the following problem



**Fig. 1** Evolution of the optimization for the linear poisson problem (4.1)

**Table 1** Evolution of the geometries for the linear Poisson problem (4.1)

Algorithm	50 iterations	100 iterations	500 iterations	Reference
Sphere combination				
Convex combination				
Gradient descent				
BFGS				

$$\begin{aligned}
 \min_{\Omega} J(\Omega, u) &= \frac{1}{2} \int_D (u - u_{\text{des}})^2 \, dx \\
 \text{s.t.} \quad -\Delta u + \alpha_{\Omega} u &= f_{\Omega} \quad \text{in } D, \\
 u &= 0 \quad \text{on } \partial D,
 \end{aligned}
 \tag{4.1}$$

where  $\alpha_{\Omega}(x) = \chi_{\Omega}(x)\alpha_{\text{in}} + \chi_{\Omega^c}(x)\alpha_{\text{out}}$  and  $f_{\Omega}(x) = \chi_{\Omega}(x)f_{\text{in}} + \chi_{\Omega^c}(x)f_{\text{out}}$  with  $\alpha_{\text{in}}, \alpha_{\text{out}} > 0$  as well as  $f_{\text{in}}, f_{\text{out}} \in \mathbb{R}$  are constant in  $\Omega$  and  $\Omega^c$ . In our setting,  $u_{\text{des}}$  is given as the solution of the PDE constraint on a desired domain  $\Omega_{\text{des}}$ . Hence, the above problem can be interpreted as an inverse problem of identifying the unknown domain  $\Omega_{\text{des}}$  using the measurement  $u_{\text{des}}$ . The generalized topological derivative for this problem is derived, e.g., in Amstutz (2022) and is given by

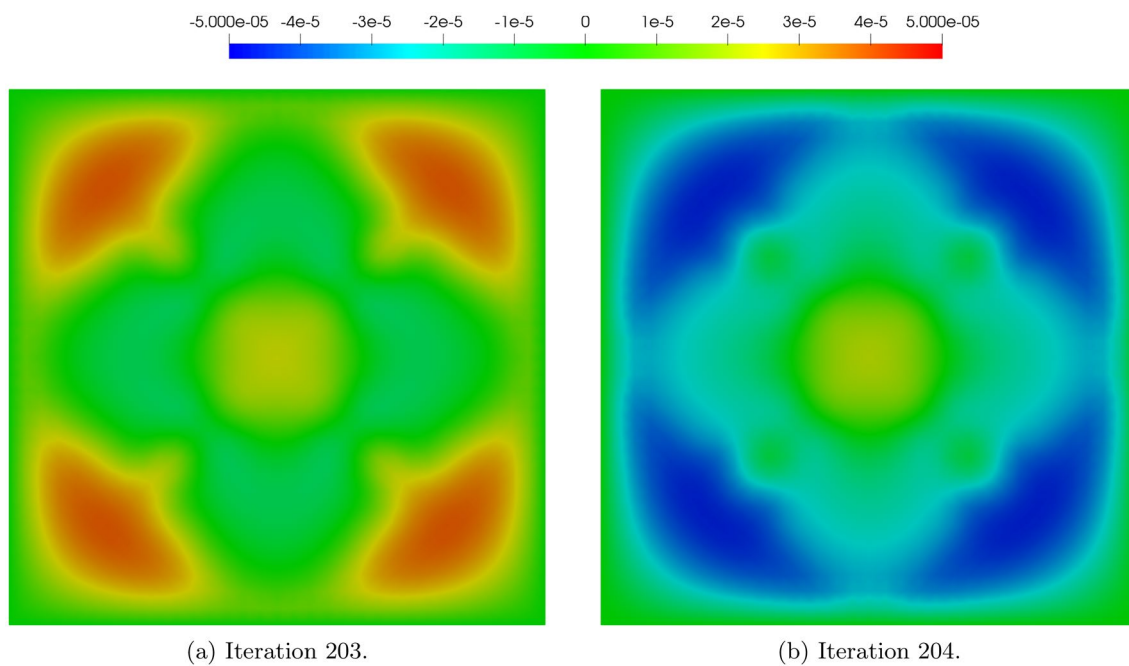
$$DJ(\Omega)(x) = (\alpha_{\text{out}} - \alpha_{\text{in}})u(x)p(x) - (f_{\text{out}} - f_{\text{in}})p(x) \quad \text{for all } x \in D \setminus \Gamma,$$

where  $u$  solves the state equation (4.1) and  $p$  solves the following adjoint equation

$$\begin{aligned}
 -\Delta p + \alpha_{\Omega} p &= -(u - u_{\text{des}}) \quad \text{in } D, \\
 p &= 0 \quad \text{on } \partial D.
 \end{aligned}$$

**Remark 4.1** Usually, the term ‘‘inverse problem’’ is used to denote a identification problem which aims to reconstruct some unknown domain  $\Omega_{\text{des}} \subset D$  using measurements obtained on (parts of) the boundary of  $D$ , see, e.g., Canelas et al. (2015); Hintermüller and Laurain (2008); Laurain et al. (2013); Beretta et al. (2017). However, for our model problems (4.1) and (4.2), we use (artificial) measurements in the entire domain  $D$ . Still, we refer to these problems as inverse problems for the sake of simplicity.





**Fig. 2** Plots of the topological derivative at two iterations of the sphere combination algorithm for problem (4.1)

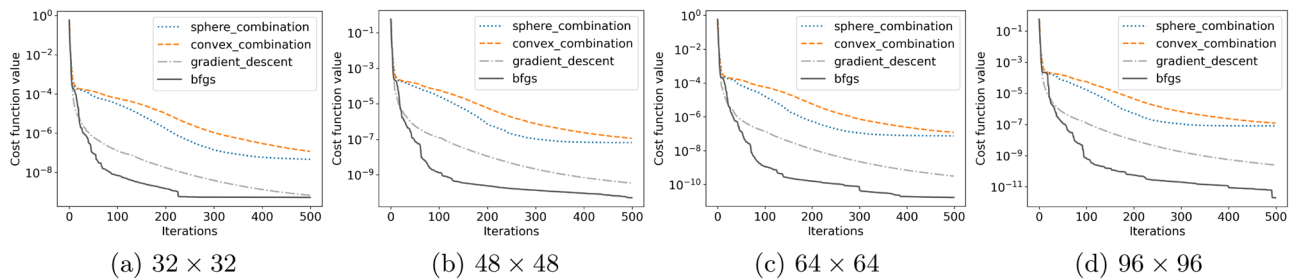
In the following, we solve this problem with the four optimization algorithms described in Sects. 2 and 3, where we consider a hold-all domain of  $D = (-2, 2)^2$ . We discretize the geometry using four different mesh sizes to investigate the dependence of the algorithms on the discretization. The considered meshes are generated by creating a uniform quadrangular grid with  $32 \times 32$ ,  $48 \times 48$ ,  $64 \times 64$ , and  $96 \times 96$  squares, where each square is in turn subdivided into four triangles, so that the meshes consist of 2113 nodes and 4096 triangles ( $32 \times 32$ ), 4705 nodes and 9216 triangles ( $48 \times 48$ ), 8321 nodes and 16384 triangles ( $64 \times 64$ ), and 18625 nodes and 36864 triangles ( $96 \times 96$ ), respectively. Moreover, we discretize both the state and adjoint variables with linear Lagrange finite elements. For the parameters, we use  $\alpha_{in} = f_{in} = 10$  and  $\alpha_{out} = f_{out} = 1$ . The sought design  $\Omega_{des}$  is chosen as the one corresponding to a clover shape, which can be seen in the right-most column of Table 1.

The results of the optimization can be seen in Fig. 1, where we show the evolution of the cost functional, the angle criterion, and the norm of the projected gradient (2.5) over the optimization for the finest discretization. Here, we observe that our proposed gradient descent and BFGS methods perform significantly better than the established methods. We observe that the convergence behavior, based on the cost functional and the norm of the projected gradient, is best for the BFGS method, followed by the gradient descent and sphere combination methods, and that the convex combination algorithm performs worst. Particularly, the

BFGS method reaches stationarity in the cost functional and norm of the projected gradient after only about 125 iterations, whereas all remaining methods continue to decrease the respective measures until the final iteration.

