

# Shape optimization approaches towards scalable algorithms allowing large deformations



Cumulative Dissertation

with the aim of achieving a doctoral degree at the Faculty of  
Mathematics, Informatics and Natural Sciences, Department  
of Mathematics from Universität Hamburg

Submitted by

**José Alfonso Pinzón Escobar**  
from Merida, México

Hamburg, 2024



Supervisor:

Dr. Martin Siebenborn (From August  
2020 to October 2022)

Prof. Winnifried Wollner (From October  
2022 to September 2023)

Date of oral defense:

05.03.2024

Chair of the examination commission:

Prof. Dr. Ingenuin Gasser

Reviewers:

Prof. Dr. Winnifried Wollner

Dr. Martin Siebenborn

Prof. Dr. Ing. Thomas Rung



## **Official Declaration**

Hereby I declare, that I have not submitted this thesis in this or similar form to any other examination at the Universität Hamburg or any other institution or university.

I officially ensure, that this paper has been written solely on my own. I herewith officially ensure, that I have not used any other sources but those stated by me. Any and every parts of the text which constitute quotes in original wording or in its essence have been explicitly referred by me by using official marking and proper quotation. This is also valid for used drafts, pictures and similar formats.

I also officially ensure that the printed version as submitted by me fully confirms with my digital version. I agree that the digital version will be used to subject the paper to plagiarism examination. Not this English translation, but only the official version in German is legally binding.

## **Eidesstattliche Erklärung**

Ich erkläre, dass ich keine Arbeit in gleicher oder ähnlicher Fassung bereits für eine andere Prüfung an der Universität Hamburg oder einer anderen Hochschule eingereicht habe.

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Die Stellen, die anderen Quellen dem Wortlaut oder dem Sinn nach entnommen sind, habe ich unter Angabe der Quellen kenntlich gemacht. Dies gilt sinngemäß auch für verwendete Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen.

Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Ich erkläre mich damit einverstanden, dass die digitale Version dieser Arbeit zwecks Plagiatsprüfung verwendet wird.

---

PLACE, DATE

---

JOSE A. PINZON ESCOBAR



# Abstract

Shape optimization is an area that aims at improving the design of an object with respect to a given physical quantity expressed by an objective function. In most meaningful applications the objective function requires the solution to a partial differential equation (PDE). There are empirical techniques for improving existing designs. However, recent developments in the mathematical theory used to formulate these kind of problems has triggered new research in this area. Moreover, the development of progressively faster computers has lead to applications involving very large number of degrees of freedom (DoFs).

The goal of this work is to investigate, develop, and implement state-of-the-art techniques in shape optimization. The main area of interest is in applications which require large deformations to optimize a given domain. Here, the formulation of the problem is done in a continuous setting, and its followed by a discretization of the domain via the finite element method (FEM). This discrete setting describes the geometry which is to be optimized by applying a series of deformation fields across the nodes of the grid. It is during this process, that the discrete elements might undergo mesh quality losses. Therefore, this thesis is part of the ongoing interest of the research community on techniques that allow for large deformations, while preserving mesh quality.

In the context of finite element analysis, complex geometries usually require highly refined computational meshes to yield accurate results. Thus, here we also focus on implementing case studies with a very high number of DoFs. With this purpose, the proposed algorithms are implemented using the parallel-computing simulation framework UG4. Additionally, the results are generated using massive distributed-memory systems, i.e. supercomputers.

A literature-based, fluid dynamics case study is used to benchmark the shape optimization methods developed in this thesis. The used example is that of an object located in the center of a flow tunnel, whose surface must be optimized with respect to a certain quantity such as the drag or lift. For the studies within this work, the shape of the object is set to include geometric singularities -e.g. edges and corners- which are to be removed during the optimization. Furthermore, based on the literature, it is known that the optimized obstacle must include newly generated geometric singularities. Thus, this is a good benchmarking case for large deformation techniques in shape optimization.

# Zusammenfassung

Formoptimierung ist ein Bereich, der darauf abzielt, das Design eines Objekts in Bezug auf einen gegebenen physikalischen Größe zu verbessern, die durch eine Zielfunktion ausgedrückt wird. In den meisten nicht-trivialen Anwendungen erfordert die Zielfunktion die Lösung einer partiellen Differentialgleichung (englisch: partial differential equation, PDE). Es existieren empirische Techniken zur Verbesserung bestehender Designs. Allerdings haben neue Entwicklungen in der mathematischen Theorie, die zur Formulierung dieser Art von Problemen verwendet wird, neue Forschung in diesem Bereich ausgelöst. Darüber hinaus hat die Entwicklung von zunehmend schnelleren Computern zu Anwendungen geführt, die eine sehr große Anzahl von Freiheitsgraden (englisch: degrees of freedom, DoFs) umfassen.

Das Ziel dieser Arbeit besteht darin, hochmoderne Techniken in der Formoptimierung zu untersuchen, zu entwickeln und umzusetzen. Das Hauptinteressengebiet liegt in Anwendungen, die große Verformungen erfordern, um eine gegebene Domäne zu optimieren. Hier wird das Problem in einem kontinuierlichen Rahmen formuliert und anschließend durch die Methode der finiten Elemente (FEM) diskretisiert. Dieser diskrete Rahmen beschreibt die Geometrie, die durch die Anwendung einer Reihe von Deformationsfeldern über die Knoten des Gitters optimiert werden soll. Während dieses Prozesses können die diskreten Elemente möglicherweise Qualitätsverluste im Gitter erleiden. Daher ist diese Arbeit Teil des fortwährenden Interesses der Forschung an Techniken, die große Verformungen ermöglichen und gleichzeitig die Gitterqualität erhalten.

Im Kontext der Finite-Elemente-Analyse erfordern komplexe Geometrien normalerweise hochfeine Berechnungsgitter, um genaue Ergebnisse zu liefern. Daher konzentrieren wir uns hier auch auf die Implementierung von Fallstudien mit einer sehr großen Anzahl von DoFs. Zu diesem Zweck werden die vorgeschlagenen Algorithmen mit einem parallelisierten Framework, UG4, implementiert. Darüber hinaus werden die Ergebnisse mit massiven verteilten Speichersystemen, d.h. Supercomputern, erzeugt.

Eine auf der Literatur basierende Fallstudie zur Fluidodynamik wird verwendet, um die in dieser Arbeit entwickelten Formoptimierungsmethoden zu benchmarken. Das verwendete Beispiel ist das eines Objekts in der Mitte eines Strömungstunnels, dessen Oberfläche hinsichtlich einer bestimmten Größe wie dem Strömungswiderstand oder dem Auftrieb optimiert werden muss. Für die Studien innerhalb dieser Arbeit ist die Form des Objekts so festgelegt, dass sie geometrische Singularitäten wie Kanten und Ecken enthält, die während der Optimierung entfernt werden sollen. Darüber hinaus ist aus der Literatur bekannt, dass das optimierte Hindernis neu generierte geometrische Singularitäten enthalten muss. Daher handelt es sich um einen guten Benchmark-Fall für Techniken zur Formoptimierung bei großen Verformungen.

# Acknowledgments

First and foremost, I want to thank the funding provided by the state of Hamburg through the project “Simulation-Based Design Optimization of Dynamic Systems Under Uncertainties” (SENSUS) under the aegis of the Landesforschungsförderungs-Projekt LFF-GK11. Specially I want to thank Prof. Thomas Rung, not only for agreeing to be a reviewer of my thesis, but also for his leading role within the SENSUS project.

Of course, I would like to acknowledge Prof. Winnifried Wollner for receiving me in his working group, and thank him for all his support, including reviewing my thesis. I know this took time and work from him, for which I am extremely grateful. Above everyone else, I am forever indebted to (former Jun. Prof. )Dr. Martin Siebenborn, who offered me the opportunity to join the Math department at the University of Hamburg and do my PhD. He was not only a supervisor of my thesis, but also a leader through the different research topics we tackled during my PhD. To me, he is an example of passion for research and professionalism, and I have the privilege of calling him my friend. The research in this thesis would not have been possible without the collaboration of highly talented and smart researches, therefore I want to thank my co-authors and colleagues Dr. Peter Marvin Müller and Dr. Philip J. Herbert. The scientific discussions certainly enriched me as a researcher, and the casual conversations made my stay at the university a fun time to look back on.

I also want to thank my parents and my brother. They have been there for me always, and none of this would be possible without them. And thank you, Barbara, for me carrying me through the finish line.

The results presented in this thesis were obtained using some of the most powerful and modern supercomputers in Germany. So I would like to acknowledge the computing time on the national supercomputer Lise at the North German Supercomputing Alliance (HLRN) under the project hhm00006 (An optimal shape matters), and 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).



*A mi papá, el mejor ingeniero del mundo*



# List of publications

## Published research articles

1. Müller, Peter Marvin, Pinzon Escobar, Jose A., Rung, Thomas, Siebenborn, Martin. “A Scalable Algorithm for Shape Optimization with Geometric Constraints in Banach Spaces.” *SIAM Journal on Scientific Computing*, vol. 45, no. 2, Apr. 2023, pp. B231–51. DOI.org (Crossref), <https://doi.org/10.1137/22M1494609>.
2. Pinzon, Jose, and Martin Siebenborn. “Fluid Dynamic Shape Optimization Using Self-Adapting Nonlinear Extension Operators with Multigrid Preconditioners.” *Optimization and Engineering*, vol. 24, no. 2, June 2023, pp. 1089–113. DOI.org (Crossref), <https://doi.org/10.1007/s11081-022-09721-8>.

## Peer-reviewed conference proceedings

1. Pinzon Escobar, Jose Alfonso, and Martin Siebenborn. “Shape Optimization Algorithms for Fluid Dynamics Applications.” *PAMM*, vol. 22, no. 1, Mar. 2023, p. e202200279. DOI.org (Crossref), <https://doi.org/10.1002/pamm.202200279>.
2. Pinzon, J., Siebenborn, M., and Vogel, A. (2023). Scalable multigrid algorithm for fluid dynamic shape optimization. In *High Performance Computing in Science and Engineering '21* (pp. 481–495). Springer International Publishing.

## Unpublished manuscripts

1. Herbert, P. J., Pinzon Escobar, J. A., and Siebenborn, M. (2023). Shape optimization in  $W^{1,\infty}$  with geometric constraints: A study in distributed-memory systems. ArXiv preprint, submitted (in review). arxiv:2309.15607.
2. J. Pinzon, M. Siebenborn, and A. Vogel. Geometric Constrained Scalable Algorithm for PDE-Constrained Shape Optimization. Accepted for publication: *Proceedings of the High Performance Computing in Science and Engineering - 25th Results and Review Workshop* (2022).



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	PDE-constrained shape optimization . . . . .	2
1.2	An algorithmic framework . . . . .	4
1.3	Case study . . . . .	6
1.4	Numerical methodology . . . . .	8
<b>2</b>	<b>Synthesis Essay</b>	<b>10</b>
2.1	Nonlinear extension equation . . . . .	12
2.1.1	Background . . . . .	12
2.1.2	Results . . . . .	14
2.2	$p$ -Laplace relaxation . . . . .	17
2.2.1	Background . . . . .	17
2.2.2	Results . . . . .	20
2.3	$W^{1,\infty}$ -descent directions . . . . .	24
2.3.1	Background . . . . .	24
2.3.2	Results . . . . .	26
2.4	Discussion . . . . .	29
<b>3</b>	<b>Publications</b>	<b>31</b>
	Paper I . . . . .	32
	Paper II . . . . .	58
	Paper III . . . . .	80
<b>4</b>	<b>Conclusion and Outlook</b>	<b>101</b>
4.1	Concluding remarks . . . . .	101
4.2	Outlook . . . . .	102
<b>5</b>	<b>Declaration of personal contribution</b>	<b>104</b>

# Chapter 1

## Introduction

There is ongoing research in the field of PDE-constrained shape optimization. An introduction to the area can be found in the major monographs [16, 76, 31]. In [61], shape optimization is introduced in the context of elliptical PDEs. The recent book [33] and monograph [2], together with their respective bibliographies, provide an extensive overview. Several applications can be found in the fields of structural mechanics [21, 4, 3]; fluid dynamics [67, 66, 42, 12]; electrostatics [24]; and acoustics [79], to mention some. Renewed interest in the field has resulted in further advances in shape optimization [71, 14, 63].

This work proposes novel algorithmic schemes for solving optimization problems, where the objective function depends on the domain, the so-called field of shape optimization. Within the optimization process, one key step in a shape optimization algorithm is finding a deformation field that can generate a better, optimized shape, see e.g. [2, Chapter 5.1] for a general overview. In this thesis we are interested in the algorithms used to obtain these descent directions, i.e. the deformation fields, that are iteratively applied to the geometry to find an optimized domain. There is extensive ongoing research in this field, as illustrated in [14, 38, 10], to mention some recent examples. We place particular interest in shape functionals, which are a function both of the domain and of a solution to a partial differential equation (PDE).

On this topic, the preservation of mesh quality during the optimization process has been the subject of recent research. By this, it is meant the appearance of phenomena such as the overlapping of the domain with itself, as well as the degeneration of the discrete elements in the computational grid. Some efforts in this field are related to extending the deformation across the domain [38]. Another approach is based on remeshing the domain either after a fixed number of iterations, or when the mesh quality goes below a certain threshold [82, 78, 65]. Or, for instance, restricting the deformation in the discrete setting [22]. Recently, advances on the simultaneous optimization of the shape and mesh quality [45, 46] have been proposed. This thesis aims at contributing to this general line of research.

In this work we make use of two different approaches for the generation of an optimized shape. As already mentioned, one approach is based on an iterative process that results in a sequence of shape iterates, e.g. [56]. On the other hand, the second approach is based on the method of mappings [52], that allows us to work on the reference configuration by not requiring explicit domain deformations. As explained in [32, 54], this formulation provides more control over the mesh quality of the optimal shape. That is, the achieved optimal shape can fulfill constraints over the deformation, which would otherwise not be fulfilled if a concatenation of deformations would be used. It was used, for instance, within the context of shape optimization on  $W^{1,\infty}$  in [15]. Therefore, this thesis is mainly focused on developing and analyzing new techniques for obtaining the deformation fields in case studies where large deformations are required to obtain an optimized shape. And where

it is important to preserve the mesh quality of the computational grid across the optimization process.

Our research dealt with algorithms in the widely used Hilbertian framework [2, 54, 44], as well as in the novel approaches that aim at finding descent directions in Banach spaces [14, 15, 50]. These algorithms are benchmarked using a well-know example from fluid mechanics, the so-called rugby ball case study [59]. It was used recently for example in [69], in the context of shape optimization on Riemannian manifolds. Given that geometric constraints are required for the formulation of this problem, in this thesis we propose a method to handle these constraints. Additionally, our experiments are implemented in a high performance computing setting as a stepping stone for more complex, industrial applications.

The outline is as follows: in the remainder of Chapter 1, the theoretical background, algorithmic framework, as well as the case study and numerical setting used in this thesis are presented. In Chapter 2, the publications that compose this thesis are discussed via a theoretical introduction and some meaningful results, followed by a discussion where the advantages and disadvantages between them are compared. Finally, Section 4.1 provides the concluding remarks and an outlook for further research in this field.

## 1.1 PDE-constrained shape optimization

The goal of shape optimization is to find the shape  $\Omega^*$ , within a set of admissible shapes, that minimizes the value of an objective function. An unconstrained problem of this type can be abstractly defined as

$$\min_{\Omega \in \mathcal{S}_{\text{ad}}} J(\Omega), \quad (1.1)$$

where  $\mathcal{S}_{\text{ad}}$  is a subset of all subsets of  $\mathbb{R}^d$ , defined as the set of admissible shapes. The objective function  $J: \mathcal{S}_{\text{ad}} \mapsto \mathbb{R}$  defines some property over the domain which is used as a measure of performance, it is sometimes referred to as the shape functional. These terminology will be used interchangeably from now on. The set of admissible shapes  $\mathcal{S}_{\text{ad}}$  is assumed to be a collection of open and bounded domains in  $\mathbb{R}^d$ , where we limit ourselves to  $d \in \{2, 3\}$ . Here  $\Omega$  is assumed to be a Lipschitz-bounded domain, with boundary  $\Gamma$ . The boundary might include regions that are not part of the optimization procedure. For instance, when a section of the boundary has boundary constraints.

It is not our intention to give here a complete theoretical introduction to the topic of shape optimization, but interested readers are directed to the major monographs [16, 76]. In particular, topics such as existence of solutions, differentiability of shape functionals, or convergence of shape optimization algorithms are out of the scope of this thesis, but interested readers can refer to, e.g., [33]. An engineering perspective, based on automatic differentiation, can be found in [44]. In this section, only the theoretical aspects of shape optimization, which are required to introduce an the algorithmic approach used in this thesis, are mentioned.

Assuming that a solution to (1.1) exists, an optimal shape  $\Omega^* \in \mathcal{S}_{\text{ad}}$  can be found as the limit to a series of shape iterates. This can be understood as finding a perturbed shape  $\Omega_u$ , starting from an initial configuration  $\Omega_0$ , such that  $J(\Omega_u) < J(\Omega_0)$  is fulfilled. With this purpose, a mechanism to generate shape variations has

to be defined. And in this sense, a perturbed domain can be parametrized by the deformation field  $u : \mathbb{R}^d \mapsto \mathbb{R}^d; x \mapsto u(x)$ , which is applied over the whole domain. From now on,  $u(x)$  will be written as  $u$ . Thus, for a small enough  $u$ , a perturbed domain can be obtained as

$$\Omega_u = \{x + u(x), \quad x \in \Omega\}. \quad (1.2)$$

This is understood as in [2, 33], i.e. as moving all the points in  $\Omega$  according to the descent direction  $u$  to generate a variation of  $\Omega$ , where the topology remains the same.

With this purpose, it is necessary to define a mapping, which transforms the domain, such that  $F : \Omega \mapsto \Omega_u$ . In this work,  $F$  is defined as

$$F := \text{id} + u, \quad (1.3)$$

called the perturbation of identity, where  $u$  is a sufficiently smooth deformation field which parametrizes the transformed shape as  $\Omega_u = F(\Omega)$ . It is important for the transformation  $F$  to be of Lipschitz type, with Lipschitz inverse [2, 33]. This is the case for a sufficiently small  $u$ , see [76]. Therefore, the deformation field  $u$  is considered to belong to the space  $W^{1,\infty}(\mathbb{R}, \mathbb{R})$  with  $\|u\|_{W^{1,\infty}} < 1$ , see [33, section 5.2.2].

In meaningful applications the shape functional is a function of the solution to a PDE, known as the state equation. Let

$$J(\Omega, y) := \int_{\Omega} j(y) dx, \quad (1.4)$$

where  $y$  is referred to as the state variable and is the solution to the aforementioned PDE defined on the domain  $\Omega$ . Therefore, the optimization problem presented in Eq. (1.1) is extended to include a constraint. It can be formulated as

$$\begin{aligned} \min_{\Omega \in \mathcal{S}} \quad & J(\Omega, y) \\ \text{s.t.} \quad & e(\Omega, y) = 0, \end{aligned} \quad (1.5)$$

where  $e(\Omega, y)$  -the PDE constraint- depends on the state  $y$  and the shape of the domain. Throughout this work, the state equation will be referred as  $e$  and  $J(\Omega)$  as  $J$ , for readability.

One common optimization method used to solve problem (1.5) is based on the introduction of Lagrange multipliers [5, 40]. Assuming that  $y$  is the solution to (1.1), as explained in [36], this can be viewed as obtaining the reduced objective function, which can be formulated in terms of the Lagrangian function

$$J(\Omega)_{\text{red}} = \mathcal{L}(\Omega, y, \lambda_y) = J(\Omega, y) + \langle \lambda_y, e(y, \Omega) \rangle, \quad (1.6)$$

where  $\lambda_y$  are the multipliers associated to  $y$ . Moreover, based on control theory [36], it is assumed that the state is unique on  $\Omega$  and that the control-to-state mapping  $\Omega \mapsto y(\Omega)$  exists. This is an important assumption used to define optimality conditions. The  $\lambda_y$  can be used as the adjoint variable to  $y$ , and obtained by solving the corresponding adjoint equation. With it,  $J'(\Omega)_{\text{red}}$  can be computed as described in [2, 36].

This leads us to the concept of shape derivative. It is usually defined in terms of the Eulerian derivative

$$J'(\Omega)u := dJ(\Omega)u = \lim_{t \rightarrow 0} \frac{J(\Omega_{tu}) - J(\Omega)}{t},$$

and, here, it is assumed to exist for all directions  $u$ . The mapping  $u \mapsto dJ(\Omega)$  is assumed to be linear. Under these conditions,  $dJ(\Omega)$  is called the shape derivative of  $J$  in the direction  $u$ . Moreover, the value of the shape functional evaluated on the shape variation is defined by the expansion

$$J(\Omega_u) = J(\Omega) + J'(\Omega)u + o(u) \text{ where } \frac{o(u)}{\|u\|_{W^{1,\infty}(\mathbb{R},\mathbb{R})}} \xrightarrow{\|u\| \rightarrow 0} 0$$

as in [2], where it is explained that the differentiability at  $\Omega$  stems from the mapping  $u \mapsto J(\Omega_u)$  being Fréchet differentiable at  $u = 0$ . The shape derivative in the direction  $u$  will be denoted, interchangeably, as  $J'$  or  $J'(\Omega)$  from now on, and it will refer to the derivative of the reduced problem  $J_{\text{red}}$ .

In this work, the shape derivative  $J'$  is obtained via the method of Lagrange multipliers. The problem is formulated in the deformed domain  $\Omega_u$ , and then pulled back to the reference configuration. Then, the derivative is evaluated. It is computed by obtaining the solutions to the state and adjoint equations. Under the mentioned conditions, the deformation field minimizes the functional  $J$  for a descent direction, so that the condition  $J'(\Omega)u < 0$  is fulfilled. This means that for a sufficiently small descent direction the value of  $J(\Omega)$  is decreased when deformed in the direction of  $u$ . This can be expressed as  $J(\Omega_u) = J(\Omega) - tJ'(\Omega)u + o(t) < J(\Omega)$  for a small  $t > 0$ . Thus, finding a descent direction that fulfills the previously mentioned conditions is a nontrivial task.

## 1.2 An algorithmic framework

As explained in Section 1.1, the Lagrangian multipliers method can be used to obtain the shape derivative, whose computation depends on the state  $y$  and its adjoint variable. With it, descent directions that fulfill  $J'(\Omega)u < 0$  can be found. Therefore, this can be used to formulate a steepest descent-like method, with which the initial shape can be iteratively optimized by applying the aforementioned deformation field inferred from  $J'(\Omega)$ .

Several different methodologies have been proposed for this. In [2] an overview of methods based on a Hilbertian framework is found. These methods are based on computing the descent direction via the Riesz representation theorem, which allows to relate a bilinear form to the shape derivative. The inner product of a Hilbert space is then used to obtain regular deformation fields, for instance across the whole domain.

Although in this work the shape derivative is defined in its volume integral form, it is also possible to use its boundary form as in [2, 77, 19]. This implies some differences which are discussed in [37]. It is explained there, that in a discrete setting -such as FEM- the equivalence between the two formulations is lost. Nevertheless, in a continuous setting both forms can be used without loss of generality.

---

**Algorithm 1** Descent Method

---

**Require:** Initial configuration  $\Omega^0$

- 1: **for**  $n=0, \dots$ , until convergence **do**
  - 2:    $y \leftarrow$  Solution of state equation
  - 3:    $\lambda_y \leftarrow$  Solution of adjoint equation to the state
  - 4:   Compute the shape gradient  $J'(\Omega^n)$ , i.e. the shape derivative
  - 5:   Find a descent direction  $u^n$  that fulfills  $J'(\Omega^n)u^n < 0$
  - 6:   Apply the deformation to obtain  $\Omega^{n+1}$ , so that  $J(\Omega^{n+1}) < J(\Omega^n)$  is fulfilled
  - 7: **end for**
  - 8: **return** Optimized geometry  $\Omega^n$
- 

A generic methodology for PDE-constrained shape optimization is given in Algorithm 1. This can be used to solve problem (1.5). Starting from an arbitrarily shaped domain  $\Omega_0$ , a series of deformation fields is computed and applied partly or completely over the geometry. This generates a series of shape iterates  $\Omega^{n+1}$ , as seen in line 6. The condition  $J(\Omega^{n+1}) < J(\Omega^n)$  verifies that the newly generated geometry represents a decrease of the objective function. On each iteration, the state and adjoint equations must be solved with the purpose of computing the shape derivative  $J'(\Omega)$ , which might prove both computationally challenging and expensive. Here, however, we ignore the costs associated with finding the solution of the state and adjoint equations, since it is not the focus of this thesis. Finally, and assuming that the state and adjoint variables have been approximated, most of the computational work is spent finding a descent direction  $u$ , as in line 5 of Algorithm 1. The latter procedure is the core of this thesis.

The framework given in Algorithm 1 is an abstract construct which is not usable in nontrivial applications, such as the ones covered here. Although it illustrates the general workflow of how a shape optimization simulation could be carried out, in our work a slightly more complex algorithm had to be implemented. For instance, a step size control mechanism can be included to fulfill the condition in line 6. In [51], we discuss why this proves challenging when the problem includes geometrical constraints. Additionally, a routine to revert step 6 might be necessary when  $J$  evaluated on  $\Omega^{n+1}$  does not decrease with respect to the previous value. Also, no constraints are considered in Algorithm 1, which would require a formulation in terms of a Lagrangian function together with an update rule for the multipliers. The general algorithmic scheme used in this work can be found repeatedly in [51, 34].

It is also worth mentioning another type of formulation, which we used in [55]. The so-called method of mappings [52, 74, 43, 75], where the problem is formulated on the reference domain and the deformation step in line 6 is not performed. This is achieved by introducing a transformation  $F$ , with which the current configuration is pulled back to the reference domain. In this way,  $F$  is parametrized on every step of the outer loop of Algorithm 1 by a deformation field  $u$ . The optimized grid, and the sequence of shape iterates, can be generated in post-processing.

### 1.3 Case study

The work presented in this thesis uses a case study taken from fluid mechanics to benchmark the different strategies used to obtain a descent direction  $u$ . This case study is used in [60] for a flow described by Stokes equations, where the shape of an object is to be minimized with respect to the energy dissipation. An approximation is given for the optimal shape of an object with unit-volume, however this is not obtained through an algorithmic implementation. Nevertheless, the optimal profile of a 2d obstacle is referred to as the prolate spheroid, with conical front and back in the direction of the flow. In [59] this analysis is extended to Navier-Stokes flow, where the Reynolds number is higher. Moreover, the resulting optimal profile is referred to as a rugby ball with a wedge-like front end. Throughout this work, we use this terminology.

The topic of optimal design for energy dissipation, or drag reduction, has also been explored for other types of flows in [62]. Some computational aspects, such as automatic differentiation, are discussed in [49, 44]. A topic-specific monograph [48] provides an introduction to shape optimization in fluid mechanics, together with many examples and case studies. Formulations oriented to industrial applications in wing and jet design can be found in [40, 41, 26], where aerodynamic phenomena, such as transonic flow, are studied. Together with the aforementioned sources, further research of shape optimization in fluid mechanics can be found in [30, 11] and [8, 23].

In this thesis, the geometry is configured as in Fig. 1.1. It represents a flow tunnel where an object is placed inside. There is a known flow in the inlet of the tunnel  $\Gamma_{\text{in}}$ , which generates energy dissipation over the surface of the obstacle. Therefore, the task is to optimize  $\Gamma_{\text{obs}}$  for the energy dissipation.

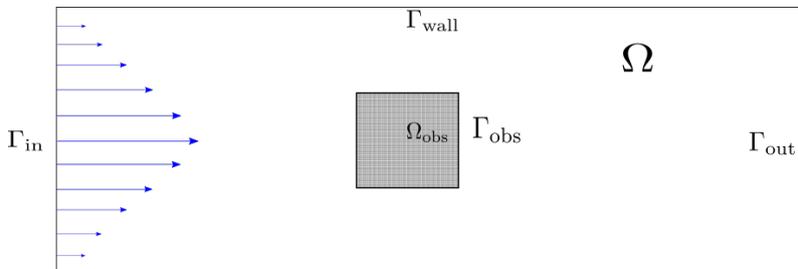


Figure 1.1: Schematic of the holdall domain,  $\Omega$ . The obstacle  $\Omega_{\text{obs}}$  includes geometric constraints which must be removed. The surface of the obstacle,  $\Gamma_{\text{obs}}$  is to be optimized, and the outer boundary  $\Gamma_{\text{wall}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$  is fixed.

