ÖAW AUSTRIAN ACADEMY OF SCIENCES

RICAM – JOHANN RADON INSTITUTE FOR COMPUTATIONAL AND APPLIED MATHEMATICS

# Homotopy methods for higher order shape optimization: A globalized shape-Newton method and Pareto-front tracing

A. Cesarano, B. Endtmayer, P. Gangl

RICAM-Report 2024-08

# Homotopy methods for higher order shape optimization: A globalized shape-Newton method and Pareto-front tracing

A. Cesarano[1], B. Endtmayer[2,3], and P. Gangl[1]

[1]Johann Radon Institute for Computational and Applied Mathematics, Altenbergerstr. 69, A-4040 Linz, Austria

[2]Leibniz Universität Hannover, Institut für Angewandte Mathematik, AG Wissenschaftliches Rechnen, Welfengarten 1, 30167 Hannover, Germany

[3]Cluster of Excellence PhoenixD, (Photonics, Optics, and Engineering - Innovation Across Disciplines), Leibniz Universität Hannover, Germany

**Abstract**

First order shape optimization methods, in general, require a large number of iterations until they reach a locally optimal design. While higher order methods can significantly reduce the number of iterations, they exhibit only local convergence properties, necessitating a sufficiently close initial guess. In this work, we present an unregularized shape-Newton method and combine shape optimization with homotopy (or continuation) methods in order to allow for the use of higher order methods even if the initial design is far from a solution. The idea of homotopy methods is to continuously connect the problem of interest with a simpler problem and to follow the corresponding solution path by a predictor-corrector scheme. We use a shape-Newton method as a corrector and arbitrary order shape derivatives for the predictor. Moreover, we apply homotopy methods also to the case of multi-objective shape optimization to efficiently obtain well-distributed points on a Pareto front. Finally, our results are substantiated with a set of numerical experiments.

***Keywords:*** continuation, homotopy, path following, nonlinear problem, shape optimization, shape Newton, Pareto tracing, multi-objective shape optimization

***2020 MSC:*** 49Q10, 49M15, 49M41, 58E17

## 1 Introduction

Over the past decades, mathematical and numerical methods for shape optimization have been thoroughly investigated in both the mathematical and the engineering community and have been applied to a wide range of real world problems. We refer to the monographs [8, 38] for detailed introductions to the main concepts of shape optimization and, exemplarily, mention concrete applications from solid [12] or fluid mechanics [30], multi-physics problems [15, 16], electrical engineering [17], acoustics [34] or inverse problems [23].

While most of these works are based on first order shape derivatives, also second order shape derivatives have been investigated from both an analytical and numerical viewpoint [9, 31, 33, 13, 40, 18]. A key difficulty when employing Newton methods in shape optimization is the kernel of the shape Hessian operator which includes all interior and tangential shape perturbations. In order to approximately solve the shape Newton system, often regularization techniques are employed which, however, may prevent the desired superlinear or even quadratic convergence unless devised carefully [33]. As an alternative, the shape Newton equation can be solved for normal perturbations only as proposed in [13]. In [40], the author proves superlinear convergence of a shape Newton method by employing approximately normal functions.

Another well-known difficulty in the application of Newton's method for (typically non-convex) shape optimization problems is the method's local convergence properties, meaning that Newton's method is only applicable if the initial design is sufficiently close to a solution of the optimization problem. As a simple remedy, one could couple first and second order methods by first running a gradient descent method until a certain residual is reached before switching to a second order method yielding fast convergence. In this approach, however, the threshold that needs to be reached by the first order method may vary from problem to problem, making the method unrobust.

In this paper, we propose to couple shape optimization with homotopy (or continuation) methods [2, 10] in order to improve convergence properties of shape-Newton methods even if the initial design is far from the sought solution. Homotopy methods have been introduced as a method for solving difficult systems of equations, see e.g. [24, 43] or [11, 42] for the context of optimization problems. These methods rely on the idea of smoothly connecting a problem of interest whose solution is sought with a much simpler problem whose solution is known or can easily be computed. Given such a connection, called the homotopy map, one can follow the path of solutions to intermediate problems to reach the problem of interest by means of, e.g., a predictor-corrector method. As shown in [10, 41], Newton's method just ensures quadratic convergence if the initial guess is in a neighborhood of the solution. The basic idea of the predictor corrector method is to stay in this neighbourhood when following the path. We will apply this method to the problem of finding stationary shapes with vanishing shape derivative. The employed corrector method will be a shape Newton method and we will use polynomial predictors of first and higher order.

Beside their globalizing effect, there are situations where continuation methods are useful for their own sake as the choice of the homotopy allows to implicitly control the path that is taken during the optimization. Often, optimization problems involve parameters coming from the mathematical model or the algorithm to be used, for which a gradual variation can be beneficial in terms of the obtained solution. As an example we mention the penalization parameter in density-based topology optimization [21]. Moreover, as we will elaborate in Section 5, homotopy methods can be used for finding points on a Pareto front in the case of multi-objective optimization problems [19, 20], and are also useful for systematically exploring the design space by, e.g., deflation methods [29]. Preliminary numerical results related to Section 5 have been submitted to a conference proceeding, see [6].

The novelty of this paper is threefold: On the one hand, we present a new way of solving the unregularized Newton system which, to the best of the authors' knowledge, has not been reported in the literature. The main contribution of this work is the coupling of shape optimization with homotopy methods including higher order predictors. Finally, as a by-product, we also obtain an efficient way for tracing Pareto-optimal shapes of multi-objective shape optimization problems.

The remainder of this work is organized as follows: In Section 2, we present the concept of homotopy methods along with different predictor choices and step size strategies. We briefly discuss concepts of shape optimization in Section 3 before presenting the connection of these two concepts in Section 4.1. We comment on its use in the context of multi-objective optimization in 5 before presenting a set of numerical experiments in Section 6.

## 2 Homotopy methods

The idea of homotopy methods for solving a problem of interest is to establish a smooth connection with a simple problem whose solution is known or can easily be computed, and to follow the path of solutions to the arising intermediate problems until the original problem is reached. For better illustration, consider a function $F : \mathbf{R}^n \to \mathbf{R}^n$ and let $x^{(0)} \in \mathbf{R}^n$ be given. We look for a root satisfying

$$F(x) = 0. \tag{1}$$

Often, a solution to problem (1) cannot be found from the given initial guess $x^{(0)}$. A homotopy map $H : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}^n$ defines a family of equations

$$H(x, t) = 0 \tag{2}$$

parametrized by the homotopy parameter $t \in [0, 1]$. The homotopy map should be chosen in such a way that solution to the initial problem $H(x, 0) = 0$ can be easily obtained from the initial guess $x^{(0)}$, and that the equation $H(x, 1) = 0$ is equivalent to the original equation (1). We mention one widely used class of homotopy maps which is also considered throughout this article.

**Example 2.1** (convex homotopy). *Let $G : \mathbf{R}^n \to \mathbf{R}^n$ and $x^{(0)}$ given with $G(x^{(0)}) = 0$. Then the convex homotopy of $F$ and $G$ can be defined as*

$$H(x, t) = tF(x) + (1 - t)G(x). \tag{3}$$

*Examples for the choice of $G$ include the Keller model $G(x) := F(x) - F(x^{(0)})$ or the choice $G(x) := x - x^{(0)}$.*

Given a homotopy map $H$ for the problem of interest $F$ and a suitable initial guess $x^{(0)}$, the solution path can be followed by a predictor-corrector method. The main ingredients are summarized in Algorithm 1. Here, one iteratively chooses a step size by which the current homotopy value is increased, defines an improved starting guess for the subsequent homotopy value (predictor step) and tries to solve the problem for the updated homotopy value (corrector step). The latter step is often realized by Newton's method, which converges only if the starting guess is sufficiently close to the solution. Thus, the choice of the step size has a large impact on the efficiency of the overall method. If the step $\Delta t^{(k)}$ at iteration $k$ is chosen too large, the corrector step may fail and one may have to repeat the steps for the current homotopy value $t^{(k)}$ with a decreased step size. We comment on the important issue of step size choice in Section 2.2. Similarly, a good predictor can yield better starting guesses for the corrector step and thus allow for larger step sizes and shorter total computation times. We comment on this topic in Section 2.1.

**Algorithm 1** Predictor-corrector algorithm

```
Input: $x^{(0)} \in \mathbf{R}^n$, $t^{(0)} = 0$, $k = 0$.
while $t^{(k)} < 1$ do
    Choose step size $\Delta t^{(k)}$ and set $t^{(k+1)} = \min(t^{(k)} + \Delta t^{(k)}, 1)$.
    Predictor: $\tilde{x}^{(k+1)} = \mathrm{Pred}(x^{(k)}, t^{(k)}, t^{(k+1)} - t^{(k)})$
    Corrector: Obtain $x^{(k+1)}$ as solution to $H(x, t^{(k+1)}) = 0$ by iterative method with starting guess $\tilde{x}^{(k+1)}$.
    $k \leftarrow k + 1$
end while
```

**Remark 2.2** (Homotopy methods for optimization)**.** *The same principles can also be adapted to the solution of optimization problems rather than nonlinear equations. In that setting, $H(\cdot, t)$ would represent a family of optimization problems and the corrector step of Algorithm 1 would be replaced by an iterative optimization algorithm, see also [11].*

**Remark 2.3.** *Throughout this paper, we assume that the solution path can be parametrized by the homotopy parameter $t$ and that no turning points exist. Moreover, we assume the absence of bifurcations. Then, for a given $t \in [0, 1]$, there exists a unique solution to the equation $H(\cdot, t) = 0$ on the path of interest, which we denote by $x(t)$.*

## 2.1 On the choice of predictors

Recall that, for $t \in [0, 1]$ fixed, $x(t)$ denotes a solution to the homotopy equation for $t$, i.e., it holds

$$H(x(t), t) = 0. \tag{4}$$

In this section, we assume $x(t)$ to be sufficiently smooth such that all arising derivatives exist. This can be inferred from a sufficient smoothness of $H$ by the implicit function theorem. For more information on predictor methods for homotopy, we refer to [2]. In what follows, we discuss different predictor choices which are also illustrated in Figure 1.

### 2.1.1 Zero order predictors

The most simple predictor is the identity, i.e

$$\mathrm{Pred}^{(0)}(x^{(k)}, t^{(k)}, \Delta t^{(k)}) = x^{(k)}. \tag{5}$$

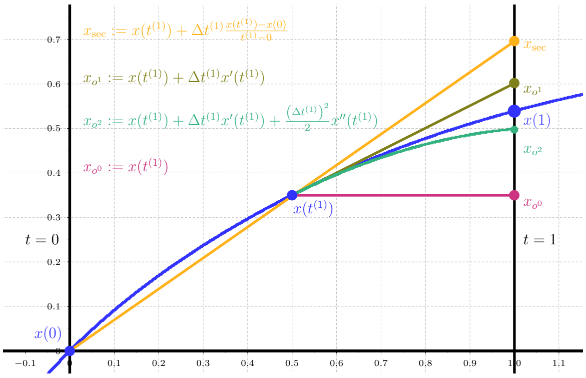


Figure 1: Visualization of a solution curve $(x(t), t)$ (blue) along with different ways of predicting an initial guess starting from $t^{(1)} = 0.5$ with $\Delta t^{(1)} = 0.5$: Step (magenta), secant (orange), first order (olive), second order (green). Illustration is for problem (3) with $F(x) := x^5 + x - e^{-x}$ and $G(x) := x$.

This is an zero order predictor. In some literature, this is also called classical continuation; see [10]. Another zero order predictor is the so called secant predictor. In contrast to (5) this predictor also depends on the solution of the previous step, i.e.

$$\text{Pred}^{\text{sec}}(x^{(k)}, t^{(k)}, x^{(k-1)}, t^{(k-1)}, \Delta t^{(k)}) = x^{(k)} + \Delta t^{(k)} \frac{x^{(k)} - x^{(k-1)}}{t^{(k)} - t^{(k-1)}}. \tag{6}$$

For the first step any other predictor can be used. We mention that this predictor does not fit into the class of predictors which are used in Algorithm 1.

### 2.1.2 First order predictors

Differentiating (4) with respect to $t$ yields

$$H_x(x(t), t)x'(t) + H_t(x(t), t) = 0, \tag{7}$$

where $H_x : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}^{n \times n}$ denotes the Jacobian of $H(\cdot, t)$ and $H_t : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}^n$ the partial derivative of the homotopy map with respect to the homotopy parameter $t$. Assuming invertibility of $H_x(x(t), t)$, this defines the first order derivative

$$x'(t) = -\left(H_x(x(t), t)\right)^{-1} H_t(x(t), t) =: g_{\text{Dav}}(x(t), t) \tag{8}$$

which gives rise to a first order predictor

$$\text{Pred}^{(1)}(x^{(k)}, t^{(k)}, \Delta t^{(k)}) = x^{(k)} + \Delta t^{(k)} g_{\text{Dav}}(x^{(k)}, t^{(k)}). \tag{9}$$

This is also known as tangent predictor our tangent continuation as explained in [10]

**Remark 2.4.** *Instead of the predictor-corrector scheme described above, one could also consider the related initial value problem given by* (8) *together with the initial condition* $x(0) = x^{(0)}$ *and apply numerical techniques for ordinary differential equations, see [4]. Equation* (8) *is called the Davidenko differential equation.*

### 2.1.3 Higher order predictors

As it was mentioned above, a better predictor can speed up the overall computation as it may allow for larger step sizes and reduce the number of iterations needed in the corrector step. For smooth enough problems one may thus approximate the unknown solution $x(t + \Delta t)$ by higher order Taylor polynomials involving higher order derivatives of the mapping $t \mapsto x(t)$, see Fig. 1 for an illustration. These higher order derivatives $x^{[n]}(t)$, $n \geq 2$ can be obtained by further differentiating (7). Note the difference between the employed notation $x^{(k)} = x(t^{(k)})$ and the $n$-th derivative $x^{[n]}(t)$ of the mapping $t \mapsto x(t)$.

