

A Scalable Algorithm for Shape Optimization with Geometric Constraints in Banach Spaces

P. Marvin Müller², J. Pinzón¹, T. Rung², and M. Siebenborn¹

¹Hamburg University

²Hamburg University of Technology

May 5, 2022

Abstract

This work develops an algorithm for PDE-constrained shape optimization based on Lipschitz transformations. Building on previous work in this field, the p -Laplace operator is utilized to approximate a descent method for Lipschitz shapes. In particular, it is shown how geometric constraints are algorithmically incorporated avoiding penalty terms by assigning them to the subproblem of finding a suitable descent direction. A special focus is placed on the scalability of the proposed methods for large scale parallel computers via the application of multigrid solvers. The preservation of mesh quality under large deformations, where shape singularities have to be smoothed or generated within the optimization process, is also discussed. It is shown that the interaction of hierarchically refined grids and shape optimization can be realized by the choice of appropriate descent directions. The performance of the proposed methods is demonstrated for energy dissipation minimization in fluid dynamics applications.

Keywords: Shape optimization, Lipschitz transformations, p -Laplace, geometric multigrid, parallel computing

1 Introduction

In this paper we present a numerical scheme for the efficient treatment of geometrical constraints in shape optimization within the context of fluid dynamics applications. Several advantages over other well-known techniques are described, particularly how the need for penalty terms is relinquished in favor of a more robust approach. Additionally, we argue how the presented algorithm is well-suited for geometrical constraints of integral form, which are preserved up to a numerical tolerance during the optimization process.

Constraints on the volume and barycenter are often required in fluid dynamics. This is particularly true for the minimal drag problem of a free floating body, see [25, 18, 17, 20]. Another example is in [3], where a volume and perimeter constraints are considered for a structural optimization problem. A constraint for minimum and maximum thickness is formulated in [2, 8] in order to meet requirements stemming from the manufacturing process.

In order to preserve these constraints, we include them in the process of finding descent directions in Banach spaces. This is especially challenging, since the geometrical constraints are of a different type than that of the state equation, i.e. the governing partial differential equation (PDE). Meaning, that the flow field is characterized by the stationary, incompressible Navier-Stokes equations, which lead to a PDE constraint optimization problem over infinite dimensional Banach spaces. Whereas, the geometrical constraints are given by a finite number of integral type constraints, independently of a finite element model (see Section 2).

We focus on a well-established benchmark problem in the field of shape optimization constrained by Navier-Stokes equations, where the shape of an obstacle located within a flow channel is to be optimized with respect to the drag generated over its surface. In general, the optimization problem can be summarized as determining a geometry $\Omega \subset \mathbb{R}^d$ that minimizes a shape functional J . The functional represents a physical quantity, e.g. drag of an obstacle or the energy dissipation associated with the flow around an obstacle such that a fixed number of geometric constraints $g(\Omega) = 0 \in \mathbb{R}^m$ is also fulfilled. There are two major challenges in this problem, one concerning admissibility and optimality, and another

related to the regularity of the obstacle’s shape. In addition, the aforementioned geometrical constraints have to be preserved in order to avoid trivial and non-feasible solutions.

Many popular approaches rely on strategies to simultaneously update state variables and Lagrange multipliers of the constraints, for an overview see [4]. In other words, optimality and admissibility are established simultaneously within one iteration. However, the optimization problems addressed here tend to be non-meaningful or even unsolvable, provided that the constraints are not precisely fulfilled. For instance, if the barycenter of the obstacle is not fixed, then the object would leave the domain. In the same way the optimization procedure usually yields a trivial solution if the volume is not preserved, because then the obstacle would be contracted to a single point. The second issue is the regularity of initial and optimized shapes. On one hand, it might be essential that singularities can be represented in the optimal shapes, e.g. kinks and sharp edges. On the other, it is necessary for the scalability of the algorithm to apply multigrid methods as a grid-independent preconditioner for a Krylov subspace solver. In addition, the discretization of the domain with a coarse grid must be able to adequately represent both the surface of the obstacle and far field boundaries on the base level. Therefore during the optimization process the descent directions and shape updates have to feature non-smooth characteristics.

A common approach in shape optimization is to map a reference domain $\Omega \subset \mathbb{R}^d$ with $d = 2$ or $d = 3$, to a perturbed domain $(\text{id} + u)(\Omega) := \{x + u(x) \in \mathbb{R}^d : x \in \Omega\}$ with $u \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ such that $\text{id} + u$ is a Lipschitz homeomorphism, cf. [26, 7, 2]. Note that this does not require any parametrization of the geometry, e.g. like in a CAD description with NURBS-surfaces. For the shape deformation we follow [6, 13] and consider the steepest descent direction in $W^{1,\infty}$ -topology with a p -Laplace relaxation and the deformation vector field u then is the solution to a minimization problem. In contrast here we consider a constraint optimization problem in order to take the geometric constraints into account.

The well-known fluid dynamic example for a minimal drag problem in [22] considers the volume constraint, which is one dimensional. As the shape update is performed solely in the surface normal direction, the corresponding Lagrange multiplier is given by the mean value of the deformation. In [16], constraints as maximum thickness and volume have been taken into account via penalization of the cost function. Even though this approach allows also for more general shape deformations the shape update has to be rather small in order to keep the procedure numerically stable. Besides this one, two other approaches have successfully been applied to shape optimization problems of this kind. Firstly, an augmented Lagrange method can be used to determine the Lagrangian multipliers associated with the geometric constraints, [3, 25, 2, 17]. However, this approach has difficulties that are challenging to overcome, e.g. several parameters are problem dependent and have to be assigned to appropriate initial values. Furthermore, the constraints first have to be violated in order to determine the desired multipliers and the whole shape optimization problem has to be solved repetitively until the multipliers converge. This can lead to unfeasible shapes throughout the optimization procedure, as was previously mentioned. Secondly, the method of mappings [18, 11] enables for the fulfillment of the geometric constraints up to machine accuracy. Therefore, the shape optimization problem only has to be solved once and within each iteration only feasible geometries are computed. Here the state as well as the adjoint variables are determined on the transformed domain by applying the perturbation of identity. Thus, the descent vector field couples to all the constraints. Nevertheless, it is necessary to solve the fully coupled optimality system as a whole which is challenging, not only from an implementation point of view, but because it is computationally expensive.

In contrast, in this approach we use a second order method only to determine the steepest descent direction in a first order shape optimization. This reduces the dimensions of the linear systems to be solved. However, our optimization scheme requires that the geometric constraints only depend on the descent vector field u and not on the physical state variables, e.g. velocity or pressure. Therefore, the optimality system can be solved sequentially starting with the state, then the adjoints to the state, and finally the descent direction. This gives us the ability to handle problems with very large degrees of freedom (DoFs), while fulfilling the geometric constraints up to an arbitrary tolerance. It may be mentioned that the approach presented has analogies to optimization on manifolds and the investigations carried out in [19, 24], although it is based on Hilbert space settings. We also want to mention the Uzawa iteration [28] for solving a saddle point problem which occurs in the presented approach.

The remainder of this paper has the following structure: In section 2, the physical problem is introduced and the basics of shape optimization in Banach spaces are recalled. Section 3 proposes an algorithm which determines admissible shape deformation descent directions. In section 5, we demonstrate a scalable multigrid implementation for a fluid dynamics benchmark problem, while the performance of the method is investigated in section 5.3. In section 6 the presented algorithm and numerical experiments are recalled and summarized.

Regarding the notation in the upcoming equations $D(\cdot)$ denotes the Jacobian, for the spatial Euclidean gradient operator we use $\nabla(\cdot)$ and the directional derivatives with respect to a specific variable are indicated via, for instance $\frac{\partial}{\partial u}(\cdot)\delta_u$ in direction δ_u . The shape derivative of the functional $J(\Omega)$ in direction u is denoted by $J'(\Omega)u$ as defined in (7).

2 Model Equations

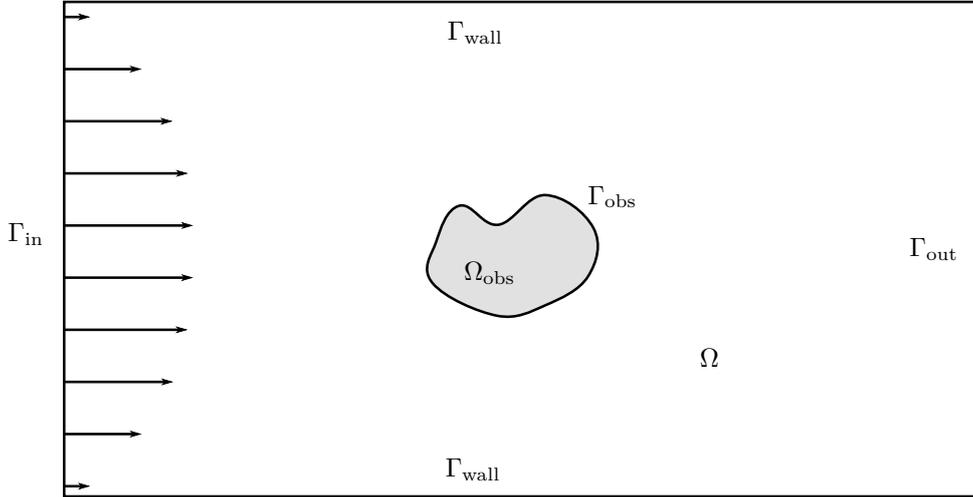


Figure 1: Schematic view on a flow tunnel-like domain Ω with the obstacle Ω_{obs} encircled by its surface Γ_{obs} , wall boundaries Γ_{wall} , inflow Γ_{in} , and outflow Γ_{out} . The height of the flow tunnel is defined by $\gamma > 0$.