However, none of the methods converged based on the angle criterion after 500 iterations. On the contrary, the angle between level-set function and generalized topological derivative remains bounded from below by about  $30^\circ$ , so that none of the methods can be considered as converged by this criterion. However, if we take a look at the evolution of the relative norm of the projected gradient, we observe a steep decrease for all methods. The convex and sphere combination methods are able to decrease the norm of the projected gradient by about five and six orders of magnitude, respectively. The gradient descent and BFGS methods are even more efficient and decrease the norm of the projected gradient by about seven and eight orders of magnitude, respectively. Therefore, based on the second convergence criterion, the algorithms can be considered converged. Moreover, for this example, it seems like the angle criterion is too strict as a convergence criterion, whereas the norm of the projected gradient seems to be better suited.

**Remark 4.2** A possible explanation for this behavior is the fact, that problem (4.1) is well-known to be ill-conditioned. This means that topological derivatives become flatter the closer we get to the optimal solution of the problem, which is also what we have observed in our numerical experiments.



**Fig. 3** Evolution of the cost functional for the linear Poisson problem (4.1) for several discretization levels

The topological derivative becomes successively smaller and flatter over the course of the optimization.

In Fig. 2, we show plots of the topological derivative during the middle of the optimization for the sphere combination method. In Fig. 2a, the topological derivative is shown at iteration 203 and for this iterate, the angle between topological derivative and level-set function is comparatively small with  $29^\circ$ . On the other hand, the topological derivative in the next iteration, which is shown in Fig. 2b, has a much larger angle of  $130^\circ$  with the level-set function. A possible reason for this behavior could be the line search, which, at first, uses successively larger step sizes up to iteration 204, after that the accepted step size drops down. This behavior repeats and causes the oscillations in the angle criterion which can be seen in Fig. 1.

Moreover, we remark that the angle criterion only provides a necessary condition for optimality, as can be easily seen by the discussion in Sect. 2.1 (cf. (2.2) and (2.3)). There may be other minimizers that satisfy (2.3) without satisfying the angle criterion. A thorough investigation of these issues is an interesting direction for future work.

The evolution of the geometry during the optimization algorithms is depicted in Table 1, where we show the geometries after 50, 100, and 500 iterations for all four considered optimization algorithms on the finest discretization. In addition, we also show the reference shape for comparison. Note, that the reference shape has four larger inclusions in the middle and a smaller one in the center, making the problem particularly hard to solve. Comparing the obtained shapes after 50 iterations, we observe that the sphere combination and convex combination algorithms only start to form the four major inclusions, whereas they are already present for the gradient descent and BFGS methods. Moreover, the geometry obtained by BFGS method already has an inclusion in the center of the geometry, making it already very similar to the reference solution. After 100 iterations, the established methods have formed the major inclusions, but their position and shape is still quite wrong. The gradient descent method has improved the shapes of the major inclusions so that they are rather similar to the desired ones and

is starting to locate the inclusion in the center. For the BFGS method, there is no visible difference anymore between the obtained and desired geometry, indicating that the method has already converged after 100 iterations, whereas the other methods are still quite far away from the desired shape. After 500 iterations, we observe that the sphere combination and convex combination methods have corrected the shape of the four major inclusions so that they are now quite similar to the desired ones. However, neither of these methods was able to reconstruct the inclusion in the center of the geometry. The gradient descent method, on the other hand, was able to reconstruct the center inclusion after 500 iterations, and there are no visual distinctions between the obtained and desired geometry anymore. The same is, of course, also true for the BFGS method, whose corresponding shape does not change visually between iterations 100 and 500.

Let us finally investigate the dependence of the algorithms on the discretization. We do so by comparing the evolution of the cost functional for all methods on the different discretization levels. The results are depicted in Fig. 3. Here, we observe that the performance of the algorithms, which we have discussed previously, is not dependent on the mesh size. In particular, the BFGS and gradient descent methods always perform best, whereas the sphere and convex combination methods perform significantly worse. Overall, the cost functional decreases further when a finer discretization is chosen. This is due to the fact, that a finer discretization also allows for a more detailed resolution of the desired shape, which in turn results in a lower cost functional value for the optimum. Therefore, we conclude that all algorithms behave mesh-independently, i.e., they do not require more iterations for finer levels of discretization to reach the same level of approximation of the optimal solution.

These results show the great potential of the proposed BFGS methods as they performed best of all considered methods, significantly outperforming the remaining algorithms. Moreover, we have shown that the methods also show mesh-independent behavior, which is due to the fact that the presented algorithms are discretizations of optimization algorithms acting in infinite-dimensional spaces.

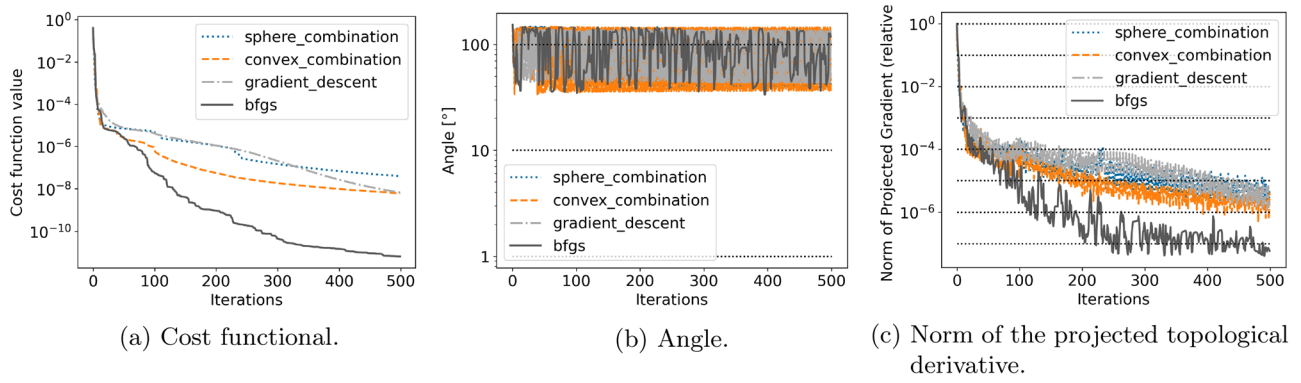


Fig. 4 Evolution of the optimization for the semilinear Poisson problem (4.2)

Table 2 Evolution of the geometries for the semilinear Poisson problem (4.2)

Algorithm	100 iterations	200 iterations	500 iterations	Reference
Sphere combination				
Convex combination				
Gradient descent				
BFGS				

### 4.2 Semilinear poisson problem

To showcase the methods’ performance for nonlinear problems, we now investigate a semilinear variant of the Poisson problem we considered in Sect. 4.1. Our setting is like before, so  $D \subset \mathbb{R}^d$  with  $d \in \mathbb{N}_{>0}$  and  $d \leq 3$  denotes the holdall domain,  $\partial D$  is its boundary,  $\Omega \subset D$  is an open subset of  $D$ ,  $\Gamma = \partial\Omega \setminus \partial D$  is the interior boundary of  $\Omega$  in  $D$ , and  $\Omega^c = D \setminus \overline{\Omega}$  is the complement of  $\Omega$  in  $D$ . Our semilinear version of the problem reads

$$\begin{aligned} \min_{\Omega} J(\Omega, u) &= \frac{1}{2} \int_D (u - u_{\text{des}})^2 \, dx \\ \text{s.t.} \quad -\Delta u + \alpha_{\Omega} u^3 &= f_{\Omega} \quad \text{in } D, \\ u &= 0 \quad \text{on } \partial D, \end{aligned} \tag{4.2}$$

where we again have  $\alpha_{\Omega}(x) = \chi_{\Omega}(x)\alpha_{\text{in}} + \chi_{\Omega^c}(x)\alpha_{\text{out}}$  and  $f_{\Omega}(x) = \chi_{\Omega}(x)f_{\text{in}} + \chi_{\Omega^c}(x)f_{\text{out}}$  with  $\alpha_{\text{in}}, \alpha_{\text{out}} > 0$  as well as  $f_{\text{in}}, f_{\text{out}} \in \mathbb{R}$ . The topological derivative for this problem is given by

$$DJ(\Omega)(x) = (\alpha_{\text{out}} - \alpha_{\text{in}})u(x)^3 p(x) - (f_{\text{out}} - f_{\text{in}})p(x) \quad \text{for all } x \in D \setminus \Gamma,$$

where  $u$  solves the state equation and  $p$  solves the following adjoint equation

$$\begin{aligned} -\Delta p + 3\alpha_{\Omega} u^2 p &= -(u - u_{\text{des}}) \quad \text{in } D, \\ p &= 0 \quad \text{on } \partial D. \end{aligned}$$

We again solve this problem with the four solution algorithms under consideration and use the hold-all domain  $D = (-2, 2)^2$ . The desired state  $u_{\text{des}}$  is obtained by solving the semilinear Poisson problem on a reference domain, which is given by the same clover shape as considered in Sect. 4.1. The geometry is discretized by dividing it into  $96 \times 96$  squares, which are subdivided into four triangles each, so that we use 18625 nodes and 36864 triangles. Again, we employ linear Lagrange elements for the discretization of the state and adjoint variables. Finally, we use the same setting for  $\alpha$  and  $f$  as before, so that  $\alpha_{\text{in}} = f_{\text{in}} = 10$  and  $\alpha_{\text{out}} = f_{\text{out}} = 1$ .