Additionally, we are interested on studying applications that require large deformations of the initial configuration to obtain an optimized domain. The performance of the algorithms in Chapter 2 is partly tested with respect to their capacity to remove and create geometric singularities. For this reason, some of the studies in this work [55, 51, 34] were carried out using an initial obstacle shape that included geometric singularities. These must be removed in the optimized surface, and the conic front and back must be generated. And this must occur while preserving mesh quality, inasmuch as the convergence of the used iterative solvers is not affected. The outer boundaries of the domain are fixed throughout the optimization.

In this context, let  $\Omega^d$  be a Lipschitz-bounded domain, with boundary  $\Gamma = \Gamma_{\text{obs}} \cup$

$\Gamma_{\text{wall}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ . The spatial dimension is chosen as  $d \in \{2, 3\}$  and  $\Omega^d$  is written as  $\Omega$  from now on for simplicity. As previously mentioned,  $\Gamma_{\text{obs}}$  is considered to be variable and the outer boundary of  $\Omega$  remains fixed. The minimization is done with respect to the energy dissipation functional, defined as

$$J(\Omega) := j(\Omega, v) = \frac{\nu}{2} \int_{\Omega} Dv : Dv \, dx, \quad (1.7)$$

where  $\nu$  refers to the kinematic viscosity and the flow velocity is given by  $v$ . The state variable corresponds to the solution of the incompressible, steady-state Navier-Stokes equations

$$\begin{aligned} -\nu \Delta v + (v \cdot \nabla)v + \nabla \mathbf{p} &= 0 \text{ in } \Omega \\ \operatorname{div} v &= 0 \text{ in } \Omega \\ v &= 0 \text{ on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \\ v &= v_{\infty} \text{ on } \Gamma_{\text{in}} \\ \nu Dv \cdot n &= \mathbf{p}n \text{ on } \Gamma_{\text{out}}. \end{aligned} \quad (1.8)$$

where  $\mathbf{p}$  is the pressure, and the state variable  $y$  is described by the pair  $v, \mathbf{p}$ . For simplicity, we will refer to the state variable either by  $y$  or by the appropriate variable  $v$  or  $\mathbf{p}$  from now on. The state equation (1.8) is used in its weak form, as given for instance in [18]. As shown in Fig. 1.1, in  $\Gamma_{\text{in}}$  there is a known flow  $v_{\infty}$ , whose profile is known 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  $\delta$  is the height of the entry surface. It reaches a peak uni-value at the center of the inlet plane. The dimensions of the holdall and obstacle are known.

The addition of geometric constraints is found for instance in [48], where a constant volume constraint in optimization with respect to the lift is found. This is extended in some formulations, for instance [69, 10, 38, 54, 50], to include a barycenter constraint. In this case study, the volume and barycenter of the obstacle are fixed. The purpose being to avoid trivial solutions, i.e. design optimizations which are nonmeaningful or that do not provide an improved design. For instance, the object can be reduced to a point if its volume is decreased on every iteration of Algorithm 1. Or the object could move towards a wall or outside of the domain if the barycenter is not fixed. Therefore, the constraints considered here, are formulated as

$$\int_{\Omega} (x + u) \det(DF) \, dx = 0, \quad (1.9)$$

$$\int_{\Omega} \det(DF) - 1 \, dx = 0. \quad (1.10)$$

Where (1.9) and (1.10), are the barycenter and volume constraints respectively. The constraints are used in their volume form, as in [54], although it is also possible to use them in their boundary form, see [32]. The volume was preferred over the boundary based on the studies presented in [37], where it is explained that the volume form provides better accuracy in a finite element discretization. As explained [55, 51], in this work it is assumed that the barycenter of the initial domain is the origin

$0 \in \mathbb{R}$ . The volume is considered to be a constant, therefore it can be taken out of the integral (1.9).

Finally, the fundamental problem studied in this work can be formulated as

$$\begin{aligned} \min_{u \in V} & \quad (1.7) \\ \text{s.t.} & \quad (1.8) \\ & \quad (1.9) \\ & \quad (1.10), \end{aligned}$$

where  $V$  is some vector space of admissible deformations, through which the set of admissible shapes is defined.

This gives an abstract overview of the optimization problem used within this thesis to benchmark the proposed methods. As explained in Section 1.1, the adjoint method is used to obtain the reduced problem and the shape derivative. Additionally, it is discussed throughout Chapter 2 that the geometric constraints can be handled via an augmented Lagrangian function. As explained in [55], this has some downsides which lead to an improved strategy in [51, 34].

## 1.4 Numerical methodology

The different implementations in this thesis were done using a parallel-simulation framework, UG4 [81, 80]. Based on MPI communication [47], UG4 is oriented towards applications in large supercomputers. It includes parallel implementations for many iterative solvers, preconditioners, and discretization schemes. In addition, it is possible to extend UG4 through its plugin capabilities, therefore we made use of this to implement the required functionalities. Regarding the generation of geometries, we made use of GMSH [25] and ProMesh [64]. In all cases, triangular and tetrahedral elements were used in 2d and 3d, respectively. The visualization of the results was done with ParaView [1].

We used the finite element method [20, 13] for the discretization of the continuous systems of equations that appear throughout this work. The state equation, i.e. the incompressible, steady-state Navier-Stokes equations, was discretized with a stable  $P_2 - P_1$  scheme, when possible. Given that its adjoint possesses the same structure, the same set of finite elements were used. If required, a pressure-gradient stabilization term was as used given in [18]. Similarly, most of the systems of equations used for the computation of the descent directions were discretized with  $P_1$  Lagrange shape functions.

Since this work deals with applications in a high-performance computing setting, it was important to measure the parallel scalability performance of the proposed algorithms. Standard metrics include the weak and strong scalability [29], which are related to how the computational work behaves when the size of the problem or the number of processors change. If the size of the problem -DoFs- is fixed, an increment in computer processors should speedup the solution up to a theoretical limit. Or if both the size of the problem and computational power increase proportionally, the execution time should theoretically remain constant. Alas, here we were concerned with weak scalability. To achieve this, it was important to appropriately precondition the iterative solvers used in this work. Thus, the geometric

multigrid included in UG4 was used as a preconditioner, taking advantage of its grid-size independent convergence properties [28]. With this, the iterative solvers require an approximately constant number of iterations to solve the involved equation systems. Moreover, the solution time for these systems increases only linearly with the number of DoFs.

Another important algorithmic property is mesh independence [36], where the geometry converges to a similar optimized domain as the discretization is refined. This was also studied in this work by using different levels of refinement for 2d and 3d calculations. For instance, the resulting profile of the obstacle in 2d can be compared across refinements. Or the distance between shapes in 3d can be measured for highly refined grids. With this, also the quality of the mesh was measured in 2d. The reason for this was that the visualization of element degeneracy was evident for the triangular elements. This is accompanied by figures of the computational mesh from critical areas of the domain. Many quality metrics are given in [72] and a good analysis in the context of finite elements is given in [9], both of which were consulted throughout this work. In a sense, the selected mesh quality metrics were used as measure of how the computed descent directions affected the elements of the computational grid.

## Chapter 2

# Synthesis Essay

The main focus of this thesis is the development of algorithms that find descent directions for shape optimization schemes. This was illustrated in line 5 of Algorithm 1, as finding a deformation field using the shape derivative  $J'(\Omega)$ . Particular interest is placed on applications where large deformations are needed to reach an optimized shape, and which can lead to loss of mesh quality in a discrete setting. For this reason, an initial configuration with preexisting geometric singularities was selected as the benchmarking geometry for the used case study. Additionally, this work deals with systems of equations with very large numbers of DoFs. Therefore, the research was carried out using distributed-memory systems, which implied tailoring the different implementations to a parallel setting.

This chapter presents the results of three research articles published (or submitted for publication) as part of the work of this thesis:

- [55]: Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners
- [51]: A Scalable Algorithm for Shape Optimization with Geometric Constraints in Banach Spaces
- [34]: Shape optimization in  $W^{1,\infty}$  with geometric constraints: a study in distributed-memory systems

Each of these publications presents a different approach to finding a descent direction from the shape derivative. The case study described in Section 1.3 is used in all articles to do a performance benchmark of the proposed techniques.

The outline is as follows. In Section 2.1, the nonlinear extension equation approach presented in [55] is discussed. It is based on the previous work found in [32, 54], and expands the problem's formulation to allow for a self-adapting scheme which can identify domain regions where large deformations are required. In Section 2.2, the topic of Banach spaces is addressed by presenting the work done in [51], where the descent directions are found in  $W^{1,p}$  as an approximation to  $W^{1,\infty}$ . In this work, the  $p$ -Laplace relaxation is combined with the geometric constraints described in Section 1.3. An algorithm is proposed to generate shape iterates that fulfill  $g$  on every step. Emphasis is placed on achieving a simpler formulation of the optimization problem. Finally, Section 2.3 discusses the methodology presented in [34] for descent directions in  $W^{1,\infty}$ . This combines the strategy in [51], which is used to handle the geometric constraints, with the ADMM formulation given [15]. The proposed methodology allows for large deformations, without pronounced negative impacts on the convergence of the iterative solvers nor on the mesh quality.

The following sections are each composed of a theoretical description, highlighting the meaningful aspects of each optimization method, which is followed by a selection of meaningful results which illustrate the capabilities of each respective approach.

The developed techniques are compared at the end of this chapter. The complete articles are included as part of this thesis.

## 2.1 Nonlinear extension equation

### 2.1.1 Background

This section covers the work presented in [55], where a nonlinear elliptic PDE is used to find a descent direction in a Hilbertian framework. It is based on the common approach [24, 70] of relating the shape derivative,  $J'(\Omega)u$ , to a bilinear product. For instance, with  $u \in H$ , where  $H$  is a suitable Hilbert space and typically  $H \subset H^1$ . Thus, letting

$$a(u, \delta_u) = J'(\Omega)\delta_u \quad \forall \delta_u \in H, \quad (2.1)$$

a descent direction -i.e., deformation field- can be obtained by solving the resulting system of equations. Then the shape derivative evaluated in the direction  $\delta_u$  can be understood as the right-hand side vector to an operator. In the extension equation approach, a deformation field over the complete domain  $\Omega$  is related to a control variable defined over the surface of the object  $\Gamma_{\text{obs}}$ . This addresses the issues associated with loss of mesh quality when the deformation field is applied only over  $\Gamma_{\text{obs}}$  and the surrounding mesh either becomes highly degenerate or remeshing is required.

The mathematical argumentation can be found in [32], where a linear extension operator  $S$  was proposed and where several choices of  $S$  are explored, for instance by combining the solution of the Laplace-Beltrami operator to that of an elliptic equation. This approach is based on having a control variable defined over the surface of the obstacle,  $c \in L^2(\Gamma_{\text{obs}})$ . The boundary control is imposed as a Neumann boundary condition for the extension operator  $S$ , whose solution links  $c$  to a deformation field across the whole domain  $u : \Omega \mapsto \mathbb{R}$ .

Unlike the optimization scheme presented in Algorithm 1, which is based on the generation of shape iterates by computing a sequence of deformation fields and applying them over the geometry, the method in this section is based on the method of mappings [52, 74, 61]. It reformulates the problem over the reference domain  $\Omega_0$  by using a transformation  $F$  to pull back the deformed configuration back to the reference domain. The transformation is defined as  $F = \text{id} + u$ , the perturbation of identity (1.3), and  $u$  is found as the solution to the operator  $S(c, \Omega)$ .

The shape optimization problem is now understood as finding an optimal transformation  $F$  in a set of admissible transformations  $\mathcal{F}_{\text{adm}}$ , so that a deformation from reference to optimized configuration is obtained. The optimization is done without explicitly deforming the domain on every iteration of the outer loop of Algorithm 1.

The optimization problem has to be enriched by the nonlinear constraint

$$\det(DF) \geq b \quad (2.2)$$

to guarantee local injectivity, with  $b > 0$ . Fulfilling this constraint allows for the reference domain to be mapped to a Lipschitz domain by the transformation  $F$ . Opposite to the iterative approach, here it is guaranteed that a sequence of deformations  $\tilde{F}_k := F_k \circ \dots \circ F_n$  does not violate condition (2.2)

Some of the caveats of a linear extension operator are studied in [54]. Mainly that, under large deformations, a linear  $S$  might be too restrictive. And that combined with (2.2), the set of admissible transformations might limit the reachable optimal shape. With the intention of extending the set of admissible shapes  $\mathcal{S}_{\text{ad}}$ , a nonlinear equation was proposed in [54], where a nonlinear term  $\eta(u \cdot \nabla)u$  was

added to  $S$ . It is described as a nonlinear advective term, which promotes node displacements along large gradients. In this way, the compression provoked by the deformations on the direction normal to the surface of the obstacle is avoided. The advection's influence is controlled by the constant scalar  $\eta$ .

As part of this thesis, a nonlinear extension operator with variable nonlinearity control is proposed [55]. The scalar field  $\eta \in L^2(\Omega)$  thus plays the role of a control for the amount of nonlinearity in the deformation field per element. It is introduced as another control variable in the optimization problem. Within the context of removal and creation of singularities,  $\eta$  helps to identify the regions which require larger deformations. Additionally, through this functionality, the extension equation can self-adapt to the reference domain by promoting large deformations through an element-specific value of  $\eta$ .

The self-adapting, nonlinear extension operator  $u = S(\eta, c, \Omega)$  is defined to be the solution of the PDE

$$\int_{\Omega} (Du + Du^T) : D\delta_u + \eta(Du u) \cdot \delta_u dx = \int_{\Gamma_{\text{obs}}} cn \cdot \delta_u ds \quad (2.3)$$

given in variational form. Where  $u, \delta_u \in W$ , with  $W := \{u \in H^1(\Omega, \mathbb{R}) : u|_{\partial\Omega \setminus \Gamma_{\text{obs}}} = 0 \text{ a.e.}\}$ . This is added as another constraint to the minimization problem.

For this approach, the optimization problem is formulated as

$$\begin{aligned} \min_{(c, \eta) \in L^2(\Gamma_{\text{obs}}) \times L^2(\Omega)} \quad & J_{\text{aug}} := j(y, F(\Omega)) + \frac{\alpha}{2} \|c\|_{L^2(\Gamma_{\text{obs}})}^2 \\ & + \frac{\theta}{2} \left\| \eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}}) \right\|_{L^2(\Omega)}^2 + \frac{\beta}{2} \left\| [b - \det(DF)]^+ \right\|_{L^2(\Omega)}^2 + \tau \|g(u)\|_2^2, \\ \text{s.t.} \quad & e(y, F(\Omega)) = 0 \\ & u = S(\eta, c, \Omega) \text{ as in (2.3)} \\ & F = id + u \\ & \eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}} \text{ in } \Omega \\ & g(u) = 0, \end{aligned} \quad (2.4)$$

where  $\alpha$  is a regularization parameter,  $\theta$  is the penalty factor associated to the box constraints on  $\eta$ , and  $\tau$  is the penalty associated to the geometric constraints. The determinant inequality constraint (2.2) is introduced to the objective function via the penalty term  $\beta$ , where  $(\cdot)^+$  denotes the positive part function. This implies the use of a semi-smooth Newton's method, due to the non-differentiability introduced by constraint (2.2).

A descent algorithm is formulated using the sensitivities of the objective function given in (2.4). The reduced gradient must be found with respect to the variables  $(c, \eta)$ , for which the chain of mappings is differentiated as  $(c, \eta) \mapsto u \mapsto y \mapsto J_{\text{aug}}$ . And the sensitivities of  $(c, \eta)$  are found by computing the Lagrange multipliers corresponding to  $u$ . The system of equations follows from the augmented Lagrangian

function

$$\begin{aligned} \mathcal{L}(u, y, c, \eta, \lambda_u, \lambda_y, \lambda_g) = & \frac{\nu}{2} \int_{\Omega} (Dv((DF)^{-1})) : (Dv((DF)^{-1})) \det(DF) dx \\ & \frac{\alpha}{2} \int_{\Gamma_{\text{obs}}} c^2 ds + \frac{\beta}{2} \int_{\Omega} ([\eta_{\text{det}} - \det(DF)]^+) dx + \frac{\eta}{2} \int_{\Omega} [\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}})]^2 dx \\ & + \tau \|g(u)\|_2^2 + \langle \lambda_u, S(\eta, c) \rangle + \langle \lambda_y, e(y, F(\Omega)) \rangle + \langle \lambda_g, g(u) \rangle, \end{aligned} \quad (2.5)$$

where  $\lambda_*$  indicates the respective Lagrange multipliers to each variable. Notice that  $\lambda_g$  starts with non-converged initial values, thus violations of geometric constraints  $g(u)$  at the onset of the simulation are prevented by the penalty term  $\tau$  and altogether avoided once the Lagrange multipliers converge. This is related to Algorithm 1 in [55].

The core the optimization scheme is the solution of the PDE systems. These consist of the derivatives of the Lagrangian (2.5), and they are solved in a block-wise manner. This is shown in Algorithm 2, which yields a new deformation field  $u$ , as well as the updates for  $(\eta, c)$  which define the nonlinear operator  $S$ .

---

**Algorithm 2** Computation of reduced gradient, as in [55]

---

- 1: **function** GRADIENT( $c, \eta, g_{\text{def}}$ )
  - 2:    $c \mapsto u$  solving equation (2.3)
  - 3:    $u \mapsto (y)$  solving the state equation  $e$
  - 4:    $(y, u) \mapsto (\lambda_y)$  solving the adjoint state
  - 5:    $(y, u, \lambda_y, \lambda_g) \mapsto \lambda_u$
  - 6:    $\lambda_g \mapsto (\gamma, \kappa)$
  - 7:   **return**  $(u, \gamma, \kappa)$
  - 8: **end function**
- 

The outer iteration, given in [55, Alg.1], consists of the updates for  $\tau$  and  $\lambda_g$ . The latter update is critical to the fulfillment of the geometrical constraints  $g(u) = 0$ . Their initial, non-constrained values result on a violation of these constraints, until sufficiently converged values are reached.

### 2.1.2 Results

The case study described in Section 1.3 was used to benchmark the 2d simulations in [55]. This was done in a distributed-memory setting. Some fundamental results are presented to illustrate the performance of the optimization process. A grid with 421 888 triangular elements was used.

The nonlinearity control  $\eta$  accumulates over the regions of the domain where large deformations are needed. In this configuration, these regions correspond to the surface of the obstacle, because this is where geometric singularities must be removed or created. In this sense, it can be seen at a later stage of the optimization process, that  $\eta$  adapts via the element-specific values to the given initial geometry. As shown in Fig. 2.1, large deformations are promoted over critical parts of the domain.

It is important to mention that no mesh deformations are performed during the optimization process. The method of mappings allows us to work over a virtual domain defined by the transformation  $F$ . However, the shape iterates can be generated during post-processing. For instance, knowing the optimal values that define

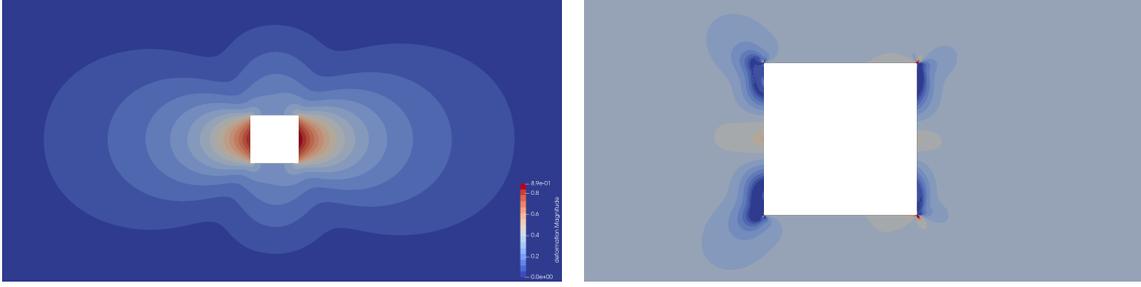


Figure 2.1: Obtained deformation field  $u$  over the reference domain is shown (left), where the associated scalar field  $\eta$  (right) defines the operator  $S$  together with the control variable  $c$ . Taken from [55].

$S(c, \eta)$ , the deformation field could be computed by solving (2.3) over the reference domain and by applying the obtained  $u$  the optimized grid could be generated. This is featured in Fig. 2.1, where the magnitude of the deformation field is shown to have larger values for the points in  $\Gamma_{\text{obs}}$  where the tips must be removed and generated.

Initial steps would show that the self-adapting  $\eta$  targets the corners of the initial configuration, a box, within the first steps. As the simulation progresses, high values can be found at the regions where the front and back are to be generated. However, a lot of computational effort is required to fully smooth out the initial singularities.

The optimized front tip is shown in Fig. 2.2. As mentioned in Section 2.1.1, one of the goals of this approach is to preserve mesh quality. This can be seen as the even distribution of the triangular elements over the surface  $\Gamma_{\text{obs}}$ , as well as in the surrounding domain. It is seen that the elements have not experienced noticeable degeneration, also that no overlapping elements are present. This is related to the constraint (2.2) which preserves local injectivity. Additionally, it is seen that the

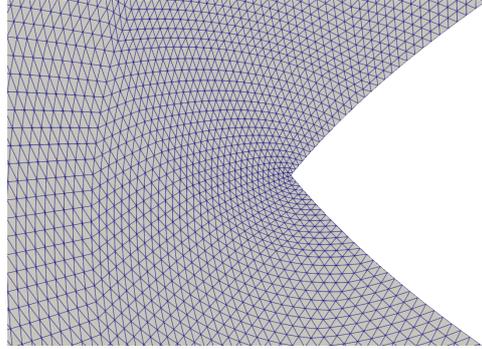


Figure 2.2: Singularity generated at the front of the reference configuration. An even distribution of the elements is observed. Taken from [55].

front tip is generated without compressing the elements that conform it.

An important property is that of mesh independence. As part of our experiments, the optimized obstacle profile  $\Gamma_{\text{obs}}$  was compared over several levels of refinement. The expected result would be that as the discretization is refined, the obtained profiles converge to the same shape. The results are shown in Fig. 2.3 for a series of simulations using a grid refined up to 1 687 552 elements, running for 400 outer loop steps. As mentioned before, these shapes are generated only in post-processing. It can be seen that this approach approximates to the same optimized profile for all

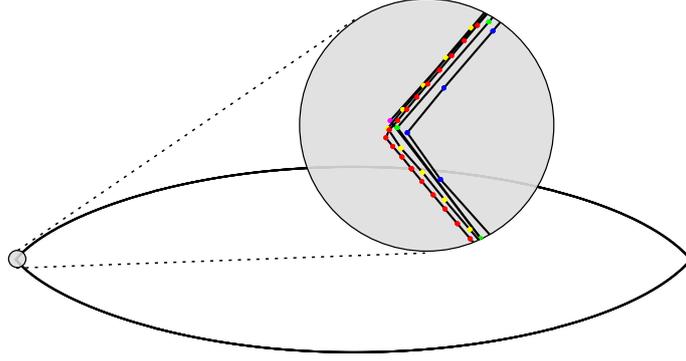


Figure 2.3: Mesh independence for several refinement levels, signaled by different colors over the nodes. A detail of the front tip is given. Taken from [55].

refinement levels for a given number of steps. The magnified tip shows how the singularities appear at approximately the same point over  $\Gamma_{\text{obs}}$ , as well as how the preexisting singularities are removed in the optimized  $\Gamma_{\text{obs}}$ .

A disadvantage of this approach is related to the convergence of the Lagrange multipliers associated to the geometric constraints,  $\lambda_g$ . Given that these are updated via a simple update rule, the constraints  $g$  are not fulfilled on every step. Particularly at the initial steps, the augmented objective function  $J_{\text{aug}}$  in (2.4) does not necessarily decrease.

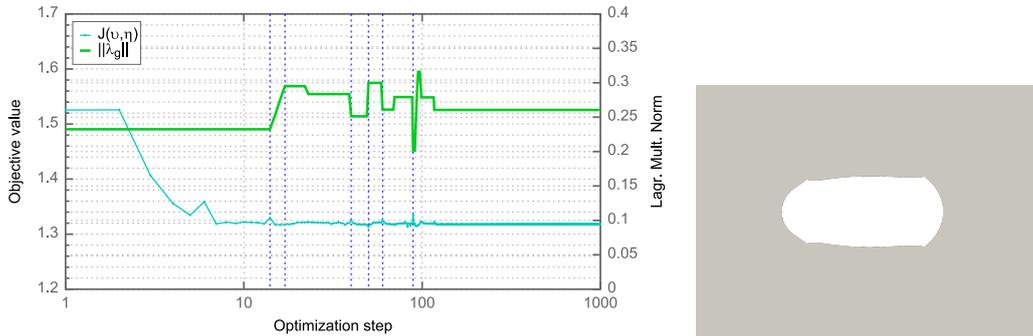


Figure 2.4:  $J_{\text{aug}}$  plot (blue) compared against the norm of the Lagrange multipliers  $\lambda_g$ ,  $\|\lambda_g\|$ . The vertical dashed lines (dark blue), indicate major changes in  $\lambda_g$ . A non-optimal shape (right), related to the initial non-converged values of  $\lambda_g$ . (Left) Taken from [55]. (Right) Taken from [58].

This can lead to a bouncing of the shapes, which can result in reduced performance of the iterative solvers. Figure 2.4 presents this by comparing the value of  $J_{\text{aug}}$  against the Euclidean norm of  $\lambda_g$ ,  $\|\lambda_g\|$ . Some major changes in  $\|\lambda_g\|$  correspond to increments of  $J_{\text{aug}}$ , indicated by the dashed vertical line. However, the objective function and norm of the multipliers converge towards the end of the simulation.

## 2.2 $p$ -Laplace relaxation

### 2.2.1 Background

In this section the work published in [51] is discussed. It is based on a  $p$ -Laplace relaxation problem, which aims at approximating descent directions in the  $W^{1,\infty}$ -topology. An algorithmic approach is presented which makes the implementation less dependent on heuristically-determined parameters. It allows for large deformations without requiring a constraint on the deformation, e.g. by limiting the Frobenius norm of  $\|Du\|$  as in [73] or constraining the determinant of the deformation gradient [32].

This approach addresses some of the shortcomings of the extension equation methodology in Section 2.1. For instance, by leaving the Hilbertian framework in favor of a vector space where the Lipschitz property of transformation is preserved, a constraint on the deformation, for instance (2.2), is no longer needed. It was explained in [32], as well as in Section 2.1, that when condition (2.2) is fulfilled, in the context of the method of mappings, it might limit the set of admissible transformations. This was further investigated in [54], by using a nonlinear extension operator to enable large deformations. Based on the studies carried therein, it was observed that (2.2) limits the set of admissible shapes in such a way that the benchmark, rugby-ball solution to the optimization problem in Section 1.3 might not be reached depending on the value assigned to  $b$  in constraint (2.2). Therefore, it would be profitable to remove a constraint on  $Du$ , while preserving Lipschitz transformations. This could be achieved by finding a descent direction directly on  $W^{1,\infty}$ .

Another benefit is related to the amount of heuristically-determined parameters required for the augmented Lagrangian approach used in Section 2.1. This caveat in the formulation given in [55] is evidenced by the high amount of penalty and regularization terms in the augmented Lagrangian (2.5). In which each of these parameters must be determined for a given geometry. The determination of all these parameters, not only consumes computational budget, but also can have a negative effect on the set of admissible shapes, as previously described. It is then of interest to find a formulation which requires a minimal amount of such parameters, or which does not have them at all.

A similar concern is that of the convergence of the Lagrange multipliers associated to the geometric constraints. As shown in Fig. 2.4, non-converged multipliers might lead to shapes that do not necessarily fulfill  $g(u) = 0$ . Starting from an arbitrary initial guess, the convergence of the multipliers is usually slow. Additionally, the step-size could also play a negative role on the convergence of the multipliers. In this line of thought, this work aims at proposing a better methodology, where the convergence of the geometric constraints is incorporated to the convergence of the descent direction at the possible cost of an increase in computational work.

The  $p$ -Laplace relaxation was presented in [39] as an approximation for the limit case  $p = \infty$  case of the variational problem for the functional  $I_\infty = \int_\Omega f(x)u(x)dx$  where  $\{u \in W_0^{1,\infty}(\Omega), \|Du\|_\infty \leq 1\}$ . Moreover, the work presented here is based on the novel methodology for PDE-constrained shape optimization with  $W^{1,\infty}$  descent

directions presented in [14, 50]. The problem under consideration is

$$\begin{aligned} \min_{u \in V_0^p} \quad & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} dx + J'(\Omega) u \\ \text{s.t.} \quad & g(F(\Omega)) = 0, \\ & F = \text{id} + u, \end{aligned} \tag{2.6}$$

where the vector space  $V_0^p$  is defined as

$$V_0^p = \{u \in W^{1,p}(\Omega, \mathbb{R}^d) : u = 0 \text{ a.e. on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}\}. \tag{2.7}$$

The relaxation term, the first integral term in problem (2.6), falls back to the Hilbert space framework for  $p = 2$ . As before, the transformation  $F$  stands for the perturbation of identity and  $g(F(\Omega))$  for the geometric constraints (1.9) and (1.10). The shape derivative is computed with the Lagrange multiplier method, by taking the derivative with respect to  $u$  and the limit as  $u \mapsto 0$ . As explained in Section 1.1,  $J'(\Omega)u$  stands for derivative of the reduced problem, so its computation requires the solution of the state and adjoint equations.