In the present work, we propose an optimization methodology for PDE constraint shape optimization problems of the abstract form

$$\min_{\Omega \in \mathcal{S}} j(\Omega, y) \quad (1)$$

$$\text{s.t. } e(\Omega, y) = 0 \quad (2)$$

$$g(\Omega) = 0 \quad (3)$$

where e denotes the PDE constraint on a state variable y . The mapping g refers to some finite dimensional geometric constraints on the bounded Lipschitz domain Ω with boundary $\Gamma = \Gamma_{\text{obs}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}$, where Γ_{obs} is to be optimized. Furthermore, \mathcal{S} denotes an abstract set of admissible shapes, as explained for instance in [26, 7].

For some $m \in \mathbb{N}$ the geometric constraint is thus given as $g : \mathcal{S} \rightarrow \mathbb{R}^m$. We assume the existence of the mapping $\Omega \mapsto y(\Omega)$. Thereby, we obtain the reduced cost functional $J(\Omega) := j(\Omega, y(\Omega))$. In order to obtain sensitivities of the objective J we follow [26, 7, 3, 5]. For this purpose, the domain Ω is parameterized in the sense of the perturbation of identity with the displacement field $u : \mathbb{R}^d \rightarrow \mathbb{R}^d$. For a sufficiently small u we thus obtain deformed configurations

$$\tilde{\Omega} := \{x + u : x \in \Omega\} \quad (4)$$

of the reference shape Ω . For the sake of readability we abbreviate the perturbation of identity as

$$F : \Omega \rightarrow \tilde{\Omega}; \quad F := \text{id} + u. \quad (5)$$

Moreover, the previously mentioned abstract set of admissible shapes \mathcal{S} can be further specified as

$$\mathcal{S} := \{F(\Omega) : F = \text{id} + u, u \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)\} \quad (6)$$

As parts of the boundary of Ω shall remain fixed, the displacement field u is chosen in such a way that it vanishes on all boundaries of Ω which are not to be optimized. The directional shape derivative of J evaluated in Ω in the direction u is then given in a natural way by

$$J(\tilde{\Omega}) = J(\Omega) + J'(\Omega)u + o(\|u\|) \quad \text{where} \quad \frac{o(u)}{\|u\|} \xrightarrow{\|u\| \rightarrow 0} 0. \quad (7)$$

Then we can interpret the shape optimization problem (1) to (3) locally, as a problem in $U_{ad} \subseteq W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$. In the present work, U_{ad} is the admissible set of displacements u defining the transformation F , which inherently fulfills the geometric constraints (3) in the sense that $g(F(\Omega)) = 0$. Note that $U_{ad} \neq \emptyset$ as $u = 0$ is an admissible transformation if the initial geometry Ω fulfills the geometrical constraints, i.e. $g(\Omega) = 0$. The crucial aspect of the present method, is to separate the geometric constraints (3) from the remaining PDE-constrained shape optimization problem (1) and (2), and move it to the admissible set U_{ad} of descent directions. In contrast to other popular approaches, where admissibility is only guaranteed in the optimal configuration, we ensure that (3) is fulfilled in each optimization step. This can be done, because the geometric constraints do not depend on the state y and rely on the properties of the shape only.

For the computation of the shape derivative $J'(\Omega)u$, we utilize formally the method of C ea, see for instance [2, section 4.6]. In the following, we consider the particular problem of minimizing energy dissipation of the fluid flow mainly caused by an obstacle in a laminar, stationary flow, where the function

$$J(\Omega) := j(\Omega, v) = \frac{\nu}{2} \int_{\Omega} Dv : Dv \, dx \quad (8)$$

is to be minimized. Here and in the following, we denote velocity v , the density-specific pressure \mathbf{p} , viscosity ν , and an inflow velocity v_{∞} . As the PDE constraint e , with state variable $y = (v, \mathbf{p})$, we consider the stationary, incompressible Navier-Stokes equations

$$\begin{aligned} -\nu \Delta v + (v \cdot \nabla)v + \nabla \mathbf{p} &= 0 \text{ in } \Omega \\ \operatorname{div} v &= 0 \text{ in } \Omega \\ v &= 0 \text{ on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \\ v &= v_{\infty} \text{ on } \Gamma_{\text{in}} \\ Dv \cdot n &= \mathbf{p}n \text{ on } \Gamma_{\text{out}}. \end{aligned} \quad (9)$$

Here, we consider the adjoint approach for determining the directional shape derivative $J'(\Omega)u$. For details on the adjoint Navier-Stokes equations, see e.g. [12, 27, 18, 20]. For details on the shape derivative $J'(\Omega)u$ of the objective function in (8), see e.g. [15, 18]. In order to approximate the steepest descent direction u in

$$V_0^{\infty} := \{u \in W^{1,\infty}(\Omega \cup \Omega_{\text{obs}}, \mathbb{R}^d) : \|u\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1, u = 0 \text{ a.e. } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}\}, \quad (10)$$

following [2, Proposition 4.1], we introduce a p -Laplace relaxation with $p > 2$ with the corresponding minimization problem

$$\begin{aligned} \min_{u \in V_0^{\infty}} \quad & J'(\Omega)u \\ \text{s.t.} \quad & g(F(\Omega)) = 0, \\ & F = \text{id} + u, \end{aligned} \quad (11)$$

inspired by [13] and [6]. This makes F a Lipschitz transformation, not only on the obstacle's surface Γ_{obs} , but also on the entire domain Ω . Thus, in a discretization the elements undergo deformations that preserve mesh quality, as illustrated later in section 5. Therefore, let

$$V_0^p = \{u \in W^{1,p}(\Omega, \mathbb{R}^d) : \|Du\|_{L^p(\Omega, \mathbb{R}^d)} \leq 1, u = 0 \text{ a.e. on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}\} \quad (12)$$

and consider

$$\begin{aligned} \min_{u \in V_0^p} \quad & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} \, dx + J'(\Omega)u \\ \text{s.t.} \quad & g(F(\Omega)) = 0, \\ & F = \text{id} + u \end{aligned} \quad (13)$$

where it is assumed that $g(F(\Omega)) : V_0^p \rightarrow \mathbb{R}^m$, $u \mapsto g((\text{id} + u)(\Omega))$, $m \geq 1$. Notice that this is consistent with (1) to (3) for g over a fixed Ω and a variable displacement field u . In order to address the inconsistency between (10) and (12), we additionally assume that all displacements $u \in V_0^p$ are sufficiently small. Thus, the admissible set \mathcal{S} is locally parameterized by V_0^p -deformations of Ω . In the present work, $m = d + 1$ refers to the barycenter and volume constraints

$$\int_{\Omega} (x + u) \det(DF) \, dx = 0, \quad (14)$$

$$\int_{\Omega} \det(DF) - 1 \, dx = 0, \quad (15)$$

and without loss of generality, we assume that the barycenter of the initial domain Ω is located at the origin $0 \in \mathbb{R}^d$ of the domain, cf. [20]. For computational reasons, the mathematical domain is restricted to the wetted domain, i.e. Ω without the shape Ω_{obs} itself. Furthermore, due to the constant volume constraint (15), we can omit in (14) the division by the reference volume and deformed domain Ω and $F(\Omega)$, respectively. For the derivation of the optimality conditions of the steepest descent problem (13), we define the Lagrangian function

$$\begin{aligned} L(u, \lambda) = & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} dx + J'(\Omega) u \\ & + \sum_{i=1}^d \lambda_i \int_{\Omega} (x_i + u_i) \det(DF) dx + \lambda_{d+1} \int_{\Omega} \det(DF) - 1 dx \end{aligned} \quad (16)$$

with $\lambda = (\lambda_1, \dots, \lambda_d, \lambda_{d+1})^T$, where $\lambda_1, \dots, \lambda_d$ are associated with the barycenter (14) and λ_{d+1} with the volume constraint (15). In the following we want to recall some rules of differentiation. Therefore, let $\delta_u, \mu_u : \Omega \rightarrow \mathbb{R}^d$, $B : \Omega \rightarrow \mathbb{R}^{d \times d}$ and

$$DF = \left(\frac{\partial}{\partial x_j} F_i \right)_{1 \leq i, j \leq d} = I + Du \quad (17)$$

the Jacobian of F . We specify the following useful formulae by applying the product and chain rule:

$$\begin{aligned} \frac{\partial}{\partial u} DF \delta_u &= D\delta_u, \\ \frac{\partial}{\partial u} \det(DF) \delta_u &= \text{tr}((DF)^{-1} D\delta_u) \det(DF), \\ \frac{\partial}{\partial u} (\text{tr}(DF B)) \delta_u &= B^T : \left(\frac{d}{du} DF \delta_u \right) = B^T : D\delta_u, \\ \frac{\partial}{\partial u} ((DF)^{-1}) \delta_u &= -(DF)^{-1} D\delta_u (DF)^{-1}, \\ \frac{\partial}{\partial u} (\text{tr}((DF)^{-1} D\delta_u)) \mu_u &= -D\delta_u^T : (DF)^{-1} D\mu_u (DF)^{-1} \\ &= \text{tr}(-D\delta_u (DF)^{-1} D\mu_u (DF)^{-1}) \\ &= \text{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u). \end{aligned} \quad (18)$$