The evolution of the cost functional, angle criterion, and norm of the projected topological derivative over the course of the optimization are shown in Fig. 4. Analogously to our previous findings, we observe that the BFGS method significantly outperforms the remaining methods as it decreases the cost functional and the norm of the projected topological derivative most. For this problem, we observe that the gradient descent method does not perform as well as previously, but is rather comparable to the sphere and convex combination methods in performance. Again, for all considered algorithms, the angle between level-set function and generalized topological derivative remained bounded away from zero.

Let us investigate the geometries obtained by the methods, which are depicted in Table 2 after 100, 200, and 500 iterations of the methods. Here, we observe that the BFGS method again performs substantially better than the other methods. Even after 100 iterations, all inclusions of the geometry are found and the geometry exhibits the correct topology. However, the shape of the inclusions is still a bit off. The geometry obtained with the remaining algorithms, on the other side, is still far away from the optimal geometry. The same is true after 200 iterations: There, the geometry obtained by the BFGS method is visually identical to the reference solution, whereas the other methods produce geometries that still deviate significantly, both in topology and shape, from the optimal solution. Finally, the results after 500 iterations show, that the sphere and convex combination methods are able to locate the four major inclusions of the reference solution, but their shape is still a bit off. Moreover, they were unable to detect the inclusion in the center. The gradient descent method, on the other hand, was able to find all inclusions and, hence, to reconstruct the sought topology. However, the shape of the inclusions could still be improved. For the BFGS method, there are no major changes in the geometry between 200 and 500 iterations, as the geometry showed no visual differences to the reference solution already after 200 iterations.

Again, these results highlight the potential and efficiency of the BFGS methods for solving topology optimization problems, as they significantly outperformed all other methods considered in this paper.

### 4.3 Compliance minimization in linear elasticity

In this section, we consider the problem of compliance minimization in linear elasticity which has been previously investigated, e.g., in Amstutz and Andrä (2006); Allaire et al. (2004, 2005). Let  $D \subset \mathbb{R}^d$ ,  $d \in \mathbb{N}_{>0}$  be an open and bounded domain with boundary  $\partial D$ , which is the disjoint union of the Dirichlet boundary  $\Gamma_D$  and Neumann boundary  $\Gamma_N$ . Further, let  $\Omega \subset D$  and denote its complement by  $\Omega^c = D \setminus \overline{\Omega}$ . The compliance minimization problem is given by

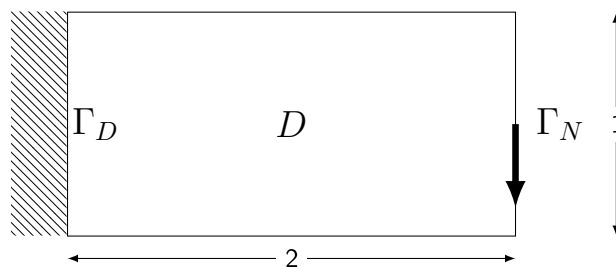


Fig. 5 Schematic of the cantilever problem

$$\begin{aligned}
 \min_{\Omega} J(\Omega, u) &= \int_D \alpha_{\Omega} \sigma(u) : \epsilon(u) \, dx + l|\Omega| \\
 \text{s.t.} \quad & -\operatorname{div}(\alpha_{\Omega} \sigma(u)) = f \quad \text{in } D, \\
 & u = 0 \quad \text{on } \Gamma_D, \\
 & \alpha_{\Omega} \sigma(u) n = g \quad \text{on } \Gamma_N.
 \end{aligned}
 \tag{4.3}$$

Here,  $u$  is the deformation of a linear elastic material,  $\sigma(u) = 2\mu e(u) + \lambda \operatorname{tr}e(u)I$  is Hooke's tensor,  $e(u) = 1/2(\nabla u + (\nabla u)^T)$  is the symmetric gradient of  $u$ , and  $A : B$  denotes the Frobenius inner product between matrices  $A, B \in \mathbb{R}^{d \times d}$ , i.e.,  $A : B := \sum_{i,j=1}^d a_{ij} b_{ij}$ . Here,  $\mu$  and  $\lambda$  are the so-called Lamé parameters for which we assume  $\mu > 0$  and  $2\mu + d\lambda > 0$ . Moreover,  $\alpha_{\Omega}(x) = \chi_{\Omega}(x)\alpha_{\text{in}} + \chi_{\Omega^c}(x)\alpha_{\text{out}}$  is, again, constant in  $\Omega$  and  $\Omega^c = D \setminus \overline{\Omega}$  with  $\alpha_{\text{in}}, \alpha_{\text{out}} > 0$ . The first term in the cost functional measures the compliance of the structure and the second term is a regularization