The solution of the optimization problem follows from the Lagrangian

$$\begin{aligned} L(u, \lambda_g) = & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} dx + J'(\Omega) u \\ & + \sum_{i=1}^d \lambda_{g,i} \int_{\Omega} (x_i + u_i) \det(DF) dx + \lambda_{g,d+1} \int_{\Omega} \det(DF) - 1 dx, \end{aligned} \tag{2.8}$$

which is used to derive the optimality system.

The optimality system consists of the derivatives with respect to  $u$  and  $\lambda_g$ . The resulting nonlinear system is solved using Newton's method, for which the linearization must be computed. This leads to an equation system of the form

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta \lambda_g \end{pmatrix} = \begin{pmatrix} r_u \\ r_{\lambda_g} \end{pmatrix} \tag{2.9}$$

given in discrete form using, as in [51], the abbreviate notation for readability

$$\begin{aligned} A\delta_u &:= \frac{\partial^2}{\partial u^2} L(u^k, \lambda_g^k)(\mu_u, \delta_u) & \forall \mu_u \in V_{0,h}^p \\ B\delta_{\lambda_g} &:= \frac{\partial}{\partial u \, d\lambda_g} L(u^k, \lambda_g^k)(\mu_u, \delta_{\lambda_g}) & \forall \mu_u \in V_{0,h}^p \\ B^T\delta_u &:= \frac{\partial}{\partial \lambda_g \, \partial u} L(u^k, \lambda_g^k)(\delta_u, \mu_{\lambda_g}) & \forall \mu_{\lambda_g} \in \mathbb{R}^d \\ r_u &:= -\frac{\partial}{\partial u} L(u^k, \lambda_g^k) \mu_u & \forall \mu_u \in V_{0,h}^p \\ r_{\lambda_g} &:= -\frac{\partial}{\partial \lambda_g} L(u^k, \lambda_g^k) \mu_{\lambda_g} & \forall \mu_{\lambda_g} \in \mathbb{R}^d. \end{aligned} \tag{2.10}$$

The space  $V_{0,h}^p$  is the discrete approximation of  $V_0^p$ .

From the structure seen in the discrete system (2.9), a saddle point problem has to be solved. Therefore (2.9) is reformulated in terms of the Schur complement

operator  $S := -B^T A^{-1} B$ , which is used to obtain the increment  $\delta_{\lambda_g}$ . After this, the computation of  $\delta_u$  follows from solving  $(-B^T A^{-1} B)\delta_{\lambda_g} = (r_{\lambda_g} - B^T A^{-1} r_u)$  for  $\delta_{\lambda_g}$ . As mentioned before, additional computational effort is needed for this implementation, because a numerical solve is used to represent  $A^{-1}$  in the computation of  $S$  with appropriate right-hand-side vectors, see [51].

The core of the optimization method is the computation of the descent directions, carried out as shown in Algorithm 3. This process corresponds to line 5 of Algorithm 1. It is worth noticing that the shape derivative  $J'(\Omega)$  is included in the derivative with respect to  $u$  of the Lagrangian (2.8), the defect  $r_u$ .

The optimization process used to find a descent direction using the  $p$ -Laplace relaxation is given in Algorithm 3. Because the optimality system is highly nonlinear, a very good initial guess is needed for a given value of  $p$ . This is obtained by solving the nonlinear optimality system via Newton's method for increasing values of  $p$ , up to a preset  $p_{\max}$ . Each successive solve, uses the computed  $u_p$  as initial guess. This follows an increase rule  $p_k = p_{\text{init}} + k p_{\text{inc}}$ , for a preset  $p_{\text{inc}}$ . The increment of  $p$  is chosen so that a good initial guess is obtained for the following value of  $p$ . Starting at an initial value of  $p_{\text{init}} = 2$ , Newton's method is called for each update of  $p$ , and within it the Schur complement system (2.9) is solved. The loop finishes when the maximum value of  $p$  is reached, and the descent direction is used as a deformation field to generate a new shape iterate.

---

**Algorithm 3**  $p$ -Laplace Descent Direction, as in [51]

---

**Require:**  $p_{\max}$

- 1:  $p \leftarrow 2$
  - 2:  $\bar{u} \leftarrow 0$
  - 3: **while**  $p < p_{\max}$  **do**
  - 4:    $(u_p, \lambda_g) \leftarrow \text{NewtonSolver}$ , as in [51, alg. 3.2]
  - 5:    $\bar{u} \leftarrow u_p$
  - 6:   Increase  $p$
  - 7: **end while**
  - 8: **return**  $(u_{p_{\max}}, \lambda_g)$
- 

One downside of this methodology is related to the operator  $A$ , i.e. the second derivative of the function (2.8) with respect to  $u$ . The first term of this derivative,

$$\int_{\Omega} (p-2)(Du: Du)^{\frac{p-4}{2}} (Du: D\delta_u)(Du: D\mu_u) \dots,$$

can lead to division-by-zero when  $Du = 0$ . Therefore, in [51] a modification using a Heaviside function and a small number  $\epsilon > 0$  is implemented. This modification affects the linearization, and not the defects, therefore a solution to the original problem is obtained.

Additionally, the described Schur complement strategy implies that a large number of numerical solves is required within the linearization. This is the cost of incorporating the geometric constraints to the descent directions, so that only shapes that fulfill the geometric constraints are obtained. However, because this introduces the advantage of requiring no parameters in the Lagrangian, no computational time is spent in tuning or determining appropriate values. The advantages in the implementation can be seen by comparing the Lagrangian function (2.8) to the augmented Lagrangian function (2.5) of the extension equation approach given in Section 2.1.

Moreover, since all the shape iterates fulfill the geometric constraints, no shape overshoots occur.

### 2.2.2 Results

The capabilities of the  $p$ -Laplace relaxation approach are discussed via some meaningful results originally presented in [51]. For the case study described in Section 1.3, 2d and 3d simulations were performed in a distributed-memory system. The results illustrate how the  $W^{1,p}$  descent directions remove the preexisting singularities, while fulfilling the geometric constraints and preserving mesh quality. In this work, Algorithm 3 was configured with  $p_{\max} = 4.8$  in 2d and  $p_{\max} = 4.1$  in 3d. These values were chosen based on the studies carried out in [50], where  $p \leq 4$  is given as a rule of thumb. Additionally,  $p_{\max} = 4.8$  provided good mesh quality results in the 2d, cf. Table 2.1. For the 3d case studies, the selected  $p_{\max} = 4.1$  was a good compromise between the values used in the literature and the administration of the assigned computational time within the project. For all cases  $p_{\text{inc}} = 0.19$  was used, which allowed for the convergence of Newton’s method by providing a good initial guess up to  $p_{\max}$ .

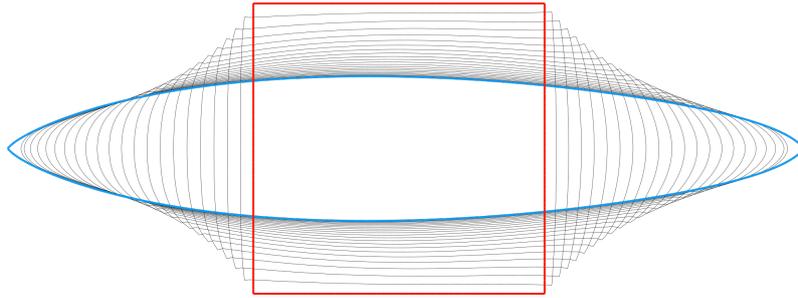


Figure 2.5: Deformation sequence for a 2d simulation, the reference configuration (red) is compared to the optimized domain (blue). Taken from [51]

The workflow follows that of Algorithm 1, where a series of shape iterates is generated based on the computed descent direction. Unlike the approach in [55], here the reference configuration changes on every optimization step, not in post-processing. Based on the discussion in Section 2.2.1, it is expected that the application of the deformation results in only shapes that fulfill  $g(F(\Omega)) = 0$ .

This is visualized in Fig. 2.5 by overlapping the reference configuration with the optimized  $\Gamma_{\text{obs}}$ , together with the intermediate shape iterates. In Fig. 2.5 this is shown for a 2d computational mesh with 70 656 triangular elements. The creation and removal of the geometric singularities is shown as a process that does not result in shape overshoots, as some of the shape iterates generated by the nonlinear extension approach. As a comparison, see the right-hand side figure given in Fig. 2.4, for which the geometric constraints are not necessarily fulfilled.

A detailed view of the singularity smoothing process for several levels of refinement is shown in Fig. 2.6. The computational grid around the upper-left corner of the reference configuration is shown. The several levels of refinement show that the smoothing process is similar across all grids. In order to remove a preexisting singularity, large deformations are required. As previously discussed, this can present several disadvantages, for instance the loss of mesh quality or the overlapping of

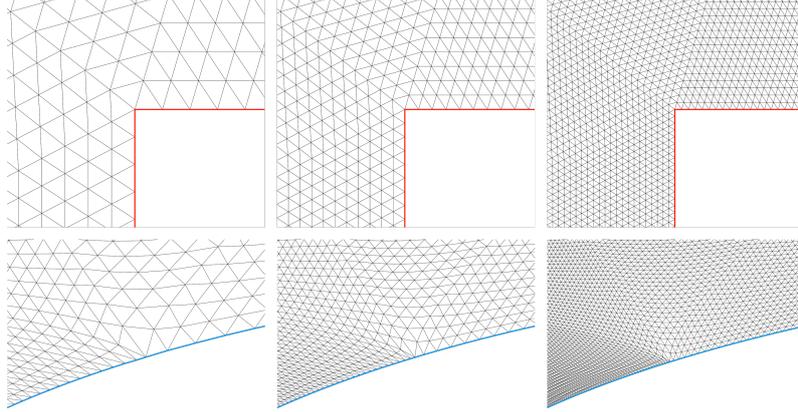


Figure 2.6: Removal of the upper-left corner of the reference configuration for  $\{4, 5, 6\}$  levels of refinement. Taken from [51].

individual elements. The stretching experienced by the mesh is clearly visible on the coarsest level in Fig. 2.6. Finer meshes display similar deformations of the elements that form the corner. It is seen that, even under large deformations, the elements are not highly degenerate, and that at the end of the optimization process, the singularities are completely removed.

Similar results were obtained in 3d, shown in Fig. 2.7. It follows the process described in Algorithm 1, where the shape is deformed on each step of the outer loop. Here a grid with 4 980 736 tetrahedral elements was used. The surface of the obstacle was discretized by 49 152 triangular elements. The simulation was set to run for a 100 steps. It can be seen how the initial singularities are removed as the object is stretched in the direction of the flow. All the shape iterates fulfill the geometrical constraints. As the preexisting edges are smoothed out, the front and back tips are generated. Already at step 50, the rugby ball shape is evident. Finally, at step 100 the tips are completely formed and the edges and corners of the reference configuration are fully removed.

A mesh quality study in 2d was performed for the optimized domain. Although the optimization scheme allows for a preservation of mesh quality at each step, it does not improve on the quality of the initial grid -i.e., the original mesh. Therefore, for the triangular elements in the mesh, the radius ratio between the initial configuration,  $\rho_0 = 1.468$ , and the optimized grid is computed,  $\frac{\rho_\infty}{\rho_0}$ . The data in Table 2.1

Refs	Elements	Minimum angle	Maximum angle	Radius ratio $\rho_\infty$	$\frac{\rho_\infty}{\rho_0}$
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 2.1: Mesh quality measurements across several levels of refinement in 2d. The minimum and maximum interior angles, largest radius ratio, and ratio of largest to initial ( $\rho_0$ ) are shown for the optimized shape. Taken from [51].

reflects the approximation of  $p = \infty$  in terms of the mesh quality. In this way, the deformations with vector fields  $u \in W^{1,p}$  result in a mesh preserving deformation at the end of the optimization process, even under the large deformations experienced by the grid.

One of the disadvantages of Algorithm 3 is related to the amount of computa-

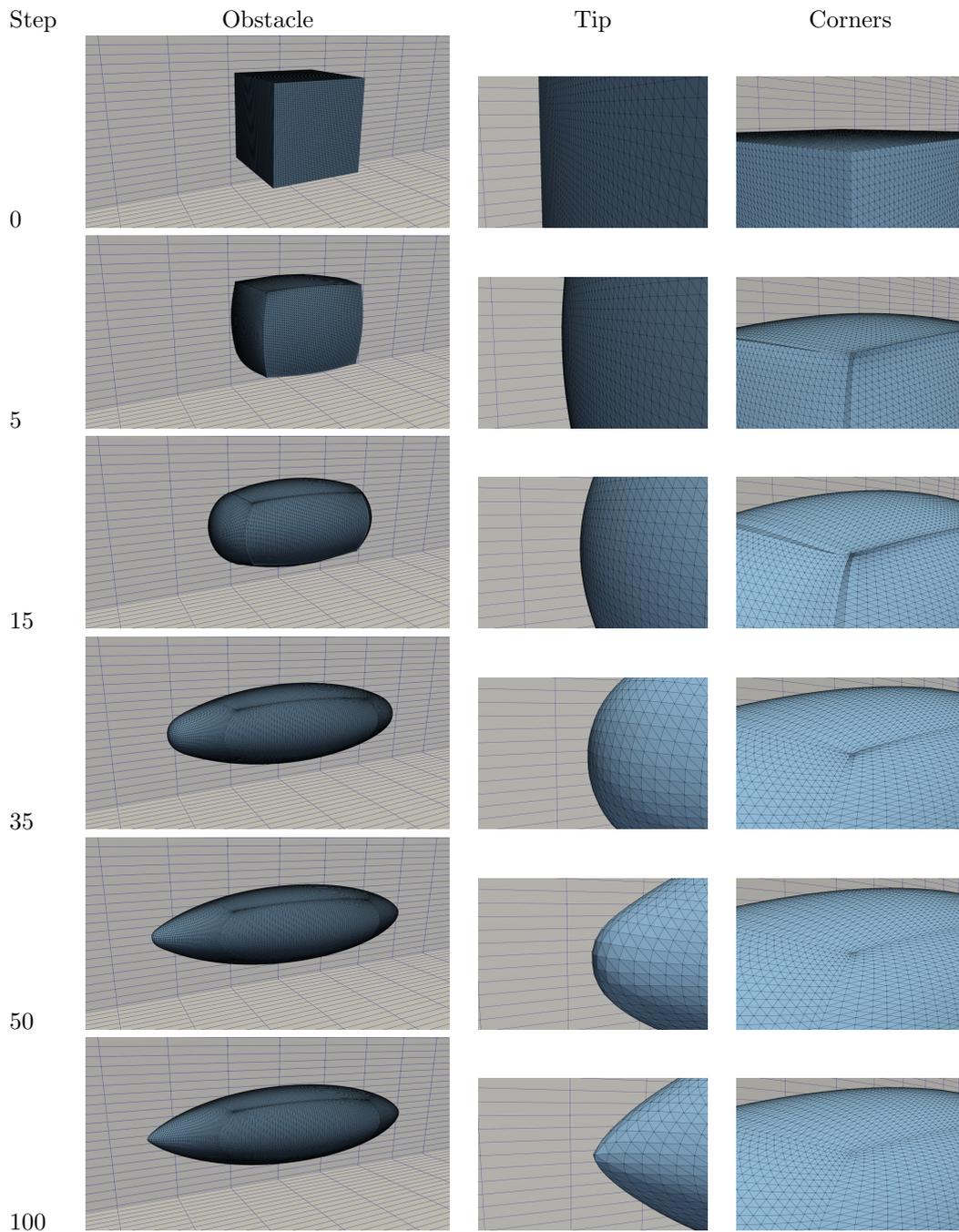


Figure 2.7: Deformation sequence for a 3d example. The surface of an object is deformed until an optimized shape is reached. Several optimization steps are shown. Taken from [51].

tional work. It was mentioned that this simpler implementation, compared to the Lagrangian (2.5), relies on iterative solves to compute the Schur complement operator via applications of  $A^{-1}$ . For this reason, it is of interest to analyze the linear iterations for the solution of the optimality system for all  $p$  increments up to  $p_{\max}$  across several levels of refinements. This was done in [51] as part of a weak scalability study in 3 dimensions for the first optimization step. These results are presented in Table 2.2.

Procs	Refs	Number of Elements	DoFs	Total.Lin. Its.
512	2	77 824	44730	2080
4096	3	622 592	334 158	2458
32 768	4	4 980 736	2 581 014	2606
262 144	5	39 845 888	20 283 942	2912

Table 2.2: Linear iterations within Algorithm 3. Includes all  $A^{-1}$  required to compute  $u$  and  $\lambda_g$ . Results computed for a 3d simulation with  $p = 4.1$ . Taken from [51].

In Table 2.2 it is taken into account that Algorithm 3 is called thirteen times, moreover Newton’s method in line 4 of Algorithm 3 is called several times across a single step. This implies a call to the linear solver for as many calls to Newton’s method until convergence of the nonlinear solver is reached. As explained in [51], the high demands in computational work illustrate the importance of using preconditioners with grid-size independent scaling. With this purpose, the geometric multigrid is used to allow for an efficient solution of the linear systems. As shown in the last column of Table 2.2, the preconditioner bounds the number of iterations even when the number of DoFs increases by three orders of magnitude.

## 2.3 $W^{1,\infty}$ -descent directions

### 2.3.1 Background

The  $p$ -Laplace relaxation approximates a descent direction in  $W^{1,\infty}$  by calculating  $u \in W^{1,p}(\Omega)$  for a sufficiently large  $p$ . This is necessary to preserve the Lipschitz property of the transformation, which is of interest in terms of mesh quality preservation and shape admissibility. This section is based on the work presented in [34], where the alternate direction method of multipliers (ADMM) is used to obtain a deformation field in  $W^{1,\infty}$ .

The methodology discussed in this section is based on [15], where the ADMM was used in an academic setting for shape optimization. Based on the benchmark case used throughout this work, see Section 1.3, this approach introduces the geometric constraints to the problem formulation. This results in a computationally demanding optimization method, which is solved in a distributed-memory system.

Here, the challenge of finding a solution to the optimization problem

$$\min_{u \in W^{1,\infty}(\Omega, \mathbb{R}^d), \|u\|_{W^{1,\infty}} < 1} J'(\Omega) u \quad (2.11)$$

is addressed. The main interest being that when  $u \in W^{1,\infty}$  with  $\|Du\|_\infty < 1$  the mapping  $F$  in (1.3) is Lipschitz with Lipschitz inverse. However, this is mathematically and numerically difficult. For instance, in [14] it was suggested to solve the problem  $u^* \in \arg \min\{J'(\Omega)u : u \in W^{1,\infty}(\Omega, \mathbb{R}^d), |Du| \leq 1 \text{ a.e. in } \Omega\}$  as the limit case of the  $p$ -Laplace relaxation problem [39].

Thus, building on the work in [14, 15], we propose in [34] a methodology to find a minimizer for the problem

$$\begin{aligned} \min_{u \in V_0^\infty} \quad & J'(\Omega) u \\ \text{s.t.} \quad & g(\Omega(u)) = 0, \\ & \|Du\|_{L^\infty(\Omega)} \leq \sigma. \end{aligned} \quad (2.12)$$

where  $\sigma \in (0, 1)$ . As in previous sections, the geometric constraints are given by  $g$ . These have to be fulfilled on every iteration. The shape derivative  $J'(\Omega)u$  corresponds to the derivative of the reduced problem, as explained in Section 1.1. In contrast to the formulation in [15], where the condition  $|Du| < 1$  is fulfilled with  $|\cdot|$  the spectral norm; here,  $J'(\Omega)u$  does not necessarily equal  $-\|J'(\Omega)\|$ . This discussion, however, is not within the scope of this work.

The algorithmic approach is based on the Lagrangian

$$\begin{aligned} L(u, \lambda_g, q, \lambda) := & J'(\Omega) u \\ & + \sum_{i=1}^d \lambda_{g,i} \int_{\Omega} (x_i + u_i) \det(DF) \, dx + \lambda_{g,d+1} \int_{\Omega} \det(DF) - 1 \, dx \\ & + \frac{\tau}{2} \|Du - q\|_{L^2}^2 + \tau(Du - q, \lambda)_{L^2}, \end{aligned} \quad (2.13)$$

where the two last terms are taken from the methodology given in [15, 7, 6]. Following the same strategy as in [51], a descent-like method can be formulated using the derivatives with respect to  $u$  and  $\lambda_g$ . As mentioned before, the geometric constraints

are fulfilled on every optimization step. With this purpose, algorithm 3.3 from [51] is also used in [34].

The process of finding a descent direction in  $W^{1,\infty}$  is computationally expensive. The use of the ADMM requires that two minimization problems and one update be performed [15]. As part of the Lagrangian (2.13), two variables were introduced. A slack variable  $q \in Q_h$

$$Q_h := \{q \in L^2(\Omega; \mathbb{R}^{d \times d}) : q|_T \in \mathbb{P}^0(T; \mathbb{R}^{d \times d}), |q| \leq \sigma\}, \quad (2.14)$$

and the corresponding Lagrange multiplier  $\lambda$  that brings  $q$  close to  $Du$ . What follows is an alternating minimization of  $L$  with respect to  $q$  and  $u$ , and an update of  $\lambda$  via a simple update norm. These operations correspond to line 4, 5, and 7 from Algorithm 4. The variable  $\tau := 1$  can be seen as a step size, but here its assumed to be constant.

---

**Algorithm 4** Descent Direction in  $W^{1,\infty}$ . Taken from [34].

---

**Require:**  $J'(\Omega), \sigma, \epsilon_2, \epsilon_3, N$

```

1:  $\tau \leftarrow 1$ 
2:  $u_0 \leftarrow 0$ 
3: for  $i = 0, 1, \dots, N$  do
4:   Find  $q \leftarrow \arg \min\{L(u, q, \lambda) : q \in Q_h, |q| \leq \sigma\}$ 
5:   Find  $u \leftarrow \arg \min\{L(u, q, \lambda) : u \in V_h, g(\Omega(u)) = 0\}$ 
6:    $\Delta_\lambda \leftarrow \tau(Du - q)$ 
7:    $\lambda \leftarrow \lambda + \Delta_\lambda$ 
8:    $\Delta_\sigma \leftarrow \sigma - \max(|Du|_{L^\infty})$ 
9:    $\Delta u \leftarrow u - u_0$ 
10:   $u_0 \leftarrow u$ 
11:  if  $(\|\Delta_\lambda\|_{L^2}^2 + \|\Delta u\|_{L^2}^2 < \epsilon_2)$  and  $\Delta_\sigma > -\epsilon_3$  then
12:    if  $\Delta_\sigma > \epsilon_3$  then
13:       $t \leftarrow 2t$ 
14:       $J'(\Omega) \leftarrow tJ'(\Omega)$ 
15:    else
16:      break
17:    end if
18:  end if
19: end for

```

---

It is worth mentioning, as in [34], that the system of equations solved to minimize with respect to  $q$ , can be split into two steps. First, solving a mass matrix system

$$\frac{\partial}{\partial q} L(u, \lambda_g, q, \lambda)(\delta_q - \tilde{q}) = 0 \quad \forall \delta_q \in Q_h. \quad (2.15)$$

It incurs a low computational cost, due to the resulting structure of the matrix, which can be solved for instance with a Jacobi method. Second, a pointwise projection

$$\tilde{q} \mapsto \frac{\tilde{q}}{\max\left(1, \frac{|\tilde{q}|}{\sigma}\right)} =: q. \quad (2.16)$$

The optimality system for the minimization of  $L$  with respect to  $u$  leads to the same saddle point structure as described in Section 2.2.1. Nevertheless, one advantage of this approach is that the second derivative of the Lagrangian (2.13) with respect to  $u$  does not require a modification, because there is no risk of division by

zero. One downside of using Algorithm 4, is the high computational cost associated to calling the loop in lines 3-19 several times in the same optimization step. In each iteration, line 4 requires the call to Newton’s method. Therefore, a high number of linear solves might also be required in proportion to the number  $N$  of times the ADMM loop is called.

### 2.3.2 Results

Special interest is placed here on discussing the results taken from [34], where the  $p$ -Laplace and ADMM methods are compared. As mentioned before, a descent direction in  $W^{1,\infty}$  or an approximation to this space, leads to a Lipschitz transformation  $F$  of the domain. In this work, the optimized shape for the case study Section 1.3 includes the creation of tips and the initial configuration implies that geometric singularities must be removed. As will be observed, the ADMM approach leads to a quick removal and creation of corners and edges by applying large deformations on the very early steps. In contrast, the  $p$ -Laplace formulation is constrained by the need to use small step-size to avoid convergence issues in the the solver used for the highly nonlinear problem.

A direct comparison between the initial five shape iterates of the  $W^{1,\infty}$  and  $W^{1,p}$  is presented in Fig. 2.8. While the initial configuration is the same in both cases, the differences are evident since the first generated shape. It is seen how  $u \in W^{1,\infty}$  allows for a larger deformation and earlier removal of the preexisting corners. The  $W^{1,p}$  directions are smaller in comparison. At the last shown shape iterate,  $\Omega_5$ , the creation of tips is seen for the ADMM, and the corners of the box have been smoothed out almost entirely.

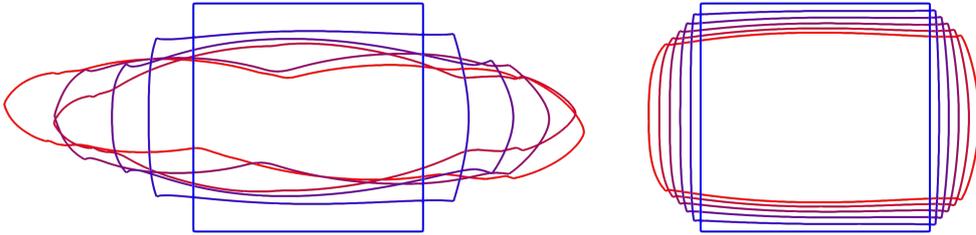


Figure 2.8: The initial five shape iterates for the  $W^{1,\infty}$  (left) and the  $W^{1,p}$  (right) shape optimization schemes. The initial configuration (blue) is deformed by applying the computed deformation field until  $\Omega_5$  (red) is obtained. Taken from [34].

These high deformations can be linked to a reduction of the objective function (1.7). As seen in Fig. 2.9, an aggressive reduction of  $J$  is observed for the  $W^{1,\infty}$  case. The values of  $J$  have been divided by the computed value  $J_0$  on the initial grid  $\Omega_0$ . This shows that more than 15% reduction is achieved with the first computed  $W^{1,\infty}$  deformation field. On the other hand, the  $W^{1,p}$  sequence display a slower reduction of  $J(\Omega)$ .

The larger deformations allowed by the descent directions in  $W^{1,\infty}$  have a direct impact on the mesh quality. In Table 2.3, the quality of the edges that conform the surface of the obstacle  $\Gamma_{\text{obs}}$  is studied. The initial five and last four shape iterates of a simulation, configured to run for 50 iterations, are analyzed by presenting the ratio between the longest and shortest edge length in  $\Gamma_{\text{obs}}$ . This is done across several

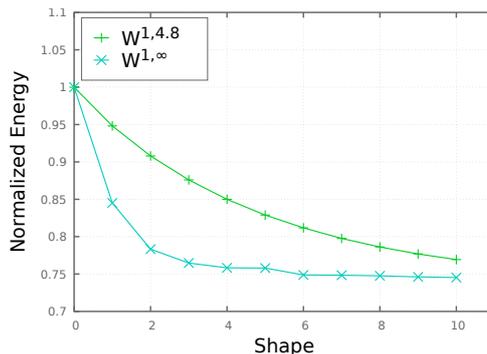


Figure 2.9: Values of the dissipated energy, divided by  $J_0$ , for the initial 10 shape iterates. Taken from [34].

levels of refinement to study the mesh independence of the  $W^{1,\infty}$  approach.

Table 2.3: Ratio between longest and shortest edge length in  $\Gamma_{\text{obs}}$  for different refinement levels. Taken from [34].