By making use of the rules above, we obtain the derivatives of the Lagrangian (16) with respect to u in the direction $\mu_u \in V_0^p$:

$$\begin{aligned} \frac{\partial}{\partial u} L(u, \lambda) \mu_u &= \int_{\Omega} (Du : Du)^{\frac{p-2}{2}} (Du : D\mu_u) dx + J'(\Omega) \mu_u \\ &+ (\lambda_1, \dots, \lambda_d)^T \cdot \int_{\Omega} \mu_u \det(DF) + (x + u) \text{tr}((DF)^{-1} D\mu_u) \det(DF) dx \\ &+ \lambda_{d+1} \int_{\Omega} \text{tr}((DF)^{-1} D\mu_u) \det(DF) dx. \end{aligned} \quad (19)$$

Together with the usual derivative with respect to λ into direction $\mu_\lambda \in \mathbb{R}^{d+1}$, the optimality system reads

$$\begin{aligned} \frac{\partial}{\partial u} L(u, \lambda) \mu_u &= 0 \quad \forall \mu_u \in V_0^p \\ \frac{\partial}{\partial \lambda} L(u, \lambda) \mu_\lambda &= 0 \quad \forall \mu_\lambda \in \mathbb{R}^{d+1}. \end{aligned} \quad (20)$$

As the derivatives with respect to λ can directly be taken from (16), we omit the details here. In order to solve the nonlinear system (20) we require the linearization

$$\frac{\partial^2}{\partial u^2} L(u^k, \lambda^k)(\mu_u, \delta_u) + \frac{\partial}{\partial \lambda \partial u} L(u^k, \lambda^k)(\mu_u, \delta_\lambda) = -\frac{\partial}{\partial u} L(u^k, \lambda^k) \mu_u \quad \forall \mu_u \in V_0^p \quad (21)$$

$$\frac{\partial}{\partial u \partial \lambda} L(u^k, \lambda^k)(\delta_u, \mu_\lambda) = -\frac{\partial}{\partial \lambda} L(u^k, \lambda^k) \mu_\lambda \quad \forall \mu_\lambda \in \mathbb{R}^m \quad (22)$$

and the updates

$$u^{k+1} = u^k + \delta_u, \quad \lambda^{k+1} = \lambda^k + \delta_\lambda \quad (23)$$

where

$$\begin{aligned} \frac{\partial^2}{\partial u^2} L(u, \lambda)(\delta_u, \mu_u) = & \int_{\Omega} (p-2)(Du : Du)^{\frac{p-4}{2}} (Du : D\delta_u)(Du : D\mu_u) + (Du : Du)^{\frac{p-2}{2}} (D\delta_u : D\mu_u) dx \\ & + (\lambda_1, \dots, \lambda_d)^T \cdot \int_{\Omega} \left(\delta_u \operatorname{tr}((DF)^{-1} D\mu_u) + \mu_u \operatorname{tr}((DF)^{-1} D\delta_u) \right. \\ & \left. + (x+u) \left(\operatorname{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u) + \operatorname{tr}((DF)^{-1} D\delta_u) \operatorname{tr}((DF)^{-1} D\mu_u) \right) \right) \det(DF) dx \\ & + \lambda_{d+1} \int_{\Omega} \left(\operatorname{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u) + \operatorname{tr}((DF)^{-1} D\delta_u) \operatorname{tr}((DF)^{-1} D\mu_u) \right) \det(DF) dx. \end{aligned} \quad (24)$$

Reviewing the first integral in (24), one observes that these terms do not exist for $p < 4$, where $Du : Du = 0$ holds on a set of non-zero measure. However, this issue does not appear in the defect equation (19), since there all exponents are non-negative. We thus modify the first integral in (24) to

$$\begin{aligned} \int_{\Omega} (p-2)(Du : Du + \epsilon \Theta(4-p))^{\frac{p-4}{2}} (D\mu_u : Du)(D\delta_u : Du) \\ + (Du : Du + \epsilon)^{\frac{p-2}{2}} (D\mu_u : D\delta_u) dx \end{aligned} \quad (25)$$

where Θ denotes the Heaviside function and $\epsilon > 0$ a sufficiently small constant. Notice that, within Newton's method in (21) and (22), this modification only affects the linearization and not the defect. Thus, solutions of the original problem (13) are still obtained upon convergence. Adding ϵ in (25) serves two purposes. On the one hand guarantees invertibility and on the other it prevents divide-by-zero operations in the first term.

3 Optimization Algorithm

In this section we describe an algorithm for the solution of (13). By the restriction of descent directions to maintain $g(F(\Omega)) = 0$, it is guaranteed that the geometric constraints are fulfilled up to a given tolerance at each iteration of the optimization process and not only on the limit. The geometric constraints considered here, i.e. barycenter and volume of a free floating obstacle are particularly challenging to handle. In an augmented Lagrangian or even pure penalty approach, the violation of $g = 0$ in one iteration might lead to a strong overshoot of the shape deformation. This causes oscillation of the shape because the geometry is unfeasible in each iteration. For example, in particular at low Reynolds number flows, a major influence to the minimization of the energy dissipation is associated with the displacement of the flow by the obstacle. Also minimizing the volume minimizes the energy dissipation. At higher Reynolds number flows a descent direction is to move the obstacle downstream. From a practical point of view this can only be solved by carefully adjusting initial values of the multipliers λ , the penalty factors, and the penalty increment values. Thus, the practical attractiveness of the approach outlined here is that there are less heuristic and problem-dependent quantities to be adjusted. The user only has to provide the convergence criteria, the parameters of the step size control, and the values corresponding to the sequence of p , i.e. p_{\max} and p_{inc} .

From a mathematical point of view, the computational price one has to pay is the following: The set of admissible descent directions is not convex anymore, but the solution manifold of the nonlinear equation $g(F(\Omega)) = 0$. For example, having computed an admissible step u_p does not imply that $\frac{1}{2}u_p$ is also admissible. This makes a step size control expensive, since the geodesics on the solution manifold are not straight lines in this case. In algorithm 1 the step size control is thus handled by scaling the shape sensitivity $J'(\Omega)$ with a decreasing sequence $\sigma = (1, \frac{1}{2}, \frac{1}{4}, \dots)$.

As a note, for numerical reasons it might prove profitable to multiply equation (20) with $1/\sigma$. For

Algorithm 1 Shape Optimization Steepest Descent Method

Require: Ω, p_{\max}

- 1: $y \leftarrow$ Solve primal problem
- 2: $y_0 \leftarrow y$
- 3: Compute objective $\Phi_0 = J(\Omega)$
- 4: **repeat**
- 5: $y^* \leftarrow$ Solve adj. problem
- 6: $\sigma \leftarrow 1$
- 7: **while** True **do**
- 8: $p \leftarrow 2$
- 9: $\bar{u} \leftarrow 0$
- 10: **while** $p \leq p_{\max}$ **do**
- 11: $(u_p, \lambda) \leftarrow$ NEWTONSOLVER(\bar{u}, σ, y, y^*)
- 12: $\bar{u} \leftarrow u_p$
- 13: Increase p
- 14: **end while**
- 15: Update geometry Ω with $u_{p_{\max}}$
- 16: $y \leftarrow$ Solve primal problem
- 17: Compute objective $\Phi = J(\Omega)$
- 18: **if** $\Phi \geq \Phi_0$ **then**
- 19: Update geometry Ω with $-u_{p_{\max}}$
- 20: $\sigma \leftarrow \sigma/2$
- 21: $y \leftarrow y_0$
- 22: **else**
- 23: $\Phi_0 \leftarrow \Phi$
- 24: $y_0 \leftarrow y$
- 25: **break**
- 26: **end if**
- 27: **end while**
- 28: **until** $\|u_{p_{\max}}\|_{W^{1,p}(\Omega)} < \epsilon_1$