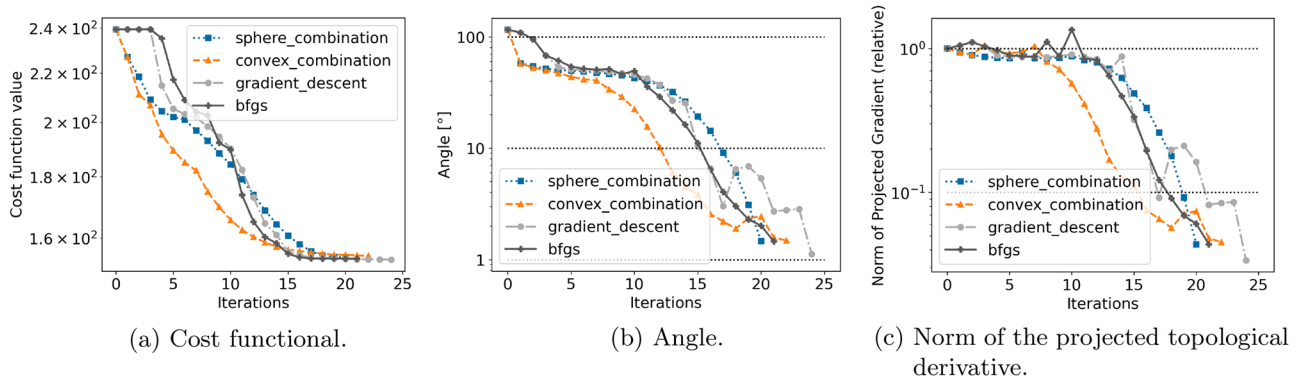


Fig. 6 Evolution of the optimization for the cantilever problem

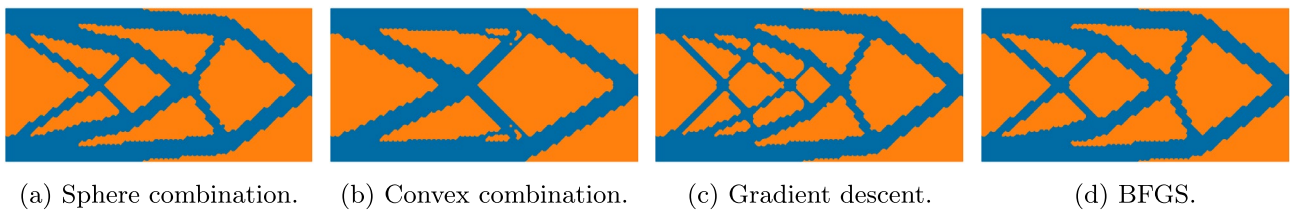


Fig. 7 Optimized geometries for the cantilever problem

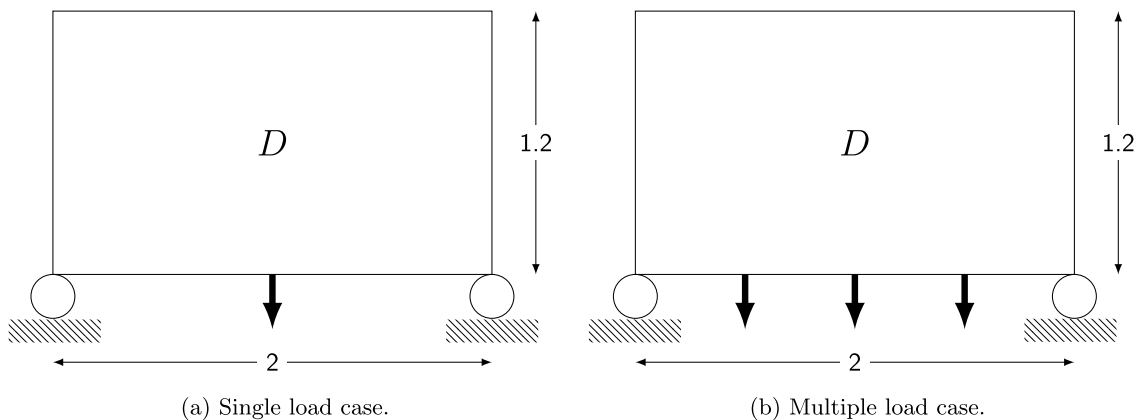


Fig. 8 Schematic setup of the bridge problem



parameter which penalizes large domains  $\Omega$ , so that the optimization is not trivial.

The topological derivative for problem (4.3) can be found, e.g., in Amstutz and Andr  (2006) and is given by

$$DJ(\Omega)(x) = \begin{cases} -\alpha_{\text{in}} \frac{r_{\text{in}}-1}{\kappa r_{\text{in}}+1} \frac{\kappa+1}{2} \left( 2\sigma(u) : e(u) + \frac{(r_{\text{in}}-1)(\kappa-2)}{\kappa+2r_{\text{in}}-1} \text{tr}\sigma(u)\text{tr}e(u) \right) - l & \text{for } x \in \Omega, \\ -\alpha_{\text{out}} \frac{r_{\text{out}}-1}{\kappa r_{\text{out}}+1} \frac{\kappa+1}{2} \left( 2\sigma(u) : e(u) + \frac{(r_{\text{out}}-1)(\kappa-2)}{\kappa+2r_{\text{out}}-1} \text{tr}\sigma(u)\text{tr}e(u) \right) + l & \text{for } x \in \Omega^c = D \setminus \bar{\Omega}, \end{cases}$$

where  $r_{\text{in}} = \frac{\alpha_{\text{out}}}{\alpha_{\text{in}}}$ ,  $r_{\text{out}} = \frac{\alpha_{\text{in}}}{\alpha_{\text{out}}}$ , and  $\kappa = \frac{\lambda+3\mu}{\lambda+\mu}$ .

In the following, we consider two test cases for this problem, namely the so-called cantilever and bridge problems, which are taken from Amstutz and Andr 