	3-refs		4-refs		5-refs	
Triangles in $\Omega$	17 664		70 656		282 624	
Edges in $\Gamma_{\text{obs}}$	128		256		512	
Step	ADMM	PLAP	ADMM	PLAP	ADMM	PLAP
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.89	1.20	1.91	1.23	1.96	1.28
2	3.62	1.42	3.67	1.49	3.61	1.59
3	5.58	1.64	5.27	1.76	6.45	1.91
4	10.54	1.87	10.57	2.03	11.01	2.24
5	10.52	2.09	10.53	2.29	9.71	2.56
⋮						
47	13.49	8.33	14.96	9.56	13.69	8.86
48	13.50	8.51	15.05	9.89	13.64	8.98
49	13.50	8.54	15.03	9.73	13.64	9.09
50	13.50	8.56	15.12	10.17	13.60	9.30

It can be seen that the larger deformations provoke an increase on the edge length ratio. This can be observed in Table 2.3 for the initial shape iterates generated with the ADMM approach, in comparison to the  $p$ -Laplace. However, as shown in [34], the resulting grids do not display large element degeneracy over the surface of the obstacle. Moreover, the large deformations do not prevent the iterative solvers from converging, as shown in the iteration counts in Table 2.4. Given that a small step-size was used for  $J'$  in the  $p$ -Laplace approach, see [51], the large deformations obtained with the ADMM could not be achieved.

The computational cost in terms of the number of iterations for Algorithm 4, and the solvers used therein, is shown in Table 2.4. The average iteration counts for the first five optimization steps are presented for several levels of refinement. The average ADMM iterations refer to the number of times the loop in Algorithm 4 had to run before the convergence conditions in lines 11 and 12 were fulfilled. Within it, Newton's method is used and the average number of calls per step is measured

Table 2.4: Iteration counts for first five steps in a 2d simulation. Average per step number of calls to Algorithm 4, together with the iteration counts for Newton’s method and the linear solver used in the linearization, are presented. Taken from [34].

Procs	Refs	Number of Elements	DoFs	AVG ADMM Its.	AVG Newton Its.	AVG Lin.Its	AVG Time per Shape[s]
48	3	17 664	18 016	31	166	5267	59
192	4	70 656	71 360	37	197	6825	76
768	5	282 624	284 032	25	139	5068	56
3072	6	1 130 496	1 133 312	25	141	5463	65

together with the average number of linear solver iterations needed to solve for the corresponding increment of  $u$  within the linearization. It is seen that the computational workload per step is larger than the reported values for the  $p$ -Laplace approach in [51]. However, the use of adequately preconditioned iterative solvers allows for a bounded number of iterations across several levels of refinement.

## 2.4 Discussion

In this section a comparison between the used optimization schemes is provided. As a reminder, the general line of research was in shape optimization applications where large deformations were required, while preserving some notion of mesh quality at the optimal shape. The overall goal is to obtain an optimized domain in the fastest, most efficient manner possible. To illustrate this goal, a case study that required the removal and creation of geometric singularities was selected and used to benchmark the developed optimization techniques. In this section, the results obtained using this case study are used to draw comparisons between the advantages and disadvantages of each scheme.

The self-adapting nonlinear extension equation is presented in [55] and discussed in Section 2.1. Our research, in comparison to the work in [32, 54], was carried out over a domain as described in Fig. 1.1; and a variable scalar field was used to control the nonlinearity. Thus, it was expected to see larger absolute values of the nonlinearity field over the edges and corners of the surface of the obstacle. The values of  $\eta$  would promote large deformations over the critical regions of the grid. This is shown in Fig. 2.1, where  $\eta$  is plotted over the surface  $\Gamma_{\text{obs}}$ . The largest absolute values correspond to the regions where the tips must be generated and the corners removed. However, a large number of optimization steps was required to reach the optimized shape. This is illustrated in detail in [55], where it can be seen that many steps are needed to completely remove the preexisting corners. Though unexpected, this could be partly related to the highly viscous flow used in our experiments. Additionally, the augmented Lagrangian (2.5) implied that the formulation required a high number of heuristically determined parameters. This was in part related to the augmented Lagrangian approach used to handle the constraints, for instance the geometric constraints and the determinant condition (2.2). Not only did these parameters complicate the implementation, but also they had to be determined by carrying out several test runs in the supercomputer, which depleted our assigned computational budget.

Our research in [51] built on the short comings of the previously described approach. Based on the  $p$ -Laplace relaxation, the proposed algorithmic scheme allowed for approximating descent directions in  $W^{1,\infty}$ , rather than being limited to a Hilbertian framework. Therefore, it was expected to obtain Lipschitz transformations of the domain without requiring a constraint related to the descent direction, e.g. (2.2). Approximating  $W^{1,\infty}$  required large enough values of  $p$ , this was done using the incremental approach for  $p$  described in Algorithm 3. In comparison to the extension equation, a larger number of calls to the iterative solvers per step was expected. On the other hand, the used strategy for handling the geometric constraints allowed for a reduction of the heuristically determined parameters. The downside was that this came at the expense of more computational work, as described in Section 2.2 and [51]. Good results were obtained for 2d and 3d case studies. Compared to the extension equation, less optimization steps were required in 2d to reach an optimized shape where the previously existing geometric singularities are completely removed, see [58, 57], albeit at the cost of more linear iterations per step. Moreover, all the generated shape iterates fulfilled the geometric constraints, which could have contributed to the lower number of steps required before the end of the optimization. One of the unexpected disadvantages was the scaling of the vector  $J'$ , which

was required as a step-size control mechanism, see [51]. The scaling resulted in smaller steps, which could have increased the required number of iterations of the optimization loop. Still, the number of test runs in the supercomputer was certainly decreased by the Lagrangian formulation used in this approach. This enabled us to spend the computational budget on the production runs, with which we obtained valuable results such as the 3d examples presented in Section 2.2.

The topic of descent directions in  $W^{1,\infty}$  using the ADMM was recently explored in [15], on which our research was built. Here we extend the formulation to include geometric constraints, therefore a rather large number of linear iterations were expected for the solution of the equation systems on every optimization step. This was certainly the case, as given in Table 2.4. Larger deformations were expected compared to the  $p$ -Laplacian strategy. This was evident in the comparison of the early steps of the 2d case studies in Fig. 2.8. The prolate shape was reached within the first few steps, leading to a quick reduction of the objective function. On the other hand, it was observed that the mesh quality was affected due to this large deformations. For our selected applications, this did not prevent the iterative solvers from converging throughout the optimization process. One advantage, in terms of the implementation, was that the computation of the second derivative with respect to  $u$  no longer required a modification as in the  $p$ -Laplace. This is related to the fact that in the second derivative of the  $W^{1,\infty}$  formulation, see the Lagrangian (2.13), there are no risks of dividing by zero. This can be seen in [34, Eq. 16]. The technique proved useful for the removal and creation of geometric singularities, and can be considered a good candidate for more complex applications.

It is worth mentioning that our observations on the advantages and disadvantages of each scheme can only be seen as rules of thumb, and not as a definitive set of rules. The different results obtained for each technique were subject to the availability of the computational budget. Because different grids were used across our work, comparing the respective mesh quality results might prove difficult. Still, such an attempt is presented in [34] for the  $p$ -Laplacian and ADMM methods. A comparison of the number of iterations or time-per-step provides some insight regarding the computational costs of each technique. Although, it does not account for the magnitude of the computed deformation field. For instance, this trade-off is described in Section 2.3. The mentioned limitations show that a proper ranking of these methods is not a trivial task. However, some helpful insight for further applications can be obtained from the discussion presented in this chapter.

# Chapter 3

## Publications

The following listed publications are considered as part of this cumulative thesis. They are given with their publication information, as well as with the contributions from the doctoral candidate.

- [55]: Pinzon, J., & Siebenborn, M. (2023). Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners. *Optimization and Engineering*, 24(2), 1089-1113. **Contributions:** Conceptualization (75%), Software (85%), Formal Analysis (80%), Investigation (90%), Writing – Original Draft (85%), Writing – Review and Editing (90%), Visualization (90%)
- [51]: Müller, P. M., Pinzón, J., Rung, T., & Siebenborn, M. (2023). A scalable algorithm for shape optimization with geometric constraints in banach spaces. *SIAM Journal on Scientific Computing*, 45(2), B231-B251. **Contributions:** Conceptualization (70%), Software (90%), Formal Analysis (80%), Investigation (85%), Writing – Original Draft (75%), Writing – Review and Editing (75%), Visualization (85%)
- [34]: Herbert, P. J. , Pinzon Escobar, J., & Siebenborn, M. (2023). Shape optimization in  $W^{1,\infty}$  with geometric constraints: a study in distributed-memory systems. ArXiv preprint, submitted. **Contributions:** Conceptualization (75%), Software (90%), Formal Analysis (90%), Investigation (80%), Writing – Original Draft (75%), Visualization (90%)

This section is composed by the mentioned articles in their published or pre-print form.

# PAPER I



# Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners

Jose Pinzon<sup>1</sup> · Martin Siebenborn<sup>1</sup>

Received: 18 August 2021 / Revised: 15 March 2022 / Accepted: 23 March 2022 /  
Published online: 9 April 2022  
© The Author(s) 2022

## Abstract

In this article we propose a scalable shape optimization algorithm which is tailored for large scale problems and geometries represented by hierarchically refined meshes. Weak scalability and grid independent convergence is achieved via a combination of multigrid schemes for the simulation of the PDEs and quasi Newton methods on the optimization side. For this purpose a self-adapting, nonlinear extension operator is proposed within the framework of the method of mappings. This operator is demonstrated to identify critical regions in the reference configuration where geometric singularities have to arise or vanish. Thereby the set of admissible transformations is adapted to the underlying shape optimization situation. The performance of the proposed method is demonstrated for the example of drag minimization of an obstacle within a stationary, incompressible Navier–Stokes flow.

**Keywords** Aerodynamic shape optimization · Method of mappings · Geometric multigrid · Parallel computing

**Mathematics Subject Classification** 35Q93 · 49Q10 · 65M55 · 65Y05 · 65K10

## 1 Introduction

PDE-constrained shape optimization is a mathematical tool to obtain an optimal contour for a randomly-shaped obstacle subject to physical phenomena described by a partial differential equation (PDE). This is achieved by the evaluation of sensitivities of a shape functional  $j(y, \Omega)$ , which depends on the state variable  $y$  and

---

✉ Jose Pinzon  
jose.pinzon@uni-hamburg.de

Martin Siebenborn  
martin.siebenborn@uni-hamburg.de

<sup>1</sup> University Hamburg, Bundesstr. 55, 20146 Hamburg, Germany

the domain  $\Omega$ . The set of admissible shapes,  $G_{\text{adm}}$ , will be implicitly defined within the method of mappings by the set of admissible transformations. The functional is constrained by one or several PDEs, among them a state equation  $E(y, \Omega) = 0$ , which is fulfilled by the state variable  $y$ . In this article, we focus on the incompressible flow case, which is described by Navier–Stokes equations. Here the objective is to minimize the energy dissipation around an obstacle in terms of its shape and additional geometrical constraints. Building on a long history ranging back several decades (see for instance Jameson 2003; Giles and Pierce 2000; Mohammadi and Pironneau 2010), the field of shape optimization with fluid dynamics applications is still very active today following various approaches (e.g. Schmidt et al. 2013; Müller et al. 2021; Garcke et al. 2016; Fischer et al. 2017). For an overview of shape optimization constrained by PDEs we refer the reader to Sokolowski and Zolesio (2012), Allaire et al. (2021), Delfour and Zolésio (2001). The iterative application of deformation fields to a finite element mesh can lead to distortions and loss of mesh quality, as studied by Dokken et al. (2019), Etling et al. (2018), Iglesias et al. (2018), Blauth (2021). This becomes particularly disruptive for numerical algorithms if there are large deformations leading from the reference domain to the optimal configuration. Several approaches, especially in recent studies, have been proposed to prevent this. For instance, the use of penalized deformation gradients in interface-separated domains helps maintain mesh quality but might still lead to element overlaps when taken to the limit Siebenborn and Vogel (2021). Other approaches rely on remeshing the domain, as for instance in Wilke et al. (2005). More recent efforts on this area make use of pre-shape calculus to allow for the simultaneous optimization of both the shape and mesh quality of the grid Luft and Schulz (2021a).

Although the variable of the optimization problem is only the contour of the shape, the surrounding space plays a crucial role since it describes the domain for the physical effects. Due to the Hadamard–Zolésio structure theorem (see for instance Sokolowski and Zolesio 2012) changes of the objective function under consideration can be traced back to deformations of the shape, which are orthogonal to its boundary. This has been recognized as a source for decreasing mesh quality and is addressed by many authors, for instance by also allowing displacements tangent to the shape surface (cf. Luft and Schulz 2021b). In contrast to the statement of the structure theorem, from a computational point of view, it can be favorable to extend surface deformations into the entire surrounding domain instead of building a new discretization around the improved shape after each descent step, i.e. avoid remeshing on each new iteration. In recent works it has become popular to reuse the domain around the shape, which describes the domain of the PDE, for the representation of Sobolev gradients (e.g. Schulz et al. 2016). Typically, elliptic equations are solved in this domain in order to represent the shape sensitivity as a gradient with respect to a Hilbert space defined therein Dokken et al. (2020), Gangl et al. (2015). The benefit of this approach is that the resulting deformation field not only serves as a deformation to the obstacle boundary, but can also be utilized as a domain deformation. Thus, the discretization for the next optimization step is obtained without additional computational cost.

At this point, two different approaches can be distinguished within the context of the finite element method. On the one hand, the computed gradient can be used

directly as a deformation to the domain after each optimization step which can be seen as changing the reference domain iteratively. On the other, in the so called method of mappings Murat and Simon (1976), the reference domain is fixed and the shape optimization problem is interpreted via the theory of optimal control. These are implemented through the definition of a variable around the surface of the shape to be optimized and its connection with the deformation field affecting the whole domain through a so called extension operator. The solution of which results in the optimal deformation field for both the target shape and its surrounding domain. For an application of this method to aerodynamic shape optimization see for instance Fischer et al. (2017).

We can oppose these two approaches as

$$\begin{aligned} \min_{\Omega \in G_{\text{adm}}} j(y, \Omega) & \quad \min_{F \in F_{\text{adm}}} j[y, F(\Omega)] \\ \text{s.t. } e(y, \Omega) = 0 & \quad \text{s.t. } e[y, F(\Omega)] = 0 \end{aligned} \quad (1)$$

where either a set of admissible domains  $G_{\text{adm}}$  or a set of admissible transformations  $F_{\text{adm}}$  has to be defined. A link between these two can then be established via

$$G_{\text{adm}} := \{F(\Omega) : F \in F_{\text{adm}}\} \quad (2)$$

in terms of a given reference configuration  $\Omega$ .

The approach we propose here is based on the research carried out in Haubner et al. (2021). Compared to iteratively updating the shape, it offers the possibility to require properties of the deformation from reference to optimal shape. Moreover, it reformulates the optimization over a set of admissible transformations  $F \in F_{\text{adm}}$ , which enables us to carry out the optimization procedure on the reference domain. Additionally, it has been documented that such extension operators are possible without the need for heuristic parameter tuning. In Onyshkevych and Siebenborn (2021) an additional nonlinear term is introduced to the elliptic extension equation allowing for large deformations while preserving mesh quality, and preventing element self-intersections and degeneration. In shape optimization this occurs when trying to obtain an optimal shape for an obstacle surrounded by media, for which the creation or removal of a singularity on the obstacle's boundary is necessary.

In the present work we focus on applying parallel solvers for the solution of PDEs in large distributed-memory systems. This stems from the fact that the discretizations will lead to a very large number of degrees of freedom (DoFs), for which the application of the geometric multigrid method (see for instance Hackbusch 1985) guarantees mesh-independent convergence on the simulations. Its application within the context of parallel computing towards the solution of PDEs is an area of ongoing research Reiter et al. (2013), Gmeiner et al. (2014), Baker et al. (2011). The feature of mesh-independent convergence is a necessary condition towards weak scalability of the entire optimization algorithm, which is why in this article we apply the multigrid method as a preconditioner for the solution of the PDE constraints. This requires to provide a sequence of hierarchically refined discretizations. However, the shape optimization problem is a fine grid problem, which means that the contour of the obstacle has to be representable within the entire grid hierarchy. This leads to

undesired effects and conceptual challenges that have been addressed for instance in Nägel et al. (2015), Siebenborn and Welker (2017), Pinzon et al. (2020).

The finest grid in the hierarchy typically represents a high-resolution discretization of a polygonal shape, which—besides the aforementioned sources for mesh degeneracy—also introduces challenges to the shape optimization. Given that the grid hierarchy stems from an initially coarse grid, the polygonal shape represented by the computational mesh includes geometrical singularities such as edges and corners. As the base level is refined, these singularities become more pronounced. This is an undesired effect that has a negative impact on the discretization, for instance on the mesh quality, which in turn can impact the convergence of the used numerical solvers. Particularly for fluid dynamics applications this is problematic. If the considered domain transformation is too smooth, i.e. the descent directions are chosen in an inappropriate space, it is not possible to remove or form new singularities. The latter is particularly important when using the geometric multigrid method, due to the hierarchical grid structure.

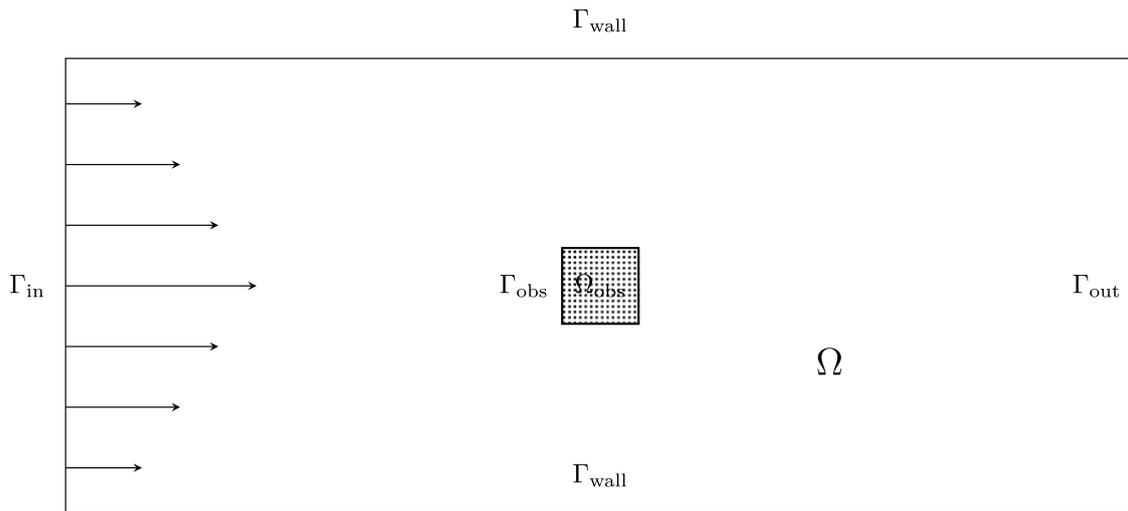
In this work, we propose an approach that is able to identify these regions and adapt the set of admissible transformations  $F_{\text{adm}}$  as part of the optimization problem. In Onyshkevych and Siebenborn (2021) it is illustrated how adding a nonlinear convection term to the extension model that defines  $F_{\text{adm}}$  affects forming singularities in optimal shapes. We thus study in this article how the non-linearity can be adjusted according to the shape of the reference domain.

The rest of this article is structured as follows: In Sect. 2 the optimization problem is formulated and the underlying fluid experiment is outlined. Section 3 is devoted to the optimization algorithm and the computation of the reduced gradient via the adjoint method. Subsequently, in Sect. 4 the performance of the proposed method is discussed by presenting numerical tests. In Sect. 5 the numerical scalability of the method is discussed. The article closes with a conclusion in Sect. 6.

## 2 Problem formulation and mathematical background

The model problem under consideration is sketched in Fig. 1 in a bounded holdall domain  $G := \Omega \cup \Omega_{\text{obs}}$ , where  $\Omega$  is assumed to have a Lipschitz boundary. In  $\Omega$  we consider a stationary, incompressible flow. It surrounds an obstacle  $\Omega_{\text{obs}}$  with variable boundary  $\Gamma_{\text{obs}}$ , but fixed volume and barycentric coordinates. Throughout this article, the original setting of the domain will be referred to as the reference configuration or domain.

In this section we present the theoretical background that culminates with the algorithm presented in Sect. 3. The problem is first formulated in terms of classical shape optimization, to be then reformulated as an optimal control problem. Later on, it is pulled back to the reference domain through the method of mappings. The weak form of the extension operator, presented in this section, is used to formulate the augmented Lagrangian. Finally, the Lagrangian is used to obtain the sensitivities necessary for the descent direction and for the approximation to the Hessian used. For an in-depth discussion of the underlying theory we refer the reader, as previously mentioned, to Onyshkevych and Siebenborn (2021),



**Fig. 1** 2d Holdall reference domain of the flow field with square obstacle

Brandenburg et al. (2009), Haubner et al. (2021), the use of the adjoint method in fluid dynamics can be reviewed in Giles and Pierce (2000), Jameson (2003), and Hinze et al. (2009) can be consulted for a mathematical review of the theory used here.

Let  $X = L^2(\Gamma_{\text{obs}}) \times L^2(\Omega)$ ,  $0 < \eta_{\text{lb}} < \eta_{\text{ub}}$ ,  $b > 0, \alpha > 0, \theta > 0$  and consider the optimization problem

$$\min_{(u,\eta) \in X} j[y, F(\Omega)] + \frac{\alpha}{2} \|u\|_{L^2(\Gamma_{\text{obs}})}^2 + \frac{\theta}{2} \|\eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}})\|_{L^2(\Omega)}^2 \tag{3}$$

$$\text{s.t. } e[y, F(\Omega)] = 0 \tag{4}$$

$$F = \text{id} + \mathbf{w} \tag{5}$$

$$\mathbf{w} = S(\eta, u, \Omega) \tag{6}$$

$$\det(DF) \geq b \quad \text{in } \Omega \tag{7}$$

$$\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}} \quad \text{in } \Omega \tag{8}$$

$$g(\mathbf{w}) = 0. \tag{9}$$

where  $g(\mathbf{w})$  represents geometric constraints. Throughout this work  $D$  denotes the Jacobian matrix, while  $\nabla$  is the Euclidean gradient operator, and  $\text{grad}$  is used for the gradient related to the inner product of the corresponding space.  $S$  denotes an extension operator, which links the boundary control variable  $u \in L^2(\Gamma_{\text{obs}})$  to a displacement field  $\mathbf{w} : \Omega \rightarrow \mathbb{R}^d$ . Examples of possible choices of  $S$  are given and investigated in Haubner et al. (2021). Therein a discussion of the properties of  $S$  which guarantee a certain regularity of  $\mathbf{w}$  can be found, as well as the resulting regularity of the domain transformation. Here we enrich this operator with an additional control

variable  $\eta \in L^2(\Omega)$ , which plays the role of a nonlinearity switch. In the following we assume that  $S$  is defined such that  $\mathbf{w}|_{\Gamma_{in} \cup \Gamma_{wall} \cup \Gamma_{out}} = 0$  almost everywhere.

In the experiment presented here, the geometric constraints require the barycenter and volume to be the origin and constant, respectively. This excludes trivial solutions where the obstacle shrinks to a point or moves towards a position where the objective functional is minimized. Thus, the principal geometric constraints are given by

$$\hat{g}(\mathbf{w}) = \left( \int_{F(\Omega)} 1 \, dx - \int_{\Omega} 1 \, dx, \frac{\int_{F(\Omega)} x \, dx}{\int_{F(\Omega)} 1 \, dx} - \frac{\int_{\Omega} x \, dx}{\int_{\Omega} 1 \, dx} \right)^T \tag{10}$$

This simplifies to

$$g(\mathbf{w}) = \left( \int_{\Omega} \det(DF) - 1 \, dx, \int_{\Omega} F(x) \det(DF) \, dx \right)^T, \tag{11}$$

assuming, without loss of generality, that the barycenter of the reference domain is  $0 \in \mathbb{R}^d$  and the volume is precisely retained.

The condition (7) is approximated via a non-smooth penalty term. This results in an approximation via the objective function

$$J(y, u, \mathbf{w}, \eta) = j[y, F(\Omega)] + \frac{\alpha}{2} \|u\|_{L^2(\Gamma_{obs})}^2 + \frac{\theta}{2} \|\eta - \frac{1}{2}(\eta_{ub} + \eta_{lb})\|_{L^2(\Omega)}^2 + \frac{\beta}{2} \|[b - \det(DF)]^+\|_{L^2(\Omega)}^2. \tag{12}$$

In contrast to the PDE constraints (5), (6), (7), the geometric constraints (9) are fixed dimensional (here it is  $d + 1$  where  $d \in \{2, 3\}$ ). Thus, the multipliers associated to these conditions are not variables in the finite element space but a  $d + 1$ -dimensional vector. This is incorporated into the optimization algorithm in the form of an augmented Lagrange approach. This leads to the augmented objective function

$$J_{aug}(y, u, \mathbf{w}, \eta) := J(y, u, \mathbf{w}, \eta) + \tau \|g(\mathbf{w})\|_2^2, \tag{13}$$

where  $\tau > 0$  is a penalty factor for the geometric constraints and  $\|\cdot\|_2$  refers to the Euclidean 2-norm since  $g(\mathbf{w})$  is finite dimensional. The basic concept of the augmented Lagrange method is to optimize the objective (13). By contrast to a pure penalty method, the geometric constraints (9) are not entirely moved to the objective function, but the corresponding multipliers  $\lambda_g$  are assumed to be approximately known and iteratively updated.

We consider the PDE constraint  $e[y, F(\Omega)]$  to be the stationary, incompressible Navier–Stokes equations in terms of velocity and pressure  $(\mathbf{v}, p)$ . In the following it is distinguished between PDE solutions defined on the reference domain  $\Omega$  denoted by  $(\mathbf{v}, p)$  and on the transformed domain  $F(\Omega)$  as  $(\hat{\mathbf{v}}, \hat{p})$ . We thus consider

$$-\nu \Delta \hat{\mathbf{v}} + (\hat{\mathbf{v}} \cdot \nabla) \hat{\mathbf{v}} + \nabla \hat{p} = 0 \quad \text{in } F(\Omega) \tag{14}$$

$$\operatorname{div} \hat{\mathbf{v}} = 0 \quad \text{in } F(\Omega) \tag{15}$$

$$\hat{\mathbf{v}} = \mathbf{v}_\infty \quad \text{on } \Gamma_{\text{in}} \tag{16}$$

$$\hat{\mathbf{v}} = 0 \quad \text{on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \tag{17}$$

$$\hat{p}n - \nu \frac{\partial \hat{\mathbf{v}}}{\partial n} = 0 \quad \text{on } \Gamma_{\text{out}}, \tag{18}$$

where for compatibility it is assumed that  $\int_{\Gamma_{\text{in}}} \mathbf{v}_\infty \cdot n \, ds = 0$  holds for the inflow velocity profile  $\mathbf{v}_\infty$ . Notice that the boundaries  $\Gamma_{\text{in}}, \Gamma_{\text{out}}, \Gamma_{\text{wall}}$  in (15), (16), (17) are unchanged since the displacement  $\mathbf{w}$  is assumed to vanish here. This assumption reflects that the outer boundaries of the experiment domain are not a variable in the optimization problem.

The variational formulation of the PDE (14) to (18) pulled back to the reference domain  $\Omega$  is given by: Find  $(\mathbf{v}, p) \in V \times Q$  such that for all  $(\delta_{\mathbf{v}}, \delta_p) \in V_0 \times Q$  it holds

$$\int_{\Omega} \nu (D\mathbf{v}(DF)^{-1}) : (D\delta_{\mathbf{v}}(DF)^{-1}) + (D\mathbf{v}(DF)^{-1}\mathbf{v}) \cdot \delta_{\mathbf{v}} - p \operatorname{Tr} (D\delta_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0 \tag{19}$$

$$- \int_{\Omega} \delta_p \operatorname{Tr} (D\mathbf{v}(DF)^{-1}) \det(DF) \, dx = 0, \tag{20}$$

where trial and test functions are chosen in

$$\begin{aligned} V &:= \{ \mathbf{v} \in H^1(\Omega, \mathbb{R}^d) : \mathbf{v}|_{\Gamma_{\text{in}}} = \mathbf{v}_\infty, \mathbf{v}|_{\Gamma_{\text{wall}} \cup \Gamma_{\text{obs}}} = 0 \text{ a.e.} \}, \\ V_0 &:= V \text{ with } \mathbf{v}_\infty = 0, \\ Q &:= \left\{ p \in L^2(\Omega) : \int_{\Omega} p \, dx = 0 \right\}. \end{aligned} \tag{21}$$

In the equations above  $\operatorname{Tr}$  denotes the trace operator,  $:$  the double contraction, and  $\det$  the determinant.