Algorithm 2 Newton's Method for p -Laplacian Problem

- 1: **function** NEWTONSOLVER(u_p, σ, y, y^*)
- 2: $\lambda \leftarrow 0$
- 3: **repeat**
- 4: $(A, B, r_u, r_\lambda) \leftarrow$ Assemble(u, λ, y, y^*) according to (26)
- 5: $(\delta_{u_p}, \delta_\lambda) \leftarrow$ SCHURSOLVER($A, B, r_u, r_\lambda, \delta_{u_p}, \delta_\lambda$)
- 6: $u_p \leftarrow u_p + \delta_{u_p}$
- 7: $\lambda \leftarrow \lambda + \delta_\lambda$
- 8: **until** $\|\delta_{u_p}\|_{W^{1,p}(\Omega)} + \|\delta_\lambda\|_2 < \epsilon_2$
- 9: **return** (u_p, λ)
- 10: **end function**

Algorithm 3 Schur Complement Product

- 1: **function** SCHURCOMPLEMENTPRODUCT(A, B, w)
- 2: **for** $i = 1, \dots, m$ **do**
- 3: $b \leftarrow b + B(:, i)w_i$
- 4: **end for**
- 5: Solve $Az = b$
- 6: **for** $i = 1, \dots, m$ **do**
- 7: $b_i \leftarrow -B(:, i)^T z$
- 8: **end for**
- 9: **return** b
- 10: **end function**

the sake of readability we abbreviate the linearized optimality system (21) and (22) using the symbols

$$\begin{aligned}
A\delta_u &:= \frac{\partial^2}{\partial u^2} L(u^k, \lambda^k)(\mu_u, \delta_u) & \forall \mu_u \in V_{0,h}^p \\
B\delta_\lambda &:= \frac{\partial}{\partial u} \frac{\partial}{\partial \lambda} L(u^k, \lambda^k)(\mu_u, \delta_\lambda) & \forall \mu_u \in V_{0,h}^p \\
B^T \delta_u &:= \frac{\partial}{\partial \lambda} \frac{\partial}{\partial u} L(u^k, \lambda^k)(\delta_u, \mu_\lambda) & \forall \mu_\lambda \in \mathbb{R}^d \\
r_u &:= -\frac{\partial}{\partial u} L(u^k, \lambda^k) \mu_u & \forall \mu_u \in V_{0,h}^p \\
r_\lambda &:= -\frac{\partial}{\partial \lambda} L(u^k, \lambda^k) \mu_\lambda & \forall \mu_\lambda \in \mathbb{R}^d.
\end{aligned} \tag{26}$$

With $V_{0,h}^p$ a discrete approximation of V_0^p used for a finite element discretization of (21) and (22), which then leads to the saddle point problem.

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_u \\ r_\lambda \end{pmatrix} \tag{27}$$

where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. In order to solve for the increments δ_u and δ_λ , we formally apply one block wise Gauss elimination and obtain

$$\begin{pmatrix} A & B \\ 0 & -B^T A^{-1} B \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_u \\ r_\lambda - B^T A^{-1} r_u \end{pmatrix} \tag{28}$$

where $S := -B^T A^{-1} B$ is the so-called Schur complement operator. In order not to explicitly compute A^{-1} a equation system with A is solved instead. In general, the optimality system (20) of problem (1) to (3) is highly nonlinear. Especially with increasing values of p the solution process becomes more challenging unless a good initial guess u_p^0 is provided. To overcome this issue, and to reduce computational effort, we consider a finite sequence $p_k := p_{\text{init}} + k p_{\text{inc}}$ where $p_{\text{init}} := 2$. First, the solution for p_{init} with initial $u_{p_{\text{init}}} = 0$ and $\lambda = 0$ is computed. Thereafter the solution of the constraint p_k -Laplacian problem (13) is used as an initial guess for the p_{k+1} -Laplacian problem, cf. [17]. Here, with the choice of p_{max} we adjust the approximation quality of Lipschitz deformations.

The overall optimization procedure is outlined in algorithm 1. The steepest descent method is reflected in the loop spanning from lines 4 to 29, where the necessary optimality condition is checked. Here, y again denotes the state variable of the PDE constraint e , which we refer to as the primal problem. Nested within this loop, a step-size control operates in the lines 8 to 28. It checks whether the proposed next shape $F(\Omega)$ leads to an improvement of the objective function, in terms of the displacement field u_p . If not, then the parameter σ is reduced. Note that, in contrast to the classical backtracking line search in linear spaces, we have to recompute the descent direction u_p hereafter. This is due to the fact that by shortening the step-length, we can not follow straight lines towards $0 \in V_0^p$, but have to stay within the solution manifold of the non-linear geometric constraints $g(F(\Omega)) = 0$. In line 5 the adjoint PDE is solved, which yields the adjoint state y^* . After this, the shape sensitivity $J'(\Omega)$ can be evaluated in line 6, which depends on y and y^* .

In line 11 of algorithm 1, the nonlinear solver for the steepest descent problem (13) is called, which can be seen in algorithm 2. The key part of this solver is the solution to the saddle point problem (27) in the Schur complement form (28). This could be realized with a variety of iterative solvers, which are not further specified here. Popular approaches for these kind of problems are the Uzawa iteration and the Arrow-Hurwicz algorithm. For this purpose, algorithm 3 outlines the computational steps for a matrix-vector product with S .

4 Numerical Methodology

Results of the present study are obtained from the open-source toolbox UG4 [29]. This simulation framework has MPI-based parallelization, and features a geometrical multigrid preconditioner [10]. The grid partitioning and load balancing scheme is based on ParMetis [14].

Stable $P_2 - P_1$ finite elements were used to discretize the governing nonlinear Navier-Stokes equations (9) and their linearization, therefore no additional stabilization is required. Moreover, the viscosity is $\nu = 0.02$ in all cases. The same setting was used to discretize the linear adjoint problem, cf. [20, 18] and

the references therein. As regards the p-Laplace relaxation problem, for which the optimality system is described in (28), P_1 Lagrange shape functions were employed. Computational grids consist of triangular (2d) and tetrahedral elements (3d). They were generated using GMSH [9].

The simulations followed the workflow proposed in algorithm 1. At the beginning of each optimization step, the steady, incompressible flow, described by Navier-Stokes equations (9), was computed followed by the solution of the corresponding adjoint system. The p-Laplacian descent algorithm algorithm 1 initially employed $p_{\text{init}} = 2.0$ and incremented p by $p_{\text{inc}} = 0.19$. The given maximum values of p read $p_{\text{max}} = 4.8$ [$p_{\text{max}} = 4.1$] for the computed 2d [3d] test cases. Termination criteria of algorithm 1 and algorithm 2 were always set as $\epsilon_1 = 1\text{E}-5$ and $\epsilon_2 = 1\text{E}-8$, respectively. The modification term introduced in (25) reads $\epsilon = 1\text{E}-8$ for all cases.

As a practical note, care must be taken to correctly interpolate the values of v , \mathbf{p} , and their respective adjoints. These are involved in the assembly of J' , which is present in (28). The geometrical constraints are part of this system of equations. It has been described that they lead to an $m \times m$ system of equations, so their discretization is not within a finite element space but in \mathbb{R}^m . For the investigated case cases, $m \in \{3, 4\}$ in 2d and 3d, respectively. Thus, we use a direct solver to find the solution of the Schur complement system.

The corresponding codes used for these results can be found in the online repository [21].

5 Results

This section presents results for 2d and 3d fluid dynamics applications. They either refer to an initial square (2d) or cube (3d) centrally placed in a rectangular flow domain at low Reynolds number, i.e. $Re = 1 \cdot H/\nu = 20$, where H refers to the length of the initial edges. The employed box-domain is outlined in Fig. 1. It spans 20 units in length and $\gamma = 6$ units in height (2d, 3d) and depth (3d), respectively, and the flow enters the domain through the left vertical boundary. The inflow profile on Γ_{in} features a peak unit-value in the center of the inlet plane and is described by

$$v_{\infty} = \left(\max \left\{ 0, \prod_{i=2}^d \cos\left(\frac{\pi|x_i|}{\delta}\right) \right\}, 0, \dots, 0 \right) \in \mathbb{R}^d$$

where δ corresponds to the inlet height.

The central aspect of the paper is the creation and removal of geometrical singularities. Emphasis is placed on illustrating and explaining how the corners of the obstacle are removed during the optimization process, as well as how tips are generated to reach an optimal shape. A crucial aspect is the evolution of the mesh quality during an optimization. We utilize the 2d studies, to compare the mesh quality of the optimal and the initial design by means of the ratio ρ between the radii of circumcircle and incircle, and report the extreme values of the interior angles of the triangulation. Moreover, we describe the behavior of the proposed algorithm in two different 3d configurations, where the surface of the obstacle is highly resolved.

Mind that the geometrical constraints are preserved during each optimization step for every value of p , since they are incorporated to the system of equations. Their fulfillment is included in the convergence condition set for Newton's method in algorithm 2, therefore there is no need to provide results for their fulfillment per step. Solving the nonlinear system of (20) implies solving the geometrical constraints (14) and (15) to the error reduction tolerance set for Newton's method. The major portion of the computational effort in algorithm 1 is spent on solving the p -Laplace relaxed problem via the scheme described in section 3. Particularly, lines 11-15 of algorithm 1 are computationally expensive, as will be explained here and in section 5.3.