For that purpose, let $X^{(k)}(t) := \left\{x^{[k]}(t), x^{[k-1]}(t), \ldots, x^{[2]}(t), x'(t)\right\}$ denote the set of all derivatives of $t \mapsto x(t)$ up to order $k$. Moreover, let $\alpha_k := \{(l, m) \in (\mathbb{N} \cup \{0\})^2, l + m \leq k\}$ denote all multi-indices of order $k$ or less and $H_{\alpha_k} := \left\{\frac{\partial^l}{\partial x^l}\frac{\partial^m}{\partial t^m}H : (l, m) \in \alpha_k\right\}$ the set of all derivatives of $H$ corresponding to $\alpha_k$.

**Theorem 2.5.** *Let $H(x(t), t)$ be sufficiently smooth and assume that $H_x(x(t), t)$ is invertible for all $t \in [0, 1]$. Then the $n$-th derivative of $x(t)$, $x^{[n]}(t)$, is given as the solution to the system*

$$H_x(x(t), t)x^{[n]}(t) = b^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n}(x(t), t)\right), \tag{10}$$

*where $b^{(n)}(\cdot, \cdot)$ is a semilinear form which satisfies the recurrence relation*

$$\begin{aligned} b^{(n+1)}\left(X^{(n)}(t), H_{\alpha_{n+1}}(x(t), t)\right) :=& b_X^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n}(x(t), t)\right) X_t^{(n-1)}(t) \\ &+ b^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n x}(x(t), t)x'(t) + H_{\alpha_n t}(x(t), t)\right) \\ &- H_{xx}(x(t), t)\left[x^{[n]}(t), x'(t)\right] - H_{xt}(x(t), t)x^{[n]}(t). \end{aligned}$$

*with $b^{(1)}(X^{(0)}(t), H_{\alpha_1}(x(t), t)) = -H_t(x(t), t)$.*

*Proof.* We prove this statement by induction. The base case $n = 1$ was already derived in (8). Now let us assume the statement is already valid for $n$, i.e., we assume

$$H_x(x(t), t)x^{[n]}(t) = b^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n}(x(t), t)\right).$$

Again differentiating with respect to $t$ yields

$$\begin{aligned} H_x(x(t), t)x^{[n+1]}(t) + H_{xx}(x(t), t)\left[x^{[n]}(t), x'(t)\right] + H_{xt}(x(t), t)x^{[n]}(t) \\ = b_X^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n}(x(t), t)\right) X_t^{(n-1)}(t) \\ + b^{(n)}\left(X^{(n-1)}(t), H_{\alpha_n x}(x(t), t)x'(t) + H_{\alpha_n t}(x(t), t)\right), \end{aligned}$$

which, after rearranging, yields

$$H_x(x(t), t)x^{[n+1]}(t) = b^{(n+1)}\left(X^{(n)}(t), H_{\alpha_n}(x(t), t)\right).$$

□

**Remark 2.6.** *Of course there is a trade-off between the reduction of computational costs by the use of additional predictor terms and the cost of computing these terms. We note, however, that the system matrix for computing the $n$-th derivative of $t \mapsto x(t)$ in* (10) *is the same for all $n$. Thus, if the system matrix can be factorized, the factorization has to be done only once and additional derivatives can be computed at the cost of assembling the right hand sides of* (10)*.*

Given the first $q$ derivatives of $t \mapsto x(t)$, an $q$-th order predictor at $x^{(k)} = x(t^{(k)})$ can be defined as

$$\mathrm{Pred}^{(q)}(x^{(k)}, t^{(k)}, \Delta t^{(k)}) := x^{(k)} + \sum_{i=1}^{q} \frac{1}{i!} \left(\Delta t^{(k)}\right)^{i} x^{[i]} \left(t^{(k)}\right). \tag{11}$$

**Corollary 2.7.** *In addition to the system defining the first order derivative of $t \mapsto x(t)$ stated in* (7) *which corresponds to* (10) *for $n = 1$, we also give the problems defining the second and third order derivatives:*

$$H_x[x''(t)] = -\, H_{xx}[x'(t), x'(t)] - 2H_{xt}[x'(t)] - H_{tt}, \tag{12}$$

$$\begin{aligned} H_x[x'''(t)] = -\, \Bigg[ & H_{xxx}[x'(t), x'(t), x'(t)] + 3H_{xxt}[x'(t), x'(t)] \\ & + 3H_{xx}[x''(t), x'(t)] + 3H_{xt}[x''(t)] + 3H_{xtt}[x'(t)] + H_{ttt} \Bigg]. \end{aligned} \tag{13}$$

*Note that the arguments $(x(t), t)$ were omitted here for brevity.*

## 2.2 Step size choice

The choice of the step sizes $\Delta t^{(k)}$ in Algorithm 1 has a big impact on the overall efficiency of the method. While larger step sizes reduce the number of necessary homotopy steps to reach the solution of the original problem at $t = 1$, the quality of the starting guesses for the corrector phase is decreased and the corrector step may be unsuccessful.

A straightforward way to define step sizes would be to equidistant steps, i.e., to choose $\Delta t^{(k)} = 1/m$ for a fixed number $m$ of steps. This choice, however, can be inefficient when small steps are necessary for a certain part of the homotopy path, but larger steps could be made in other parts. Thus, we propose three different ways of adapting the step sizes along the way:

### 2.2.1 Fixed adaptation of step sizes

We start with $t^{(0)} = 0$ and an initial step size $\Delta t^{(0)} > 0$. If, for $k \geq 1$, the combination of predictor and corrector step to get from homotopy value $t^{(k-1)}$ to $t^{(k)}$ by the increment $\Delta t^{(k-1)}$ was successful, we increase the step size for the next step by a constant factor $\overline{\gamma} > 1$, i.e., $\Delta t^{(k)} = \overline{\gamma} \Delta t^{(k-1)}$. If, on the other hand, the corrector step is not successful, we reduce the step size by a constant factor $\underline{\gamma} < 1$ and repeat the corrector step starting from the adapted predicted value.

### 2.2.2 Agile adaptation of step sizes based on predictors

While the strategy of Section 2.2.1 is capable of adapting the step size to the problem, this adaptation happens only gradually. In particular, sudden changes in the desirable step size or the detection of a suitable initial step size $\Delta t^{(0)}$ are not possible in this way. For that reason, we propose to choose the step size depending on the estimated distance between the solution at $t = t^{(k+1)}$ and its prediction of a given order. More precisely, given a predictor of order $q$ via (11), by Taylor expansion it holds

$$\left| x\left(t^{(k+1)}\right) - \mathrm{Pred}^{(q)}\left(x^{(k)}, t^{(k)}, \Delta t^{(k)}\right) \right| = \frac{(\Delta t^{(k)})^{q+1}}{(q+1)!} \left| x^{[q+1]}\left(t^{(k)}\right) \right| + \mathcal{O}((\Delta t^{(k)})^{q+2})$$

as $\Delta t^{(k)} \to 0$. The idea now is, at the cost of computing one additional derivative, to choose $\Delta t^{(k)}$ such that the leading term on the right hand side becomes a constant $\alpha > 0$ (which is in particular independent of the curve $t \mapsto x(t)$), i.e., we choose

$$\Delta t^{(k)} = ((q+1)!\,\alpha)^{1/(q+1)} \left| x^{[q+1]}\left(t^{(k)}\right) \right|^{-1/(q+1)}. \tag{14}$$

By the constant $\alpha$, the user can determine the level of conservativeness. Smaller values of $\alpha$ will increase the success probability of the corrector step, but will increase the total number of homotopy steps.

### 2.2.3 Agile adaptation with adaptive parameter choice

While the strategy of Section 2.2.2 allows for an automatic detection of the order of magnitude of a step size, the best possible choice of the parameter $\alpha$ is again problem-dependent. For that reason, we suggest to combine the strategy of Section 2.2.2 with the parameter adaptation described in Section 2.2.1, i.e. to multiply $\alpha$ by constants $\underline{\alpha} < 1$ or $\overline{\alpha} > 1$ in case of failure or success of the corrector step, respectively.

## 3 Shape optimization

We briefly review the main concepts needed for first and second order shape optimization algorithms. For a more thorough introduction, we refer the reader to the monographs [8, 38] or the review papers [1, 18].

Numerical shape optimization algorithms are based on the concept of shape derivatives, i.e., sensitivities of a shape-dependent cost function with respect to a variation of the domain represented by the action of deformation vector fields. To be more precise, let $\mathsf{D} \subset \mathbf{R}^d$ denote a hold-all domain, let $\mathcal{A} \subset \mathcal{P}(\mathsf{D})$ be a set of admissible shapes and consider a shape function $\mathcal{J} : \mathcal{A} \to \mathbf{R}$. For a shape $\Omega \in \mathcal{A}$, a smooth vector field $\mathbf{V} : \mathbf{R}^d \to \mathbf{R}^d$ and a parameter $s > 0$, let $\Omega_s := (\mathrm{id} + s\mathbf{V})(\Omega)$ denote the corresponding perturbed domain. The first order shape derivative of the shape function $\mathcal{J}$ is defined as

$$d\mathcal{J}(\Omega)(\mathbf{V}) := \left( \frac{d}{ds} \mathcal{J}(\Omega_s) \right)\bigg|_{s=0} = \lim_{s \to 0} \frac{\mathcal{J}(\Omega_s) - \mathcal{J}(\Omega)}{s} \tag{15}$$

if this limit exists and the mapping $\mathbf{V} \mapsto d\mathcal{J}(\Omega)(\mathbf{V})$ is linear. Similarly, we define the symmetric second order shape derivative by considering smooth vector fiels $\mathbf{V}, \mathbf{W} : \mathbf{R}^d \to \mathbf{R}^d$, two parameters $s, r > 0$ and the corresponding perturbed domain $\Omega_{s,r} := (\mathrm{id} + s\mathbf{V} + r\mathbf{W})(\Omega)$. The symmetric second order shape derivative is defined as

$$d^2\mathcal{J}(\Omega)[\mathbf{V},\mathbf{W}] := \left(\frac{d^2}{dsdr}\mathcal{J}(\Omega_{s,r})\right)\Big|_{s=0,r=0}. \tag{16}$$

if it exists and the mapping $(\mathbf{V},\mathbf{W}) \mapsto d^2\mathcal{J}(\Omega)[\mathbf{V},\mathbf{W}]$ is bilinear. We mention that there also exists a non-symmetric second order shape derivative which is obtained by repeated shape differentiation, i.e.,

$$D^2\mathcal{J}(\Omega)[\mathbf{V}][\mathbf{W}] := \lim_{r\to 0}\frac{1}{r}\Big(d\mathcal{J}((\mathrm{id}+r\mathbf{W})(\Omega))(\mathbf{V}) - d\mathcal{J}(\Omega)(\mathbf{V})\Big) \tag{17}$$

with the relation [8, 40, 33, 37]

$$D^2\mathcal{J}(\Omega)[\mathbf{V}][\mathbf{W}] = d^2\mathcal{J}(\Omega)[\mathbf{V},\mathbf{W}] + d\mathcal{J}(\Omega)(\partial\mathbf{V}\mathbf{W}).$$

### 3.1 First order methods

The main idea of most first order shape optimization algorithms consists in repeatedly finding a descent direction $\mathbf{V}$ such that $d\mathcal{J}(\Omega)(\mathbf{V}) < 0$ and updating the current domain by the action of this vector field to get $\tilde{\Omega} = (\mathrm{id}+s\mathbf{V})(\Omega)$ as the new iterate. Here, the parameter $s > 0$ is chosen sufficiently small in order to guarantee a descent, $\mathcal{J}(\tilde{\Omega}) < \mathcal{J}(\Omega)$. A descent direction can be obtained by solving an auxiliary boundary value problem to find $\mathbf{V} \in H$ satisfying

$$b(\mathbf{V},\mathbf{W}) = -d\mathcal{J}(\Omega)(\mathbf{W}) \tag{18}$$

for all $\mathbf{W} \in H$ for a given vector-valued Hilbert space $H$ and a positive definite bilinear form $b(\cdot,\cdot) : H\times H \to \mathbf{R}$. Possible choices for $b(\cdot,\cdot)$ include a vector Laplacian or elasticity problem on (a subdomain of) $H^1(\mathsf{D})^d$, the latter being advantageous in terms of mesh quality [36]. Other means to ensure mesh quality of deformed finite element meshes include the use of nearly conformal mappings [22] or solving nonlinear versions of (18) [28, 26] or even employing $W^{1,\infty}$ descent directions [7].

### 3.2 Second order methods

Replacing the bilinear form $b(\cdot,\cdot)$ in (18) by the (symmetric) second order shape derivative defined in (16) for the problem at hand and the current domain $\Omega$ yields the shape-Newton system to find $\mathbf{V} \in H$ satisfying

$$d^2\mathcal{J}(\Omega)(\mathbf{V},\mathbf{W}) = -d\mathcal{J}(\Omega)(\mathbf{W}) \tag{19}$$

for all $\mathbf{W} \in H$ for a suitable Hilbert space $H$. The operator on the left hand side of (19), however, is not invertible. In fact, its kernel contains all vector fields whose normal component on the boundary $\partial\Omega$ vanishes [33, 27, 40]. This includes vector fields corresponding to deformations which only move interior points as well as tangential movements. In order to obtain an approximate shape-Newton descent direction, problem (19) is often modified either by regularizing the operator on the full domain [31] or only on the boundary [18].