In the experiment considered in this work the physical part of the objective function (12) is given by the energy dissipation in terms of the velocity  $\mathbf{v}$ , thus  $y = (\mathbf{v}, p)$  and

$$\hat{j}(\hat{\mathbf{v}}, F[\Omega]) = \nu \int_{F(\Omega)} D\hat{\mathbf{v}} : D\hat{\mathbf{v}} \, dx, \tag{22}$$

which can be pulled back to the reference domain  $\Omega$  as

$$j(\mathbf{v}, \mathbf{w}) = \nu \int_{\Omega} (D\mathbf{v}(DF)^{-1}) : (D\mathbf{v}(DF)^{-1}) \det(DF) dx. \quad (23)$$

The extension  $S(\eta, u, \Omega)$  is defined to be the solution operator of the PDE

$$\operatorname{div}(D\mathbf{w} + D\mathbf{w}^{\top}) + \eta(\mathbf{w} \cdot \nabla)\mathbf{w} = 0 \quad \text{in } \Omega \quad (24)$$

$$(D\mathbf{w} + D\mathbf{w}^{\top})\mathbf{n} = \mathbf{u}\mathbf{n} \quad \text{on } \Omega_{\text{obs}} \quad (25)$$

$$\mathbf{w} = 0 \quad \text{on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}. \quad (26)$$

Consider the space

$$W := \{ \mathbf{w} \in H^1(\Omega, \mathbb{R}^d) : \mathbf{w}|_{\partial\Omega \setminus \Gamma_{\text{obs}}} = 0 \text{ a.e.} \}. \quad (27)$$

Then the variational formulation of (24), (25), (26) is obtained by: Find  $\mathbf{w} \in W$  such that for all  $\delta_{\mathbf{w}} \in W$  it holds

$$\int_{\Omega} (D\mathbf{w} + D\mathbf{w}^{\top}) : D\delta_{\mathbf{w}} + \eta(D\mathbf{w}\mathbf{w}) \cdot \delta_{\mathbf{w}} dx = \int_{\Gamma_{\text{obs}}} \mathbf{u}\mathbf{n} \cdot \delta_{\mathbf{w}} ds. \quad (28)$$

Finally, we can formulate the approximate optimization problem, which is then solved via the augmented Lagrange approach in Sect. 3, as

$$\min_{(u, \eta) \in X} J_{\text{aug}}(y, u, \mathbf{w}, \eta) \quad (29)$$

$$\text{s.t. (14) to (18) and (29)} \quad (30)$$

$$F = \text{id} + \mathbf{w} \quad (31)$$

$$\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}} \quad \text{in } \Omega \quad (32)$$

$$g(\mathbf{w}) = 0, \quad (33)$$

where the multipliers for conditions (33) are assumed to be known in each iteration.

In order to formulate a gradient-based descent algorithm, we have to compute the sensitivities of the final objective function  $J_{\text{aug}}$  in (13) with respect to the variables  $(u, \eta)$ . This means to differentiate the chain of mappings

$$(u, \eta) \mapsto \mathbf{w} \mapsto (\mathbf{v}, p) \mapsto J_{\text{aug}}(\mathbf{v}, u, \mathbf{w}, \eta) \quad (34)$$

and obtain the sensitivities in reverse order

$$J_{\text{aug}}(\mathbf{v}, u, \mathbf{w}, \eta) \mapsto (\lambda_{\mathbf{v}}, \lambda_p) \mapsto \lambda_{\mathbf{w}} \mapsto (\lambda_u, \lambda_{\eta}). \quad (35)$$

The derivatives of the mappings mentioned in (34) have been omitted here for brevity, but can be found in Onyshkevych and Siebenborn (2021), Haubner et al. (2021). The differentiability of the mapping is discussed in Sect. 3. Access to the adjoint gradient formulation can be obtained via the corresponding Lagrangian, which is given by

$$\begin{aligned}
 \mathcal{L}(\mathbf{w}, \mathbf{v}, p, u, \eta, \lambda_{\mathbf{w}}, \lambda_{\mathbf{v}}, \lambda_p, \lambda_{bc} \lambda_{vol}) &= \frac{\nu}{2} \int_{\Omega} (D\mathbf{v}(DF)^{-1}) : (D\mathbf{v}(DF)^{-1}) \det(DF) \, dx \\
 &+ \frac{\alpha}{2} \int_{\Gamma_{obs}} u^2 \, ds + \frac{\beta}{2} \int_{\Omega} ([\eta_{det} - \det(DF)]^+)^2 \, dx + \frac{\theta}{2} \int_{\Omega} \left[ \eta - \frac{1}{2}(\eta_{ub} + \eta_{lb}) \right]^2 \, dx + \tau \|g(\mathbf{w})\|_2^2 \\
 &+ \int_{\Omega} (\nu [D\mathbf{v}(DF)^{-1}] : [D\lambda_{\mathbf{v}}(DF)^{-1}] + [D\mathbf{v}(DF)^{-1}] \cdot \lambda_{\mathbf{v}} - p \operatorname{Tr} [D\lambda_{\mathbf{v}}(DF)^{-1}]) \det(DF) \, dx \\
 &- \int_{\Omega} \lambda_p \operatorname{Tr} [D\mathbf{v}(DF)^{-1}] \det(DF) \, dx + \int_{\Omega} [(D\mathbf{w} + D\mathbf{w}^T) : D\lambda_{\mathbf{w}} + \eta(D\mathbf{w} \cdot \mathbf{w})] \, dx \\
 &- \int_{\Gamma_{obs}} \mathbf{un} \cdot \lambda_{\mathbf{w}} \, ds + \lambda_{bc} \cdot \int_{\Omega} F(x) \det(DF) \, dx + \lambda_{vol} \int_{\Omega} [\det(DF) - 1] \, dx
 \end{aligned} \tag{36}$$

under the assumption that the barycenter of  $\Omega$  is  $0 \in \mathbb{R}^d$ .

From the Lagrangian (36) the adjoint Navier–Stokes equations follow as: Find  $(\lambda_{\mathbf{v}}, \lambda_p) \in V_0 \times Q$  such that for all  $(\delta_{\mathbf{v}}, \delta_p) \in V_0 \times Q$  it holds

$$\begin{aligned}
 &\int_{\Omega} \nu (D\lambda_{\mathbf{v}}(DF)^{-1}) : (D\delta_{\mathbf{v}}(DF)^{-1}) + (D\delta_{\mathbf{v}}(DF)^{-1}\mathbf{v}) \cdot \lambda_{\mathbf{v}} \\
 &+ (D\mathbf{v}(DF)^{-1}\delta_{\mathbf{v}}) \cdot \lambda_{\mathbf{v}} - \lambda_p \operatorname{Tr} (D\delta_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0
 \end{aligned} \tag{37}$$

$$- \int_{\Omega} \delta_p \operatorname{Tr} (D\lambda_{\mathbf{v}}(DF)^{-1}) \det(DF) \, dx = 0, \tag{38}$$

The adjoint displacement equation is obtained by: Find  $\lambda_{\mathbf{w}} \in W$  such that for all  $\delta_{\lambda_{\mathbf{w}}} \in W$  it holds

$$\int_{\Omega} (D\lambda_{\mathbf{w}} + D\lambda_{\mathbf{w}}^T) : D\delta_{\lambda_{\mathbf{w}}} + \eta(D\mathbf{w} \lambda_{\mathbf{w}}) \cdot \delta_{\lambda_{\mathbf{w}}} \, dx = R(\mathbf{w}, \mathbf{v}, p, \lambda_{\mathbf{v}}, \lambda_p). \tag{39}$$

In (39)  $R$  denotes the derivative of the Lagrangian (36) w.r.t.  $\mathbf{w}$ . This is obtained after straightforward computations and omitted here for the sake of brevity. Finally, the reduced gradient is obtained as: Find  $(\gamma, \kappa) \in X$  such that for all  $(\delta_u, \delta_{\eta}) \in X$  it holds

$$\int_{\Gamma_{\text{obs}}} \gamma \delta_\gamma + \alpha u \delta_u - \lambda_w \cdot n \delta_u \, ds = 0, \quad (40)$$

$$\int_{\Omega} \kappa \delta_\eta + \theta \left( \eta - \frac{1}{2}(\eta_{\text{ub}} + \eta_{\text{lb}}) \right) \delta_\eta - (D\mathbf{w} \cdot \mathbf{w}) \cdot \lambda_w \delta_\eta \, ds = 0. \quad (41)$$

With the sensitivity equations (40), (41) we are now prepared to apply a descent method.

### 3 Optimization algorithm

In Sect. 2 we present the approximate optimization problem (29), (30), (31), (32), (33), which is solved via the augmented Lagrange approach shown in Algorithm 1. An initial guess is given to the Lagrange multipliers  $\lambda_g$  associated to the geometrical constraints. These in turn are iteratively updated in each optimization step subject to the condition that the norm of the defect of the geometrical constraints is smaller than a prescribed tolerance  $\epsilon_g > 0$ . In Algorithm 1 only the problem-dependent parameters have to be provided, which are not the ones immediately related to the extension operator  $S(\eta, u, \Omega)$ . The parameters that define the set of admissible shapes, e.g.  $\eta$ , are simultaneously optimized. This is a significant improvement to previous approaches, where the parameters had to be manually determined, e.g. Blauth (2021), Schulz and Siebenborn (2016).

Most of the computational time is consumed for solving the PDE systems presented in Sect. 2. This is carried out by Algorithm 3 in a block-wise manner, where the output consists of the new displacement field to update the transformation (5), as well as the results of the reduced gradient  $\text{grad } u^{k,\ell}$ ,  $\text{grad } \eta^{k,\ell}$ , which will be further used to obtain updates for the current control and extension factor,  $u^{k,\ell+1}$ ,  $\eta^{k,\ell+1}$ , respectively. A reference for the gradient method can be found for instance in (Hinze et al. 2009, p. 94).

**Algorithm 1** Augmented Lagrange Outer Optimization Algorithm

**Require:**  $\lambda_g, \lambda_{\text{inc}} > 0, 0 \leq \eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}, b > 0, \epsilon_g > 0, \tau > 0, \tau_{\text{inc}} > 0, \epsilon_{\text{outer}} > 0, \epsilon_{\text{inner}} > 0$

```

1: repeat
2:    $\ell \leftarrow 0$ 
3:   repeat
4:      $(w, \text{grad } u^{k,\ell}, \text{grad } \eta^{k,\ell}) \leftarrow \text{REDUCED GRADIENTS}(g_{\text{def}}, u^{k,\ell}, \eta^{k,\ell})$ 
5:      $(u^{k,\ell+1}, \eta^{k,\ell+1}) \leftarrow \text{LBFGS-B}(m, \ell, g_{\text{def}}, \tau, u^{k,\ell}, \eta^{k,\ell}, \text{grad } u^{k,\ell}, \text{grad } \eta^{k,\ell})$ 
6:      $g_{\text{def}} \leftarrow g(w) - g(0)$ 
7:      $\ell \leftarrow \ell + 1$ 
8:   until  $\|(\text{grad } u^{k,\ell-1}, P_{(\eta_{\text{lb}}, \eta_{\text{ub}})}(\eta^{k,\ell-1} - \text{grad } \eta^{k,\ell-1}) - \eta^{k,\ell-1})\|_X < \epsilon_{\text{inner}}$ 
9:   if  $\|g_{\text{def}}\|_2 < \epsilon_g$  then
10:      $\tau \leftarrow \tau_{\text{inc}} \tau$ 
11:   else
12:      $\lambda_g \leftarrow \lambda_g + \lambda_{\text{inc}} g_{\text{def}}$ 
13:   end if
14:    $k \leftarrow k + 1$ 
15: until  $k \geq 1$  and  $\|u^{k,\ell} - u^{k-1,\ell}\|_{L^2(\Gamma_{\text{obs}})} < \epsilon_{\text{outer}}$ 

```

The objective function is non-differentiable due to the presence of the positive part mapping in  $R$  in (39), which is discussed in depth in Haubner et al. (2021). Moreover, a discussion on quasi-Newton methods for semi-smooth objective functions and can be found in Mannel and Rund (2020).

**Algorithm 2** Limited Memory BFGS Algorithm with Box Constraints Inner Iteration

---

```

1: function LBFGS-B( $m, \ell, g_{\text{def}}, \tau, w, u^\ell, \eta^\ell, \text{grad } u^\ell, \text{grad } \eta^\ell$ )
2:   Determine  $\chi_\eta$  according to (43) and  $(\cdot, \cdot)_{\hat{X}}$  according to (45)
3:    $q = (q_1, q_2) \leftarrow (\text{grad } u^\ell, \text{grad } \eta^\ell)$ 
4:   if  $\ell > 0$  then
5:      $s^{\ell-1} \leftarrow (u^\ell, \eta^\ell) - (u^{\ell-1}, \eta^{\ell-1})$ 
6:      $z^{\ell-1} \leftarrow (\text{grad } u^\ell, \text{grad } \eta^\ell) - (\text{grad } u^{\ell-1}, \text{grad } \eta^{\ell-1})$ 
7:     for  $i = \ell - 1, \dots, \max\{\ell - m, 0\}$  do
8:        $\alpha_i \leftarrow \rho_i(s^i, q)_{\hat{X}}$ 
9:        $q \leftarrow q - \alpha_i z^i$ 
10:    end for
11:     $q \leftarrow \frac{(s^{\ell-1}, z^{\ell-1})_{\hat{X}}}{(z^{\ell-1}, z^{\ell-1})_{\hat{X}}} q$ 
12:    for  $i = \max\{\ell - m, 0\}, \dots, \ell - 1$  do
13:       $q \leftarrow q + (\alpha_i - \rho_i(y^i, q)_{\hat{X}}) s^i$ 
14:    end for
15:  end if
16:  return  $(u + q_1, P_{(\eta_{\text{lb}}, \eta_{\text{ub}})}(\eta + q_2))$ 
17: end function

```

---

The use of box-constraints for the extension factor  $\eta$  make it necessary to implement the BFGS method similarly to what can be found in Byrd et al. (1995), from which Algorithm 2 is partly inspired. For the box-constrained limited memory BFGS method, we introduce the indicator function for the condition  $\eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}$  as

$$\chi_\eta(x) := \begin{cases} 1, & \text{if } \eta_{\text{lb}} \leq \eta(x) - \sigma \text{grad } \eta(x) \leq \eta_{\text{ub}} \\ 0, & \text{else} \end{cases} \quad (42)$$

for some small  $\sigma > 0$ . Recall that the canonical inner product on  $X$  is given as

$$X = (u_1, u_2)_{L^2(\Gamma_{\text{obs}})} + (\eta_1, \eta_2)_{L^2(\Omega)}. \quad (43)$$

This is now modified to take the active box-constraints into account by introducing  $\chi_\eta$  into the second term and thereby reducing the integration to the region of inactive constraints

$$\hat{X} = (u_1, u_2)_{L^2(\Gamma_{\text{obs}})} + (\eta_1, \chi_\eta \eta_2)_{L^2(\Omega)} \quad (44)$$

Eq. (44) defines the inner product appearing in lines 8, 11, 13 of Algorithm 2. Whereas, in line 16 the operator  $P$  refers to the projection with respect to the box constraints on  $\eta$ .

Conceptually the optimization scheme presented consists of outer and an inner iterations. The outer iteration, seen in Algorithm 1, updates either  $\lambda_g$  or the penalty factor  $\tau$  by increment factors  $\lambda_{inc}$  and  $\tau_{inc}$ , respectively. In each cycle of the inner loop, a complete optimization is solved using BFGS updates as seen in Algorithm 2.

---

**Algorithm 3** Computation of Reduced Gradient

---

```

1: function GRADIENT( $g_{def}, u, \eta$ )
2:    $u \mapsto w$  via (29)
3:    $w \mapsto (v, p)$  via (20) and (21)
4:    $(v, p, w) \mapsto (\lambda_v, \lambda_p)$  via (40)
5:    $(v, p, w, \lambda_v, \lambda_p, \lambda_g) \mapsto \lambda_w$  via (40)
6:    $\lambda_w \mapsto (\gamma, \kappa)$  via (41) and (42)
7:   return  $(w, \gamma, \kappa)$ 
8: end function
    
```

---

### 4 Shape optimization applications

In this section we present shape optimization applications with the incompressible, stationary Navier–Stokes equations as state equation. The purpose of the featured case studies in this section is to show the application of the algorithm presented in Sect. 3, which includes the effect of the nonlinearity control variable  $\eta$  on the extension operator  $S$ . The obstacle shape deformations demonstrate the algorithm’s capabilities at the detection, smoothing and creation of domain singularities such as tips and edges. Aspects of the multigrid preconditioner’s effects are discussed. Moreover, a grid independence study illustrates that the optimal shape is reached regardless of the number of refinements in the grid hierarchy. The latter result is a fundamental stepping stone towards a scalable parallel implementation of the methodology proposed in Sect. 5.

The flow tunnel is depicted as the holdall domain in Fig. 1 with

$$G_{2d} = (-7, 7) \times (-3, 3) \text{ and } G_{3d} = (-7, 7) \times (-3, 3) \times (-3, 3)$$

for the 2d and 3d cases respectively, taking into account that for the 3d case the obstacle has a spherical shape. Thus, in 2d we have  $\Omega_{obs} = (-0.5, 0.5)^2$  and  $\Omega_{obs} = \{x \in \mathbb{R}^3 : \|x\|_2 < 1\}$  in 3d, respectively.

The boundary conditions at the inflow boundary  $\Gamma_{in}$  are set as

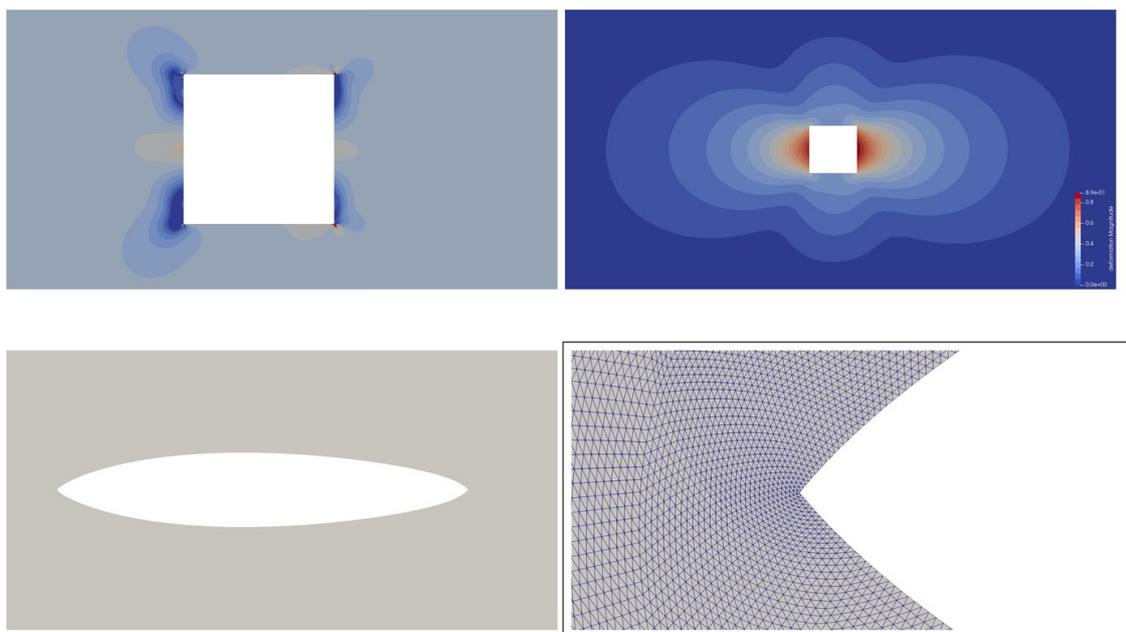
$$v_\infty = \left( \max \left\{ 0, \prod_{i=2}^d \cos\left(\frac{\pi|x_i|}{\delta}\right) \right\}, 0, \dots, 0 \right) \in \mathbb{R}^d$$

with  $\delta$  the diameter of the flow tunnel. The side length of the square obstacle is  $d = 1$ , whereas the radius of the sphere in the 3d case is  $r = 0.5$ . The simulations are performed using UG4 Vogel et al. (2013). We expand UG4 through its C++ based

plugin functionality. The code used for the studies here presented can be consulted at the online repository in Pinzon and Siebenborn (2021). The 2d and 3d grids are generated using the GMSH toolbox Geuzaine and Remacle (2009).

#### 4.1 2d Results

In this section we present 2d simulations for a flow with viscosity  $\nu = 0.03$ . All PDEs are discretized using a  $P_1$  approximation, except for the Navier–Stokes equations and its adjoint which are solved with a stable  $P_2 - P_1$  finite element discretization. For this example,  $\eta$  has initial value of 0.5 and box constraints are  $0 \leq \eta \leq 1.0$  and  $b = 0.001$ . The grid consists of 421,888 triangular elements, with 5 refinement levels. Figure 2 shows results for the optimization of a square obstacle subject to an incompressible, stationary flow. The reference configuration with the extension factor  $\eta$  and optimal displacement field are shown together with the transformed domain and a closeup of the front tip where the element edges are depicted. Regarding the reference configuration, it can be seen that the extension factor approaches the imposed values of the box-constraints at two places, the corners of the square and the sections where new singularities have to be created. If we recall the weak form of the extension factor (28),  $\eta$  controls the nonlinearity in each element. Given that the same initial value of  $\eta$  is set for all elements at the beginning of the simulation and the  $\theta$ -term in (12) penalizes the deviation from the average of  $\frac{1}{2}(\eta_{ub} + \eta_{lb})$ , the different extension factor values, particularly the ones close to the obstacle's surface  $\Gamma_{obs}$ , show that equation (28) adapts depending on the current iterate for the displacement field  $w$ . This ensures that the  $w$  promotes both the generation of new



**Fig. 2** At the top, the reference configuration is shown with the optimal  $\eta$  (left) and  $w = S(u)$ . At the bottom, the transformed grid  $F(\Omega)$  with resulting singularities (left) is shown, altogether with a zoom on the singularity where mesh quality is preserved due to the choice of  $S$

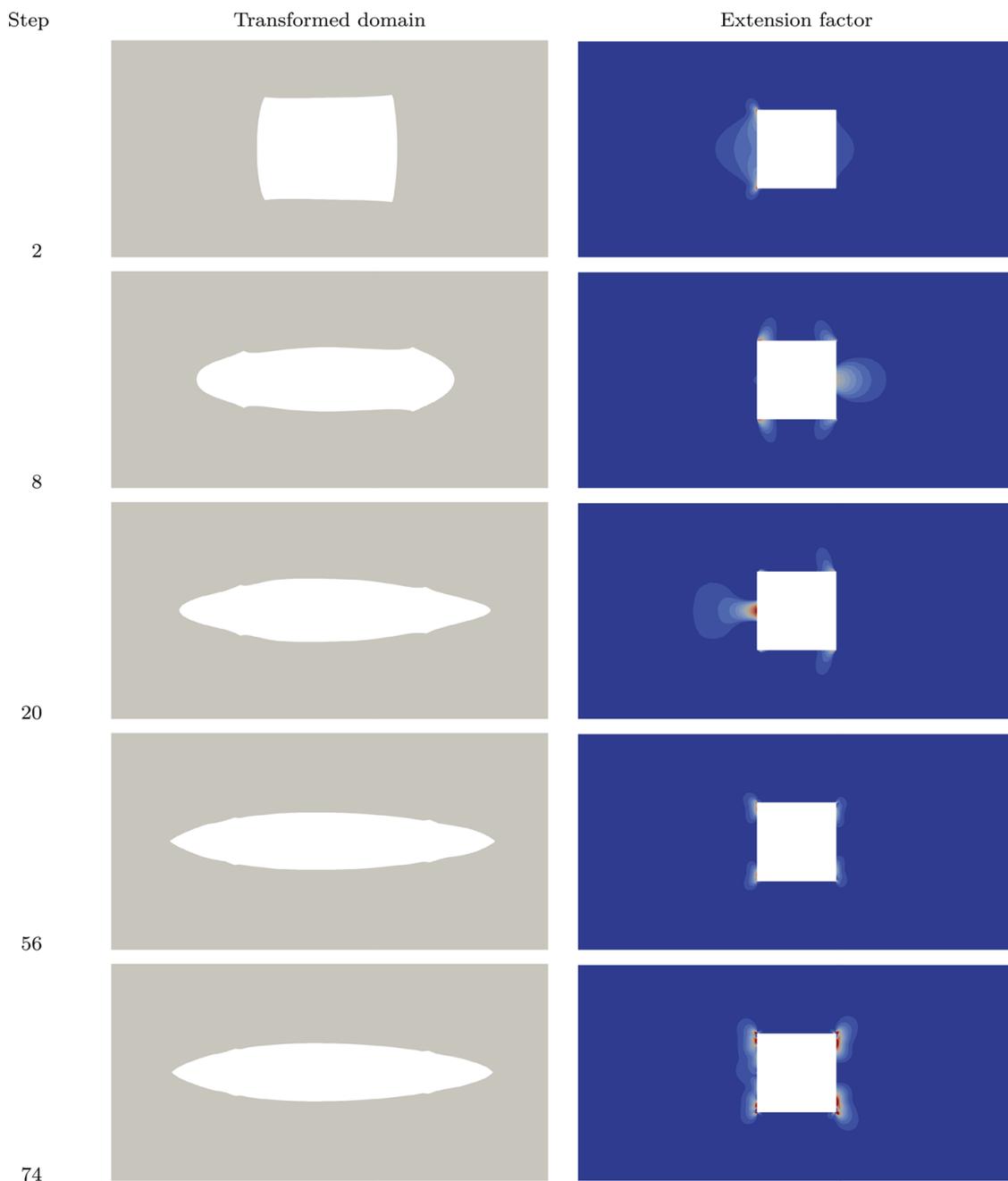
non-smooth points on the boundary, as well as the smoothing of such points introduced by the choice of the reference domain, i.e. the four corners of the box inclusion  $\Omega_{\text{obs}}$ . This can be observed in Fig. 2, where high valued displacements are present at the sections where the tips and corners are generated or smoothed. Which in turn leads to achieving large deformations  $F(\Omega)$  without loss of convergence of the iterative solvers.

In Sect. 2 we already mention that explicit mesh deformations are avoided. This comes from the fact that all optimization steps are solved on the reference domain through the method of mappings. Therefore we speak of obtaining an optimal deformation field  $F$ , which is used to transform the domain  $\Omega \mapsto F(\Omega)$  according to (5). This is used to obtain the optimal shape, as in Fig. 2. The transformed domain shows the smoothed corners and the generated front and back surface singularities, which are in accordance to the previously mentioned properties of  $\eta$  and  $\mathbf{w}$ . However, throughout the optimization process the proposed algorithm does not require the nodal positions to be redefined, since the reference grid transformation is only performed as part of the post-processing and not of the optimization. The close-up corresponds to the front singularity with respect to the direction of flow. Figure 2 also shows that elements around the generated tip show no distortion and no significant loss of quality. This stems from both the effect of the nonlinear term in the extension operator  $S$  and the imposed upper bound  $b$  on the determinant of the deformation gradient  $\det(DF)$ , given in (7). The latter condition is what preserves local injectivity, thus avoiding the loss of mesh quality. In Fig. 2 this is shown as the absence of collapsed or overlapping elements, as is previously mentioned, even for the elements that clearly undergo large deformations, i.e. the ones that conform the generated tips and the smoothed square corners. Moreover, in Sects. 4.3 and 5 this can be understood as a mesh independent preservation of the geometrical and numerical convergence in terms of the final optimal shape achieved and the total iteration counts of the iterative solvers.

On the other hand, the extension equation adapts where the tips have to be created to reach an optimal value of the objective function. This is illustrated through the changing value of the nonlinearity switch  $\eta$  on each optimization step. Figure 3 shows the plot of  $\eta$  over the reference domain compared to the domain transformed by the displacement field  $\mathbf{w}$ .

At the start of the simulation the extension equation is already adapted to find the corners of the reference configuration, this is shown as the concentrated value of  $\eta$ . As the obstacle's initial singularities are removed, necessary ones are created. This causes a concentration of the extension factor at the reference configuration locations where new geometrical singularities have to be created. Afterwards, the optimization scheme works towards smoothing the obstacle's surface, therefore  $\eta$  goes through no major concentration values across the grid, as can be seen in step 74 of Fig. 3.