5.1 Two-dimensional studies

Simulations in the 2d domain were performed for several levels of refinement, to better describe the removal of the geometrical singularities, as well as the mesh quality. Figure 2 compares the initial design (red) with the converged design (blue), together with a contour plot of a deformation sequence (gray). A robust removal of the box corners is clearly visible, as well as the creation of the tips in the rear and the aft sections. As described in section 2, the geometrical constraints are preserved in all optimization steps. This feature can be observed by the continuous transition between shape iterates until an optimum is obtained. In contrast, in [20] bouncing of the shapes during the early stages of the optimization is reported, which is related to an approximate solution of the geometrical constraints.

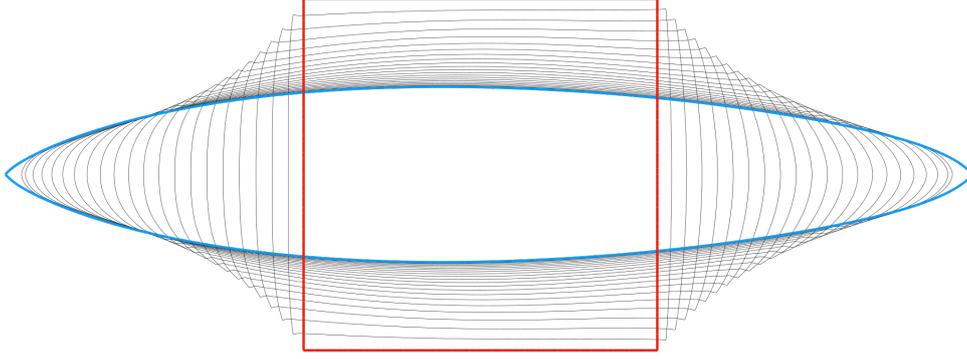


Figure 2: Superposition of the deformation sequence for a 2d configuration. The obstacle's initial shape **red** is presented superimposed to the sequence of generated shapes **gray** until an optimal shape **blue** is obtained upon convergence.

Figure 3 magnifies, the geometry and the mesh in the upper-left corner of the obstacle. The initial and final shapes are presented on the top and bottom, respectively, for different grid refinement levels from left to right. The figure displays that the smoothing occurs similarly on all grids, and the elements around the initial singularity are not dramatically degenerated during the optimization. Towards the last step, no indication of the initial geometric singularity is visible on the obstacle's surface.

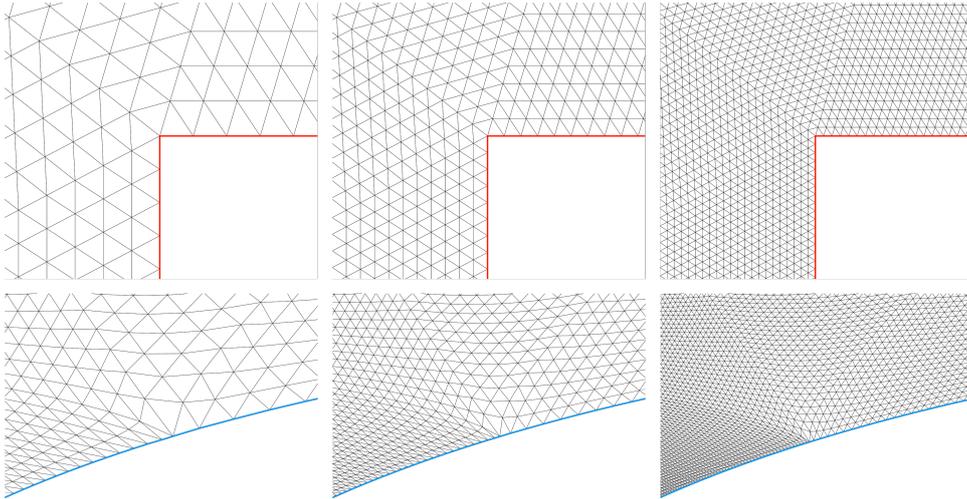


Figure 3: Removal of the geometrical singularity in the obstacle's initial configuration across several levels of refinement. For 4, 5, and 6 refinements the upper left corner of the box is smoothed via updating the geometry Ω iteratively, as stated in line 16 in algorithm 1.

The mesh quality is investigated for the final step using 4, 5, and 6 levels of refinement. As outlined in section 3, a series of shape iterates are obtained until an optimum, with respect to (8), is found. Mind that the geometric multigrid preconditioner, which is used to allow for numerical scalability, requires the generation of a grid hierarchy, of which we provide the base level, i.e. the coarsest mesh. This implies that the simulations are based upon a predetermined mesh quality, and while the optimization we propose in algorithm 1 aims at preserving grid quality, it doesn't contemplate improving it with respect to the initial geometry. Table 1 provides quality measurements for the final step, when the optimal shape is found, using several grid refinement levels. Assessed data refers to the worst triangular elements extracted from the 2d grid, i.e. the observed minimum and maximum interior angles, and the largest radius ratio. We also compare the radius ratio between the last and first configurations. The value of $\rho_0 = 1.468$ indicates that the initial mesh does not have an ideal quality. Results also demonstrate that, if p is high enough, the approximation of Lipschitz transformations, as seen in (10) and (12), prevents a significant loss of mesh quality over mesh refinements. For the presented 2d cases, a value of $p = 4.8$ yielded a sufficient approximation to $p = \infty$ in terms of the mesh quality, while allowing for the creation and removal of geometrical singularities. The mesh refinement study might reveal that a higher

Refinements	Elements	Minimum angle	Maximum angle	Radius ratio ρ_∞	$\frac{\rho_\infty}{\rho_0}$
4	70 656	13.41	132.32	3.20	2.18
5	282 624	11.93	139.03	4.24	2.89
6	1 130 496	9.94	145.04	5.76	3.92

Table 1: Assessment of mesh quality evolution for several refinement levels observed in 2d; Displayed data for minimum and maximum interior angles supplemented by the largest radius ratio of the triangulation extracted for the last optimization step, where an optimal shape is reached. The last column compares the largest radius ratio of the optimal shape (subscript ∞) and the initial configuration (subscript 0, $\rho_0 = 1.468$).

maximum p-values are necessary for the finer grids, since the quality slightly deteriorates. Nevertheless, numerical stability must be taken into account when increasing this value, given that it is used in (24) as an exponent. The latter fact turned out to be a limiting factor in our numerical simulations. However, the measurement of the worst minimum and maximum angles express that the triangles, which have undergone the largest deformation, are still not close to being critical.

5.2 Three-dimensional studies

Results for the 3d simulations refer to 4 levels of grid refinement. The computational grid has a total of 4 980 736 tetrahedrons, and 49 152 triangles discretize the surface of the obstacle, Γ_{obs} , on the highest refinement level. Our optimization scheme generates a series of deformation fields u_p that, applied to the domain Ω , results in an optimal shape with respect to the energy dissipation (8).

Figure 4 presents iterated shapes from the initial to the final optimization step. It shows the downstream part of the geometry. For the reference shape the streamlines visualize a region where the flow direction points backwards w.r.t. the main flow direction. Since this effect contributes to the energy dissipation it vanishes during the optimization at an early stage. This phenomenon can be quantified by observing the shear stress acting on the surface of the obstacle $\tau \cdot e_1 = (v(Dv + Dv^T) \cdot n) \cdot e_1$. Here e_1 is the first unit vector describing the main flow direction. For the 3d (cf. section 5.2) case $\tau \cdot e_1 \in [-11.06, 1.27]$ for the initial shape and $\tau \cdot e_1 \in [-20.53, -1.12]$ for the final one, respectively.

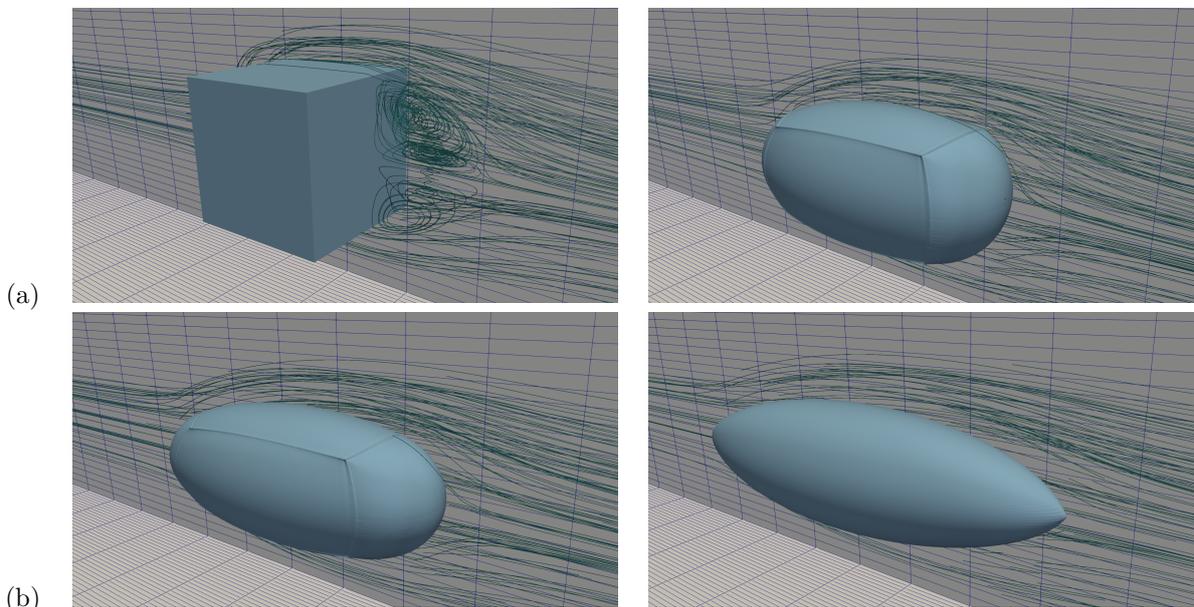


Figure 4: Streamlines for the rear of the obstacle located in the wind tunnel.