### 3.3 Numerical shape-Newton method

In this work, we propose to solve the finite-element discretized version of (19) without regularization, by filtering out the full kernel of the shape Hessian. To that end, consider a given initial domain $\Omega \subset \mathbf{R}^d$ which is discretized by a conforming simplicial finite element mesh $\mathcal{T}_h$. We introduce the space of piecewise linear and globally continuous functions on $\mathcal{T}_h$,

$$V_h := \{v \in H^1(\Omega) \cap C(\bar{\Omega}) : v|_T \in P^1(T) \text{ for all } T \in \mathcal{T}_h\} \tag{20}$$

where $P^1(T)$ denotes the space of linear functions on a simplex $T$ (i.e. a triangle if $d = 2$ or a tetrahedron if $d = 3$). We denote the number of vertices of $\mathcal{T}_h$ by $N$ and the number of vertices that lie on the boundary of $\Omega$ by $N_{\text{bdy}}$. The linear hat basis function of $V_h$ corresponding to vertex $x_k$ is denoted by $\varphi_k$, $k = 1, \dots, N$. Without loss of generality, we assume the basis functions to be numbered in such a way that the first $N_{\text{bdy}}$ basis functions correspond to the vertices on the boundary.

Our goal is now to find a solution $\mathbf{V}_h \in V_h^d$ to the discretized version of (19). We deal with the kernel of this problem as follows:

- Interior deformations are removed by choosing the Hilbert space $H$ in (19) to be defined solely on the surface, $H := (V_h)^d \cap H^1(\partial\Omega)^d$ with $\dim(H) = dN_{\text{bdy}}$ and basis functions (for $k = 1, \dots, N_{\text{bdy}}$): $\Phi_{2k-1} = (\tilde{\varphi}_k, 0)^\top$, $\Phi_{2k} = (0, \tilde{\varphi}_k)^\top$ in the case $d = 2$ and $\Phi_{3k-2} = (\tilde{\varphi}_k, 0, 0)^\top$, $\Phi_{3k-1} = (0, \tilde{\varphi}_k, 0)^\top$ and $\Phi_{3k} = (0, 0, \tilde{\varphi}_k)^\top$ in the case $d = 3$ where $\tilde{\varphi}_k := \varphi_k|_{\partial\Omega}$.
- In order to filter out tangential movements on the boundary, we impose the condition $\int_{\partial\Omega} (\mathbf{V}_h \cdot \tau)(x)\varphi_k(x) \, \mathrm{d}S = 0$, $k = 1, \dots, N_{\text{bdy}}$, which implies the pointwise condition $(\mathbf{V}_h \cdot \tau)(x_k) = 0$ for all mesh vertices $x_k$ on the boundary of $\Omega$. Here, the unit tangential vector $\tau$ on $\partial\Omega$ at a boundary vertex $x_k$ is obtained by averaging of the adjacent edges/faces.

Summarizing, we solve the problem to find $\mathbf{V}_h = \sum_{j=1}^{dN_{\text{bdy}}} \mathbf{V}_j \Phi_j \in H$ such that $\underline{V} := (\mathbf{V}_j)_{j=1}^{dN_{\text{bdy}}}$ together with $\underline{\xi} \in \mathbf{R}^{N_{\text{bdy}}}$ satisfies

$$\begin{pmatrix} \left[d^2\mathcal{J}(\Omega)(\Phi_j, \Phi_i)\right]_{i,j=1}^{dN_{\text{bdy}}} & B_{\tau,\Omega} \\ B_{\tau,\Omega}^\top & 0 \end{pmatrix} \begin{pmatrix} \underline{V} \\ \underline{\xi} \end{pmatrix} = \begin{pmatrix} -\left[d\mathcal{J}(\Omega)(\Phi_i)\right]_{i=1}^{dN_{\text{bdy}}} \\ 0 \end{pmatrix}. \tag{21}$$

with the matrix $B_{\tau,\Omega}$ defined by

$$\left[B_{\tau,\Omega}\right]_{i,k} = \int_{\partial\Omega} (\Phi_i \cdot \tau)\varphi_k \mathrm{d}S, \quad i = 1, \dots, dN_{\text{bdy}}, \; k = 1, \dots N_{\text{bdy}}. \tag{22}$$

The solution of (21) yields a vector field $\mathbf{V}_h \in H$ that is normal to $\partial\Omega$. In a next step, we compute a smooth extension $\hat{\mathbf{V}}_h$ of $\mathbf{V}_h$ by solving the Dirichlet problem of linear elasticity type to find $\hat{\mathbf{V}}_h \in \{\mathbf{v} \in V_h^d : \mathbf{v}|_{\partial\Omega} = \mathbf{V}_h\}$ such that

$$\int_\Omega 2\mu\varepsilon(\hat{\mathbf{V}}_h) : \nabla\mathbf{v} + \lambda \mathrm{div}(\hat{\mathbf{V}}_h)\mathrm{div}(\mathbf{v}) \, \mathrm{d}x = 0 \quad \text{for all } \mathbf{v} \in V_h^d \tag{23}$$

with the linearized strain tensor $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^\top)$ and Lamé parameters $\mu, \lambda$.

The proposed procedure is summarized in Algorithm 2. As a stopping criterion, we set a tolerance to the $L^2(\partial\Omega)$ norm of the Newton update $\mathbf{V}_h$. We mention that our algorithm is related to the method proposed in [13].

**Algorithm 2** Shape-Newton algorithm without regularization

```
Input: $\Omega^{(0)} \in \mathcal{A}$, tol, $iter_{\max}$, $j = 0$.
while $j < iter_{\max}$ do
  Assemble system matrix and right hand side $d\mathcal{J}(\Omega^{(j)})$ of (21)
  Solve system (21) to obtain $\mathbf{V}_h \in H$
  Extend $\mathbf{V}_h$ to $\hat{\mathbf{V}}_h \in V_h$ by (23)
  Update shape: $\Omega^{(j+1)} \leftarrow (\mathrm{id} + \hat{\mathbf{V}}_h)(\Omega^{(j)})$, $j \leftarrow j + 1$
  if $\|\mathbf{V}_h\|_{L^2(\partial\Omega)} <$ tol then
    break
  end if
end while
```

### 3.4 Arbitrary order shape differentiation

In order to treat higher order predictors, we will also deal with higher order shape derivatives.

**Definition 3.1.** *Let a domain $\Omega$, $k$ smooth vector fields $\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(k)}$, and $s \in \mathbf{R}^k$ with $s_i \geq 0$, $i = 1, \dots, k$ be given. Let the corresponding perturbed domain $\Omega_s := \left(\mathrm{id} + \sum_{i=1}^k s_i \mathbf{V}^{(i)}\right)(\Omega)$. The symmetric k-th order shape derivative of a shape function $\mathcal{J}(\Omega)$ at $\Omega$ in the directions of $\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(k)}$ is defined as*

$$d^k \mathcal{J}(\Omega)[\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(k)}] := \frac{d^k}{ds_1 \cdots ds_k} \mathcal{J}(\Omega_s)\bigg|_{s_1 = \cdots = s_k = 0}. \tag{24}$$

We illustrate how a higher order shape derivative can be computed in practice for a simple shape optimization problem.

**Example 3.2.** *Let $f : \mathbf{R}^d \to \mathbf{R}$ smooth and consider the cost function*

$$\mathcal{J}(\Omega) = \int_\Omega f(x)\ dx \tag{25}$$

*for smooth domains $\Omega$. Given $s \in \mathbf{R}^k$, $s \geq 0$ componentwise and the smooth vector fields $\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(k)} : \mathbf{R}^d \to \mathbf{R}^d$, we define the transformation $T_s(x) = (\mathrm{id} + \sum_{i=1}^k s_i \mathbf{V}^{(i)})(x)$ with Jacobian $F_s(x) := \partial T_s(x) = I + \sum_{i=1}^k s_i \partial \mathbf{V}^{(i)}(x)$. The k-th order shape derivative can now be obtained by transforming the perturbed domain $\Omega_s$ back to the original domain $\Omega$ and repeatedly differentiating the arising integrand with respect to $T_s$ in the direction of $\frac{dT_s}{ds_i} = \mathbf{V}^{(i)}(x)$ and with respect to $F_s$ in the direction of $\frac{dF_s}{ds_i} = \partial \mathbf{V}^{(i)}(x)$ for $i = k, k-1, \dots, 1$ at $s = 0$, i.e.,*

$$\begin{aligned} d^k \mathcal{J}(\Omega)[\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(k)}] &= \frac{d^k}{ds_1 \cdots ds_k} \int_\Omega (f \circ T_s)(x) det(F_s)\ dx \bigg|_{s=0} \\ &= \frac{d^k}{ds_1 \cdots ds_{k-1}} \int_\Omega \Big( (\nabla f) \circ T_s(x) \cdot \mathbf{V}^{(k)}(x) det(F_s) \\ &\quad + (f \circ T_s)(x) \frac{d}{dF_s} \left( det(F_s) \right) \partial \mathbf{V}^{(k)}(x) \Big)\ dx \bigg|_{s=0} = \dots \end{aligned}$$

*Here it is important to note that the introduced vector fields $\mathbf{V}^{(i)}$ are functions of the unperturbed variable $x$ and not of the perturbed variable $T_s(x)$, thus they do not yield extra terms when being differentiated with respect to $T_s$. This distinguishes the symmetric higher order shape derivative computed* *here from the quantity obtained by repeated shape differentiation as in* (17)*. We refer the reader to [18, Sec. 3.2] for a more detailed description and an implementation in the finite element software package NGSolve [35]. In that framework, using symbolic differentiation of the integrands, arbitrary order symmetric shape derivatives can be computed by repeating the procedure described above. This framework will also be exploited in the numerical experiments of Section 6.*

**Example 3.3.** *As we will make use of higher order shape derivatives also in the case of PDE-constrained shape optimization, we also illustrate their computation here. We consider a problem of the form*

$$\min_{\Omega \in \mathcal{A}} J(\Omega, u) \qquad s.t. \qquad u \in V : e(\Omega, u) = 0 \ in \ V^* \tag{26}$$

*and introduce the corresponding Lagrangian*

$$L(\Omega, u, p) := J(\Omega, u) + e(\Omega, u)(p). \tag{27}$$

*Assuming unique solvability of the PDE constraint for each admissible* $\Omega$ *and denoting the corresponding solution with* $u(\Omega)$*, we can also introduce the reduced cost function*

$$\mathcal{J}(\Omega) := J(\Omega, u(\Omega)). \tag{28}$$

*It is well-known [39] that the first order shape derivative of* (26) *is given by the partial derivative with respect to the shape of the Lagrangian when the corresponding state and adjoint are inserted, i.e.,*

$$d\mathcal{J}(\Omega)[\mathbf{V}] = \partial_\Omega L(\Omega, u(\Omega), p(\Omega))[\mathbf{V}] \tag{29}$$

*where* $u(\Omega)$ *and* $p(\Omega)$*, respectively, solve*

$$\text{Find } u \in V : \partial_p L(\Omega, u, 0)[\hat{p}] = 0 \qquad \text{for all } \hat{p} \text{ in } V, \tag{30}$$

$$\text{Find } p \in V : \partial_u L(\Omega, u, p)[\hat{u}] = 0 \qquad \text{for all } \hat{u} \text{ in } V. \tag{31}$$

*Here, the partial shape derivative* $\partial_\Omega$ *is to be understood as*

$$\partial_\Omega L(\Omega, w, q)[\mathbf{V}] := \frac{d}{ds} L(T_s(\Omega), \Psi_s(w), \Psi_s(q))\Big|_{s=0}$$

*with* $T_s(x) = (\mathrm{id} + s\mathbf{V})(x)$ *where the transformation* $\Psi_s$ *depends on the functional setting (e.g.,* $\Psi_s(w) = w \circ T_s^{-1}$ *if* $w \in H^1(\Omega)$*), see also [18, Sec. 4]. The second order shape derivative is formally obtained by an application of the chain rule,*

$$d^2\mathcal{J}(\Omega)[\mathbf{V}, \mathbf{W}] = \partial_\Omega \partial_\Omega L[\mathbf{V}, \mathbf{W}] + \partial_u \partial_\Omega L[\mathbf{V}, u'[\mathbf{W}]] + \partial_p \partial_\Omega L[\mathbf{V}, p'[\mathbf{W}]] \tag{32}$$

*where the material derivatives* $u'[\mathbf{W}] \in V$ *and* $p'[\mathbf{W}] \in V$ *can be obtained by differentiation of* (30)–(31) *with respect to the shape* $\Omega$ *in the direction* $\mathbf{W}$*,*

$$\partial_u \partial_p L(\Omega, u, 0)[\hat{p}, u'[\mathbf{W}]] = -\,\partial_\Omega \partial_p L(\Omega, u, 0)[\hat{p}, \mathbf{W}] \qquad \text{for all } \hat{p} \text{ in } V, \tag{33}$$

$$\begin{aligned} \partial_p \partial_u L(\Omega, u, p)[\hat{u}, p'[\mathbf{W}]] = &-\,\partial_\Omega \partial_u L(\Omega, u, p)[\hat{u}, \mathbf{W}] \\ &-\,\partial_u \partial_u L(\Omega, u, p)[\hat{u}, u'[\mathbf{W}]] \qquad \text{for all } \hat{u} \text{ in } V. \end{aligned} \tag{34}$$