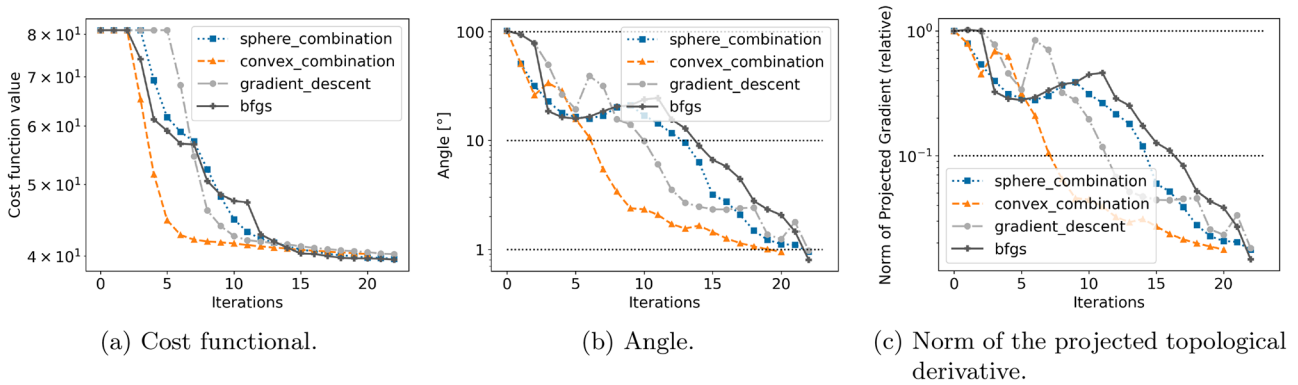


Fig. 9 Evolution of the optimization for the bridge with a single load

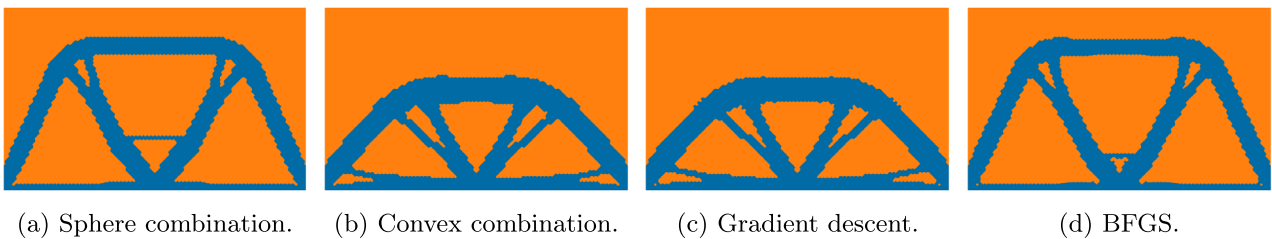


Fig. 10 Optimized geometries for the bridge with a single load

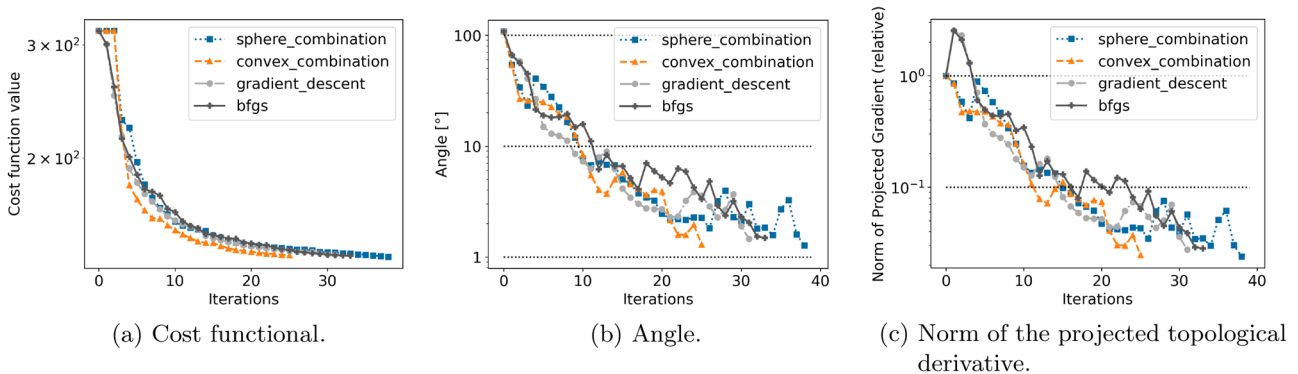


Fig. 11 Evolution of the optimization for the bridge with multiple loads



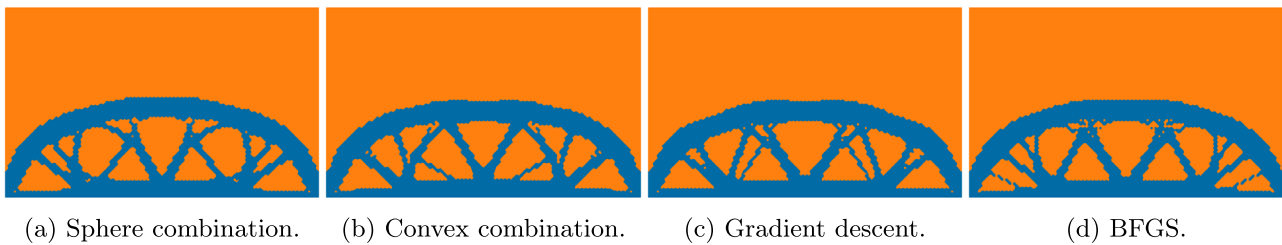


Fig. 12 Optimized geometries for the bridge with multiple load

(2006). For these test cases, we follow Amstutz and Andrä (2006) and use  $\alpha_{in} = 1$  and  $\alpha_{out} = 1 \times 10^{-3}$  as well as  $f = 0$ . Additionally, as we consider the case of plane stress, the Lamé parameters are given by  $\mu = \frac{E}{2(1+\nu)}$  and  $\lambda = \frac{2\mu\lambda^*}{\lambda^* + 2\mu}$  with  $\lambda^* = \frac{E\nu}{(1+\nu)(1-2\nu)}$  and we use a Young’s modulus of  $E = 1$  and a Poisson’s ratio of  $\nu = 0.3$  for the following numerical experiments. Finally, we discretize the state variable using linear Lagrange finite elements.

### 4.3.1 Cantilever

For our first example, we consider the so-called cantilever problem (see, e.g., Amstutz and Andrä (2006); Allaire et al. (2004, 2005)). Here, the holdall domain is given by  $D = (0, 2) \times (0, 1)$ . The Dirichlet boundary  $\Gamma_D = \{x = 0\}$  is the left side of the rectangle and for the Neumann load  $g$  we consider a unitary point load at  $(2, 0.5)$ . As in Amstutz and Andrä (2006), we discretize this domain with a uniform triangular mesh consisting of 4193 nodes and 8192 triangles and choose  $l = 100$ . A schematic of the problem setting can be seen in Fig. 5.

The evolution of the cost functional, angle criterion, and norm of the projected topological derivative are shown in Fig. 6. Here, we observe that all methods converge very fast, requiring a maximum of 25 iterations to satisfy the angle criterion with a tolerance of  $1.5^\circ$ . The performance of all methods is very comparable for this problem and no method performs significantly better or worse than the others.

However, when we investigate the optimized geometries obtained with the methods, there are some differences, as they converged to different local minimizers of the problems. Whereas the sphere combination and BFGS methods converged to the same solution that was reported in Amstutz and Andrä (2006), the gradient descent and convex combination method converged to different geometries with finer beam structures (Fig. 7).

### 4.3.2 Bridge

In this section, we consider another problem of compliance minimization in linear elasticity which corresponds to a bridge with a single and multiple loads. As before, these problems are taken from the literature (Amstutz and Andrä (2006); Allaire et al. (2004, 2005)). Here, the holdall domain is given by  $D = (0, 2) \times (0, 1.2)$ . As boundary conditions, we have zero vertical displacement at the bottom left and right corners. For the single load case, we apply a vertical downwards unitary force at  $(1, 0)$  and for the multiple load case we apply three vertical unitary loads at  $(0.5, 0)$ ,  $(1, 0)$ , and  $(1.5, 0)$ . We discretize this setup with a mesh consisting of 7809 vertices and 15360 triangles. For the volume regularization, we use the parameter  $l = 30$  in the single load case and use  $l = 120$  for the multiple load case, in analogy to Amstutz and Andrä (2006). A schematic of the problem can be seen in Fig. 8.

Let us first discuss the single load case. The history of the cost functional, angle criterion, and norm of the projected topological derivative are shown in Fig. 9. Similarly to the cantilever problem, there are no significant differences in the performance of the optimization algorithms. All methods required between 20 and 22 iterations to reach the prescribed tolerance of  $1^\circ$  of the angle criterion. The only notable difference between the methods is that the convex combination algorithm decreases the quality measures slightly earlier than the other methods, but they “catch up” during later iterations. The optimized geometries obtained with the methods are shown in Fig. 10. Here, we also observe, that the methods converge to different local minimizers. Whereas the sphere combination and BFGS method find a similar geometry, which is the same as reported in Amstutz and Andrä (2006), the convex combination and gradient method find a different local minimizer with less height and a different supporting beam structure.

The results are very similar for the multiple loads case. In Fig. 11 the evolution of the considered quality measures is depicted over the history of the optimization. Here, we

observe some slight differences in the performance of the algorithms. The convex combination method performs best as it requires only 25 iterations to satisfy the angle criterion, where we have chosen a tolerance of  $1.5^\circ$  for this example. The gradient descent and BFGS methods showed the second best performance, requiring about 30 iterations each to satisfy the stopping criterion. The sphere combination method performed worst and required 40 iterations to reach the stopping tolerance. Finally, let us briefly investigate the optimized geometries, which are shown in Fig. 12. Here, we see that all methods converge to a similar solution, which has a slightly more complicated beam structure than the one reported in Amstutz and Andr a (2006).

Altogether, for the case of linear elasticity, we observe no major differences between all considered optimization algorithms. The proposed BFGS methods show a very similar performance to the already established methods on all considered test problems. A possible reason for this behavior is that the problems are comparatively easy to solve, at least in comparison to the problems considered in Sects. 4.1 and 4.2, as the already established methods only require 20 to 50 iterations to solve these problems to a desired tolerance, so that there may not be enough iterations for the BFGS methods to show their potential, particularly if they have to be restarted often due to large changes in the topology. This topic is of interest for future research.

#### 4.4 Optimization of fluids in Navier–Stokes flow

Let us now consider another application, the optimization of fluids in Navier–Stokes flow. Our setting is similar to before, i.e., let  $D \subset \mathbb{R}^d$  be an open and bounded hold-all domain with boundary  $\partial D$ . Let  $\Omega \subset D$ , denote by  $\Omega^c = D \setminus \overline{\Omega}$  the complement of  $\Omega$  in  $D$ , and let  $\Gamma = \partial\Omega \setminus \partial D$ . We consider the following optimization problem

$$\begin{aligned} \min_{\Omega} J(\Omega, u) &= \int_D \mu \nabla u : \nabla u + \alpha_{\Omega} u \cdot u \, dx \\ &\quad -\mu \Delta u + \rho(u \cdot \nabla)u + \nabla p + \alpha_{\Omega} u = 0 \quad \text{in } D, \\ &\quad \nabla \cdot u = 0 \quad \text{in } D, \\ &\quad u = u_D \quad \text{on } \partial D, \end{aligned} \tag{4.4}$$

s.t.

$$\begin{aligned} \int_D p \, dx &= 0, \\ |\Omega| &= \text{vol}_{\text{des}}, \end{aligned}$$

where  $\alpha_{\Omega}(x) = \chi_{\Omega}(x)\alpha_{\text{in}} + \chi_{\Omega^c}(x)\alpha_{\text{out}}$  with  $\alpha_{\text{in}}, \alpha_{\text{out}} > 0$  and  $|\Omega|$  denotes the Lebesgue measure in  $\mathbb{R}^d$ . Here,  $u$  denotes the fluid’s velocity and  $p$  its pressure,  $\mu$  is its viscosity,  $\rho$  its density, and  $\alpha$  is the inverse permeability. The cost functional of the above problem models the energy dissipation of the fluid, which should be minimized. Moreover, we have a volume equality constraint, which ensures that only the

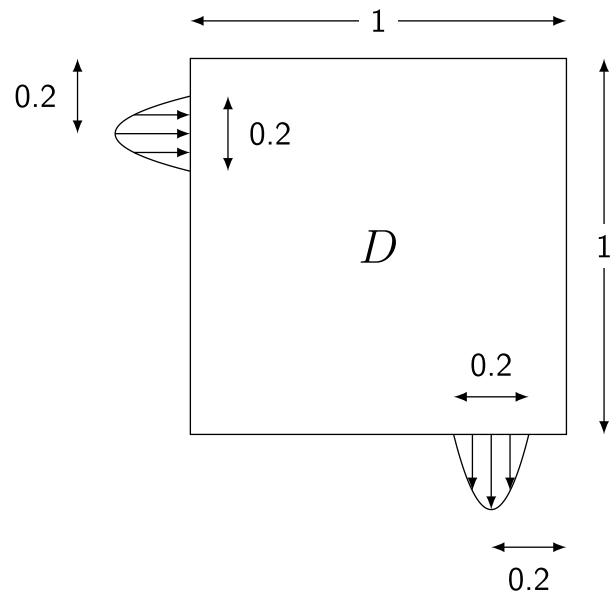


Fig. 13 Schematic setup of the pipe bend problem

desired volume of the domain is occupied by the fluid. For the topology optimization problem (4.4), the sought domain  $\Omega$  is the domain of the fluid, whereas its complement  $\Omega^c$  plays the role of a solid region. In particular, the inverse permeability  $\alpha$  is small inside  $\Omega$  and large outside of it. For a more detailed discussion, we refer the reader to Borrvall and Petersson (2003), where this model was first introduced and used for the topology optimization of fluids.