Similar to the 2d case, the edges and corners displayed by the initial geometry are gradually removed as part of the optimization process, cf. figure 5. Additionally, the tips created at the central upstream and downstream ends shapes a streamlined body that does not feature any separation, as shown in row (b) of ??.

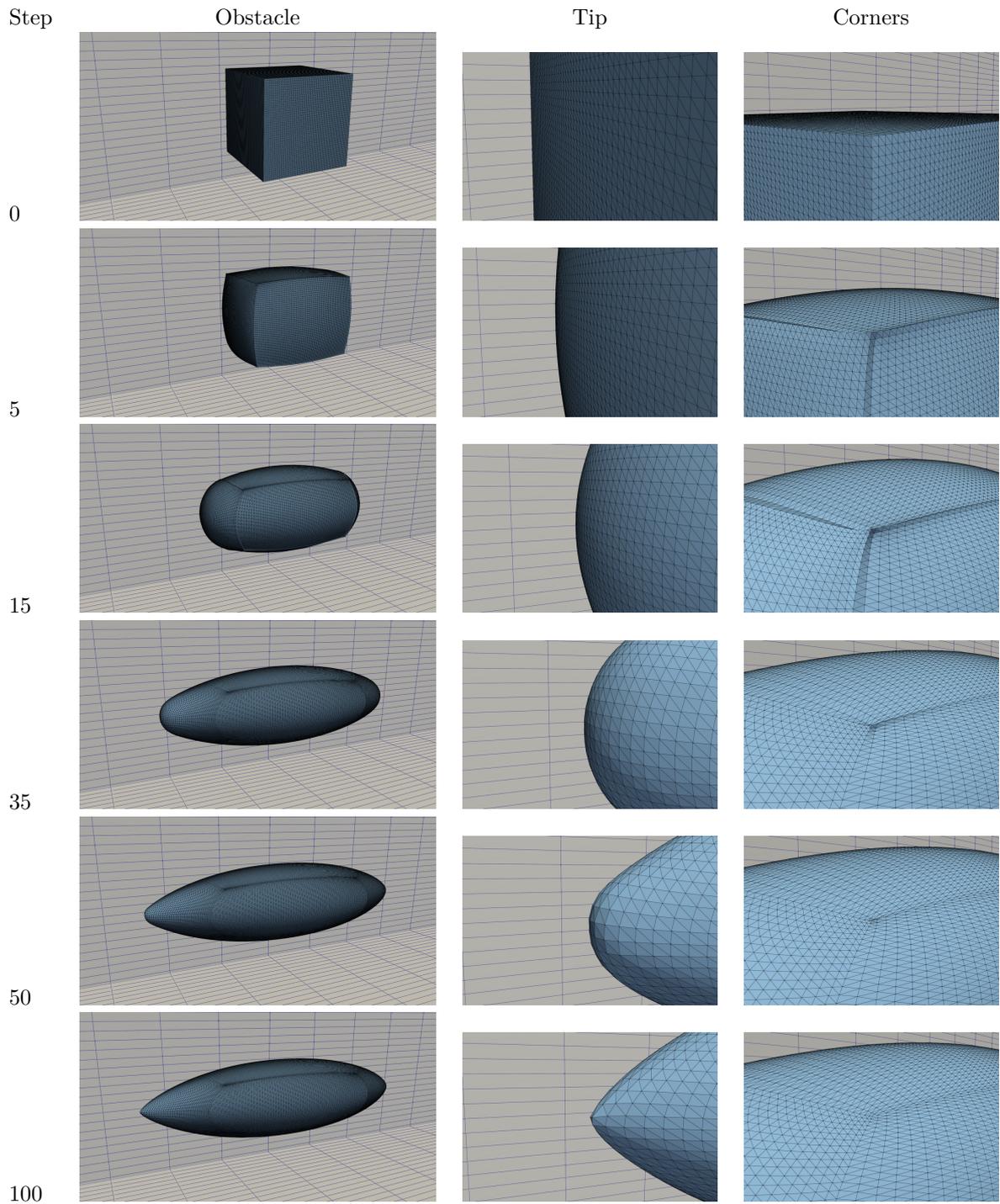


Figure 5: Deformation sequence for optimization steps $\{0, 5, 15, 35, 50, 100\}$. The complete obstacle, together with a detailed view of the geometrical singularity removal and generation process, are presented.

Figure 5 shows a deformation sequence of the 3d case, starting from the initial configuration and ending with an optimum obstacle surface. The figure focuses on the overall shape (left), an exemplary corner of the initial geometry (center), as well as the location of the upstream end of the final geometry (right). During the initial steps, the obstacle aligns to the flow, i.e. is stretched in the direction of the flow and compressed in the other two directions. Edges begin to emerge from the round upstream and downstream facing surfaces, thus creating the geometry observed in step 15. Subsequently, a round cross section starts to take form in the center and as seen in step 35, where the final tip locations also become more apparent. Recall that the mesh deformation corresponds to line 15 in algorithm 1. We again emphasize that all shape iterates meet the volume and barycenter constraints, which are deemed crucial for the success of this optimization scheme. Footprints of initial corners and edges are still visible in the mesh at later stages of the optimization, e.g. step 50. However, they are completely smoothed out towards the end of the simulation and only the macro elements, resulting from the grid hierarchy, are visible.

The front tip is shown for step 100, where also the previously existing singularities have disappeared.

As mentioned in section 2, this approach optimizes the obstacle’s shape for the functional given in (8). Therefore, results are provided in figure 6 for a 3d setting, which show how the generated shape, after convergence of algorithm 1, consists of an optimum with respect to the cost function.

Figure 6 depicts the objective function plot evolution over 120 optimization steps using 3 and 4 levels of grid refinement, respectively. The fact that the objective function (8) decreases monotonically is linked to lines 15-26 of algorithm 1, where a line search strategy is implemented. Once the deformation field is obtained for p_{max} , the geometry is updated and we get a new obstacle shape. The state equation is solved and the cost function calculated to guarantee that the new shape iterate represents a descent direction. As seen in lines 18-21, whenever the condition is false, the deformation is withdrawn and the step size control value is reduced to repeat the unsuccessful step with a scaled shape sensitivity J' . As a further indicator for the convergence, we approximate the distance between the iterated shapes Ω_{3ref}^k of the 3 refinements run to the optimal solution of the 4 refinements run Ω_{4ref}^∞ . Figure 6 shows the integrated volume that refers to the symmetric difference as

$$d(\Omega_{3ref}^k, \Omega_{4ref}^\infty) := |\Omega_{3ref}^k \setminus \Omega_{4ref}^\infty| + |\Omega_{4ref}^\infty \setminus \Omega_{3ref}^k| \quad (29)$$

The integration is carried out using the boolean filters of the VTK library [1] with which a triangulation of the surface of the volumes of interest can be obtained. A concatenation of the VTK boolean filters gives us a surface triangulation together with its normal vector. Then, utilizing divergence’s theorem the volume can be found.