*Here, we omitted the arguments* $(\Omega, u(\Omega), p(\Omega))$ *in* (32) *and* (34) *and* $(\Omega, u(\Omega), 0)$ *in* (33) *and wrote* $u'[\mathbf{W}]$*,* $p'[\mathbf{W}]$ *instead of* $u'(\Omega)[\mathbf{W}]$*,* $p'(\Omega)[\mathbf{W}]$ *for brevity. In the same manner, the third order shape derivative is obtained as*

$$\begin{aligned}
&d^3\mathcal{J}(\Omega)[\mathbf{V},\mathbf{W}] = \\
&\quad \partial_\Omega\partial_\Omega\partial_\Omega L[\mathbf{V},\mathbf{W},\mathbf{Z}] + \partial_u\partial_\Omega\partial_\Omega L[\mathbf{V},\mathbf{W},u'[\mathbf{Z}]] + \partial_p\partial_\Omega\partial_\Omega L[\mathbf{V},\mathbf{W},p'[\mathbf{Z}]] \\
&\quad + \partial_\Omega\partial_u\partial_\Omega L[\mathbf{V},u'[\mathbf{W}],\mathbf{Z}] + \partial_u\partial_u\partial_\Omega L[\mathbf{V},u'[\mathbf{W}],u'[\mathbf{Z}]] + \partial_p\partial_u\partial_\Omega L[\mathbf{V},u'[\mathbf{W}],p'[\mathbf{Z}]] \\
&\quad + \partial_\Omega\partial_p\partial_\Omega L[\mathbf{V},p'[\mathbf{W}],\mathbf{Z}] + \partial_u\partial_p\partial_\Omega L[\mathbf{V},p'[\mathbf{W}],u'[\mathbf{Z}]] \\
&\quad + \partial_u\partial_\Omega L[\mathbf{V},u''[\mathbf{W},\mathbf{Z}]] + \partial_p\partial_\Omega L[\mathbf{V},p''[\mathbf{W},\mathbf{Z}]]
\end{aligned}$$

*where the second order material derivatives* $u''[\mathbf{W},\mathbf{Z}], p''[\mathbf{W},\mathbf{Z}] \in V$ *can be obtained by further differentiating* (33)*,* (34)*, i.e., by solving*

$$\begin{aligned}
\partial_u\partial_p L[\hat{p}, u''[\mathbf{W},\mathbf{Z}]] &= -\partial_\Omega\partial_\Omega\partial_p L[\hat{p},\mathbf{W},\mathbf{Z}] - \partial_u\partial_\Omega\partial_p L[\hat{p},\mathbf{W},u'[\mathbf{Z}]] \\
&\quad - \partial_\Omega\partial_u\partial_p L[\hat{p},u'[\mathbf{W}],\mathbf{Z}] - \partial_u\partial_u\partial_p L[\hat{p},u'[\mathbf{W}],u'[\mathbf{Z}]], \\
\partial_p\partial_u L[\hat{u}, p''[\mathbf{W},\mathbf{Z}]] &= -\partial_u\partial_u L[\hat{u},u''[\mathbf{W},\mathbf{Z}]] \\
&\quad - \partial_\Omega\partial_\Omega\partial_u L[\hat{u},\mathbf{W},\mathbf{Z}] - \partial_u\partial_\Omega\partial_u L[\hat{u},\mathbf{W},u'[\mathbf{Z}]] - \partial_p\partial_\Omega\partial_u L[\hat{u},\mathbf{W},p'[\mathbf{Z}]] \\
&\quad - \partial_\Omega\partial_u\partial_u L[\hat{u},u'[\mathbf{W}],\mathbf{Z}] - \partial_u\partial_u\partial_u L[\hat{u},u'[\mathbf{W}],u'[\mathbf{Z}]] - \partial_p\partial_u\partial_u L[\hat{u},u'[\mathbf{W}],p'[\mathbf{Z}]] \\
&\quad - \partial_\Omega\partial_p\partial_u L[\hat{u},p'[\mathbf{W}],\mathbf{Z}] - \partial_u\partial_p\partial_u L[\hat{u},p'[\mathbf{W}],u'[\mathbf{Z}]],
\end{aligned}$$

*for all* $\hat{p}$ *and* $\hat{u}$ *in* $V$*, respectively. Note that we used the fact that second and higher order derivatives with respect to the third argument of the Lagrangian vanish by definition.*

# 4 Homotopy methods for second order shape optimization

In this section, we apply the concepts of homotopy methods introduced in Section 2 to the setting of solving shape optimization problems rather than solving systems of nonlinear equations. We will introduce two model problems in Section 4.1 before discussing a predictor-corrector scheme in Section 4.2.

## 4.1 Homotopies for shape optimization

Here, we discuss possible ways of defining homotopies for shape optimization problems. We begin with an academic unconstrained shape optimization problem in Section 4.1.1 and discuss the practically more relevant case of PDE-constrained shape optimization problems in Section 4.1.2.

### 4.1.1 An academic shape optimization problem

We begin by considering a shape optimization problem of the form

$$\min_{\Omega\in\mathcal{A}} \mathcal{J}_F(\Omega) = \int_\Omega f(x)\,\mathrm{d}x \tag{35}$$

for a given smooth function $f : \mathbf{R}^d \to \mathbf{R}$. Here, $\mathcal{A}$ denotes a set of admissible subsets of $\mathbf{R}^d$. It is straightforward to see that the exact solution of (35) is given by $\Omega^* = \{x \in \mathbf{R}^d : f(x) < 0\}$. Given sufficiently smooth vector fields $\mathbf{V}, \mathbf{W}$, first and second order shape derivatives are readily obtained as [18]

$$d\mathcal{J}_F(\Omega)(\mathbf{V}) = \int_\Omega \nabla f \cdot \mathbf{V} + f \,\mathrm{div}\mathbf{V} \,\mathrm{d}x, \tag{36}$$

$$\begin{aligned} d^2\mathcal{J}_F(\Omega)(\mathbf{V}, \mathbf{W}) = \int_\Omega & \nabla^2 f \,\mathbf{V} \cdot \mathbf{W} + \nabla f \cdot \mathbf{W} \,\mathrm{div}\mathbf{V} + \nabla f \cdot \mathbf{V} \,\mathrm{div}\mathbf{W} \\ & + f \,\mathrm{div}\mathbf{V} \,\mathrm{div}\mathbf{W} - f \partial\mathbf{V}^\top : \partial\mathbf{W} \,\mathrm{d}x. \end{aligned} \tag{37}$$

As noted in Remark 2.2, the concepts of Section 2 can be transfered to the context of optimzation with minor modifications. Thus, we define the convex combination of shape functionals

$$\mathcal{H}(\Omega, t) := t\mathcal{J}_F(\Omega) + (1-t)\mathcal{J}_G(\Omega) \tag{38}$$

and the corresponding family of shape optimization problems

$$\min_{\Omega \in \mathcal{A}} \mathcal{H}(\Omega, t), \tag{39}$$

where $\mathcal{J}_G(\Omega)$ is a simple shape functional with known minimizer. In particular, when starting with the initial design $\Omega^{(0)}$, one may choose

$$\mathcal{J}_G(\Omega) = \int_\Omega \psi[\Omega^{(0)}] \,\mathrm{d}x \tag{40}$$

where $\psi[\Omega^{(0)}]$ is a level set function representing $\Omega^{(0)}$, i.e., $\Omega^{(0)} = \{x \in \mathbf{R}^d : \psi[\Omega^{(0)}](x) < 0\}$. Obviously, $\Omega^{(0)}$ minimizes the functional $\mathcal{J}_G$ by construction.

**Remark 4.1.** *We remark that, of course, also other choices of homotopies are possible. In particular one could define the homotopy directly on the optimality condition of problem* (35) *and apply classical homotopies such as the Keller model to finding roots of* $H(\Omega, t) := d\mathcal{J}_F(\Omega)(\cdot) - (1-t)d\mathcal{J}_F(\Omega^{(0)})(\cdot)$. *While this approach is independent of an artificially chosen auxiliary problem, the resulting homotopy equation is posed in the dual of the space of vector fields. In order to filter out interior or tangential deformations, the equation would have to be restricted to normal vector fields on the boundary. For simplicity, we chose to restrict ourselves to the class of homotopies defined above, but believe that the Keller model deserves further investigation also in the context of shape optimization.*

### 4.1.2 A homotopy for PDE-constrained shape optimization

Similarly to the case of academic shape optimization problems of the type (35), we can also deal with PDE-constrained shape optimization problems of the form

$$\min_{\Omega \in \mathcal{A}} J_F(\Omega, u) \quad \text{s.t. } u \in V : e_F(\Omega, u) = 0 \text{ in } V^* \tag{41}$$

where $V$ denotes some Hilbert space, $J_F : \mathcal{A} \times V \to \mathbf{R}$ is the cost function and $e_F(\Omega, \cdot) : V \to V^*$ represents the PDE constraint which we assume to have a unique solution for any given $\Omega \in \mathcal{A}$. In order to define a homotopy, one can introduce an auxiliary PDE-constrained shape optimization problem whose solution can be easily obtained as

$$\min_{\Omega \in \mathcal{A}} J_G(\Omega, u) \quad \text{s.t. } u \in V : e_G(\Omega, u) = 0 \text{ in } V^* \tag{42}$$

Here, $J_G : \mathcal{A} \times V \to \mathbf{R}$ is a simple cost function and $e_G(\Omega, \cdot) : V \to V^*$ represents a simple PDE constraint which admits a unique solution for a given $\Omega \in \mathcal{A}$.

Defining $J_H(\Omega, u, t) := tJ_F(\Omega, u) + (1-t)J_G(\Omega, u)$ and $e_H(\Omega, u, t) := te_F(\Omega, u) + (1-t)e_G(\Omega, u)$ for $t \in [0, 1]$, we can now define a family of PDE-constrained shape optimization problems

$$\min_{\Omega \in \mathcal{A}} J_H(\Omega, u, t) \quad \text{s.t. } u \in V : e_H(\Omega, u, t) = 0 \text{ in } V^*. \tag{43}$$

Assuming unique solvability of the PDE constraint $e_H(\Omega, u, t) = 0$ for given $\Omega \in \mathcal{A}$ and $t \in [0, 1]$ and denoting the unique solution by $u(\Omega, t)$, we can define the reduced cost function $\mathcal{H}(\Omega, t) := J_H(\Omega, u(\Omega, t), t)$ and rewrite problem (43) as the family of shape optimization problems

$$\min_{\Omega \in \mathcal{A}} \mathcal{H}(\Omega, t). \tag{44}$$

**Remark 4.2.** *A simple concrete choice for* (42)*, which we also used in the numerical results of Section 6, is given by*

$$J_G(\Omega, u) = \int_\Omega u \; dx, \qquad e_G(\Omega, u) = \int_\Omega (u - \psi[\Omega^{(0)}])(\cdot) \; dx. \tag{45}$$

*Noting that the solution $u$ to the PDE constraint is just the $L^2$-projection of $\psi[\Omega^{(0)}]$, we see that $\Omega^{(0)}$ is very close to the solution of* (45) *in analogy to* (40).

**Remark 4.3.** *An alternative and possibly more straightforward approach would be to define the reduced cost functions $\mathcal{J}_F(\Omega)$ and $\mathcal{J}_G(\Omega)$ of* (41) *and* (42)*, respectively, and define the homotopy $\mathcal{H}(\Omega, t)$ as in* (38)*. This would mean that we just work with the averages of $u_G$ (solution to $e_G$) and $u_F$ (solution to $e_F$). In the approach defined here, we also average the equations, which is, in general, not the same. Along the same lines, one could also define the reduced cost function $\mathcal{J}_F(\Omega)$ of* (41) *and do a convex combination with the unconstrained problem* (40)*. However, we postulate that modeling a continuous transition between an artificial PDE constraint $e_G$ and the PDE of interest $e_F$ is beneficial for the arising homotopy path. We stress that a thorough comparison of different homotopies is beyond the scope of this article.*

### 4.2 A predictor-corrector scheme for shape optimization

The procedure for solving shape optimization problems (35) or (41) using homotopies such as the ones defined in (39) and (44) is summarized in Algorithm 3. Starting out from a local minimizer $\Omega(0)$ of an auxiliary problem (40) or (42), we proceed by finding stationary points of (39) and (44) for increasing values of the homotopy parameter $t$, i.e., by solving

$$d\mathcal{H}(\Omega(t), t)[\cdot] = 0 \tag{46}$$

for $\Omega(t)$. We give details about the predictor step, the step size choice and the corrector steps in Sections 4.2.1, 4.2.2 and 4.2.3, respectively.