For our numerical experiments, we regularize the equality constraint with a quadratic penalty term, leading to the following optimization problem

$$\begin{aligned} \min_{\Omega} J(\Omega, u) &= \int_D \mu \nabla u : \nabla u + \alpha_{\Omega} u \cdot u \, dx + \frac{l}{2} (|\Omega| - \text{vol}_{\text{des}})^2 \\ &\quad -\mu \Delta u + \rho(u \cdot \nabla)u + \nabla p + \alpha_{\Omega} u = 0 \quad \text{in } D, \\ &\quad \nabla \cdot u = 0 \quad \text{in } D, \\ \text{s.t.} \quad &\quad u = u_D \quad \text{on } \partial D, \\ &\quad \int_D p \, dx = 0. \end{aligned} \tag{4.5}$$

The topological derivative for problem (4.5) can be found, e.g., in S a et al. (2016) and is given by

$$\begin{aligned} DJ(\Omega)(x) &= (\alpha_{\text{out}} - \alpha_{\text{in}})u(x) \cdot (u(x) + v(x)) \\ &\quad + l(|\Omega| - \text{vol}_{\text{des}}) \quad \text{for } x \in D \setminus \Gamma, \end{aligned}$$

where  $(v, q)$  solves the adjoint Navier–Stokes system

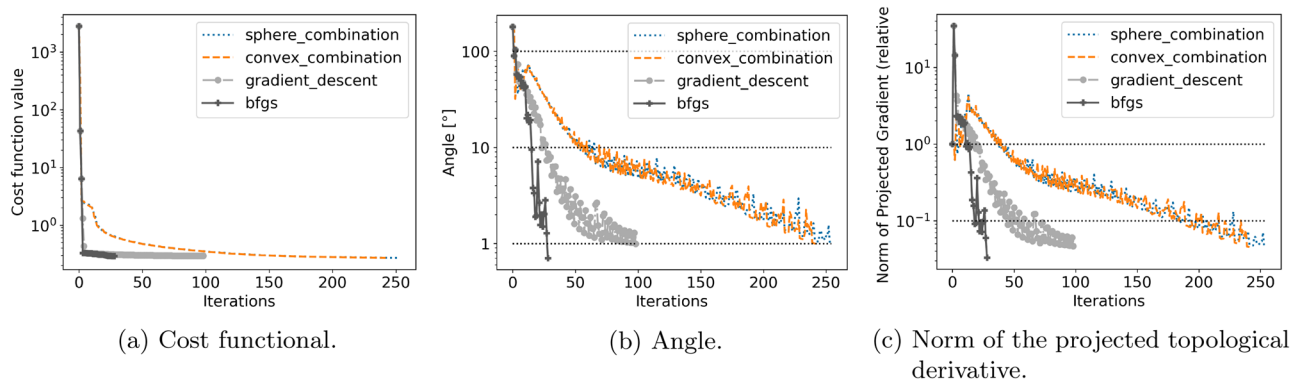


Fig. 14 Evolution of the optimization for the pipe bend problem

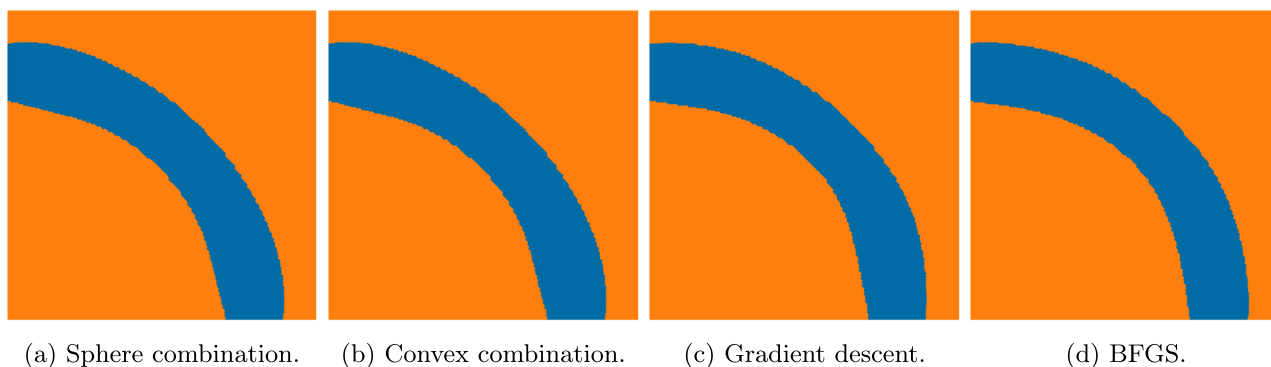


Fig. 15 Optimized geometries for the pipe bend problem

$$\begin{aligned}
 -\mu \Delta v + \rho(Du)^T v - \rho(u \cdot \nabla)v + \nabla q + \alpha_{\Omega} v &= 2(\mu \Delta u - \alpha_{\Omega} u) && \text{in } D, \\
 \operatorname{div}(v) &= 0 && \text{in } D, \\
 v &= 0 && \text{on } \partial D, \\
 \int_D q \, dx &= 0.
 \end{aligned}$$

To model an actual solid region,  $\alpha_{\text{out}}$  should tend to  $+\infty$ . For our numerical investigation, however, we follow Sá et al. (2016) and chose a finite value for  $\alpha_{\text{in}}$  and  $\alpha_{\text{out}}$ , namely we choose

$$\alpha_{\text{in}} = \frac{2.5\mu}{100^2} \quad \alpha_{\text{out}} = \frac{2.5\mu}{0.01^2}.$$

Moreover, we choose a value of  $\rho = 1$  for the fluid’s density and  $\mu = 1 \times 10^{-2}$  for its viscosity. In the following, we will consider two common benchmark problems, which are taken from Borrvall and Petersson (2003); Sá et al. (2016) and consider the problem of constructing a pipe bend and the drag minimization of an obstacle. Note, that for both problems, the hold-all domain is given by  $D = (0, 1) \times (0, 1)$ , and that we discretize this with a uniform mesh consisting of 20201 nodes and 40000 triangles. Moreover, we use the

LBB-stable Taylor-Hood finite element pair of quadratic Lagrange elements for the velocity and adjoint velocity and linear Lagrange elements for the pressure and adjoint pressure.

#### 4.4.1 Pipe bend

Let us first investigate the problem of designing a pipe bend, which is taken from Borrvall and Petersson (2003). For Dirichlet boundary conditions, we prescribe the inlet velocity with the parabolic profile

$$u_D(x) = \begin{cases} 1 - 100(x_2 - 0.8)^2, \\ 0 \end{cases} \quad \text{for } x_1 = 0 \text{ and } 0.7 \leq x_2 \leq 0.9$$

on the top part of the left boundary of  $D$ , and for the outlet velocity we use the profile

$$u_D(x) = \begin{cases} 0, \\ -(1 - 100(x_1 - 0.8)^2) \end{cases} \quad \text{for } x_2 = 0 \text{ and } 0.7 \leq x_1 \leq 0.9$$