The distribution of  $\eta$  across the grid has to be compared against the transformed grid. Given that the Lagrange multipliers are yet to converge, the initial steps can incur in violations of the geometrical constraints. This can be seen as the highly deformed shapes at the initial steps of Fig. 3. However, as the algorithm performs the multipliers' update, as in Algorithm 1 line 12, the



**Fig. 3** Transformation of the reference domain by the application of the deformation field compared to the accumulation of the extension factor, given for steps 2, 8, 20, 56, 74

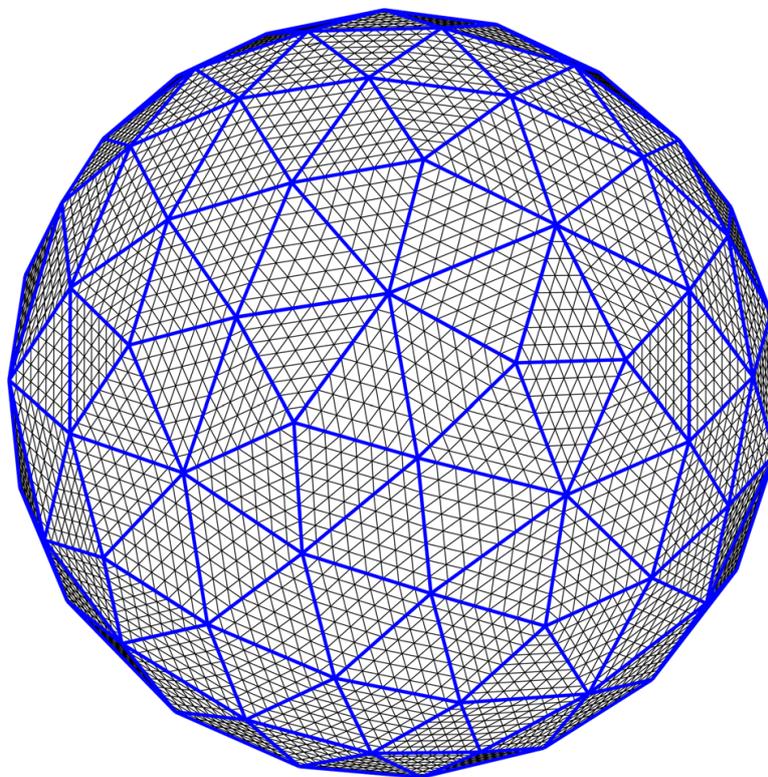
geometrical constraints are fulfilled according to the prescribed  $\epsilon_g$  and the new obstacle surface's singularities are formed. Moreover, since the reference configuration singularities are identified at the initial optimization steps, the necessary smoothing is carried out until the simulation converges or the maximum number of steps is reached. This can be seen comparing the last 2 steps of Fig. 3.

## 4.2 3d Results

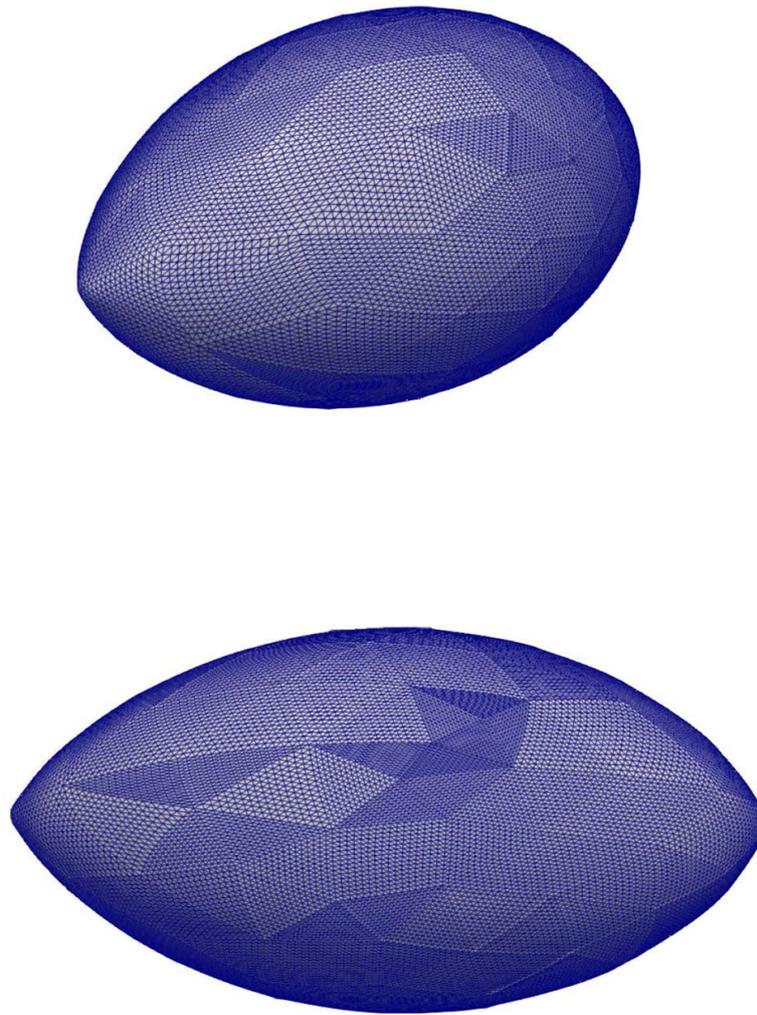
3d Results for the optimization of a unit-diameter sphere are presented here. For these results, 4 levels of refinements are used with up to 12,296,192 tetrahedral elements, while the obstacle's surface consists of 54,784 triangular elements. The viscosity is set to  $\nu = 0.1$ , with a discretization scheme as in Sect. 4.1 where, in contrast to the 2d case,  $P_1 - P_1$  mixed elements are used for the Navier–Stokes equations and its adjoint. Regarding the extension equation,  $\eta$  has initial value of 30 and the box constraints are  $0 \leq \eta \leq 60.0$ . A pressure projection stabilization term is used for the mixed finite element approximation, as given in Elman et al. (2014).

The discretized domain representing the 3d obstacle is shown in Fig. 4. Both the base level (bold blue lines) and highest refinement level are presented. As mentioned in Sect. 1, we investigate the application of the geometric multigrid method as a preconditioner in shape optimization. This implies that we strive to maintain the base level as coarse as possible, as can be seen by the underrepresented sphere shown; with the idea of solving the coarsest problem with a direct method as quickly as possible. While this is ideal for the usage and convergence of the geometric multigrid method, it has some undesired effects. As can be seen, the several refinements introduced by the creation of the hierarchical grid levels do not necessarily introduce a smoothing of the obstacle's surface. The refinements are limited to subdividing the triangular faces present on  $\Gamma_{\text{obs}}$ , while the edges from the base grid remain.

The results after 61 steps are shown in Fig. 5. Non-smooth points, i.e. the two tips, are generated on the front and back of  $\Gamma_{\text{obs}}$  with respect to the direction of flow. This



**Fig. 4** 3d Highest grid level is shown compared to the base level (bold lines) for the unit-diameter sphere obstacle



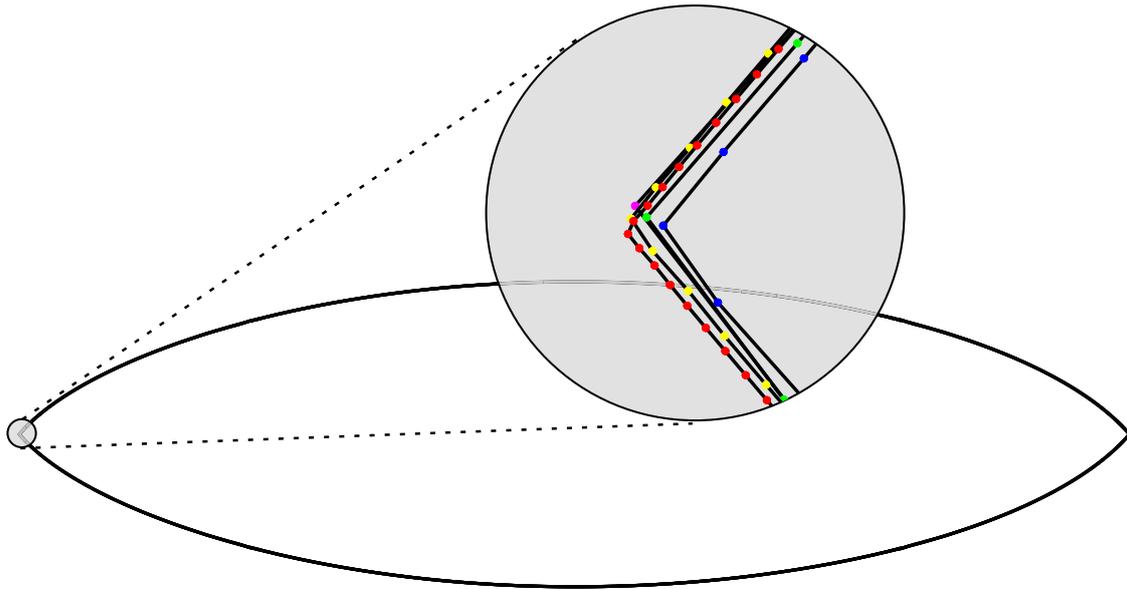
**Fig. 5** 3d optimal shape on the highest level of refinement in flow with large viscosity

is comparable to the optimal shape obtained for the 2d case in Fig. 2. The effects of the grid hierarchy can be seen as the remaining edges of the super-elements.

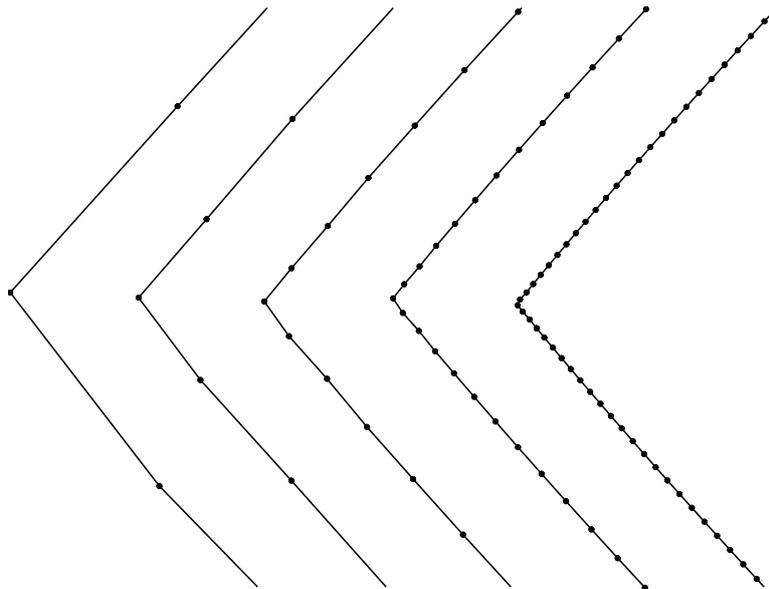
### 4.3 Grid independence study

In order for the proposed optimization scheme to be scalable in terms of time-to-solution to very high numbers of DoFs, it is necessary for the obtained obstacle shape to be independent of the initial level of refinement. In other words, besides the scalability of the finite element building blocks of the optimization algorithm, the overall convergence of the objective function has to be mesh independent. This can be understood as obtaining the same optimized shape after a given number of outer iterations in Algorithm 1, with the necessary surface singularities appearing at approximately the same locations. Therefore, in this section we provide results for a comparative study between different levels of refinement.

The grid used in this section is formed by 412 triangular elements and refined to 1,687,552 elements. The results shown go from 2 to 6 refinement levels. The simulations are set with viscosity  $\nu = 0.1$ . An equal number of 400 optimization



**Fig. 6** Optimal displacement field  $w$ , after 400 optimization steps, applied to the reference shape for several levels of refinement, indicated by colored nodes

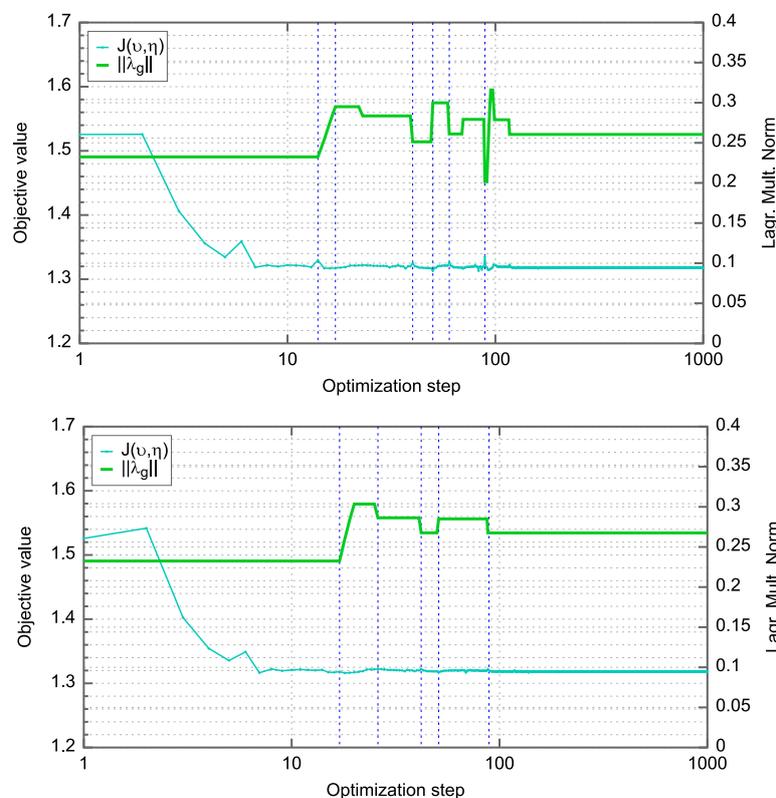


**Fig. 7** Zoom into the tips of the deformed shapes of the experiment shown in Fig. 6. Comparison of generated boundary singularities for 2, 3, 4, 5, and 6 levels of refinement (from left to right). Shapes are aligned with fixed interspace for better comparability

steps is run for all grids, to have an adequate comparison point. Figure 6 shows the superimposed contours of the obstacle for 2, 3, 4, 5, and 6 levels of refinement, indicated by the colored nodes. A magnification is used on the front tip, to emphasize that all tips appear on the same location, with slight differences owing to the discretization error introduced by different element sizes. In addition to this results, Fig. 7 shows a side-by-side comparison of the tips of the aforementioned

refinement levels. This indicates, that the singularities on the obstacle surface generated by algorithm Algorithm 1, are grid independent.

Moreover, in Fig. 8 the objective function plots for 2 different refinement levels are shown. With the same viscosity as in Figs. 6 and 7, the simulations are set to run for 1000 steps. The purpose is to demonstrate that the achieved minimal value is independent of the geometry. We thus choose this large number of optimization steps regardless of any tolerance  $\epsilon_{\text{outer}}$ . The value of (13) (in blue) is compared against the Euclidean norm of the Lagrange multipliers (in green) for each refinement, while the update of the multipliers is signaled by the dashed lines (dark blue). It is evident that before the convergence of the multipliers, the optimization process is local, which is why differences between the two plots with respect to the objective function value are present. In Algorithm 1, the condition for the update of the aforementioned Lagrange multipliers is mentioned. This is related to the set tolerance  $\epsilon_g$ . Given that the two geometries are different due to the level of refinement, the fulfillment of the geometrical tolerance is not necessarily achieved in the same optimization steps. Which in turn, as seen in Algorithm 1, has an effect as to when the multipliers are updated. Nevertheless, as previously mentioned in this section, the objective function converges altogether with the norm of the multipliers, as seen by comparing the plots in both refinement cases presented in Fig. 8. It can also be seen that in most cases, a significant update of the Lagrange multipliers is accompanied by a substantial jump, negative or positive, in the value of the objective function. This is signaled by the intersection data points between



**Fig. 8** Objective function plot for 3 (top) and 4 (bottom) refinements. Green line shows norm of geometric Lagrange multipliers. Dashed blue vertical lines indicate  $\lambda_g$  update in augmented Lagrange algorithm

the changes in the norm level of the multipliers  $\|\lambda_g\|$ , the jumps of the objective, and the marked update steps.

## 5 Algorithmic scalability

In this section we present weak scalability results for the 2d case presented in Sect. 4.1. These were carried out at *HLRS* using the modern *HPE Hawk* supercomputer. It consists of 5632 dual-socket nodes with the AMD EPYC 7742 processor. Each node has a total of 128 cores and 256GB of memory. The machine presents a 16-node hypercube connection topology, therefore the core counts are aimed at maximizing hypercube use, without significantly reducing the bandwidth. The grid partitioning is based on ParMETIS Karypis et al. (2013).

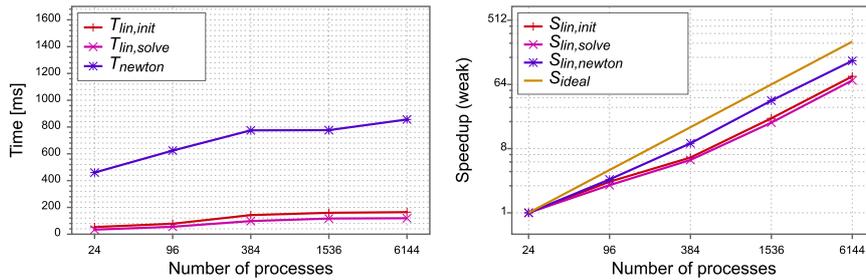
Figure 9 shows accumulated wallclock times, gained speedup relative to 24 cores and iteration counts for the first three optimization steps. These results are shown for the nonlinear extension operator (28), state (14), (15), (16), (17), (18), and the adjoint displacement (39). A  $P_1$  finite element discretization is used for extension operator and its adjoint equation, while mixed  $P_1 - P_1$  shape functions are used for the state equation Navier–Stokes equations. The nonlinear problems are solved using Newton’s method altogether with a BiCGStab solver for the underlying linearizations. The linear solver is preconditioned with the geometric multigrid, which uses a V-cycle, 3 pre- and post-smoothing steps, and an LU base solver gathered on a single core. The error reduction is set to an absolute of  $10^{-14}$  and relative  $10^{-8}$  for the nonlinear solver, and  $10^{-12}$  and  $10^{-3}$  for the linear solvers. The purely linear problem has a relative and absolute reduction of  $10^{-16}$ . A Jacobi smoother is used within the geometric multigrid for the extension equation and the derivative, whereas the Navier–Stokes equation solver features an ILU smoother [see for instance Wittum (1989)]. The results presented start at 24 cores, with a fourfold increase for each mesh refinement.

The studies show scalability and speedup for up to 6144 cores and more than 27 million triangular elements. Given that mesh refinements are performed for each core count increase, a different geometrical problem is solved therefore differences in iteration counts are expected. However, even for a significant increase in the number of geometric elements the iteration counts for the linear problems remain within moderate bounds. Moreover, it is important to point out that the total number of DOFs solved within the presented PDEs in Fig. 9 increases from about 783k to 189 million. While the total number of DOFs solved in one optimization step is close to 300 million.

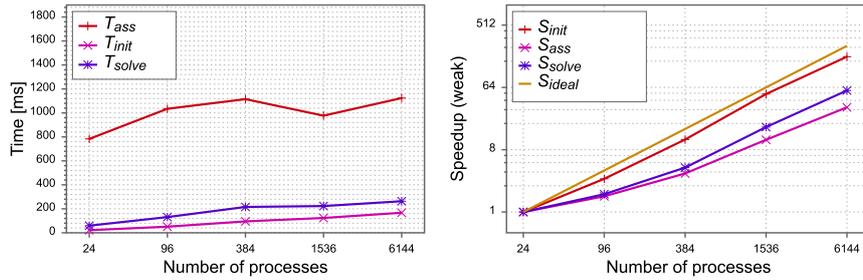
Together with the grid independence study for the outer optimization routine we thus obtain weak scalability of the overall method.

## 6 Conclusion

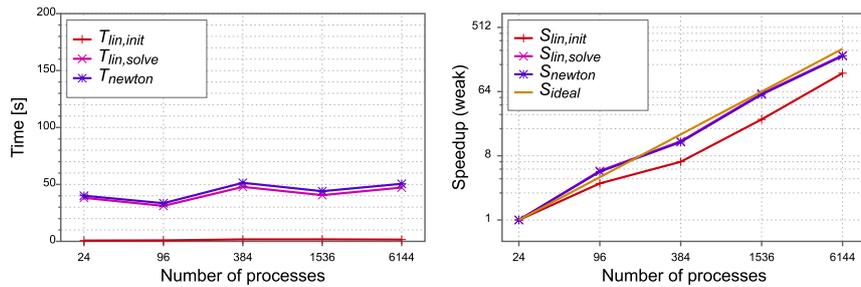
In this article we presented an optimization methodology which relies on the self-adaptation of the extension operator within the method of mappings. The results show that large deformations with respect to the reference configuration are possible while



(a) Extension equation nonlinear problem



(b) Adjoint displacement linear problem



(c) State equation nonlinear problem

Procs	Refs	NumElems	Adjoint Displacement	Displacement Field	State Equation
			(linear solver)	(linear/nonlinear solver)	(linear/nonlinear solver)
24	4	105472	56	9/12	238/12
96	5	421888	70	9/12	199/12
384	6	1687552	77	9/12	211/13
1536	7	6750208	77	9/12	194/13
6144	8	27000832	70	8/10	233/12

(d) Accumulated iteration counts for the weak scaling study

**Fig. 9** Weak scaling: For the first three optimization steps, the accumulated wallclock time is shown for: **a** the nonlinear extension equation, **b** the derivative of the objective function with respect to the displacement field. In **c**, the accumulated iteration counts are presented for the geometric multigrid preconditioned linear solver of the shape derivative, the number of Newton steps and linear iterations necessary to solve the extension equation and its linearization

preserving mesh quality. This has been studied in situations where singularities during the shape optimization process have to be smoothed out and new ones generated. It has been demonstrated that these two effects are particularly important to be tackled for applications of hierarchical multigrid solvers when experiments from fluid dynamics are considered.

The method's scalability and grid independency have been illustrated with the results of Sects. 4.3 and 5. Grid independence is necessary for applications where a high level of refinement is needed, since it guarantees that the same optimal shape is obtained regardless of the number of elements. This becomes particularly important for the weak scalability, where the grid is refined on each core count increase. The results shown in Fig. 9, in combination with the ones of Figs. 6 and 7, establish a proof of concept for industrial applicability, where a high number of DOFs are expected. Overall, in this article we have presented an algorithm towards scalable shape optimization for large scale problems with the potential to work reliably also in complex geometric situations.

**Acknowledgements** Computing time on the supercomputer *Hawk* at HLRS under the grant ShapeOpt-CompMat (ACID 44171, Shape Optimization for 3d Composite Material Models) is gratefully acknowledged. 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”.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

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

## References

- Allaire G, Dapogny C, Jouve F (2021) Chapter 1—shape and topology optimization. In: Bonito A, Nochetto RH (eds) Geometric partial differential equations—part II, handbook of numerical analysis, vol 22. Elsevier, Amsterdam, pp 1–132. <https://doi.org/10.1016/bs.hna.2020.10.004>
- Baker AH, Falgout RD, Kolev TV, Yang UM (2011) Multigrid smoothers for ultra-parallel computing. *SIAM J Sci Comput* 33:2864–2887
- Blauth S (2021) Nonlinear conjugate gradient methods for PDE constrained shape optimization based on steklov-poincaré-type metrics. *SIAM J Optim* 31(3):1658–1689. <https://doi.org/10.1137/20M1367738>
- Brandenburg C, Lindemann F, Ulbrich M, Ulbrich S (2009) A continuous adjoint approach to shape optimization for Navier Stokes flow. In: Kunisch K, Leugering G, Sprekels J, Tröltzsch F (eds) Optimal control of coupled systems of partial differential equations, vol 160. Birkhäuser, Basel, pp 35–56
- Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. *SIAM J Sci Comput* 16(5):1190–1208. <https://doi.org/10.1137/0916069>

- Delfour MC, Zolésio JP (2001) Shapes and geometries: metrics, analysis, differential calculus, and optimization. *Advances in design and control*, vol 22, 2nd edn. SIAM, Philadelphia
- Dokken JS, Funke SW, Johansson A, Schmidt S (2019) Shape optimization using the finite element method on multiple meshes with nitsche coupling. *SIAM J Sci Comput* 41(3):A1923–A1948
- Dokken JS, Mitusch SK, Funke SW (2020) Automatic shape derivatives for transient PDEs in FEniCS and Firedrake
- Elman H, Silvester D, Wathen A (2014) Finite elements and fast iterative solvers with applications in incompressible fluid dynamics, vol 1. Oxford Science Publications
- Etling T, Herzog R, Loayza E, Wachsmuth G (2018) First and second order shape optimization based on restricted mesh deformations. *SIAM J Sci Comput* 42(2):A1200–A1225
- Fischer M, Lindemann F, Ulbrich M, Ulbrich S (2017) Fréchet differentiability of unsteady incompressible Navier–Stokes flow with respect to domain variations of low regularity by using a general analytical framework. *SIAM J Control Optim* 55(5):3226–3257. <https://doi.org/10.1137/16M1089563>
- Gangl P, Laurain A, Meftahi H, Sturm K (2015) Shape optimization of an electric motor subject to non-linear magnetostatics. *SIAM J Sci Comput* 37(6):B1002–B1025
- Garcke H, Hinze M, Kahle C (2016) A stable and linear time discretization for a thermodynamically consistent model for two-phase incompressible flow. *Appl Numer Math* 99:151–171
- Geuzaine C, Remacle JF (2009) Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Methods Eng* 79(11):1309–1331. <https://doi.org/10.1002/nme.2579>
- Giles MB, Pierce NA (2000) An introduction to the adjoint approach to design. *Flow Turbul Combust* 65(3–4):393–415
- Gmeiner B, Köstler H, Stürmer M, Rüde U (2014) Parallel multigrid on hierarchical hybrid grids: a performance study on current high performance computing clusters. *Concurr Comput Pract Exp* 26(1):217–240
- Hackbusch W (1985) Multi-grid methods and applications, vol 4. Springer, Berlin
- Haubner J, Siebenborn M, Ulbrich M (2021) A continuous perspective on shape optimization via domain transformations. *SIAM J Sci Comput* 43(3):A1997–A2018. <https://doi.org/10.1137/20m1332050>
- Hinze M, Pinnau R, Ulbrich M, Ulbrich S (2009) Optimization with PDE constraints. *Mathematical modelling: theory and applications*, vol 23. Springer, Berlin
- Iglesias JA, Sturm K, Wechsung F (2018) Two-dimensional shape optimization with nearly conformal transformations. *SIAM J Sci Comput* 40(6):A3807–A3830
- Jameson A (2003) Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*
- Karypis G, Schloegel K, Kumar V (2013) Parmetis, parallel graph partitioning and sparse matrix ordering library. Available at <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>
- Luft D, Schulz V (2021a) Pre-shape calculus: foundations and application to mesh quality optimization
- Luft D, Schulz V (2021b) Simultaneous shape and mesh quality optimization using pre-shape calculus
- Mannel F, Rund A (2020) A hybrid semismooth quasi-newton method for nonsmooth optimal control with PDEs. *Optim Eng*. <https://doi.org/10.1007/s11081-020-09523-w>
- Mohammadi B, Pironneau O (2010) Applied shape optimization for fluids. Oxford University Press, Oxford
- Murat F, Simon J (1976) Etude de problèmes d’optimal design. In: Cea J (ed) Optimization techniques modeling and optimization in the service of man part 2: proceedings, 7th IFIP conference Nice, September 8–12, 1975. Springer, Berlin, Heidelberg, pp 54–62
- Müller PM, Kühl N, Siebenborn M, Deckelnick K, Hinze M, Rung T (2021) A novel p-harmonic descent approach applied to fluid dynamic shape optimization. *Struct Multidiscip Optim* 64(6), 3489–3503
- Nägel A, Schulz V, Siebenborn M, Wittum G (2015) Scalable shape optimization methods for structured inverse modeling in 3D diffusive processes. *Comput Vis Sci* 17(2):79–88. <https://doi.org/10.1007/s00791-015-0248-9>
- Onyshkevych S, Siebenborn M (2021) Mesh quality preserving shape optimization using nonlinear extension operators. *J Optim Theory Appl* 16(5):291–316. <https://doi.org/10.1007/s10957-021-01837-8>
- Pinzon J, Siebenborn M (2021) Fluidoptim. Available at <http://www.github.com/multigridshapeopt>
- Pinzon J, Siebenborn M, Vogel A (2020) Parallel 3d shape optimization for cellular composites on large distributed-memory clusters. *J Adv Simul Sci Eng* 7(1):117–135. <https://doi.org/10.15748/jasse.7.117>
- Reiter S, Vogel A, Heppner I, Rupp M, Wittum G (2013) A massively parallel geometric multigrid solver on hierarchically distributed grids. *Comp Vis Sci* 16(4):151–164

- Schmidt S, Ilic C, Schulz V, Gauger NR (2013) Three-dimensional large-scale aerodynamic shape optimization based on shape calculus. *AIAA Journal* 51(11):2615–2627
- Schulz V, Siebenborn M (2016) Computational comparison of surface metrics for PDE constrained shape optimization. *Comput Methods Appl Math* 16(3):485–496. <https://doi.org/10.1515/cmam-2016-0009>
- Schulz V, Siebenborn M, Welker K (2016) Efficient PDE constrained shape optimization based on Steklov-Poincaré -type metrics. *SIAM J Optim* 26(4):2800–2819. <https://doi.org/10.1137/15M1029369>
- Siebenborn M, Vogel A (2021) A shape optimization algorithm for cellular composites. *PINT Comput Vis Sci*. Available at [arxiv.org/1904.03860](https://arxiv.org/1904.03860)
- Siebenborn M, Welker K (2017) Algorithmic aspects of multigrid methods for optimization in shape spaces. *SIAM J Sci Comput* 39(6):B1156–B1177
- Sokolowski J, Zolesio JP (2012) Introduction to shape optimization: shape sensitivity analysis, vol 16. Springer, Berlin
- Vogel A, Reiter S, Rupp M, Nägel A, Wittum G (2013) UG 4: a novel flexible software system for simulating PDE based models on high performance computers. *Comput Vis Sci* 16(4):165–179
- Wilke DN, Kok S, Groenwold AA (2005) A quadratically convergent unstructured remeshing strategy for shape optimization. *Int J Numer Methods Eng* 65(1):1–17. <https://doi.org/10.1002/nme.1430>
- Wittum G (1989) Multi-grid methods for stokes and Navier–Stokes equations. *Numer Math* 54:543–563. <https://doi.org/10.1007/BF01396361>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## PAPER II

## A SCALABLE ALGORITHM FOR SHAPE OPTIMIZATION WITH GEOMETRIC CONSTRAINTS IN BANACH SPACES\*

PETER MARVIN MÜLLER<sup>†</sup>, JOSÉ PINZÓN<sup>‡</sup>, THOMAS RUNG<sup>†</sup>, AND  
MARTIN SIEBENBORN<sup>‡</sup>

**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.

**Key words.** shape optimization, Lipschitz transformations, geometric multigrid, parallel computing

**MSC codes.** 35Q93, 49Q10, 65M55, 65Y05, 65K10

**DOI.** 10.1137/22M1494609

**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 [30, 23, 22, 25]). 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, 12] 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).