**Algorithm 3** Predictor-corrector for shape optimization

```
Input: $\Omega^{(0)} \in \mathcal{A}, t^{(0)} = 0, k = 0$.
Choose $\Delta t^{(0)}$
while $t^{(k)} < 1$ do
  Set $t^{(k+1)} = \min(t^{(k)} + \Delta t^{(k)}, 1)$.
  Predictor: $\tilde{\Omega}^{(k+1)} = \mathrm{Pred}^{(q)}(\Omega^{(k)}, t^{(k)}, t^{(k+1)} - t^{(k)})$
  Corrector: Find stationary point $\Omega^{(k+1)}$ of $\mathcal{H}(\Omega, t^{(k+1)})$ starting from $\tilde{\Omega}^{(k+1)}$ by Algorithm 2.
  if corrector step successful then
    Choose $\Delta t^{(k+1)}$
    $k \leftarrow k + 1$
  else
    Reduce $\Delta t^{(k)}$
  end if
end while
```

### 4.2.1 Predictors in shape optimization

The results of Theorem 2.5 carry over to this case by formally replacing differentiation with respect to $x$ by shape differentiation. Exemplarily, we give the equations defining the first, second and third order derivatives $\Omega'(t)$, $\Omega^{[2]}(t)$ and $\Omega^{[3]}(t)$ of the mapping $t \mapsto \Omega(t)$, cf. Corollary 2.7,

$$d^2\mathcal{H}[\Omega'(t), \cdot] = -\, d\mathcal{H}_t[\cdot], \tag{47}$$

$$d^2\mathcal{H}[\Omega^{[2]}(t), \cdot] = -\, d^3\mathcal{H}[\Omega'(t), \Omega'(t), \cdot] - 2d^2\mathcal{H}_t[\Omega'(t), \cdot] - d\mathcal{H}_{tt}, \tag{48}$$

$$\begin{aligned} d^2\mathcal{H}[\Omega^{[3]}(t), \cdot] = &- d^4\mathcal{H}[\Omega'(t), \Omega'(t), \Omega'(t), \cdot] - 3d^3\mathcal{H}_t[\Omega'(t), \Omega'(t), \cdot] \\ &- 3d^3\mathcal{H}[\Omega^{[2]}(t), \Omega'(t), \cdot] - 3d^2\mathcal{H}_t[\Omega^{[2]}(t), \cdot] \\ &- 3d^2\mathcal{H}_{tt}[\Omega'(t), \cdot] - d\mathcal{H}_{ttt}[\cdot], \end{aligned} \tag{49}$$

where we again omitted the arguments $(\Omega(t), t)$ of $\mathcal{H}$ for brevity. Again, note that the operator on the left hand side which is to be inverted is the same in all cases and thus, after discretization, the corresponding matrix has to be assembled (and, in the case of direct solvers, also factorized) only once for given $(\Omega(t^{(k)}), t^{(k)})$. Also note that this operator is the same as the one used in each iteration of Newton's method, cf. Section 4.2.3. The computation of the higher order shape derivatives appearing in (47)–(49) is discussed in Section 3.4, see also [18]. Note that systems (47)–(49) have to be solved as discussed in Section 3.3 and the predictors $\Omega'(t)$, $\Omega^{[2]}(t)$, $\Omega^{[3]}(t)$ are actually vector fields defined only on the boundary of $\Omega$ and have to be extended to the interior by (23). Given $q$ derivatives of the domain with respect to the homotopy parameter $t$ and denoting the extension of the $i$-th derivative $\Omega^{[i]}(t)$ by $\hat{\Omega}^{[i]}(t)$, we can define the $q$-th order shape predictor at homotopy parameter $t^{(k)}$ by

$$\mathrm{Pred}^{(q)}(\Omega^{(k)}, t^{(k)}, \Delta t^{(k)}) := \left(\mathrm{id} + \sum_{i=1}^{q} \frac{1}{i!} \left(\Delta t^{(k)}\right)^i \hat{\Omega}^{[i]}(t^{(k)})\right) \left(\Omega^{(k)}\right). \tag{50}$$

**Remark 4.4.** *Note that, for $q > 1$, it can be useful to, instead of extending each of the terms $\Omega^{[i]}$*

*individually, to apply the extension operator only to the weighted sum in* (50)*. However, this entails that the extension has to be repeated every time when the homotopy step size* $\Delta t^{(k)}$ *is reduced.*

### 4.2.2 Step size choice

The step size choices presented in Sections 2.2.1–2.2.3 can be directly transfered to the setting of shape optimization. We mention that for the choice (14), in our experiments the magnitude of the predictor of order $q+1$ is measured as $\|\Omega^{[q+1]}(t^{(k)})\|_{L^2(\partial\Omega)^d}$.

### 4.2.3 Shape-Newton method

In the corrector step, given a homotopy value $t = t^{(k+1)}$ and starting out from a starting guess $\tilde{\Omega}^{(k+1)}$, we solve optimality condition (46) by Newton's method as described in Algorithm 2.

In the unconstrained case, the Newton system is as given in (21) with $\mathcal{J}(\Omega)$ replaced by $\mathcal{H}(\Omega, t)$.

In the case of PDE-constrained shape optimization, the optimality condition (46) can be written as

$$\begin{pmatrix} \partial_\Omega L_H(\Omega, u, p; t)[\cdot] \\ \partial_u L_H(\Omega, u, p; t)[\cdot] \\ \partial_p L_H(\Omega, u, 0; t)[\cdot] \end{pmatrix} = 0 \tag{51}$$

for the Lagrangian $L_H(\Omega, u, p; t) := J_H(\Omega, u; t) + e_H(\Omega, u; t)(p)$. Assuming that the state $u$ and adjoint $p$ are approximated by finite element functions $u_h, p_h \in V_h$ with $V_h$ as defined in (20), i.e., $u_h(x) = \sum_{j=1}^N u_j \varphi_j(x)$, $p_h(x) = \sum_{j=1}^N p_j \varphi_j(x)$ with coefficient vectors $\underline{u} = (u_1, \dots, u_N)^\top, \underline{p} = (p_1, \dots, p_N)^\top \in \mathbf{R}^N$, the corresponding discretized Newton system (21) reads

$$\begin{pmatrix} A_{\Omega,\Omega} & B_{\tau,\Omega} & A_{u,\Omega} & A_{p,\Omega} \\ B_{\tau,\Omega}^\top & & & \\ A_{\Omega,u} & & A_{u,u} & A_{p,u} \\ A_{\Omega,p} & & A_{u,p} & \end{pmatrix} \begin{pmatrix} \underline{V} \\ \underline{\xi} \\ \underline{u} \\ \underline{p} \end{pmatrix} = - \begin{pmatrix} \big[\partial_\Omega L_H[\Phi_i]\big]_{i=1,\dots,dN_{\mathrm{bdy}}} \\ \\ \big[\partial_u L_H[\varphi_l]\big]_{l=1,\dots,N} \\ \big[\partial_p L_H[\varphi_l]\big]_{l=1,\dots,N} \end{pmatrix} \tag{52}$$

with the matrix entries

$$\begin{aligned} (A_{\Omega,\Omega})_{i,j} &= \partial_\Omega \partial_\Omega L_H[\Phi_i, \Phi_j], & (A_{u,\Omega})_{i,k} &= \partial_u \partial_\Omega L_H[\Phi_i, \varphi_k], & (A_{p,\Omega})_{i,k} &= \partial_p \partial_\Omega L_H[\Phi_i, \varphi_k], \\ (A_{u,u})_{l,k} &= \partial_u \partial_u L_H[\varphi_l, \varphi_k], & (A_{p,u})_{l,k} &= \partial_p \partial_u L_H[\varphi_l, \varphi_k], \end{aligned}$$

for $i, j = 1, \dots, dN_{\mathrm{bdy}}$ and $k, l = 1, \dots, N$; moreover, $A_{\Omega,u} = A_{u,\Omega}^\top$, $A_{\Omega,p} = A_{p,\Omega}^\top$, $A_{u,p} = A_{p,u}^\top$ and $B_{\tau,\Omega}$ is as defined in (22). Here, again, we skipped the arguments $(\Omega, u_h, p_h; t)$ of $L_H$.

**Remark 4.5.** *For efficiency reasons, it seems reasonable to adapt the stopping criterion of the corrector step in the course of the homotopy. Since one is interested in accurate solutions of* (46) *only for* $t = 1$*, we suggest to adapt the stopping criterion by using* $((1-t)r + t)\mathrm{tol}$ *for a certain ratio* $r \gg 1$ *and the tolerance* tol *that is desired at* $t = 1$*.*

## 5 Homotopy methods for multi-objective shape optimization

In multi-objective optimizaion one wants to simultaneously minimize multiple, possibly conflicting, cost functions. Since, in most cases, there does not exist a unique minimizer for all cost functions at the same time, one is interested in locally *Pareto optimal* designs, i.e., designs which cannot be improved with respect to any of the given cost functions without deteriorating another cost function value. The set of all Pareto optimal points is usually called Pareto set and their image (under the application of the given cost functions) is called the Pareto front. Given the Pareto set, a designer can choose the most suitable solution for the particular problem at hand, often accounting for practical constraints such as manufacturability.

As a by-product of the chosen convex homotopy (38), the solutions to all intermediate problems $\mathcal{H}(\cdot,t)=0$, $0\leq t\leq 1$ with $\mathcal{H}$ defined by (38), correspond to locally Pareto optimal points of the bi-objective optimization problem to minimize $(\mathcal{J}_G(\Omega),\mathcal{J}_F(\Omega))^\top$ given by (40), (35). While this property seems to be of limited use since $\mathcal{G}$ represents an artificial, non-physical problem, we can, of course, also apply the methodology to connect two different physically relevant cost functions $\mathcal{J}_{F,1}(\Omega)$ and $\mathcal{J}_{F,2}(\Omega)$ by a homotopy $\mathcal{H}^{1,2}(\Omega,t):=(1-t)\mathcal{J}_{F,1}(\Omega)+t\mathcal{J}_{F,2}(\Omega)$. Starting out from a stationary point of $\mathcal{J}_{F,1}$ (which may itself have been obtained by a homotopy method) and $t=0$, the predictor-corrector method of Algorithm 3 computes stationary points of $\mathcal{H}^{1,2}(\cdot,t)$ for different values of $t$, each of which corresponds to a point in the Pareto set.

We mention that the described concept of Pareto front tracing by predictor-corrector-type homotopy methods has previously been applied to multi-objective optimization [19, 20, 25, 32]. Notably, [19, 20] treats the general case of an arbitrary number of cost functions, resulting in a generalized homotopy method with a multi-dimensional homotopy parameter. We also refer the interested reader to [4] for numerical integration techniques applied to the corresponding Davidenko differential equation (8). Beside the recent application [6] of some of the authors in the context nonlinear magnetostatics, to the best of our knowledge, the application in (free-form) shape optimization is novel.

**Remark 5.1.** *A common approach to obtaining a Pareto set by gradient-based optimization methods is the so-called weighted sum method where a locally Pareto optimal point is obtained as the result of a steepest descent method applied to a convex combination of given cost functions. Here, different points are obtained by varying the weight in the convex combination, i.e., a full optimization run is required in order to get one additional point on the Pareto front. With the homotopy method on the contrary, a neighboring point on the Pareto front can be obtained by solving just one corrector equation of the type* (21) *(with $\mathcal{J}(\Omega)$ replaced by $\mathcal{H}^{1,2}(\Omega,t)$), typically resulting in a significant speed-up of the procedure. Moreover, in contrast to the weighted sum method, the homotopy method is also able to determine the Pareto front if it is non-convex (more precisely: if the set in the objective space that is bounded by the Pareto front is non-convex) [19, 20].*

**Remark 5.2.** *The approach is also applicable in the case of PDE-constrained shape optimization. In this setting, however, the corresponding PDE constraints (if they are different) should not be mixed as in* (43)*, but the convex combination of the corresponding reduced cost functions should be used.*

**Remark 5.3.** *We mention that, in practice, Pareto fronts may be non-connected and multiple starting points may be necessary to find different parts of the Pareto front.*

# 6 Numerical results

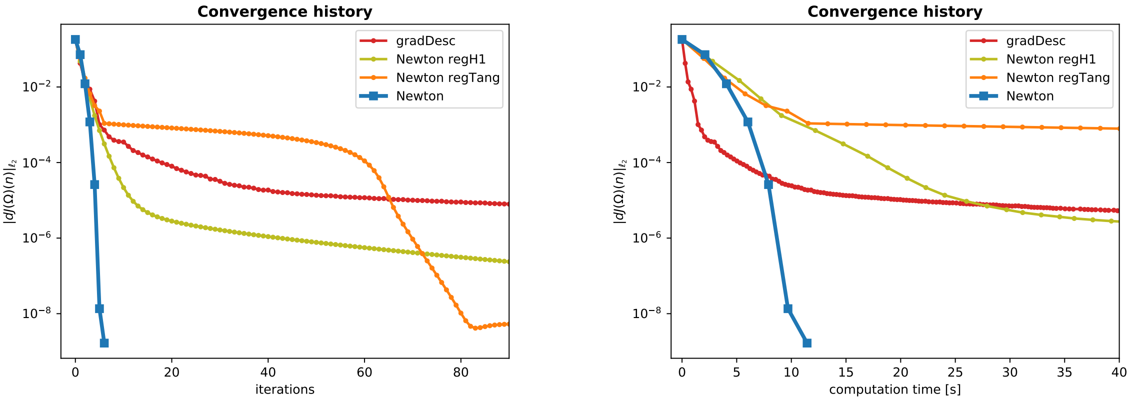
Finally, we present numerical results in which we illustrate different aspects of the method described above. We begin by illustrating the shape-Newton method proposed in Section 3.3, before showing results obtained by the homotopy method for shape optimization without and with PDE constraints. We conclude the section with a multi-objective shape optimization problem.

All numerical experiments are conducted within the finite element software NGSolve [35] using globally continuous, piecewise linear finite elements to represent both the deformations and also solutions of constraining PDEs. All arising systems of linear equations are solved by sparse direct solvers. The corrector equations $\mathcal{H}(\cdot, t) = 0$ are solved by Algorithm 3 where tol is chosen in dependence of the homotopy parameter, $\text{tol} = (1-t)10^{-4} + t10^{-10}$, since high accuracy is only required for larger values of $t$. Concerning the discussion in Remark 4.4, we chose to compute an extension for the sum of all predictors rather than for each term individually.

The code used for obtaining the results presented below is available at [5].