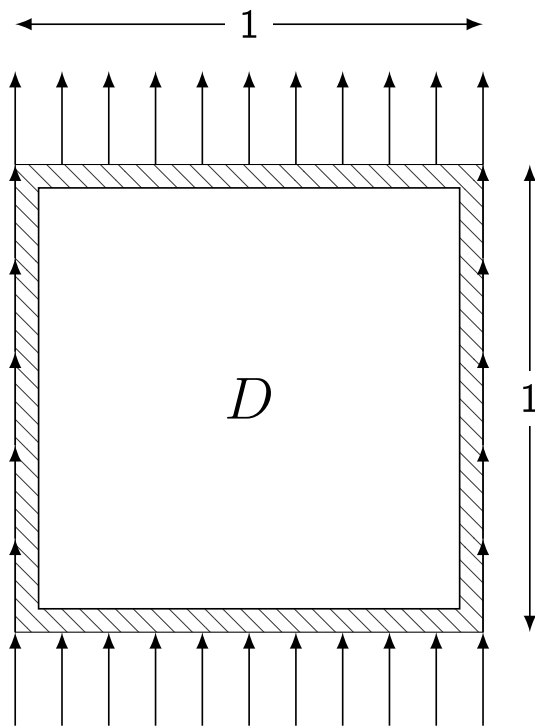


Fig. 16 Schematic of the rugby ball problem

on the right side of the bottom boundary of  $D$ . On all other parts of the boundary, we use a no-slip boundary condition, so that  $u_D = [0, 0]^T$ . The corresponding setup is shown schematically in Fig. 13. For the volume constraint, we proceed according to Borrvall and Petersson (2003) and use a value of  $\text{vol}_{\text{des}} = 0.08\pi$ . Additionally, we choose a regularization parameter of  $l = 1 \times 10^4$  to enforce the volume constraint.

The evolution of the cost functional, angle criterion, and norm of the projected topological derivative can be seen in Fig. 14. We observe that the BFGS method substantially

outperforms the remaining methods as it only required about 30 iterations to reach the stopping criterion and find a local minimizer of the problem. The gradient descent method performed worse, but still way better than the already established methods: The former required around 100 iterations to satisfy the stopping criterion, whereas the sphere and convex combination methods performed very similarly, requiring about 250 iterations to reach a local minimizer. Considering the optimized geometries obtained by the method, which are depicted in Fig. 15, we observe that all methods converge to a similar minimizer and that all optimized geometries show only the slightest visual differences.

#### 4.4.2 Rugby Ball

Let us now consider the rugby-ball problem from Borrvall and Petersson (2003), which is shown schematically in Fig. 16. Here, the goal is to design an obstacle which minimizes the energy dissipation of the flow. For the volume constraint of the problem, we choose  $\text{vol}_{\text{des}} = 0.8$  and use a regularization parameter of  $l = 1 \times 10^4$  to enforce the volume constraint. For the boundary conditions, we prescribe a constant value of  $u_D = [0, 1]^T$  on the entire boundary  $\partial D$ .

The history of the cost functional, angle between topological derivative and level-set function, and the norm of the projected topological derivative are depicted in Fig. 17. Here, we can observe that the BFGS method again outperforms all other methods significantly, as it requires slightly less than 50 iterations to find a local minimizer. The gradient descent method shows the second best performance and is able to satisfy the stopping criterion after around 80 iterations. The performance of the sphere and convex combination methods is, again, very similar and both require slightly more than 100 iterations to converge.

The obtained optimized geometries are shown in Fig. 18. We see that all methods produce the same desired shape,

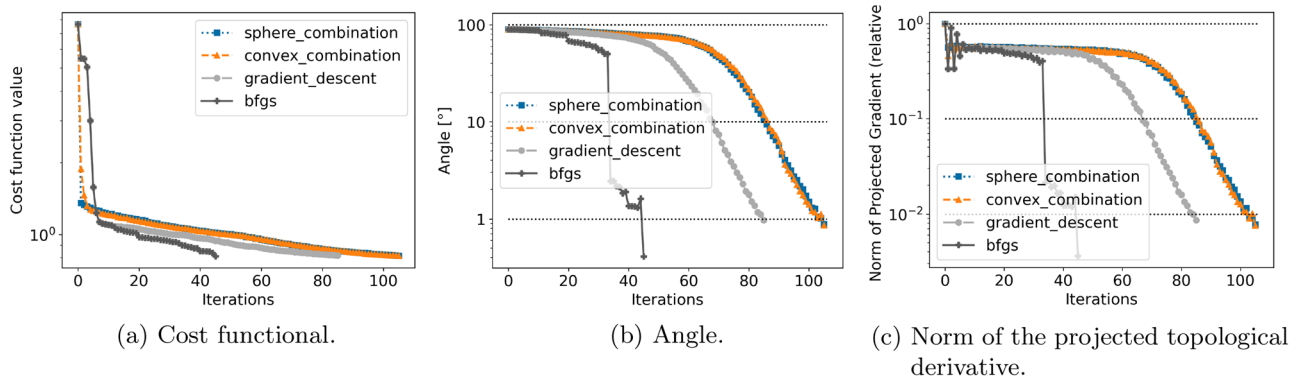
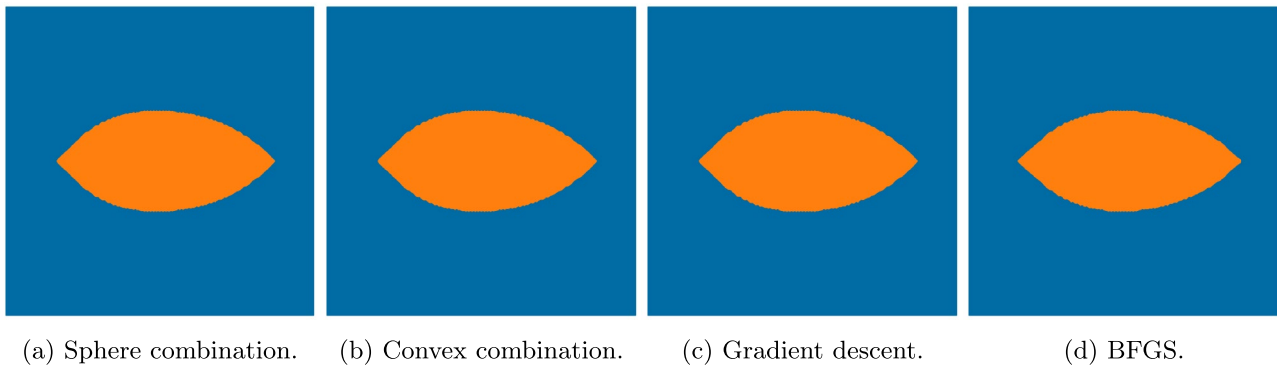


Fig. 17 Evolution of the optimization for the rugby ball



**Fig. 18** Optimized geometries for the rugby ball problem

which is reminiscent of a rugby ball, so that all of them converge to the same local minimizer of the problem.

Altogether, in the context of fluid design optimization, we observe that the BFGS method performs substantially better than the other methods considered in this paper. It reaches the desired stopping criteria with significantly less iterations than the remaining methods. Our findings highlight the efficiency and potential of the proposed BFGS method for solving topology optimization problems.

## 5 Conclusion and outlook

In this paper, we have presented novel quasi-Newton methods for topology optimization using a level-set method. We recalled the topological derivative, the level-set method for topology optimization, and the widely used optimization algorithm proposed in Amstutz and Andr  (2006). Then, we presented a new perspective on the evolution equation for evolving the level-set function according to the topological derivative, which enables an interpretation as a classical gradient descent method. This method is the basis for our derivation of quasi-Newton methods for topology optimization and we present a limited-memory BFGS method in this paper. The derivation of the BFGS methods is analogous to the finite-dimensional case and is possible due to the change in perspective described above. We investigated the performance of the proposed gradient descent and BFGS methods on four problem classes: Inverse topology optimization problems constrained by linear and semilinear Poisson problems, compliance minimization in linear elasticity, and the optimization of fluids in Navier–Stokes flow. We compared the results to current state-of-the-art solution algorithms for topology optimization with level-set methods. Our results show that

the novel BFGS methods often significantly outperform the other considered methods, requiring substantially less iterations to compute a (local) minimizer. The only exception was the problem of compliance minimization in linear elasticity, where all considered methods performed very similarly, so that the BFGS method performed at least as good as the other, already established methods. All in all, the proposed BFGS methods are efficient and attractive solution algorithms for topology optimization and show great potential for solving such problems.