---

\* Submitted to the journal's Computational Methods in Science and Engineering section May 5, 2022; accepted for publication (in revised form) November 21, 2022; published electronically April 7, 2023.

<https://doi.org/10.1137/22M1494609>

<sup>†</sup>Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Hamburg, Germany 21073 (peter.marvin.mueller@tuhh.de, thomas.rung@tuhh.de).

<sup>‡</sup>Department of Mathematics, Hamburg University, Hamburg, Germany 20146 (jose.pinzon@uni-hamburg.de, martin.siebenborn@uni-hamburg.de).

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 nonfeasible 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 nonmeaningful 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. The discretization for a multigrid approach might feature sharp kinks and edges at the initial coarse mesh. The mesh at a higher refinement level, however, should represent the actual geometry more accurately than the coarse mesh. Therefore the shape update should not only generate sharp edges and kinks, but also smooth the boundary.

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. [31, 9, 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 [8, 18] 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 [27] considers the volume constraint, which is 1d. Because the shape update in [27] is performed solely in the surface normal direction, the corresponding Lagrange multiplier is given by the mean value of the deformation. In [21], 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. First, an augmented Lagrange method can be used to determine the Lagrangian multipliers associated with the geometric constraints [3, 30, 2, 22, 11]. 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 de-

termine 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. Second, the method of mappings [23, 15] 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 [24, 29], although it is based on Hilbert space settings. We also want to mention the Uzawa iteration [33] 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 deformations or descent directions, respectively. 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  $\nabla(\cdot)$ . The directional derivatives with respect to a specific variable are indicated using  $\frac{\partial}{\partial u}(\cdot)\delta_u$  in direction  $\delta_u$ . The shape derivative of the functional  $J(\Omega)$  in direction  $u$  is denoted by  $J'(\Omega)u$  as defined in (2.7).

**2. Model equations.** In the present work, we propose an optimization methodology for PDE constraint shape optimization problems of the abstract form:

$$\begin{aligned} (2.1) \quad & \min_{\Omega \in \mathcal{S}} j(\Omega, y) \\ (2.2) \quad & \text{s.t. } e(\Omega, y) = 0 \\ (2.3) \quad & g(\Omega) = 0 \end{aligned}$$

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  $\Omega$  with boundary  $\Gamma = \Gamma_{\text{obs}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}}$ , where  $\Gamma_{\text{obs}}$  is to be optimized. Furthermore,  $\mathcal{S}$  denotes an abstract set of admissible shapes, as explained, for instance, in [31, 9], and specified in (2.6).

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  $J(\Omega) := j(\Omega, y(\Omega)) := J(\Omega) : \mathcal{S} \mapsto \mathbb{R}$ . In order to obtain sensitivities of the objective  $J$  we follow [31, 9, 3, 6]. For this purpose, the domain  $\Omega$  is parameterized in the sense of the perturbation of identity with the displacement field  $u : \mathbb{R}^d \rightarrow \mathbb{R}^d; x \mapsto u(x)$ . For the sake of readability,  $u(x)$  is written as  $u$ . For a sufficiently small  $u$  we thus obtain deformed configurations

$$(2.4) \quad \Omega_u := \{x + u : x \in \Omega\}$$

of the reference shape  $\Omega$ . For the sake of readability we abbreviate the perturbation of identity as

$$(2.5) \quad F : \Omega \rightarrow \Omega_u; \quad F := \text{id} + u.$$

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

$$(2.6) \quad \mathcal{S} := \{F(\Omega) : F = \text{id} + u, u \in U_{ad} \subseteq W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)\}.$$

As parts of the boundary of  $\Omega$  shall remain fixed, the displacement field  $u$  is chosen in such a way that it vanishes on all boundaries of  $\Omega$  which are not to be optimized. Following [2], the directional shape derivative of  $J$  evaluated in  $\Omega$  in the direction  $u$  is then defined by

$$(2.7) \quad J(\Omega_u) = J(\Omega) + J'(\Omega)u + o(u) \quad \text{where} \quad \frac{o(u)}{\|u\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)}} \xrightarrow{u \rightarrow 0} 0,$$

where  $J'(\Omega)$  is a linear operator applied to  $u$ . Then we can interpret the shape optimization problem (2.1)–(2.3) 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 (2.3) 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  $\Omega$  fulfills the geometrical constraints, i.e.,  $g(\Omega) = 0$ . The crucial aspect of the present method is to separate the geometric constraints (2.3) from the remaining PDE-constrained shape optimization problem (2.1) and (2.2) 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 (2.3) 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 formally apply the Lagrange multiplier method [17]. A rigorous derivation is beyond the scope of this paper, which, however, follows the standard approach from optimal control theory and requires to show that the control-to-state map  $\Omega \mapsto y(\Omega)$  is continuous differentiable [17, Assumption 1.47] in the sense of (2.7). This allows us to define the reduced problem, that is, minimize  $J(\Omega_u)$  on the space of transformations  $u \in U_{ad}$ , and derive first order optimality conditions. To obtain the derivative of the reduced objective function the problem is formulated on the perturbed domain  $\Omega_u$  and pulled back to the reference domain  $\Omega$ , where the derivative is evaluated. The derivative of the reduced function then can be expressed with the help of the adjoint state or Lagrange multipliers, respectively. For details the reader may be referred to [5]. 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

$$(2.8) \quad J(\Omega) := j(\Omega, v) = \frac{\nu}{2} \int_{\Omega} Dv : Dv \, dx$$

is to be minimized. Here and in the following, we denote velocity  $v$ , the density-specific pressure  $\mathbf{p}$ , viscosity  $\nu$ , and an inflow velocity  $v_{\infty}$ . As the PDE constraint  $e(\Omega, y) = 0$ , with state variable  $y = (v, \mathbf{p})$ , we consider the stationary, incompressible Navier–Stokes equations in a weak sense:

$$(2.9) \quad \begin{aligned} -\nu \Delta v + (v \cdot \nabla)v + \nabla \mathbf{p} &= 0 \text{ in } \Omega, \\ \operatorname{div} v &= 0 \text{ in } \Omega, \\ v &= 0 \text{ on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}}, \\ v &= v_{\infty} \text{ on } \Gamma_{\text{in}}, \\ \nu Dv \cdot n &= \mathbf{p}n \text{ on } \Gamma_{\text{out}}. \end{aligned}$$

In the following we consider the weak formulation [10, Chapter 8.2]; see also [7]. 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., [16, 32, 23, 25]. For details on the shape derivative  $J'(\Omega)u$  of the objective function in (2.8), see, e.g., [20, 23].

**2.1. Descent direction.** Following [2, Proposition 4.1] we want to approximate the steepest descent direction  $u$  in

$$(2.10) \quad V_0^{\infty} := \left\{ u \in W^{1,\infty}(\Omega \cup \Omega_{\text{obs}}, \mathbb{R}^d) : \|u\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1, \right. \\ \left. u = 0 \text{ a.e. } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \right\}$$

with the corresponding minimization problem

$$(2.11) \quad \begin{aligned} \min_{u \in V_0^{\infty}} \quad & J'(\Omega)u \\ \text{s.t.} \quad & g(F(\Omega)) = 0, \\ & F = \text{id} + u. \end{aligned}$$

Therefore, we introduce a  $p$ -Laplace relaxation with  $p > 2$  inspired by [18, 8]. Hence let

$$(2.12) \quad V_0^p = \left\{ u \in W^{1,p}(\Omega, \mathbb{R}^d) : \|Du\|_{L^p(\Omega, \mathbb{R}^d)} \leq 1, u = 0 \text{ a.e. on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \right\}$$

and consider

$$(2.13) \quad \begin{aligned} \min_{u \in V_0^p} \quad & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} \, dx + J'(\Omega)u \\ \text{s.t.} \quad & g(F(\Omega)) = 0, \\ & F = \text{id} + u, \end{aligned}$$

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 (2.1)–(2.3) for  $g$  over a fixed  $\Omega$  and a variable displacement field  $u$ . Thus, the admissible set  $\mathcal{S}$  is locally parameterized by  $V_0^p$ -deformations of  $\Omega$ . In the present work,  $m = d + 1$  refers to the barycenter and volume constraints

$$(2.14) \quad \int_{\Omega} (x + u) \det(DF) \, dx = 0,$$

$$(2.15) \quad \int_{\Omega} \det(DF) - 1 \, dx = 0,$$

and without loss of generality, we assume that the barycenter of the initial domain  $\Omega$  is located at the origin  $0 \in \mathbb{R}^d$  of the domain (cf. [25]) and  $\det(DF) > 0$  for  $u$  small enough [31]. Furthermore, due to the constant volume constraint (2.15), we can omit in (2.14) the division by the reference volume and deformed domain  $\Omega$  and  $F(\Omega)$ , respectively.

**2.2. Optimality system.** For the derivation of the optimality conditions of the steepest descent problem (2.13), we define the Lagrangian function

$$(2.16) \quad \begin{aligned} L(u, \lambda) := & \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} \, dx + J'(\Omega) u \\ & + \sum_{i=1}^d \lambda_i \int_{\Omega} (x_i + u_i) \det(DF) \, dx + \lambda_{d+1} \int_{\Omega} \det(DF) - 1 \, dx \end{aligned}$$

with  $\lambda = (\lambda_1, \dots, \lambda_d, \lambda_{d+1})^T$ , where  $\lambda_1, \dots, \lambda_d$  are associated with the barycenter (2.14) and  $\lambda_{d+1}$  with the volume constraint (2.15). In the following we want to recall some rules of differentiation. Therefore, let  $\delta_u, \mu_u : \Omega \rightarrow \mathbb{R}^d$  be generic differentiable vector fields,  $B : \Omega \rightarrow \mathbb{R}^{d \times d}$  and

$$(2.17) \quad DF = \left( \frac{\partial}{\partial x_j} F_i \right)_{1 \leq i, j \leq d} = I + Du$$

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

$$(2.18) \quad \begin{aligned} \frac{\partial}{\partial u} DF \, \delta_u &= D\delta_u, \\ \frac{\partial}{\partial u} \det(DF) \, \delta_u &= \text{tr}((DF)^{-1} D\delta_u) \det(DF), \\ \frac{\partial}{\partial u} (\text{tr}(DF B)) \, \delta_u &= B^T : \left( \frac{d}{du} DF \, \delta_u \right) = B^T : D\delta_u, \\ \frac{\partial}{\partial u} ((DF)^{-1}) \, \delta_u &= -(DF)^{-1} D\delta_u (DF)^{-1}, \\ \frac{\partial}{\partial u} (\text{tr}((DF)^{-1} D\delta_u)) \, \mu_u &= -D\delta_u^T : (DF)^{-1} D\mu_u (DF)^{-1} \\ &= \text{tr}(-D\delta_u (DF)^{-1} D\mu_u (DF)^{-1}) \\ &= \text{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u). \end{aligned}$$

By making use of the rules above, we obtain the derivatives of the Lagrangian (2.16) with respect to  $u$  in the direction  $\mu_u \in V_0^p$ :

$$(2.19) \quad \begin{aligned} \frac{\partial}{\partial u} L(u, \lambda) \mu_u &= \int_{\Omega} (Du : Du)^{\frac{p-2}{2}} (Du : D\mu_u) \, dx + J'(\Omega) \mu_u \\ &+ (\lambda_1, \dots, \lambda_d)^T \cdot \int_{\Omega} \mu_u \det(DF) + (x + u) \text{tr}((DF)^{-1} D\mu_u) \det(DF) \, dx \\ &+ \lambda_{d+1} \int_{\Omega} \text{tr}((DF)^{-1} D\mu_u) \det(DF) \, dx. \end{aligned}$$

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

$$(2.20) \quad \begin{aligned} \frac{\partial}{\partial u} L(u, \lambda) \mu_u &= 0 \quad \forall \mu_u \in V_0^p, \\ \frac{\partial}{\partial \lambda} L(u, \lambda) \mu_\lambda &= 0 \quad \forall \mu_\lambda \in \mathbb{R}^{d+1}. \end{aligned}$$

As the derivatives with respect to  $\lambda$  can directly be taken from (2.16), we omit the details here. In order to solve the nonlinear system (2.20) we require the linearization

$$(2.21) \quad \begin{aligned} \frac{\partial^2}{\partial u^2} L(u^k, \lambda^k)(\mu_u, \delta_u) + \frac{\partial}{\partial \lambda \partial u} L(u^k, \lambda^k)(\mu_u, \delta_\lambda) &= -\frac{\partial}{\partial u} L(u^k, \lambda^k) \mu_u \quad \forall \delta_u, \mu_u \in V_0^p, \\ \frac{\partial}{\partial u \partial \lambda} L(u^k, \lambda^k)(\delta_u, \mu_\lambda) &= -\frac{\partial}{\partial \lambda} L(u^k, \lambda^k) \mu_\lambda \quad \forall \delta_\lambda, \mu_\lambda \in \mathbb{R}^m \end{aligned}$$

and the updates

$$(2.23) \quad u^{k+1} = u^k + \delta_u, \quad \lambda^{k+1} = \lambda^k + \delta_\lambda,$$

where

$$(2.24) \quad \begin{aligned} &\frac{\partial^2}{\partial u^2} L(u, \lambda)(\delta_u, \mu_u) \\ &= \int_{\Omega} (p-2)(Du : Du)^{\frac{p-4}{2}} (Du : D\delta_u)(Du : D\mu_u) + (Du : Du)^{\frac{p-2}{2}} (D\delta_u : D\mu_u) \, dx \\ &\quad + (\lambda_1, \dots, \lambda_d)^T \cdot \int_{\Omega} \left( \delta_u \operatorname{tr}((DF)^{-1} D\mu_u) + \mu_u \operatorname{tr}((DF)^{-1} D\delta_u) \right. \\ &\quad + (x+u) \left( \operatorname{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u) \right. \\ &\quad \left. \left. + \operatorname{tr}((DF)^{-1} D\delta_u \operatorname{tr}((DF)^{-1} D\mu_u)) \right) \det(DF) \, dx \right. \\ &\quad \left. + \lambda_{d+1} \int_{\Omega} \left( \operatorname{tr}(-(DF)^{-1} D\mu_u (DF)^{-1} D\delta_u) \right. \right. \\ &\quad \left. \left. + \operatorname{tr}((DF)^{-1} D\delta_u \operatorname{tr}((DF)^{-1} D\mu_u)) \right) \det(DF) \, dx. \end{aligned}$$

Reviewing the first integral in (2.24), one observes that these terms do not exist for  $p < 4$ , where  $Du : Du = 0$  holds on a set of nonzero measure. However, this issue does not appear in the defect equation (2.19), since there all exponents are nonnegative. We thus modify the first integral in (2.24) to

$$(2.25) \quad \begin{aligned} &\int_{\Omega} (p-2)(Du : Du + \epsilon \Theta(4-p))^{\frac{p-4}{2}} (D\mu_u : Du)(D\delta_u : Du) \\ &\quad + (Du : Du + \epsilon)^{\frac{p-2}{2}} (D\mu_u : D\delta_u) \, dx, \end{aligned}$$

where  $\Theta$  denotes the Heaviside function and  $\epsilon > 0$  a sufficiently small constant. Notice that, within Newton's method in (2.21) and (2.22), this modification only affects the

linearization and not the defect. Thus, solutions of the original problem (2.13) are still obtained upon convergence. Adding  $\epsilon$  in (2.25) serves two purposes. On the one hand it 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 (2.13). 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. For 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  $\lambda$ , the penalty factors, and the penalty increment values. Thus, the practical appeal 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  $p_{\text{inc}}$ .

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 3.1 the step size control is thus handled by scaling the shape sensitivity  $J'(\Omega)$  with a decreasing sequence  $\sigma = (1, \frac{1}{2}, \frac{1}{4}, \dots)$ . The latter is possible because multiplying the objective function in (2.11) with  $\sigma > 0$  does not change the descent direction but the maximum displacement; it is sufficient to search for a minimizing function for  $\sigma J'(\Omega)u$ .

As a note, for numerical reasons it might prove profitable to multiply (2.20) with  $1/\sigma$ . For the sake of readability we abbreviate the linearized optimality system (2.21) and (2.22) using the symbols

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

with  $V_{0,h}^p$  a discrete approximation of  $V_0^p$  used for a finite element discretization of (2.21) and (2.22), which then leads to the saddle point problem.

$$(3.2) \quad \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_u \\ r_\lambda \end{pmatrix},$$

---

**Algorithm 3.1.** Shape optimization steepest descent method.
 

---

**Require:**  $\Omega, p_{\max}$ 

```

1:  $y \leftarrow$  Solve primal problem
2:  $y_0 \leftarrow y$ 
3: Compute objective  $\Phi_0 = J(\Omega)$ 
4: repeat
5:    $y^* \leftarrow$  Solve adjoint 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( $\bar{u}, \sigma, y, y^*$ )
12:       $\bar{u} \leftarrow u_p$ 
13:      Increase  $p$ 
14:    end while
15:    Update geometry  $\Omega$  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  $\Omega$  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  $\|u_{p_{\max}}\|_{W^{1,p}(\Omega)} < \epsilon_1$ 

```

---

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

$$(3.3) \quad \begin{pmatrix} A & B \\ 0 & -B^T A^{-1} B \end{pmatrix} \begin{pmatrix} \delta_u \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_u \\ r_\lambda - B^T A^{-1} r_u \end{pmatrix},$$

where  $S := -B^T A^{-1} B$  is the so-called Schur complement operator. In order to not explicitly compute  $A^{-1}$  an equation system with  $A$  is solved instead. In general, the optimality system (2.20) of problem (2.1)–(2.3) 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  $p_{\text{init}}$  with initial  $u_{p_{\text{init}}} = 0$  and  $\lambda = 0$  is computed. Thereafter the solution of the constraint  $p_k$ -Laplacian problem (2.13) is used as an initial guess for the  $p_{k+1}$ -Laplacian problem (cf. [22]). Here, with the choice of  $p_{\max}$  we adjust the approximation quality of Lipschitz deformations.

The overall optimization procedure is outlined in Algorithm 3.1. The steepest descent method is reflected in the loop spanning from lines 4 to 29, where the necessary

---

**Algorithm 3.2.** Newton's method for  $p$ -Laplacian problem.

---

```

1: function NEWTONSOLVER( $u_p, \sigma, y, y^*$ )
2:    $\lambda \leftarrow 0$ 
3:   repeat
4:      $(A, B, r_u, r_\lambda) \leftarrow \text{Assemble}(u, \lambda, y, y^*)$  according to (3.1)
5:      $(\delta_{u_p}, \delta_\lambda) \leftarrow \text{SCHURSOLVER}(A, B, r_u, r_\lambda, \delta_{u_p}, \delta_\lambda)$ 
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  $(u_p, \lambda)$ 
10: end function