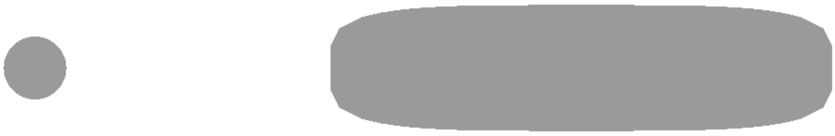
## 6.1 Local convergence of shape-Newton algorithm

First, we numerically investigate the unregularized shape-Newton method proposed in Section 3.3, which will be used as a corrector in the experiments of the subsequent subsections. Since the method is only locally convergent, we apply it to a problem whose optimal solution is known and close to the initial design. More precisely, we consider the setting of Section 4.1.1 with the integrand of (35), $f(x_1, x_2) = x_1^2/a^2 + x_2^2/b^2 - 1$, being a level set function of an ellipse centered at the origin with major and minor axes $a = 1.25$, $b = 1/a$, respectively. The initial design is the unit ball $\Omega^{(0)} = B_1(0)$ which is discretized by a triangular mesh with 3788 triangles and 1965 vertices and the integrand in (40) is given by $\psi[\Omega^{(0)}](x_1, x_2) = x_1^2 + x_2^2 - 1$. Figure 2 shows the convergence history of the unregularized shape-Newton method in comparison with other first and second order methods applied to the same problem. In particular with compare with: (i) a regular shape gradient method where the descent direction is obtained by solving a problem like (18) with a bilinear form of linear elasticity type (23); (ii) a regularized shape-Newton approach where the shape Hessian in (19) is regularized by addition of an $H^1$ inner product, i.e. $\widetilde{d^2\mathcal{J}}(\Omega)(\mathbf{V}, \mathbf{W}) = d^2\mathcal{J}(\Omega)(\mathbf{V}, \mathbf{W}) + \int_\Omega \delta_{1,a}\partial\mathbf{V} : \partial\mathbf{W} + \delta_{1,b}\mathbf{V}\cdot\mathbf{W} \, \mathrm{d}x$ as [31, 33]; and (iii) the tangentially regularized shape Hessian $\widetilde{d^2\mathcal{J}}(\Omega)(\mathbf{V}, \mathbf{W}) = d^2\mathcal{J}(\Omega)(\mathbf{V}, \mathbf{W}) + \delta_2 \int_\Omega (\mathbf{V}\cdot\tau)(\mathbf{W}\cdot\tau) \, \mathrm{d}x$ proposed in [18]. Of course the results for approaches (ii) and (iii) depend on the choices of the parameters $\delta_{1,a}$, $\delta_{1,b}$, $\delta_2$, which we chose empirically as $\delta_{1,a} = \delta_{1,b} = 0.5$ and $\delta_2 = 250$ as these values yielded the best results based on multiple conducted experiments. Figures 2(a) and (b) show the evolution of the $\ell_2$-norm of the residual in normal direction $d\mathcal{J}(\Omega)(\vec{\mathbf{n}}) := [d\mathcal{J}(\Omega)(\Phi_{2i}n_1(x_i) + \Phi_{2i+1}n_2(x_i))]_{i=1,\dots N_{\mathrm{bdy}}}$ in the course of the optimization iterations and as a function of computing time, respectively. Here $(n_1(x_i), n_2(x_i))^\top$



(a) (b)

Figure 2: Convergence history of four different methods applied to the problem described in Section 6.1: Shape derivative in normal direction $\|d\mathcal{J}(\Omega)(\vec{\mathbf{n}})\|_{\ell_2}$ as function of (a) iterations and (b) computation time.



(a) (b)

Figure 3: (a) Initial design and (b) optimal design for problem (35) with function $f$ given by (53).

denotes the normal vector at the boundary point $x_i$ (obtained by averaging from the adjacent edges). Note that, in the discretized setting, the full shape derivative vector $[d\mathcal{J}(\Omega)(\Phi_i)]_{i=1,\dots,dN}$ may contain tangential components. It can be seen that the proposed unregularized Newton method can reach high accuracy within very few iterations. Moreover, for the given problem size, the method also performs very well in terms of computing time.

**Remark 6.1.** *The poor behavior of the shape gradient method can be attributed to the fact that the discretized shape derivative may contain spurious components corresponding to interior or tangential perturbations. In [13], the authors propose to allow only for mesh deformations that correspond to normal forces acting on the boundary of the shape which can significantly improve the reached accuracy and avoid mesh degeneracy. However, even in that setting, the number of necessary iterations remains high and the authors also propose a second order method based on this principle.*

### 6.2 Application of homotopy method to simple shape optimization problem

In this subsection, we illustrate the application of the homotopy methods discussed in Section 4 to the simple shape optimization problem (35). Also here, we consider as initial guess (an approximation of) the unit ball $\Omega^{(0)} = B_1(0)$ which is represented by the level set function $\psi[\Omega^{(0)}](x_1, x_2) = x_1^2 + x_2^2 - 1$. For the function $f$ in (35) we choose a function whose zero level set describes a large $p$-ellipse ,

$$f(x_1, x_2) = (x_1/a)^p + (x_2/b)^p - R^p, \tag{53}$$

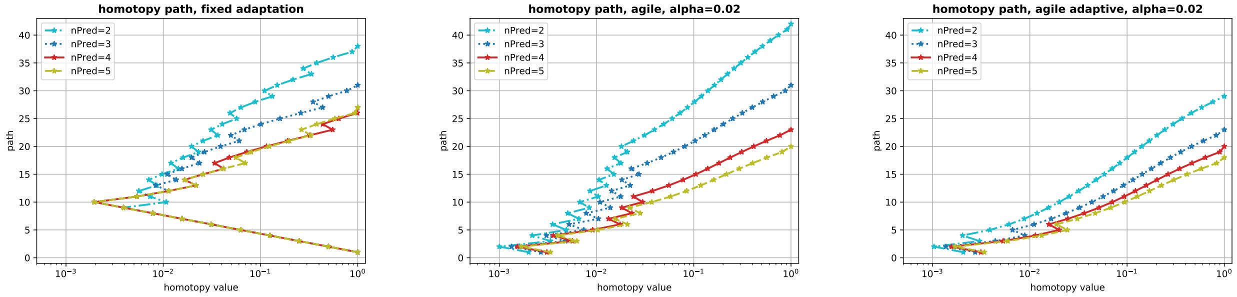


Figure 4: Homotopy paths for simple shape optimization example of Section 6.2 for $q$ = nPred = $2, 3, 4, 5$. (a) Fixed step size adaptation with initial step size $\Delta t^{(0)} = 1$ and update factors $\underline{\gamma} = 0.5$, $\overline{\gamma} = 1.75$. (b) Agile step size adaptation with $\alpha = 0.02$. (c) Agile step size strategy with adaptation of $\alpha$ with initial choice $\alpha = 0.02$ and update factors $\underline{\alpha} = 0.5$, $\overline{\alpha} = 1.1$.

with $p = 4$, semi-axes $a = 2$, $b = 0.5$ and radius $R = 4$, see Figure 3. We discretized the initial design into a triangular mesh consisting of 2992 triangles and 1707 vertices. The example is chosen such that the optimal solution is relatively far away from the initial guess. As a result, Newton's method as described in Section 6.1 fails to converge. Moreover, the evolution of a shape gradient method (with a line search for the step size) is very slow. After 10000 iterations and a runtime of 88 minutes, the residual was reduced only to about $5 \cdot 10^{-3}$ (recall Remark 6.1). For this example, we discuss the different potential ingredients to the predictor-corrector method described by Algorithm 3.

### 6.2.1 A first experiment with second order predictor

In a first experiment, we use a second order predictor, $q = 2$, together with the fixed adaptation step size choice described in Section 2.2.1. We start with $t^{(0)} = 0$ and the initial step size $\Delta t^{(0)} = 1$ and choose the reduction and increase factors $\underline{\gamma} = 0.5$ and $\overline{\gamma} = 1.75$, respectively. Note that $\Delta t^{(0)} = 1$ is the most aggressive choice which would result in a pure Newton method in case of success.

After 39 visited homotopy values, out of which 22 were successful, a solution of $\mathcal{H}(\cdot, 1) = 0$ and thus a stationary point of (35) is found, see Figure 3(b). The solution required a total of 76 Newton steps and 42 solutions of predictor equations of the form (47) or (48). While the computation time for all corrector steps was 323.21 seconds, the computation for the predictors was only 77.33 seconds. This difference can be explained by the fact that, for higher order predictors (48), the system matrix does not have to be assembled and factorized again as it coincides with the system matrix of the previously computed first order predictor (47).

### 6.2.2 Effect of higher order predictor terms

Next we investigate the effect of increasing the number of predictor terms in (50). We re-run the same experiment with $q = 3, 4, 5$. Figure 4(a) compares the different homotopy values attempted by our optimization runs. Depending on whether the corrector step for a given homotopy value was successful or not, the homotopy value is increased or decreased and the path proceeds to the

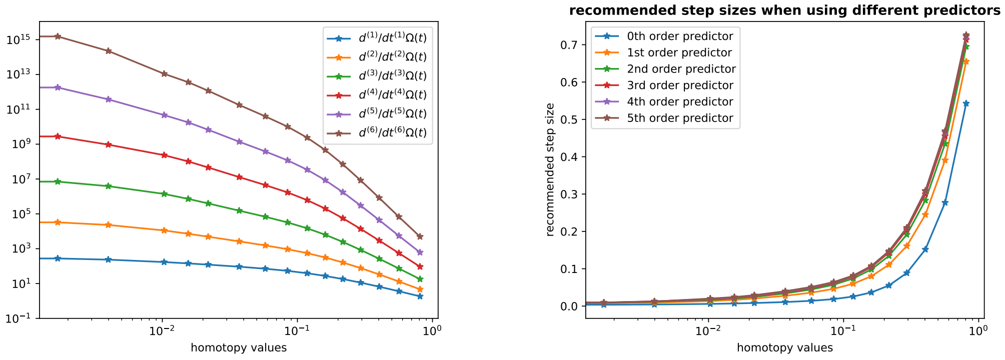
Table 1: Section 6.2: three different step size choices and predictors of order $q = 2, 3, 4, 5$, cf. Fig. 4.
"Hsteps" corresponds to visited homotopy values: successful \| insuccessful \| **total**;
"Nsteps" corresponds to number of solves of linear systems: predictor \| corrector \| **total**;
"Ntime" corresponds to computation time for linear systems in seconds: predictor \| corrector \| **total**

| $q$ | | fixed adaptation | agile $\alpha = 0.02$ | agile $\alpha = 0.02$ adaptive |
|---|---|---|---|---|
| 2 | Hsteps | 22 \| 17 \| **39** | 33 \| 10 \| **43** | 28 \| 2 \| **30** |
| | Nsteps | *42* \| 76 \| **118** | *96* \| 80 \| **176** | *81* \| 65 \| **146** |
| | Ntime | *77.33* \| 323.21 \| **400.54** | *140.75* \| 346.44 \| **487.19** | *118.24* \| 279.41 \| **397.65** |
| 3 | Hsteps | 18 \| 14 \| **32** | 24 \| 8 \| **32** | 22 \| 2 \| **24** |
| | Nsteps | *51* \| 59 \| **110** | *92* \| 57 \| **149** | *84* \| 48 \| **132** |
| | Ntime | *73.48* \| 252.61 \| **326.09** | *141.17* \| 243.11 \| **384.28** | *128.18* \| 205.32 \| **333.51** |
| 4 | Hsteps | 16 \| 12 \| **28** | 19 \| 5 \| **24** | 19 \| 2 \| **21** |
| | Nsteps | *60* \| 46 \| **106** | *90* \| 42 \| **132** | *90* \| 42 \| **132** |
| | Ntime | *101.79* \| 215.76 \| **317.55** | *209.44* \| 212.32 \| **421.76** | *189.06* \| 185.89 \| **374.95** |
| 5 | Hsteps | 16 \| 12 \| **28** | 17 \| 4 \| **21** | 17 \| 2 \| **19** |
| | Nsteps | *75* \| 42 \| **117** | *96* \| 37 \| **133** | *96* \| 37 \| **133** |
| | Ntime | *178.74* \| 212.29 \| **391.03** | *285.08* \| 157.3 \| **442.38** | *284.7* \| 157.82 \| **442.51** |

right or left, respectively. Figure 4(a) shows which homotopy values were passed for each of the four investigated optimization runs. The corresponding numbers of predictor and Newton steps as well as their timings are collected in the left column of Table 1. It can be seen from Figure 4(a) that, by employing higher order predictors, the number of necessary intermediate homotopy values can be decreased significantly. This way, the total number of Newton steps in the corrector stage gets smaller and smaller. On the other hand, however, the computational cost for computing these higher order predictors increases with their order. We emphasize that this is not primarily due to the additional solution of the corresponding predictor equations (10), but much more due to the cost of assembling their right hand sides.

### 6.2.3 Effect of step size choice

Let us now have a closer look at the step size choices discussed in Sections 2.2 and 4.2.2. Note that, for the fixed adaptation strategy according to Section 2.2.1 used in the experiments above, one has to choose three parameters (initial step size $\Delta t^{(0)}$ as well as the increase and decrease parameters $\overline{\gamma}, \underline{\gamma}$). The agile step size choice, on the other hand, automatically chooses the step size according to the Taylor remainder of the homotopy curve given a predictor of certain order. Thus, this method requires to set only one parameter $\alpha$ which is related to the allowed deviation from the homotopy curve. Application of this agile step size rule with $\alpha = 0.02$ is illustrated in Figure 4(b). When comparing with the result in Figure 4(a), it can be seen that the method can automatically detect the order of the initial step size $\Delta t^{(0)}$. For instance, for $q = 2$, the proposed step size has to be halvened only once compared to eight times in Figure 4(a) when starting from the value $\Delta t^{(0)} = 1$. The problem at hand requires comparably small step sizes in the beginning while larger step sizes can be made later on. Figure 5(a) shows the $L^2(\partial\Omega)$-norms of the different predictors $\Omega^{[j]}(t_k)$ for the



(a)

(b)

Figure 5: Example of Section 6.2: (a) $L^2(\partial\Omega)$-norms of derivatives of $t \mapsto \Omega(t)$ of different orders (see, e.g., (47)–(49)). (b) Recommended step sizes according to (14) for $\alpha = 0.02$.

derivative orders $j = 1, \ldots, 6$ for the homotopy values $t_k$ arising in the experiment of Figure 4(b). Figure 5(b) shows the step sizes to be taken according to this rule as a result of (14). Note that the step size rule automatically detects that small steps should be taken in the initial phase and larger steps are allowed later in the solution process.