For future research, there are several interesting directions. One could consider nonlinear conjugate gradient (NCG) methods for topology optimization in analogy to Blauth (2021b), whose derivation is straightforward with the new perspective on the level-set evolution equation presented in this chapter. In fact, these methods are already implemented and available in our open-source software cashocs (Blauth (2021a, 2023)). However, a thorough numerical analysis of such NCG methods is still required for understanding their performance and behavior. Moreover, a theoretical analysis of the proposed BFGS methods is of great interest. For this, it would be useful to study the properties of the approximate Hessian operator, which the BFGS methods make use of, and its relation to higher order topological derivatives (see, e.g., Baumann and Sturm (2022) and the references therein). Further, it remains an open question why the BFGS methods do not perform as well in the context of compliance minimization in linear elasticity as they do for the other problem classes considered in this paper. Finally, it is, of course, of particular interest to employ our proposed methods for solving practically relevant topology optimization problems, e.g., in the fields of fluid-dynamical optimization or the simulation of fracture evolution.



**Funding** Open Access funding enabled and organized by Projekt DEAL. No funding was received.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Replication of results** The source code for our numerical examples is available publicly on GitHub (Blauth and Sturm (2023)) and the methods presented in this paper are implemented in our open-source software cashocs (Blauth (2021a, 2023)).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Allaire G, de Gournay F, Jouve F, Toader A-M (2005) Structural optimization using topological and shape sensitivity via a level set method. *Control Cybern* 34:59–80
- Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys* 194:363–393. <https://doi.org/10.1016/j.jcp.2003.09.032>
- Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN (2015) The FEniCS project version 1.5. *Arch Numerical Softw*. <https://doi.org/10.11588/ans.2015.100.20553>
- Amstutz S (2011) Analysis of a level set method for topology optimization. *Optim Methods Softw* 26:555–573. <https://doi.org/10.1080/10556788.2010.521557>
- Amstutz S (2022) An introduction to the topological derivative. *Eng Comput* 39:3–33. <https://doi.org/10.1108/EC-07-2021-0433>
- Amstutz S, André H (2006) A new algorithm for topology optimization using a level-set method. *J Comput Phys* 216:573–588. <https://doi.org/10.1016/j.jcp.2005.12.015>
- Baumann P, Sturm K (2022) Adjoint-based methods to compute higher-order topological derivatives with an application to elasticity. *Eng Comput* 39:60–114. <https://doi.org/10.1108/EC-07-2021-0407>
- Beretta E, Manzoni A, Ratti L (2017) A reconstruction algorithm based on topological gradient for an inverse problem related to a semilinear elliptic boundary value problem. *Inverse Probl* 33:27. <https://doi.org/10.1088/1361-6420/aa5c0a>
- Blauth S (2021) cashocs: A Computational Adjoint-Based Shape Optimization and Optimal Control Software. *SoftwareX* 13:100646. <https://doi.org/10.1016/j.softx.2020.100646>
- Blauth S (2021) Nonlinear Conjugate Gradient Methods for PDE Constrained Shape Optimization Based on Steklov-Poincaré-Type Metrics. *SIAM J Optim* 31:1658–1689. <https://doi.org/10.1137/20M1367738>
- Blauth S (2022) Shape Optimization with Nonlinear Conjugate Gradient Methods. In: Melenk, J.M., Perugia, I., Schöberl, J., Schwab, C. (eds) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2020+1*. Lecture Notes in Computational Science and Engineering, vol 137. Springer, Cham. [https://doi.org/10.1007/978-3-031-20432-6\\_9](https://doi.org/10.1007/978-3-031-20432-6_9)
- Blauth S (2022) Space Mapping for PDE Constrained Shape Optimization. *SIAM J Optim* 33:1707–1733. <https://doi.org/10.1137/22M1515665>
- Blauth S (2023) *Version 2.0 – cashocs: A Computational, Adjoint-Based Shape Optimization and Optimal Control Software*. <https://arxiv.org/abs/2306.09828>
- Blauth S, Sturm K (2023) Software used in Quasi-Newton methods for topology optimization using a level-set method. GitHub. <https://doi.org/10.5281/zenodo.7773404>
- Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. *Int J Numerical Methods Fluids* 41:77–107. <https://doi.org/10.1002/flid.426>
- Canelas A, Laurain A, Novotny AA (2015) A new reconstruction method for the inverse source problem from partial boundary measurements. *Inverse Probl* 31:075009. <https://doi.org/10.1088/0266-5611/31/7/075009>
- Deckelnick K, Herbert PJ, Hinze M (2022) A novel  $W_1, \infty$  approach to shape optimisation with Lipschitz domains. *European Series in Applied and Industrial Mathematics (ESAIM): Control. Optim Cal Var* 28:2. <https://doi.org/10.1051/cocv/2021108>
- Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. *Struct Optim* 8:42–51. <https://doi.org/10.1007/bf01742933>
- Gangl P, Langer U (2012) Topology optimization of electric machines based on topological sensitivity analysis. *Comput Vis Sci* 15:345–354. <https://doi.org/10.1007/s00791-014-0219-6>
- Gangl P, Sturm K (2020) A simplified derivation technique of topological derivatives for quasi-linear transmission problems, ESAIM. *Control Optim Cal Var* 26:106. <https://doi.org/10.1051/cocv/2020035>
- Gangl P, Sturm K (2021) Asymptotic analysis and topological derivative for 3d quasi-linear magnetostatics, European series in applied and industrial mathematics (ESAIM). *Math Model Numerical Anal* 55:853–875. <https://doi.org/10.1051/m2an/2020060>
- Garreau S, Guillaume P, Masmoudi M (2001) The topological asymptotic for PDE systems: the elasticity case. *SIAM J Control Optim* 39:1756–1778. <https://doi.org/10.1137/s0363012900369538>
- Hintermüller M, Laurain A (2008) Electrical impedance tomography: from topology to shape. *Control Cybern* 37:913–933
- Kelley CT (1999) *Iterative methods for optimization frontiers in applied mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia. <https://doi.org/10.1137/1.9781611970920>
- Laurain A, Hintermüller M, Freiberger M, Scharfetter H (2013) Topological sensitivity analysis in fluorescence optical tomography. *Inverse Probl* 29:025003. <https://doi.org/10.1088/0266-5611/29/2/025003>
- Logg A, Mardal K-A, Wells GN (2012) *Automated solution of differential equations by the finite element method*. Springer. <https://doi.org/10.1007/978-3-642-23099-8>
- Sá LFN, Amigo RCR, Novotny AA, Silva ECN (2016) Topological derivatives applied to fluid flow channel design optimization problems. *Struct Multidisc Optim* 54:249–264. <https://doi.org/10.1007/s00158-016-1399-0>
- Nocedal J, Wright SJ (2006) *Springer series in operations research and financial engineering numerical optimization*, 2nd edn. Springer, New York
- Novotny AA, Sokołowski J (2020) An introduction to the topological derivative method. *Springerbriefs in mathematics*. Springer, Cham. <https://doi.org/10.1007/978-3-030-36915-6>
- Novotny AA, Sokołowski J, Zochowski A (2019) Applications of the topological derivative method, *Studies in systems*.



- decision and control. Springer, Cham. <https://doi.org/10.1007/978-3-030-05432-8>
- Novotny AA, Sokolowski J (2013) Topological derivatives in shape optimization interaction of mechanics and mathematics. Springer, Heidelberg. <https://doi.org/10.1007/978-3-642-35245-4>
- Schulz VH, Siebenborn M, Welker K (2016) Efficient PDE constrained shape optimization based on Steklov-Poincaré-type metrics. *SIAM J Optim* 26:2800–2819. <https://doi.org/10.1137/15M1029369>
- Sokolowski J, Zochowski A (1999) On the topological derivative in shape optimization. *SIAM J Control Optim* 37:1251–1272. <https://doi.org/10.1137/s0363012997323230>
- Sturm K (2020) Topological sensitivities via a lagrangian approach for semilinear problems. *Nonlinearity* 33:4310–4337. <https://doi.org/10.1088/1361-6544/ab86cb>
- Xavier M, Fancello E, Farias J, Van Goethem N, Novotny A (2017) Topological derivative-based fracture modelling in brittle materials: a phenomenological approach. *Eng Fract Mech* 179:13–27. <https://doi.org/10.1016/j.engfracmech.2017.04.005>
- Xavier M, Van Goethem N, Novotny A (2018) A simplified model of fracking based on the topological derivative concept. *Int J Solids Struct* 139–140:211–223. <https://doi.org/10.1016/j.ijsolstr.2018.02.002>
- Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.