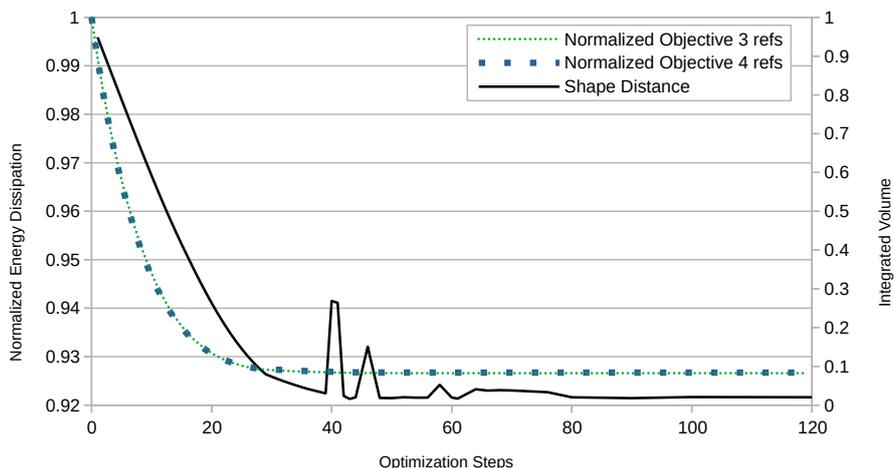


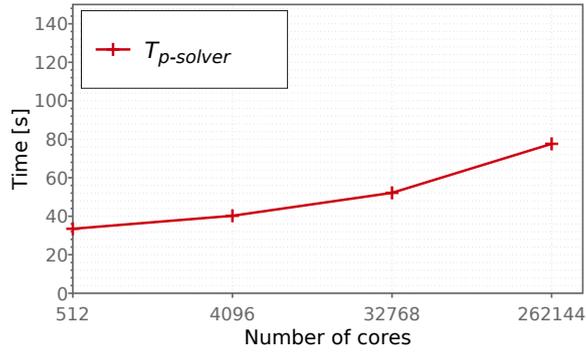
Figure 6: 3d results for 3 and 4 levels of refinement are compared. The energy dissipation, see (8), is plotted against the difference between shapes of each refinement level per step.

5.3 Scalability Study

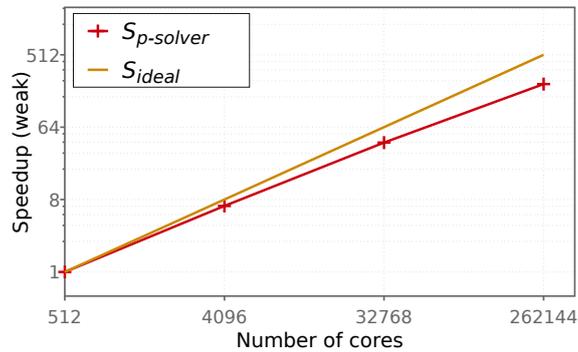
Weak scalability of the solution strategy for the p -Laplacian relaxed problem, from $p_{init} = 2.0$ up to p_{max} is presented here. This solution strategy, described in section 3, is referred in these results as the p -solver.

This corresponds to lines 11-15 of algorithm 1. It was studied for up to 262144 cores in a 3d setting. The study was carried out with the supercomputer Hawk at HLRS. It features 5632 compute nodes, each with a dual-socket architecture and a total of 128 cores. Each core with a maximum frequency of 2.25GHz, and 256GB of RAM. The runs were carried out taking into account the hypercube topology of the system to maximize core usage and minimize parallel communication overhead.

A 3d computational grid with 2 levels of refinement is used as an initial measurement in order to optimize the number of cores used at the finest level. The wallclock times, speedup, and iteration counts are shown in figure 7. An eight-fold increase in the number of cores is performed for each level of refinement. Results are presented for the solution of the nonlinear system of equations given in (20) via its linearization in (21) and (22). This system is solved using Newton’s method with a BiCGStab as a solver for the underlying linearization. The linear solver is set to absolute and relative error reductions of $1E-10$ and $1E-16$, respectively. It is preconditioned by a geometric multigrid method with 3 pre- and post smoothing steps via a Gauss-Seidel smoother within a V-cycle. An LU factorization solves the base level gathered in a single core.



(a) Time measurements



(b) Speedup relative to 512 cores

Procs	Refs	Num.Elems	DoFs	Newton Its.	Total Lin.Its.	Lin.Its. (21)
512	2	77 824	44730	68	2080	394
4096	3	622 592	334 158	68	2458	472
32 768	4	4 980 736	2 581 014	68	2606	509
262 144	5	39 845 888	20 283 942	68	2912	577

(c) Iteration counts for one optimization step of the solver used to obtain u_p , lines 11-15 in algorithm 1.

Figure 7: Weak Scaling: Results for the first optimization step. Accumulated wallclock time for all p -levels and speedup relative to 512 cores are shown. The number of Newton steps across several levels of refinement, as well as the linear solver iterations are presented in relation to the number of tetrahedrons per refinement level and the corresponding DoFs in the discretization of (27).

We measure the accumulated times and iteration counts for the routines in lines 11-15 of algorithm 1 for one optimization step. This can be understood as the time it takes to assemble the linearization, initialize the grid hierarchy necessary for the geometric multigrid preconditioner, and apply the linear solver until convergence within each call to the Newton’s solver. This is done for each value of p starting

at p_{init} up to p_{max} with p_{inc} intervals as explained in section 3. The time measurement starts for every optimization at $u_{p_{\text{init}}}$ and ends once the corresponding Newton solver for $u_{p_{\text{max}}}$ has converged. The speedup (b) is presented relative to the base measurement with 512 cores, and in (c) the iteration counts are shown in relation to the number of DoFs and tetrahedral elements. The column of the total linear iterations includes all the necessary calls to the linear solver used within the linearization. As shown in (28), for each solution of the linear system of equations it is necessary to solve $m + 2$ times with A^{-1} . These include one time for the rhs in the second equation of (28), and m for the computation of S . Additionally, the first equation of (28) has to be solved for δ_u , whose iteration counts are shown, individually, in the rightmost column of figure 7(c).

It can be seen that good scalability results are obtained for up to 262 144 cores. The communication costs impose a time overhead significantly lower to the very large increase of the number of DoFs. Altogether, the results show the need for using numerical solvers with grid independent convergence. Recall that our target is to use the solution of the p -Laplace relaxed problem for the highest value of p , i.e. p_{max} , as a deformation field to generate a series of shape iterates. Moreover, we do this by solving the same problem for lower values of p , in order to have a good initial guess as we approach the maximum p . The latter fact is necessary, since with each increment of p_{inc} , our problem becomes more nonlinear, implying it becomes more difficult to solve, particularly without a good initial guess. For the given settings, $p_{\text{init}} = 2.0$ to $p_{\text{max}} = 4.1$ and an increment of $p_{\text{inc}} = 0.19$, algorithm 2 has to be called thirteen times. Newton’s method has to call the linear solver for each of these p values. Therefore, there is an evident need for an efficient, fast, and computationally cheap preconditioner which allows for grid-size independent bounds on the convergence rate of the iterative methods. This is possible with the geometric multigrid method. One of the downsides is that this preconditioner requires a base level computational mesh that describes a geometry that can be represented by a grid hierarchy, see [23], which implies that care must be taken during the generation of the grid. Nevertheless, it is a very effective approach towards solving for u_p with increments of the p value. The results in figure 7 show that the p -relaxed problem becomes inexpensively solvable. Additionally, the benefits of the multigrid preconditioner are evident by noticing how the Newton’s method is perfectly scalable in the number of steps needed for all refinement levels, as well as in the slight increase in linear solver iterations between the initial and final runs. As seen in the table, even when the number of DoFs increases by three orders of magnitude, the timings and iteration counts are bound by the preconditioner.

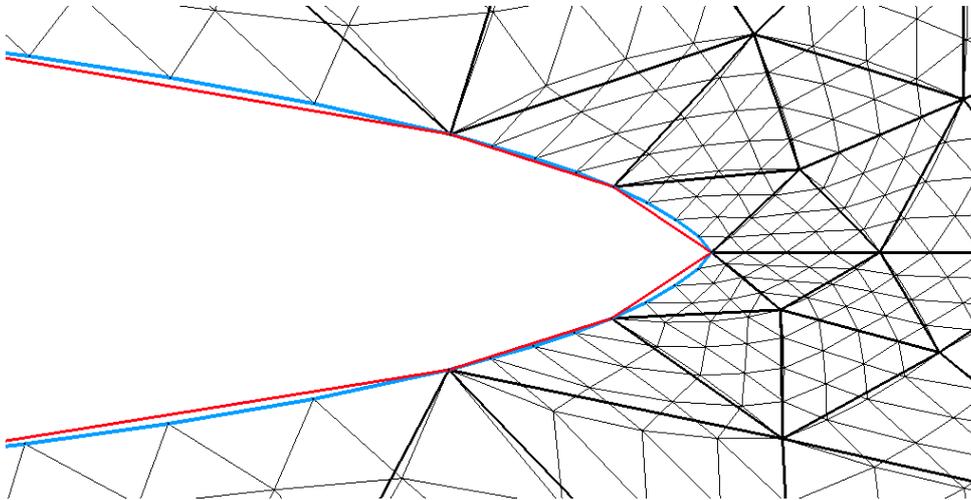


Figure 8: The base level and the second level of refinement are compared for a 2d simulation for the last step before convergence. The macro and refined triangular elements are shown in bold and thin black lines, respectively. Given that the deformation field is restricted and applied throughout the grid hierarchy, the coarsest grid is an interpolation of the finest.