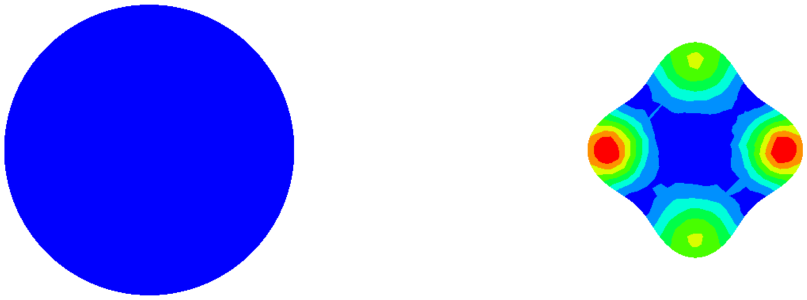
While this agile step size choice has the advantage of adapting the step size to the properties of the physical problem, it still depends on the parameter $\alpha$. As it can be seen from Figure 4(b), if the value of $\alpha$ is chosen too conservatively, the procedure may take many iterations. Therefore, we also investigate the strategy proposed in Section 2.2.3 and increase or decrease the value of $\alpha$ by constant factors $\overline{\alpha} > 1 > \underline{\alpha}$ after every successful or insuccessful homotopy step, respectively. As it can be seen from Figure 4(c) where we used $\overline{\alpha} = 1.1$, $\underline{\alpha} = 0.5$, the total number of homotopy steps can be reduced in this way. We remark, however, that for the two strategies followed in Figures 4(b) and (c), an additional predictor term has to be computed. Thus, their applicability depends on whether an additional predictor term can be obtained without too much additional computational effort, see also the discussion in Remark 6.3. A comparison of the three strategies in terms of number of predictor or Newton solves and computation time can be found in Table 1.

**Remark 6.2.** *When shape optimization is performed based on mesh deformations, the aspect of mesh quality plays an important role, in particular in the presence of large deformations. Since the computation of mesh quality preserving descent directions is an active field of research on its own, we largely exclude this aspect from the discussion here and just mention that the mesh of Figure 4(b) is distorted, but not degenerated. In order to deal with large deformations, one could employ local mesh refinement or remeshing techniques [1].*

### 6.3 Application of homotopy method to PDE-constrained shape optimization

Next, we consider the same strategies for a PDE-constrained shape optimization problem of the form (41) with $V = H^1(\Omega)$,

$$F(\Omega, u) = \int_\Omega u(x) \, \mathrm{d}x, \tag{54}$$



(a) (b)

Figure 6: (a) Initial design and (b) final design for problem (41) with $F$ and $e_F(\Omega, u)$ given by (54) and (55), respectively.

and the semilinear PDE constraint

$$\langle e_F(\Omega, u), v\rangle = \int_\Omega \left[\lambda(x)\nabla u \cdot \nabla v + u^3 v - f_{\mathrm{clov}} v\right] \,\mathrm{d}x, \; v \in H^1(\Omega), \tag{55}$$

with $\lambda(x_1, x_2) = 1/(1 + x_1^2)$ and the right-hand side function whose zero level set resembles a clover shape

$$\begin{aligned} f_{\mathrm{clov}}(x_1, x_2) = &\left(\sqrt{(x_1 - a)^2 + bx_2^2} - 1\right)\left(\sqrt{(x_1 + a)^2 + bx_2^2} - 1\right) \\ &\left(\sqrt{bx_1^2 + (x_2 - a)^2} - 1\right)\left(\sqrt{bx_1^2 + (x_2 + a)^2} - 1\right) - \epsilon, \end{aligned} \tag{56}$$

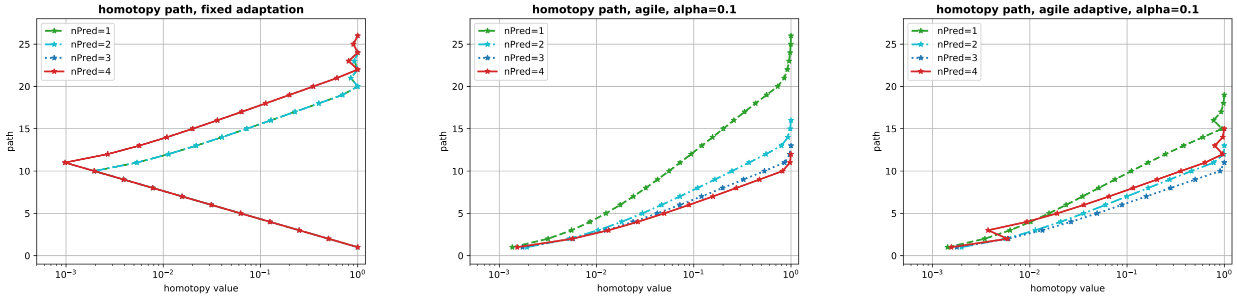
where $a = 0.8$, $b = 2$ and $\epsilon = 0.01$. We choose the homotopy defined by (43) with the simple problem (45) and the initial shape $\Omega^{(0)} = B_{2.5}(0)$ being the ball of radius 2.5 centered at the origin with the corresponding initial level set function $\psi[\Omega^{(0)}](x_1, x_2) = x_1^2 + x_2^2 - 2.5^2$. We used a mesh consisting of 650 triangles and 384 vertices. The initial design and the final design with the solution of the state equation are depicted in Figure 6. The homotopy paths for the same three different step size choices as discussed in Section 6.2 are shown in Figure 7 and corresponding numbers of predictor and Newton solves as well as timings are given in Table 2.

**Remark 6.3.** *As it can be seen from Table 2, the computation time for computing predictors grows fast with their order. This is mostly due to the assembly of the right hand sides of* (10) *where automated differentiation is used and could be reduced if higher order derivatives were provided in closed form. While the focus of this study is the investigation of the effects of higher order predictors, we mention that efficiency could be improved, e.g., by constructing predictors at* $t^{(k)}$ *by extrapolation from previously visited homotopy values* $(t^{(j)}, x(t^{(j)}))$, $j \leq k$ *as in multi-step methods for ordinary differential equations.*

### 6.4 Pareto front tracing in shape optimization

Finally, we illustrate the application of the proposed homotopy method for tracing Pareto optimal shapes. For that purpose, we consider three instances of the simple academic shape optimization problem (35) with integrands

$$f_1(x_1, x_2) = \frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 2^2, \qquad f_2(x_1, x_2) = f_{\mathrm{clov}}(x_1, x_2), \qquad f_3(x_1, x_2) = \frac{x_1^2}{b^2} + \frac{x_2^2}{a^2} - 2^2,$$



(a) (b) (c)

Figure 7: Homotopy paths for PDE-constrained example of Section 6.3 for $q = \text{nPred} = 1, 2, 3, 4$. (a) Fixed step size adaptation with initial step size $\Delta t^{(0)} = 1$ and update factors $\underline{\gamma} = 0.5$, $\overline{\gamma} = 1.75$. (b) Agile step size adaptation with $\alpha = 0.1$. (c) Agile step size strategy with adaptation of $\alpha$ with initial choice $\alpha = 0.1$ and update factors $\underline{\alpha} = 0.5$, $\overline{\alpha} = 1.1$.

Table 2: Section 6.3: three different step size choices and predictors of order $q = 1, 2, 3, 4$, cf. Fig. 7. "Hsteps" corresponds to visited homotopy values: successful | insuccessful | **total**; "Nsteps" corresponds to number of solves of linear systems: predictor | corrector | **total**; "Ntime" corresponds to computation time for linear systems in seconds: predictor | corrector | **total**

| $q$ | | fixed adaptation | agile $\alpha = 0.1$ | agile $\alpha = 0.1$ adaptive |
|---|---|---|---|---|
| 1 | Hsteps | 14 \| 11 \| **25** | 27 \| 0 \| **27** | 19 \| 1 \| **20** |
| | Nsteps | *13* \| 61 \| **74** | *52* \| 70 \| **122** | *36* \| 55 \| **91** |
| | Ntime | *18.43* \| 84.43 \| **102.86** | *43.46* \| 88.75 \| **132.2** | *30.55* \| 71.37 \| **101.92** |
| 2 | Hsteps | 14 \| 11 \| **25** | 17 \| 0 \| **17** | 14 \| 0 \| **14** |
| | Nsteps | *26* \| 48 \| **74** | *48* \| 43 \| **91** | *39* \| 38 \| **77** |
| | Ntime | *22.52* \| 60.43 \| **82.95** | *86.15* \| 55.66 \| **141.81** | *81.21* \| 60.91 \| **142.13** |
| 3 | Hsteps | 15 \| 12 \| **27** | 14 \| 0 \| **14** | 12 \| 0 \| **12** |
| | Nsteps | *42* \| 45 \| **87** | *52* \| 35 \| **87** | *44* \| 33 \| **77** |
| | Ntime | *81.18* \| 60.52 \| **141.7** | *650.94* \| 45.33 \| **696.27** | *560.82* \| 48.25 \| **609.08** |
| 4 | Hsteps | 15 \| 12 \| **27** | 13 \| 0 \| **13** | 14 \| 2 \| **16** |
| | Nsteps | *56* \| 41 \| **97** | *60* \| 34 \| **94** | *65* \| 39 \| **104** |
| | Ntime | *705.69* \| 60.44 \| **766.13** | *18587.18* \| 43.02 \| **18630.2** | *20994.39* \| 47.63 \| **21042.02** |

with $a = 1.3$, $b = 1/a$ and $f_{\text{clov}}$ as given in (56). We define the convex combination of the three corresponding cost functions $\mathcal{J}_{F,i}(\Omega) = \int_\Omega f_i(x)\,\mathrm{d}x$, $i = 1,2,3$, as $\mathcal{J}_{\text{conv}}[s_1,s_2](\Omega) = s_1\mathcal{J}_{F,1}(\Omega) + s_2\mathcal{J}_{F,2}(\Omega) + (1-s_1-s_2)\mathcal{J}_{F,3}(\Omega)$, for $s_1, s_2 \geq 0, s_1+s_2 \leq 1$. In order to trace along the Pareto front, which is a two-dimensional surface in $\mathbf{R}^3$, we define the three homotopies

$$\begin{aligned}
\tilde{\mathcal{H}}^{1,2}(\Omega,t,\delta) &:= (1-t)\mathcal{J}_{\text{conv}}[1-2\delta,\delta](\Omega) + t\mathcal{J}_{\text{conv}}[\delta,1-2\delta](\Omega)\\
\tilde{\mathcal{H}}^{2,3}(\Omega,t,\delta) &:= (1-t)\mathcal{J}_{\text{conv}}[\delta,1-2\delta](\Omega) + t\mathcal{J}_{\text{conv}}[\delta,\delta](\Omega)\\
\tilde{\mathcal{H}}^{3,1}(\Omega,t,\delta) &:= (1-t)\mathcal{J}_{\text{conv}}[\delta,\delta](\Omega) + t\mathcal{J}_{\text{conv}}[1-2\delta,\delta](\Omega)
\end{aligned}$$

which are parametrized by $t \in [0,1]$ and $\delta \in [0, \frac{1}{3}]$. Note that, for $\delta = 0$, the homotopy $\tilde{\mathcal{H}}^{i,j}$ is just the convex homotopy between $\mathcal{J}_{F,i}$ and $\mathcal{J}_{F,j}$.

In our numerical experiments, we chose $\delta \in \{0, 0.1, 0.2, 0.3\}$ and for each fixed $\delta$ followed the homotopies $\tilde{\mathcal{H}}^{1,2}$, $\tilde{\mathcal{H}}^{2,3}$ and $\tilde{\mathcal{H}}^{3,1}$ from $t = 0$ to $t = 1$. Since, here, all intermediate solutions of $\tilde{\mathcal{H}}^{i,j}(\Omega,t,\delta) = 0$ are of interest as they represent Pareto optimal shapes, we always use the full desired tolerance in the corrector stage, i.e., Remark 4.5 does not apply here. In order to solve $\tilde{\mathcal{H}}^{1,2}(\Omega,0,\delta) = 0$ for any given $\delta$, we used the homotopy defined by (38) with the simple problem (40) related to the given initial design. The solutions of $\tilde{\mathcal{H}}^{2,3}(\Omega,0,\delta) = 0$ and $\tilde{\mathcal{H}}^{3,1}(\Omega,0,\delta) = 0$ are trivial starting out from solutions of $\tilde{\mathcal{H}}^{1,2}(\Omega,1,\delta) = 0$ and $\tilde{\mathcal{H}}^{2,3}(\Omega,1,\delta) = 0$, respectively. We used second order predictors and the agile step size strategy of Section 2.2.2 with $\alpha = 0.1$. In addition, in order to promote an equispaced Pareto front, we imposed a maximum distance in objective space of any two subsequent points in a homotopy run $\tilde{\mathcal{H}}^{i,j}$, i.e., we rejected points even if the corrector was successful and continued with a reduced step size.

The resulting Pareto front is depicted in Figure 8. Here, the red, blue and green squares correspond to the homotopy runs with $\delta = 0$, i.e., they represent the Pareto fronts for the different bi-objective optimization problems (see also the two-dimensional projections in Figure 8). The points obtained by $\delta = 0.1$ are represented by the larger circles (again red, blue and green correspond to $\tilde{\mathcal{H}}^{1,2}$, $\tilde{\mathcal{H}}^{2,3}$ and $\tilde{\mathcal{H}}^{3,1}$, respectively), followed by those for $\delta = 0.2$ and $\delta = 0.3$ in the center. Black symbols denote the transitions between two different homotopies for a given value of $\delta$, i.e., they are solutions of both $\tilde{\mathcal{H}}^{i,i+1}(\Omega,1,\delta) = 0$ and $\tilde{\mathcal{H}}^{i+1,i+2}(\Omega,0,\delta) = 0$ (where indices are modulo 3). The black squares represent the minimizers of the three single-objective optimization problems. The shapes corresponding to these points as well as the solution to $\tilde{\mathcal{H}}^{1,2}(\Omega,2/3,0.2) = 0$ are shown in Figure 8. We again used a mesh consisting of 650 triangles and 384 vertices. The total computation time for obtaining the depicted Pareto front was about 35 minutes which underlines the efficiency of the method compared to conventional methods.