In order to preserve numerical scalability across all optimization steps, it is necessary to apply the deformation field across the complete grid hierarchy. This is shown in figure 8, where the base level is compared to the finest grid with two refinements. It is visible how u_p is restricted and applied to all levels, therefore generating an optimal coarse grid. Given that this implies an interpolation of the vector field, and that by definition the obstacle’s surface on the coarsest grid has less nodes than the upper levels, there is a slight mismatch between the two grids. However, this has no detrimental effects nor

adds more computational complexity to the shape optimization scheme. Our scheme works on arbitrary Lipschitz shapes. Therefore, it is not necessary to incorporate extra geometric information to the grid hierarchy.

Overall, good weak scalability results were obtained for up to 39 million elements. This represents an increase of three orders of magnitude, both in tetrahedrons as in DoFs, with a slight increase in the necessary computational work in terms of linear iterations. Although the performance dropped marginally from the ideal case, the wallclock times and speedup show that the numerical scheme we propose for the solution of the p -Laplace relaxed problem could be used for problems with large numbers of DoFs, corresponding to real-world industrial applications.

6 Conclusion

In this work we presented a steepest descent shape method based on $W^{1,p}$ approximations of $W^{1,\infty}$ for shape optimization problems with PDE and fixed-dimensional geometric constraints. We demonstrated that the algorithm works for general Lipschitz shapes since deformations allow singularities in the surface to be smoothed or newly generated. Furthermore, we incorporated fixed-dimensional constraints together with the PDE constraints into the optimization algorithm via a Schur Complement approach. Compared to approximate algorithms such as the penalty based and augmented Lagrangian approaches, we demonstrated a significant gain of robustness in the treatment of geometric constraints over the optimization steps. Additionally, this work addressed line search schemes for the steepest descent direction in $W^{1,\infty}$, which -in contrast to the aforementioned Hilbert space methods- is here a nonlinear problem. Moreover, this problem lives on the solution manifold of the nonlinear geometric constraints posing a non-convex set in general.

The essential part of this work was to investigate the application of geometrical multigrid preconditioners on hierarchical grid structures without needing any further information, such as curvature based on spline surfaces. It was demonstrated that, via the shape optimization, a body-fitted hierarchical grid structure is found for the optimal shape. Our numerical studies indicated that under these circumstances the multigrid preconditioner features a mesh-independent solver for the deformation subproblem. As a consequence, we were able to demonstrate that the proposed method exhibits weak scalability up to 262 144 CPU cores of the distributed-memory system Hawk at HLRS.

Acknowledgements

The current work is part of the research training group “Simulation-Based Design Optimization of Dynamic Systems Under Uncertainties” (SENSUS) funded by the state of Hamburg under the aegis of the Landesforschungsförderungs-Projekt LFF-GK11.

The authors acknowledge the support by the Deutsche Forschungsgemeinschaft (DFG) within the Research Training Group GRK 2583 “Modeling, Simulation and Optimization of Fluid Dynamic Applications”.

Computing time on the national supercomputer HPE Apollo Hawk at the High Performance Computing Center Stuttgart (HLRS) under the grant ShapeOptCompMat (ACID 44171, Shape Optimization for 3d Composite Material Models) is gratefully acknowledged.

References

- [1] URL: vtk.org.
- [2] G. Allaire, C. Dapogny, and F. Jouve. “Chapter 1 - Shape and topology optimization”. In: *Geometric Partial Differential Equations - Part II*. Vol. 22. Handbook of Numerical Analysis. Elsevier, 2021, pp. 1–132. DOI: 10.1016/bs.hna.2020.10.004.
- [3] G. Allaire, F. Jouve, and A.-M. Toader. “Structural optimization using sensitivity analysis and a level-set method”. In: *Journal of computational physics* 194.1 (2004), pp. 363–393. DOI: 10.1016/j.jcp.2003.09.032.
- [4] J.S. Arora, A.I. Chahande, and Paeng J.K. “Multiplier methods for engineering optimization”. In: *International Journal for Numerical Methods in Engineering* 32.7 (1991), pp. 1485–1525. DOI: 10.1002/nme.1620320706.

- [5] C. Brandenburg et al. “Advanced Numerical Methods for PDE Constraint Optimization with Application to Optimal Design in Navier Stokes Flow”. In: *Constrained Optimization and Optimal Control for Partial Differential Equations*. Vol. 160. International Series of Numerical Mathematics. Brinkhäuser, Basel, 2011, pp. 257–275. DOI: 10.1007/978-3-0348-0133-1_14.
- [6] K. Deckelnick, P. Herbert, and M. Hinze. “A novel $W^{1,\infty}$ approach to shape optimisation with Lipschitz domains.” In: (2021). arXiv: 2103.13857.
- [7] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. SIAM, 2011. DOI: 10.1137/1.9780898719826.
- [8] A. Geiser, I. Antonau, and K.-U. Bletzinger. “Aggregated Formulation of Geometric Constraints for Node-based Shape Optimization with Vertex Morphing (Eurogen 2021)”. In: *14th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control*. Vol. 14. Eccomas Procedia, 2021. DOI: 10.7712/140121.7952.18383.
- [9] C. Geuzaine and J.-F. Remacle. “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. ISSN: 0029-5981. DOI: 10.1002/nme.2579.
- [10] W. Hackbusch. *Multi-grid methods and applications*. Vol. 4. Springer, 1985.
- [11] J. Haubner, M. Siebenborn, and M. Ulbrich. “A Continuous Perspective on Shape Optimization via Domain Transformations”. In: *SIAM* 43.3 (2021), A1997–A2018. DOI: 10.1137/20M1332050.
- [12] M. Hinze and K. Kunisch. “Second order methods for optimal control of time-dependent fluid flow”. In: *SIAM* 40 (2001), pp. 925–946. DOI: /10.1137/S0363012999361810.
- [13] H. Ishii and P. Loreti. “Limits of solutions of p-Laplace equations as p goes to infinity and related variational problems”. In: *SIAM journal on mathematical analysis* 37.2 (2005), pp. 411–437. DOI: 10.1137/S0036141004432827.
- [14] G. Karypis, K. Schloegel, and V. Kumar. *Parmetis, Parallel graph partitioning and sparse matrix ordering library*. Version 4.03. 2013. URL: <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview> (visited on 03/06/2020).
- [15] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2009. DOI: 10.1093/acprof:oso/9780199546909.001.0001.
- [16] B. Mohammadi and O. Pironneau. “Shape Optimization in Fluid Mechanics”. In: *Annual Review of Fluid Mechanics* 36 (2004), pp. 255–279. DOI: 10.1146/annurev.fluid.36.050802.121926.
- [17] P. M. Müller et al. “A novel p-harmonic descent approach applied to fluid dynamic shape optimization”. In: *Struct Multidisc Optim* (2021). DOI: 10.1007/s00158-021-03030-x.
- [18] S. Onyshkevych and M. Siebenborn. “Mesh Quality Preserving Shape Optimization Using Non-linear Extension Operators”. In: *Journal of Optimization Theory and Applications* 189 (2020), pp. 291–316. DOI: 10.1007/s10957-021-01837-8.
- [19] J. Ortiz and A. Schiela. *A composite step method for equality constrained optimization on manifolds*. 2019. URL: <https://eref.uni-bayreuth.de/47908/>.
- [20] J. Pinzon and M. Siebenborn. *Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners*. 2021. arXiv: 2108.07788v1.
- [21] J. Pinzon and M. Siebenborn. *PLaplaceOptim*. <http://www.github.com/multigridshapeopt>. Version 1.0. 2021. (Visited on 08/17/2021).
- [22] O. Pironneau. “On optimum profiles in Stokes flow”. In: *Journal of Fluid Mechanics* 59.1 (1973), pp. 117–128. DOI: 10.1017/S002211207300145X.
- [23] S. Reiter et al. “A massively parallel geometric multigrid solver on hierarchically distributed grids”. In: *Comp. Vis. Sci.* 16.4 (2013), pp. 151–164.
- [24] A. Schiela and J. Ortiz. “An SQP Method for Equality Constrained Optimization on Hilbert Manifolds”. In: 31.3 (2021), pp. 2255–2284. DOI: 10.1137/20m1341325.
- [25] V. Schulz and Siebenborn M. “Computational Comparison of Surface Metrics for PDE Constrained Shape Optimization”. In: *Computational Methods in Applied Mathematics* 16.3 (2016), pp. 485–496. DOI: 10.1515/cmam-2016-0009.
- [26] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization. Shape Sensitivity Analysis*. Springer, Berlin, Heidelberg, 1992. DOI: 10.1007/978-3-642-58106-9.

- [27] M. Ulbrich. “Constrained Optimal Control of Navier-Stokes Flow by Semismooth Newton Methods”. In: *Systems & Control Letters* 48 (2003). DOI: 10.1016/S0167-6911(02)00274-8.
- [28] Hirofumi Uzawa. “Iterative methods for concave programming”. In: *Studies in linear and nonlinear programming* 6 (1958), pp. 154–165.
- [29] A. Vogel et al. “UG 4: A novel flexible software system for simulating PDE based models on high performance computers”. In: *Comp. Vis. Sci.* 16.4 (2013), pp. 165–179.