# 7 Conclusion and Outlook

In this work, we applied the concept of homotopy methods to (free-form) shape optimization problems without and with PDE constraints. We followed a predictor-corrector approach where the corrector stage is realized by an unregularized shape-Newton method. We discussed how to obtain arbitrary order predictors in shape optimization and investigated their effects numerically. Moreover, we discussed

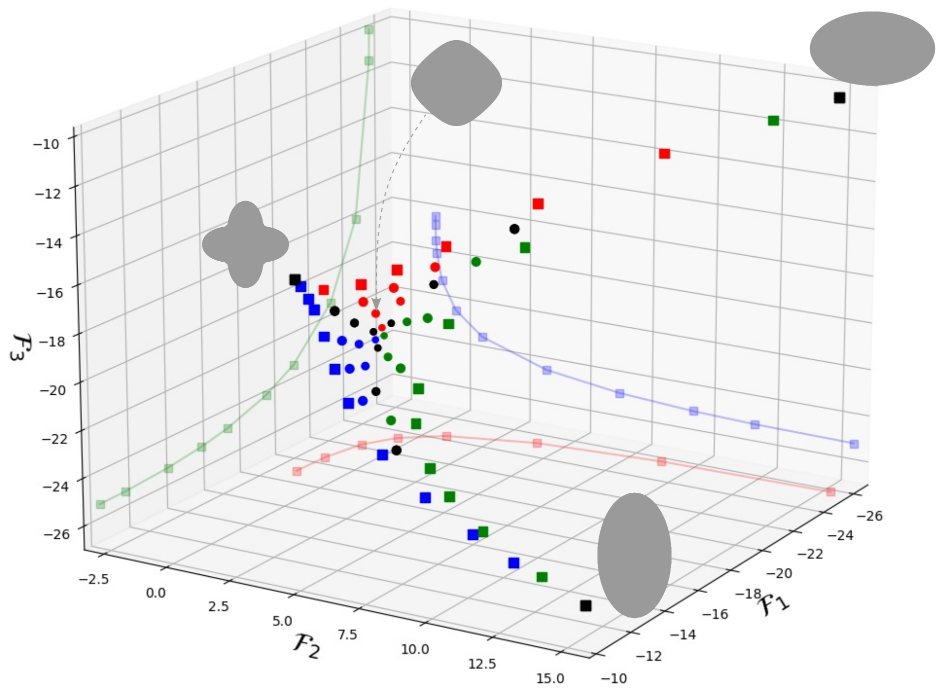


Figure 8: Pareto front for three-objective shape optimization problem obtained by Pareto method of Section 6.4. Black squares represent solutions to single-objective optimization problems; red, blue and green squares represent solutions to bi-objective optimization problems.

different strategies to automatically adapt the step sizes in the homotopy based on properties of the physical problem. We saw that homotopy methods allow for solving shape optimization problems to high accuracy in a Newton-type manner even with initial guesses that are far away from the optimal solution. The presented approach could be further extended in the following directions:

- The mentioned challenges concerning mesh quality (cf. Remark 6.2) could be circumvented when coupling the approach with a level set method and remeshing routines [1], thus also allowing for the solution of topology optimization problems in a Newton-type manner.

- For physical problems defined on more complicated geometries, the proposed simple PDE-constrained problem (45) may not be feasible. Thus, different types of homotopies such as the Keller model could be investigated.

- Often, mathematical models of physical problems involve parameters for which experience shows that gradually updating them is beneficial for the solution process. Examples include incremental loading in nonlinear elasticity or the degree of penalization of intermediate material densities in density-based topology optimization. When using a homotopy method, these parameters could either be defined as functions of the homotopy parameter, or could be treated as independent, additional homotopy parameters, amounting to a multi-dimensional homotopy method; see also [19, 20] for the case of multi-objective optimization. In combination with adaptive step size choices, updates of hard-to-tune parameters may be computed automatically in order to find the best path $\{(t(s), x(t(s))) : s \in [0,1], t(0) = 0 \in \mathbf{R}^p, t(1) = 1 \in \mathbf{R}^p\}$ from the initial design at

$t(0) = 0 \in \mathbf{R}^p$ to the solution of the problem at $t(1) = 1 \in \mathbf{R}^p$, $p > 1$.

- The mesh size of the discretized domain could be considered as an additional continuation parameter when coupled with a posteriori error estimators, meaning to start with a coarse mesh and gradually refining the mesh as one approaches the problem of interest; see e.g. [3] or [14] for related ideas.

- The concept of continuation methods allows the user to influence the path that is taken from the initial design to a (usually only locally) optimal design. This aspect can be exploited to systematically explore the design space and even detect different paths leading to different local solutions by means of deflation techniques [29].

- In the context of multi-objective shape optimiization, the concept of homotopy methods seems to be a promising tool in practice as it allows the designer to interactively steer the design into a desired direction, e.g. in order to fulfill certain manufacturability requirements.

- Finally, throughout this paper, we assumed the intermediate solutions to be parametrized by the homotopy parameter and that no turning points appear. While this assumption was fulfilled in the considered examples, it may not hold in general. Therefore, the presented approach could be extended to the case where the solution curve is parametrized by arc length [2] allowing for a wider class of problems to be treated.

## Acknowledgments

The work of A.C. and P.G. is supported by the FWF funded project P32911 as well as the joint DFG/FWF Collaborative Research Centre CREATOR (CRC – TRR361/F90) at TU Darmstadt, TU Graz, RICAM and JKU Linz. The work of B.E. is funded by a Humboldt Postdoctoral Fellowship and has been supported by the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453). The support is greatly acknowledged.

## References

[1] G. Allaire, C. Dapogny, and F. Jouve, *Chapter 1 - shape and topology optimization*, in Geometric Partial Differential Equations - Part II, A. Bonito and R. H. Nochetto, eds., vol. 22 of Handbook of Numerical Analysis, Elsevier, 2021, p. 1–132.

[2] E. L. Allgower and K. Georg, *Numerical continuation methods: an introduction*, vol. 13, Springer Science & Business Media, 2012.

[3] K. Bandara, T. Rüberg, and F. Cirak, *Shape optimisation with multiresolution subdivision surfaces and immersed finite elements*, Computer Methods in Applied Mechanics and Engineering, 300 (2016), p. 510–539.

[4] M. Bolten, O. T. Doganay, H. Gottschalk, and K. Klamroth, *Tracing locally Pareto-optimal points by numerical integration*, SIAM Journal on Control and Optimization, 59 (2021), pp. 3302–3328.

[5] A. Cesarano, B. Endtmayer, and P. Gangl, *Supplementary material to "Homotopy methods for higher order shape optimization: A globalized shape-Newton method and Pareto-front tracing*, 2024. https://zenodo.org/doi/10.5281/zenodo.11108761.

[6] A. Cesarano and P. Gangl, *Tracing pareto-optimal points for multi-objective shape optimization applied to electric machines*, 2024. https://arxiv.org/abs/2404.12205.

[7] K. Deckelnick, P. J. Herbert, and M. Hinze, *A novel $W^{1,\infty}$ approach to shape optimisation with Lipschitz domains*, ESAIM: COCV, 28 (2022), p. 2.

[8] M. C. Delfour and J. P. Zolésio, *Shapes and geometries*, Society for Industrial and Applied Mathematics, 2011.

[9] M. C. Delfour and J. P. Zolésio, *Anatomy of the shape Hessian*, Annali di Matematica Pura ed Applicata, 159 (1991), p. 315–339.

[10] P. Deuflhard, *Newton Methods for Nonlinear Problems*, vol. 35 of Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2011.

[11] D. M. Dunlavy and D. P. O'Leary, *Homotopy optimization methods for global optimization.*

[12] T. Etling and R. Herzog, *Optimum experimental design by shape optimization of specimens in linear elasticity*, SIAM Journal of Applied Mathematics, 78 (2018), pp. 1553–1576.

[13] T. Etling, R. Herzog, E. Loayza, and G. Wachsmuth, *First and second order shape optimization based on restricted mesh deformations*, SIAM Journal on Scientific Computing, 42 (2020), pp. A1200–A1225.

[14] A. Evgrafov, *State space Newton's method for topology optimization*, Computer Methods in Applied Mechanics and Engineering, 278 (2014), p. 272–290.

[15] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, and C. Dapogny, *Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework*, SeMA, 76 (2019), pp. 413–458.

[16] F. Feppon, G. Allaire, C. Dapogny, and P. Jolivet, *Body-fitted topology optimization of 2d and 3d fluid-to-fluid heat exchangers*, Computer Methods in Applied Mechanics and Engineering, 376 (2021), p. 113638.

[17] P. Gangl, U. Langer, A. Laurain, H. Meftahi, and K. Sturm, *Shape optimization of an electric motor subject to nonlinear magnetostatics*, SIAM Journal on Scientific Computing, 37 (2015), pp. B1002–B1025.

[18] P. Gangl, K. Sturm, M. Neunteufel, and J. Schöberl, *Fully and semi-automated shape differentiation in NGSolve*, Structural and Multidisciplinary Optimization, 63 (2020), p. 1579–1607.

[19] C. Hillermeier, *Generalized homotopy approach to multiobjective optimization*, Journal of Optimization Theory and Applications, 110 (2001), p. 557–583.

[20] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, Birkhäuser Basel, 2001.

[21] Z. Houta, F. Messine, and T. Huguet, *Topology Optimization for Magnetic Circuits with Adjoint Method in 3D.* working paper or preprint, May 2023.

[22] J. A. Iglesias, K. Sturm, and F. Wechsung, *Two-dimensional shape optimization with nearly conformal transformations*, SIAM Journal on Scientific Computing, 40 (2018), pp. A3807–A3830.

[23] V. A. Kovtunenko and K. Ohtsuka, *Inverse problem of shape identification from boundary measurement for Stokes equations: Shape differentiability of Lagrangian*, Journal of Inverse and Ill-posed Problems, 30 (2022), pp. 461–474.

[24] I. Malinen and J. Tanskanen, *Homotopy parameter bounding in increasing the robustness of homotopy continuation methods in multiplicity studies*, Computers & Chemical Engineering, 34 (2010), p. 1761–1774.

[25] A. Martin and O. Schütze, *Pareto tracer: a predictor–corrector method for multi-objective optimization problems*, Engineering Optimization, 50 (2018), p. 516–536.

[26] P. M. Müller, N. Kühl, M. Siebenborn, K. Deckelnick, M. Hinze, and T. Rung, *A novel p-harmonic descent approach applied to fluid dynamic shape optimization*, Structural and Multidisciplinary Optimization, 64 (2021), p. 3489–3503.

[27] A. Novruzi and M. Pierre, *Structure of shape derivatives*, Journal of Evolution Equations, 2 (2002), p. 365–382.

[28] S. Onyshkevych and M. Siebenborn, *Mesh quality preserving shape optimization using nonlinear extension operators*, Journal of Optimization Theory and Applications, 189 (2021), p. 291–316.

[29] I. P. A. Papadopoulos, P. E. Farrell, and T. M. Surowiec, *Computing multiple solutions of topology optimization problems*, SIAM Journal on Scientific Computing, 43 (2021), pp. A1555–A1582.

[30] J. Pinzon and M. Siebenborn, *Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners*, Optimization and Engineering, 24 (2022), p. 1089–1113.

[31] S. Schmidt, *Weak and strong form shape Hessians and their automatic generation*, SIAM Journal on Scientific Computing, 40 (2018), pp. C210–C233.

[32] S. Schmidt and V. Schulz, *Pareto-curve continuation in multi-objective optimization*, Pacific Journal of Optimization, 4 (2008), pp. 243–258.

[33] S. Schmidt and V. H. Schulz, *A linear view on shape optimization*, SIAM Journal on Control and Optimization, 61 (2023), pp. 2358–2378.

[34] S. Schmidt, E. Wadbro, and M. Berggren, *Large-scale three-dimensional acoustic horn optimization*, SIAM Journal on Scientific Computing, 38 (2016), pp. B917–B940.

[35] J. Schöberl, *C++11 implementation of finite elements in NGSolve*, Tech. Rep. 30, Institute for Analysis and Scientific Computing, Vienna University of Technology, 2014.

[36] V. H. Schulz, M. Siebenborn, and K. Welker, *Efficient PDE constrained shape optimization based on Steklov–Poincaré-type metrics*, SIAM Journal on Optimization, 26 (2016), p. 2800–2819.

[37] J. Simon, *Second variations for domain optimization problems*, in Control and estimation of distributed parameter systems (Vorau, 1988), vol. 91 of Internat. Ser. Numer. Math., Birkhäuser, Basel, 1989, p. 361–378.

[38] J. Sokołowski and J.-P. Zolésio, *Introduction to shape optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.

[39] K. Sturm, *Minimax Lagrangian approach to the differentiability of nonlinear PDE constrained shape functions without saddle point assumption*, SIAM Journal on Control and Optimization, 53 (2015), pp. 2017–2039.

[40] K. Sturm, *Convergence of Newton's method in shape optimisation via approximate normal functions*, 2018.

[41] E. Süli and D. F. Mayers, *An introduction to numerical analysis*, Cambridge university press, 2003.

[42] L. T. Watson and R. T. Haftka, *Modern homotopy methods in optimization*, Computer Methods in Applied Mechanics and Engineering, 74 (1989), pp. 289–305.

[43] W. Zulehner, *A simple homotopy method for determining all isolated solutions to polynomial systems*, Mathematics of Computation, 50 (1988), pp. 167–177.