```

---

**Algorithm 3.3.** 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

```

---

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  $\sigma$  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 toward  $0 \in V_0^p$ , but have to stay within the solution manifold of the nonlinear 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 3.1, the nonlinear solver for the steepest descent problem (2.13) is called, which can be seen in Algorithm 3.2. The key part of this solver is the solution to the saddle point problem (3.2) in the Schur complement form (3.3). 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 3.3 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 [34]. This simulation framework has MPI-based parallelization and features a geometrical multigrid preconditioner [14]. The grid partitioning and load balancing scheme is based on ParMetis [19].

Stable  $P_2 - P_1$  finite elements were used to discretize the governing nonlinear

Navier–Stokes equations (2.9) and their linearization, and 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. [25, 23] and the references therein. As regards the p-Laplace relaxation problem, for which the optimality system is described in (3.3),  $P_1$  Lagrange shape functions were employed. Computational grids consist of triangular (two-dimensional (2d)) and tetrahedral elements (3d). They were generated using GMSH [13].

The simulations followed the workflow proposed in Algorithm 3.1. At the beginning of each optimization step, the steady, incompressible flow, described by Navier–Stokes equations (2.9), was computed followed by the solution of the corresponding adjoint system. The p-Laplacian descent algorithm Algorithm 3.1 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 Algorithms 3.1 and Algorithm 3.2 were set as  $\epsilon_1 = 1\text{E-}5$  and  $\epsilon_2 = 1\text{E-}8$ , respectively. The modification term introduced in (2.25) 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 (3.3). 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 two and three dimensions, 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 [26].

**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 Figure 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  $\Gamma_{\text{in}}$  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  $\delta$  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. Since a high viscosity is used in our studies, the energy dissipation decrease is not strongly related to the resulting singularities. However, these were created to showcase the capabilities and properties of the optimization scheme, based on the convergence condition of Algorithm 3.1. 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  $\rho$  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.

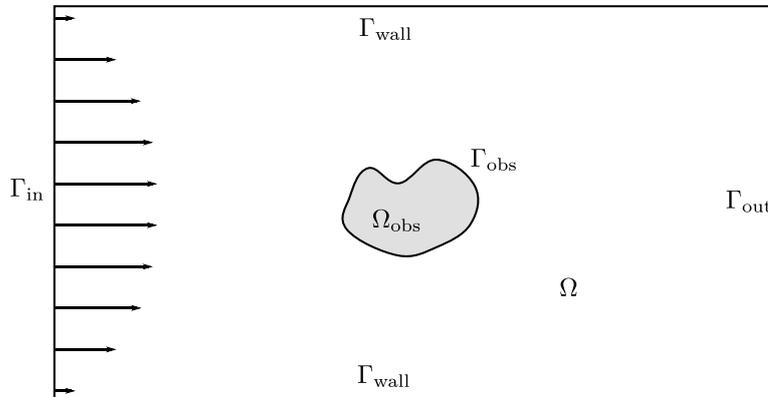


FIG. 1. Schematic view on a flow tunnel-like domain  $\Omega$  with the obstacle  $\Omega_{\text{obs}}$  encircled by its surface  $\Gamma_{\text{obs}}$ , wall boundaries  $\Gamma_{\text{wall}}$ , inflow  $\Gamma_{\text{in}}$ , and outflow  $\Gamma_{\text{out}}$ . The height of the flow tunnel is defined by  $\gamma > 0$ .

Notice 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 3.2, and therefore there is no need to provide results for their fulfillment per step. Solving the nonlinear system of (2.20) implies solving the geometrical constraints (2.14) and (2.15) to the error reduction a tolerance set for Newton's method. The major portion of the computational effort in Algorithm 3.1 is spent on solving the  $p$ -Laplace relaxed problem via the scheme described in section 3. Particularly, lines 11–15 of Algorithm 3.1 are computationally expensive, as will be explained here and in subsection 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 [25] 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 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. Toward the last step, no indication of the initial geometric singularity is visible on the obstacle's surface.

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 (2.8), is found. 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

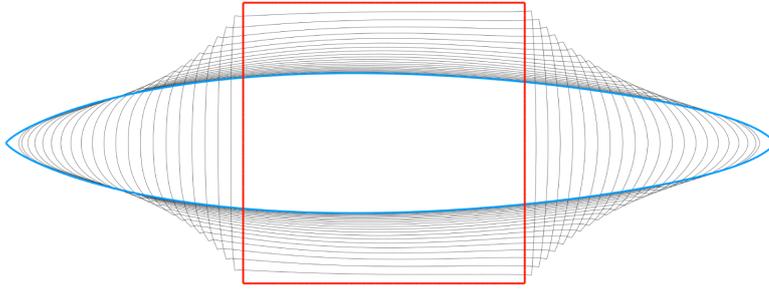


FIG. 2. Superposition of the deformation sequence for a 2d configuration. The obstacle's initial shape red is presented superimposed to the sequence of generated shapes gray until an optimal shape blue is obtained upon convergence.

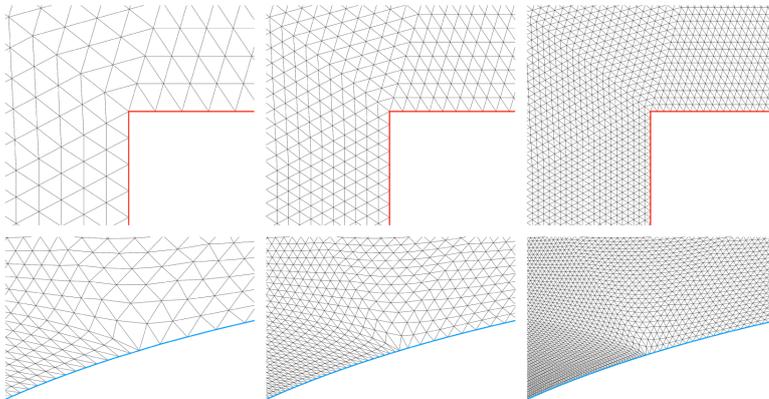


FIG. 3. Removal of the geometrical singularity in the obstacle's initial configuration across several levels of refinement. For 4, 5, and 6 refinements the upper left corner of the box is smoothed via updating the geometry  $\Omega$  iteratively, as stated in line 16 in Algorithm 3.1.

that the simulations are based upon a predetermined mesh quality, and while the optimization we propose in Algorithm 3.1 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. The presented 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 (2.10) and (2.12), 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 higher 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 (2.24) as an exponent. The latter fact turned out to be a limiting factor in our numerical simulations. However, the measurements of the worst minimum and maximum angles

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  $\infty$ ) and the initial configuration (subscript 0,  $\rho_0 = 1.468$ ).

Refs	Elements	Minimum angle	Maximum angle	Radius ratio $\rho_\infty$	$\frac{\rho_\infty}{\rho_0}$
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

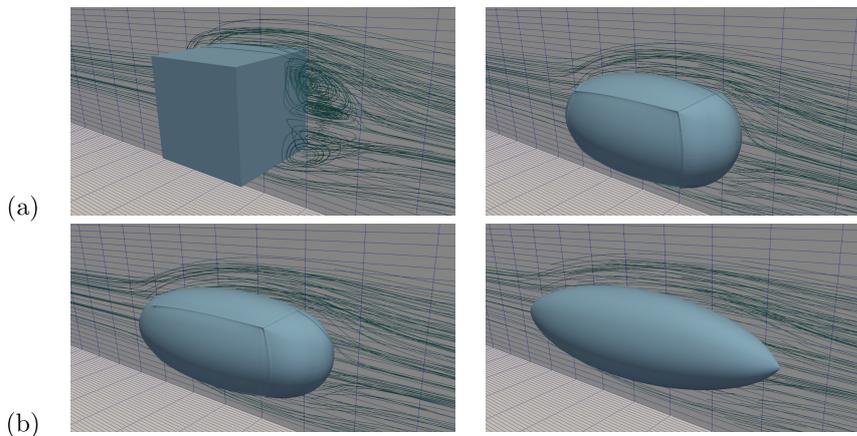


FIG. 4. Streamlines for the rear of the obstacle located in the wind tunnel.

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,  $\Gamma_{\text{obs}}$ , on the highest refinement level. Our optimization scheme generates a series of deformation fields  $u_p$  that, applied to the domain  $\Omega$ , results in an optimal shape with respect to the energy dissipation (2.8).

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 backward with respect to 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  $\tau \cdot e_1 = (\nu(Dv + Dv^T) \cdot n) \cdot e_1$ . 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.

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 form a streamlined body that does not feature any separation, as shown in row (b) of Figure 4.

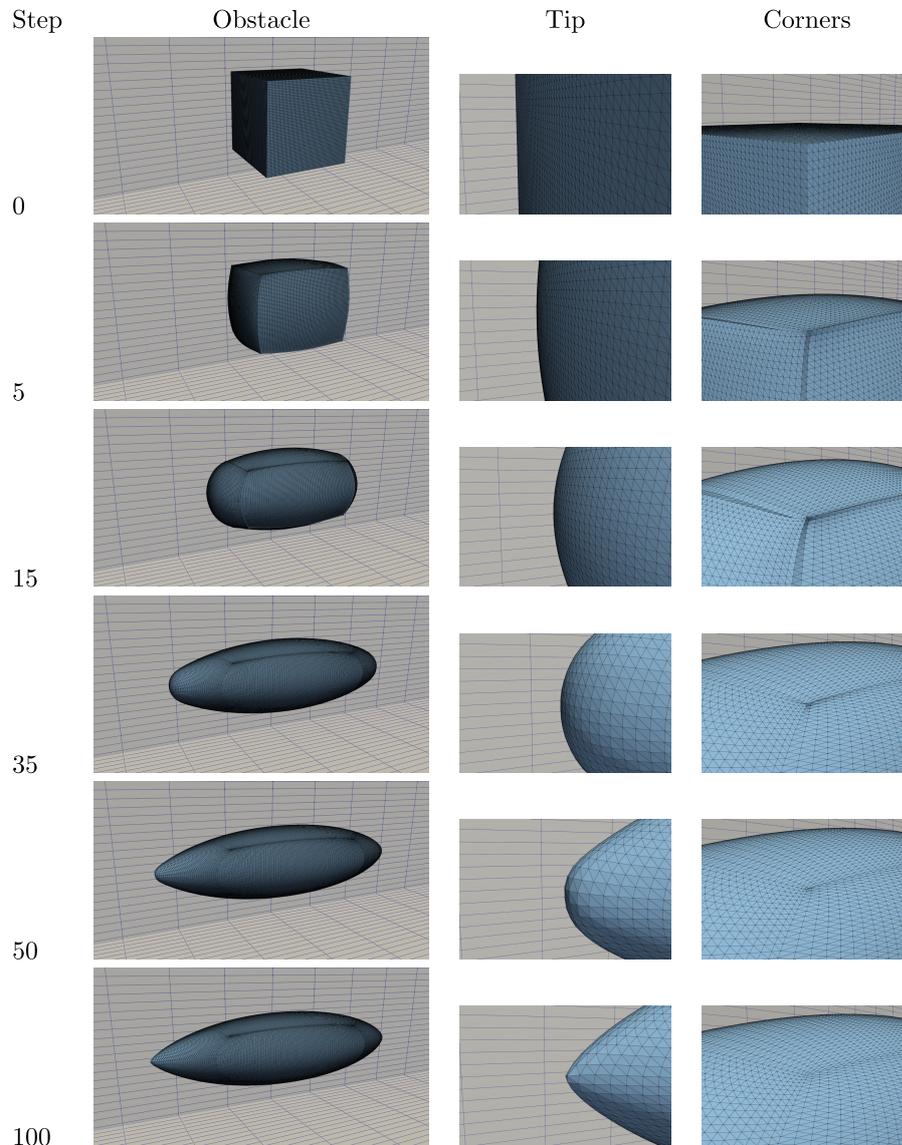


FIG. 5. Deformation sequence for optimization steps  $\{0, 5, 15, 35, 50, 100\}$ . The complete obstacle, together with a detailed view of the geometrical singularity removal and generation process, are presented.

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

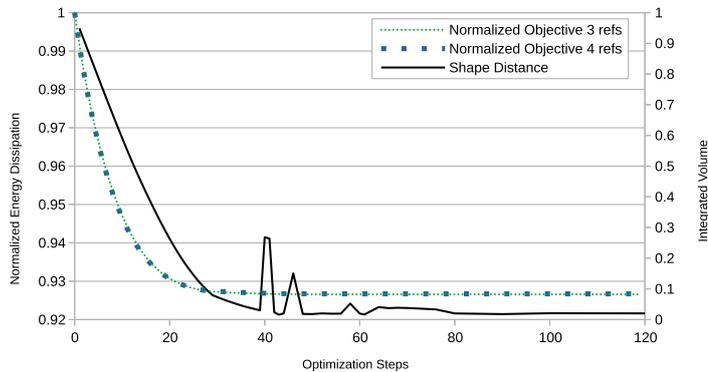


FIG. 6. 3d results for 3 and 4 levels of refinement are compared. The energy dissipation (see (2.8)) is plotted against the difference between shapes of each refinement level per step.

seen in step 35, where the final tip locations also become more apparent. Recall that the mesh deformation corresponds to line 15 in Algorithm 3.1. 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 toward 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 (2.8). Therefore, results are provided in Figure 6 for a 3d setting, which show how the generated shape, after convergence of Algorithm 3.1, 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 (2.8) decreases monotonically is linked to lines 15–26 of Algorithm 3.1, 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  $\Omega_{3ref}^k$  of the 3 refinements run to the optimal solution of the 4 refinements run  $\Omega_{4ref}^\infty$ . Figure 6 shows the integrated volume that refers to the symmetric difference as

$$(5.1) \quad d(\Omega_{3ref}^k, \Omega_{4ref}^\infty) := |\Omega_{3ref}^k \setminus \Omega_{4ref}^\infty| + |\Omega_{4ref}^\infty \setminus \Omega_{3ref}^k|.$$

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 theorem the volume can be found.

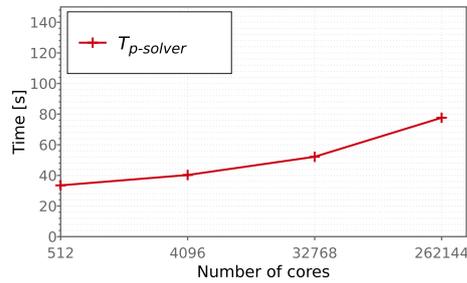
**5.3. Scalability study.** Weak scalability of the solution strategy for the  $p$ -Laplacian relaxed problem from  $p_{init} = 2.0$  up to  $p_{max}$  is presented here. This solution strategy, described in section 3, is referred in these results as the  $p$ -solver.

This corresponds to lines 11–15 of Algorithm 3.1. It was studied for up to 262,144 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 2.25 GHz, and 256 GB 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.

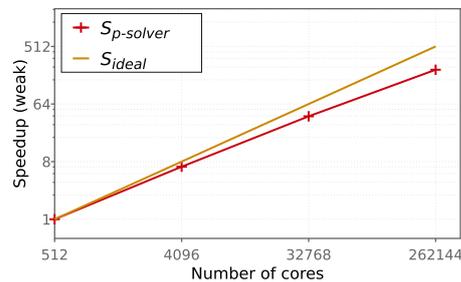
A 3d computational grid with 2 levels of refinement is used as an initial measurement in order to optimize the number of cores used at the finest level. The wallclock times, speedup, and iteration counts are shown in Figure 7. An eightfold increase in the number of cores is performed for each level of refinement. Results are presented for the solution of the nonlinear system of equations given in (2.20) via its linearization in (2.21) and (2.22). 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 of a Gauss–Seidel smoother with a V-cycle. An LU factorization solves the base level gathered in a single core.

We measure the accumulated times and iteration counts for the routines in lines 11–15 of Algorithm 3.1 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  $p_{\text{init}}$  up to  $p_{\text{max}}$  with  $p_{\text{inc}}$  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 (3.3), 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 right-hand side in the second equation of (3.3), and  $m$  for the computation of  $S$ . Additionally, the first equation of (3.3) has to be solved for  $\delta_u$ , 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.,  $p_{\text{max}}$ , 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  $p_{\text{inc}}$ , 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 3.2 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 [28]), which implies that care must be taken during the generation of the grid. Nevertheless, it is a very effective approach toward solving for  $u_p$  with increments of the  $p$  value. The



(a) Time measurements



(b) Speedup relative to 512 cores

Procs	Refs	Number of Elements	DoFs	Newton Its.	Total Lin.Its	Lin.Its. (2.21)
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 3.1.

FIG. 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 (3.2).*

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.

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

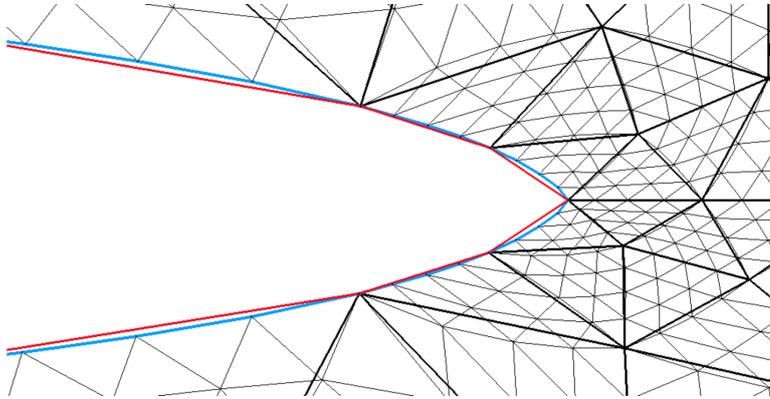


FIG. 8. The base level and the second level of refinement are compared for a 2d simulation for the last step before convergence. The macro and refined triangular elements are shown in bold and thin black lines, respectively. Given that the deformation field is restricted and applied throughout the grid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

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 fewer 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, 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 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 nonconvex 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 this 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] Kitware, *The Visualization Toolkit*, <https://vtk.org>.
- [2] G. ALLAIRE, C. DAPOGNY, AND F. JOUVE, *Shape and topology optimization*, in Geometric Partial Differential Equations—Part II, Handb. Numer. Anal. 22, Elsevier, New York, 2021, <https://doi.org/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*, J. Comput. Phys., 194 (2004), pp. 363–393, <https://doi.org/10.1016/j.jcp.2003.09.032>.
- [4] J. S. ARORA, A. I. CHAHANDE, AND J. K. PAENG, *Multiplier methods for engineering optimization*, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 1485–1525, <https://doi.org/10.1002/nme.1620320706>.
- [5] J. A. BELLO, E. FERNÁNDEZ-CARA, J. LEMOINE, AND J. SIMON, *The differentiability of the drag with respect to the variations of a Lipschitz domain in a Navier–Stokes flow*, SIAM J. Control Optim., 35 (1997), pp. 626–640, <https://doi.org/10.1137/S0363012994278213>.
- [6] C. BRANDENBURG, F. LINDEMANN, M. ULBRICH, AND S. ULBRICH, *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, Internat. Ser. Numer. Math. 160, Birkhäuser, Basel, 2011, pp. 257–275, [https://doi.org/10.1007/978-3-0348-0133-1\\_14](https://doi.org/10.1007/978-3-0348-0133-1_14).
- [7] C. BRANDENBURG, FLORIAN LINDEMANN, MICHAEL ULBRICH, AND STEFAN ULBRICH, *A continuous adjoint approach to shape optimization for Navier Stokes flow*, in Optimal Control of Coupled Systems of Partial Differential Equations, Birkhäuser, Basel, 2009, pp. 35–56, [https://doi.org/10.1007/978-3-7643-8923-9\\_2](https://doi.org/10.1007/978-3-7643-8923-9_2).
- [8] K. DECKELNICK, P. HERBERT, AND M. HINZE, *A Novel  $W^{1,\infty}$  Approach to Shape Optimisation with Lipschitz Domains*, <https://arxiv.org/abs/2103.13857>, 2021.
- [9] M. C. DELFOUR AND J.-P. ZOLÉSIO, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, SIAM, Philadelphia, 2011, <https://doi.org/10.1137/1.9780898719826>.
- [10] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Vol. 1, Oxford University Press, Oxford, 2014.
- [11] F. FEPPON, G. ALLAIRE, AND C. DAPOGNY, *Null space gradient flows for constrained optimization with applications to shape optimization*, ESAIM Control Optim. Calc. Var., 26 (2020), p. 90, <https://doi.org/10.1051/cocv/2020015>.
- [12] A. GEISER, I. ANTONAU, AND K.-U. BLETZINGER, *Aggregated formulation of geometric constraints for node-based shape optimization with vertex morphing (EUROGEN 2021)*, in

- Proceedings of the 14th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, Eccomas Proceedia 14, 2021, <https://doi.org/10.7712/140121.7952.18383>.
- [13] C. GEUZAINÉ AND J.-F. REMACLE, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331, <https://doi.org/10.1002/nme.2579>.
- [14] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer, New York, 1985.
- [15] J. HAUBNER, M. SIEBENBORN, AND M. ULBRICH, *A continuous perspective on shape optimization via domain transformations*, SIAM J. Sci. Comput., 43 (2021), pp. A1997–A2018, <https://doi.org/10.1137/20M1332050>.
- [16] M. HINZE AND K. KUNISCH, *Second order methods for optimal control of time-dependent fluid flow*, SIAM J. Control Optim., 40 (2001), pp. 925–946, <https://doi.org/10.1137/S0363012999361810>.
- [17] M. HINZE, R. PINNAU, M. ULBRICH, AND S. ULBRICH, *Optimization with PDE Constraints*, Math. Model. Theory Appl. 23, Springer, New York, 2009.
- [18] H. ISHII AND P. LORETI, *Limits of solutions of  $p$ -Laplace equations as  $p$  goes to infinity and related variational problems*, SIAM J. Math. Anal., 37 (2005), pp. 411–437, <https://doi.org/10.1137/S0036141004432827>.
- [19] G. KARYPIS, K. SCHLOEGEL, AND V. KUMAR, *Parmetis, Parallel Graph Partitioning and Sparse Matrix Ordering Library*, Version 4.03, 2013, <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview> (6 March 2020).
- [20] B. MOHAMMADI AND O. PIRONNEAU, *Applied Shape Optimization for Fluids*, Oxford University Press, Oxford, 2009, <https://doi.org/10.1093/acprof:oso/9780199546909.001.0001>.
- [21] B. MOHAMMADI AND O. PIRONNEAU, *Shape optimization in fluid mechanics*, Annu. Rev. Fluid Mech., 36 (2004), pp. 255–279, <https://doi.org/10.1146/annurev.fluid.36.050802.121926>.
- [22] 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*, Struct. Multidiscip. Optim., 64 (2021), pp. 3489–3503, <https://doi.org/10.1007/s00158-021-03030-x>.
- [23] S. ONYSHKEVYCH AND M. SIEBENBORN, *Mesh quality preserving shape optimization using nonlinear extension operators*, J. Optim. Theory Appl., 189 (2020), pp. 291–316, <https://doi.org/10.1007/s10957-021-01837-8>.
- [24] J. ORTIZ AND A. SCHIELA, *A Composite Step Method for Equality Constrained Optimization on Manifolds*, Universitat Bayreuth, 2019; also available online from <https://eref.uni-bayreuth.de/47908/>.
- [25] J. PINZON AND M. SIEBENBORN, *Fluid Dynamic Shape Optimization Using Self-Adapting Nonlinear Extension Operators with Multigrid Preconditioners*, <https://arxiv.org/abs/2108.07788v1>, 2021.
- [26] J. PINZON AND M. SIEBENBORN, *PLaplaceOptim*, Version 1.0., 2021, <http://www.github.com/multigridshapeopt> (17 August 2021).
- [27] O. PIRONNEAU, *On optimum profiles in Stokes flow*, J. Fluid Mech., 59 (1973), pp. 117–128, <https://doi.org/10.1017/S002211207300145X>.
- [28] S. REITER, A. VOGEL, I. HEPPNER, M. RUPP, AND G. WITTUM, *A massively parallel geometric multigrid solver on hierarchically distributed grids*, Comput. Vis. Sci., 16 (2013), pp. 151–164.
- [29] A. SCHIELA AND J. ORTIZ, *An SQP method for equality constrained optimization on Hilbert manifolds*, 31 (2021), pp. 2255–2284, <https://doi.org/10.1137/20m1341325>.
- [30] V. SCHULZ AND M. SIEBENBORN, *Computational comparison of surface metrics for PDE constrained shape optimization*, Comput. Methods Appl. Math., 16 (2016), pp. 485–496, <https://doi.org/10.1515/cmam-2016-0009>.
- [31] J. SOKOLOWSKI AND J.-P. ZOLÉSIO, *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer, Berlin, 1992, <https://doi.org/10.1007/978-3-642-58106-9>.
- [32] M. ULBRICH, *Constrained optimal control of Navier-Stokes flow by semismooth Newton methods*, Syst. Control Lett., 48 (2003), [https://doi.org/10.1016/S0167-6911\(02\)00274-8](https://doi.org/10.1016/S0167-6911(02)00274-8).
- [33] H. UZAWA, *Iterative methods for concave programming*, Stud. Linear Nonlinear Program., 6 (1958), pp. 154–165.
- [34] A. VOGEL, S. REITER, M. RUPP, A. NÄGEL, AND G. WITTUM, *UG 4: A novel flexible software system for simulating PDE based models on high performance computers*, Comput. Vis. Sci., 16 (2013), pp. 165–179.

## **PAPER III**

# Shape optimization in $W^{1,\infty}$ with geometric constraints: a study in distributed-memory systems

Philip J. Herbert<sup>1</sup>, Jose A. Pinzon Escobar<sup>2</sup>, and Martin Siebenborn<sup>2</sup>

<sup>1</sup>Department of Mathematics, University of Sussex, Brighton, BN1 9RF, United Kingdom

<sup>2</sup>Department of Mathematics, University Hamburg, Bundesstr. 55, 20146 Hamburg, Germany

September 28, 2023

## Abstract

In this paper we present a shape optimization scheme which utilizes the alternating direction method of multipliers (ADMM) to approximate a direction of steepest descent in  $W^{1,\infty}$ . The followed strategy is a combination of the approaches presented in Deckelnick, Herbert, and Hinze, ESAIM: COCV 28 (2022) and Müller et al. SIAM SISC 45 (2023). This has appeared previously for relatively simple elliptic PDEs with geometric constraints which were handled using an ad-hoc projection. Here, however, the optimization problem is expanded to include geometric constraints, which are systematically fulfilled. Moreover, this results in a nonlinear system of equations, which is challenging from a computational perspective. Simulations of a fluid dynamics case study are carried out to benchmark the novel method. Results are given to show that, compared to other methods, the proposed methodology allows for larger deformations without affecting the convergence of the used numerical methods. The mesh quality is studied across the surface of the optimized obstacle, and is further compared to previous approaches which used descents in  $W^{1,p}$ . The parallel scalability is tested on a distributed-memory system to illustrate the potential of the proposed techniques in a more complex, industrial setting.

**Keywords:** PDE-constrained shape optimization, Lipschitz transformations,  $W^{1,\infty}$ -descent, ADMM, parallel computing

**MSC codes:** 35Q93, 49Q10, 65Y05, 65K10

## 1 Introduction

In this paper we investigate the efficient optimization of shapes where the optimization must take geometrical constraints into consideration. We are particularly interested in so-called PDE-constrained shape optimization, where the objective functional, which depends on a domain, also depends on the solution of a PDE within that domain. Shape optimization is well studied and we refer to [35, 7, 11] for an overview. Here, special interest is placed in the case where only a subset of the boundary should be deformed, the so-called obstacle, and where it should be possible to experience large deformations with minimal degradation of the mesh. The example considered here is that of an object, which we will refer to as an obstacle, inside a flow tunnel. Inside this flow tunnel the flow is described by a PDE, such as Stokes or Navier–Stokes, and the surface of the obstacle is optimized for a given functional. We will consider only the case where the PDE constraint is given by the non-linear stationary Navier–Stokes equations and the functional is the energy dissipation. In this setting, the obstacle should maintain a fixed volume and barycenter, otherwise the minimizer would be no obstacle. There is widespread interest in this topic, for instance [28, 29] and the algorithmic approaches considered in [20, 32]. This problem is classically studied and the solution is described in [27, 26] as being a prolate-spheroid or rugby ball, given that

tips have to formed on the surface of the object as part of the optimization process. Additionally, this problem was explored more recently in a shape optimization context in [19, 4, 30].

In shape calculus, one is often interested in using a descent method to obtain a series of shapes which should approximate a minimizer of the objective,  $J$ . This requires the computation of the first derivative of the objective with respect to the domain, which we will refer to as the shape derivative,  $J'$ . A widely used approach to finding a descent direction is based on Hilbert spaces. In such an approach, one relates the shape derivative  $J'$  to the so-called shape gradient  $\nabla J$  by use of Riesz representation in the topology of the Hilbert space. This is the case, for instance, in [33, 25]. A variety of this is the so-called extension equation approach, [22, 20, 12, 33], where an elliptic PDE is solved to find a deformation field in  $H^1$ . For further details, we refer to the overview article [1] and the bibliography within.

Recent trends have involved the use of Lipschitz, or  $W^{1,\infty}$ , functions and their approximation by  $W^{1,p}$  functions. The case of  $W^{1,\infty}$ , to the best of our knowledge, was first introduced in a practical setting in [5] where it was restricted to the optimization of star-shaped domains. Generic, non-star shaped domains were considered in [6] using deformations in  $W^{1,\infty}$ . The approximation with functions in  $W^{1,p}$  is based on *relaxing* the space and choosing  $p$  *large enough* so that one is, in some sense, close to being a minimizer in  $W^{1,\infty}$ . A method which handled geometric constraints by using an augmented Lagrangian method was introduced in [20]. In order to handle *large* values of  $p$ , an iterative approach which incrementally increased  $p$  was used. The convergence, optimal achievable shape, and mesh quality were compared to a Hilbert space method approach. In [21], an algorithm for handling geometric constraints of the integral type is combined with the  $p$ -Laplace relaxation to find descent directions that preserve volume and barycenter.

For many applications, the initial configuration must go through large deformations to reach an optimal shape. While, mathematically and in the real world, there is no mesh, for computational methods, it is often utilized. Large deformations can lead to a loss of mesh quality, i.e. degeneration of discrete grid elements. Degeneration of elements has a negative impact on the convergence of iterative solvers as well as the approximation properties of the discrete solution. It is therefore important to take into account the mesh quality, if remeshing is to be prevented. Preservation mechanisms are often taken into account in computational methods. In [22, 24, 12], a constraint on the determinant of the deformation gradient was imposed, and enforced using an augmented Lagrangian. The incorporation of such constraints may reduce the space of attainable shapes.

This article firstly wishes to build upon the work of [21], which considers computational scaling to utilize the steepest descent methods in  $W^{1,\infty}$  discussed in [6]. Secondly, we investigate, to a limited extent, the quality of the produced mesh. Finally we demonstrate the computational scalability of the method, which is in no way guaranteed from the previous work which studied  $W^{1,p}$  as the scheme is entirely different.

The article is structured as follows: In Section 2, we present the essential background for shape optimization in  $W^{1,\infty}$  [6], as well as the benchmark physical model which will be the subject of study. Section 3 introduces the algorithms used within the novel components in this work. Simulation results for the  $W^{1,\infty}$  methodology appear in Section 4. We present in Section 5 a comparison between the resulting meshes and objective function for  $W^{1,\infty}$  and  $W^{1,p}$  approaches. Finally, the parallel scalability for the  $W^{1,\infty}$  method is measured in Section 6.

## 2 Shape optimization in $W^{1,\infty}$ and the model problem

This and the following sections build heavily on the work presented in [21, 5, 6]. An in-depth discussion of the theoretical aspects of shape optimizations is not within the scope of this work, therefore interested readers are referred to well-known monographs such as [35, 7, 13]. It is however necessary to recount a few aspects in order to build towards the methodology described in Section 3.

### 2.1 Shape Optimization

The task of (PDE constrained) shape optimization is to, given a collection of admissible domains  $\mathcal{S}_{ad}$  and a functional on those domains  $J: \mathcal{S}_{ad} \rightarrow \mathbb{R}$ , find  $\Omega^* \in \mathcal{S}_{ad}$  which attains the minimum. Typically  $\mathcal{S}_{ad}$  will be a collection of open and bounded domains domains in  $\mathbb{R}^d$  for  $d \in \{2, 3\}$ . It

may also be the case that one wishes to restrict the minimization to be over domains which satisfy a (geometric) constraint, say  $g(\Omega) = 0$ .

Finding minimizers in practice is a difficult task. Iterative methods are often used to find stationary points. A standard strategy is to, given the current domain  $\Omega$ , update the domain  $\Omega_{new} := \Omega(u) := \{x + u(x) : x \in \Omega\}$  for some suitably chosen  $u$ . For convenience, we denote the perturbation of the identity by  $F := \text{id} + u$ , so that  $\Omega_{new} = F(\Omega)$ . This means that our iterates yield a natural parameterization over the initial domain.

The task of choosing  $u$  in practice is non-trivial and many such choices can be made, as discussed in the introduction. In this work, we will consider the steepest descent in  $W^{1,\infty}$ . Where possible, parallels are drawn to the  $p$ -Laplace optimization scheme proposed in [20, 21]. A particular reason for using  $W^{1,\infty}$  functions is the following fact: When  $\Omega$  is convex and  $|Du| < 1$  a.e. in a sub-multiplicative norm, then the map  $F$  is bi-Lipschitz. Where  $Du$  is the Jacobian matrix of  $u$ . Two examples of sub-multiplicative norms are the spectral and Frobenius norms. In the case that  $\Omega$  is not convex, this bi-Lipschitz property need not hold, however one has that for any convex subset  $K$  of  $\Omega$ , the restriction  $F|_K$  is bi-Lipschitz onto its image. In [6] it is suggested to account for potential non-convexity by the use of a fictitious domain. For the situation we consider, this is not done. In the experiments the constructed maps remain bi-Lipschitz.

To formulate a direction of steepest descent, it is useful to have a derivative. We say that  $J$  is shape differentiable at  $\Omega$  if the map  $u \mapsto J(\Omega(u))$  is suitably differentiable at  $u = 0$  in  $W^{1,\infty}$ . We denote the directional derivative as  $J'(\Omega)u := \lim_{t \rightarrow 0} \frac{J(\Omega(tu)) - J(\Omega)}{t}$ . For the problem we consider, the derivative  $J'$  is well known, however one may wish to use the so-called Lagrange multiplier method to calculate it, see [15] for example. In order to handle the geometric constraints, we will make use of a non-linear space to find the steepest descent so that at each iteration the constraints are fulfilled to a given tolerance rather than approximated, which is the case when using, for example, an augmented Lagrange approach. This direction of descent is found as: given  $\sigma \in (0, 1)$  find

$$\begin{aligned} \min_{u \in W^{1,\infty}(\Omega; \mathbb{R}^d)} \quad & J'(\Omega)u \\ \text{s.t.} \quad & u|_{\partial D} = 0, \\ & \|Du\|_{L^\infty(\Omega)} \leq \sigma, \\ & g(\Omega(u)) = 0, \end{aligned} \tag{1}$$

where  $D$  is a sufficiently smooth hold-all domain which will be introduced later. It is worth mentioning that due to the geometric constraints,  $J'(\Omega)u$  need not equal  $-\|J'(\Omega)\|$  as in e.g. [6], where

$$\|J'(\Omega)\| := \sup\{J'(\Omega)\tilde{u} : \tilde{u} \in W^{1,\infty}(\Omega; \mathbb{R}^d), \tilde{u}|_{\partial D} = 0, \|D\tilde{u}\|_{L^\infty(\Omega)} \leq 1, g'(\Omega)\tilde{u} = 0\} \tag{2}$$

is the (dual) norm of  $J'(\Omega)$  on the subspace which corresponds to constraining  $g$ . One may however expect that  $J'(\Omega)u/\sigma$  converges to  $-\|J'(\Omega)\|$  as  $\sigma \rightarrow 0$ . Such an expectation comes from the notion of the Hadamard derivative. The existence of such a  $u$  in the continuous has not yet been developed, but will be considered in upcoming work. Let us note that obtaining a function satisfying (1) is not necessarily trivial; the construction we specifically use in the experiments is given by an ADMM. Full details are given in Section 3.3

Now that we have discussed the shape optimization framework we are interested in, let us discuss the PDE constraint.

## 2.2 Physical model

We are interested in the case that the energy  $J$  is given by

$$J(\Omega) = \frac{\nu}{2} \int_{\Omega} Dv : Dv \, dx, \tag{3}$$

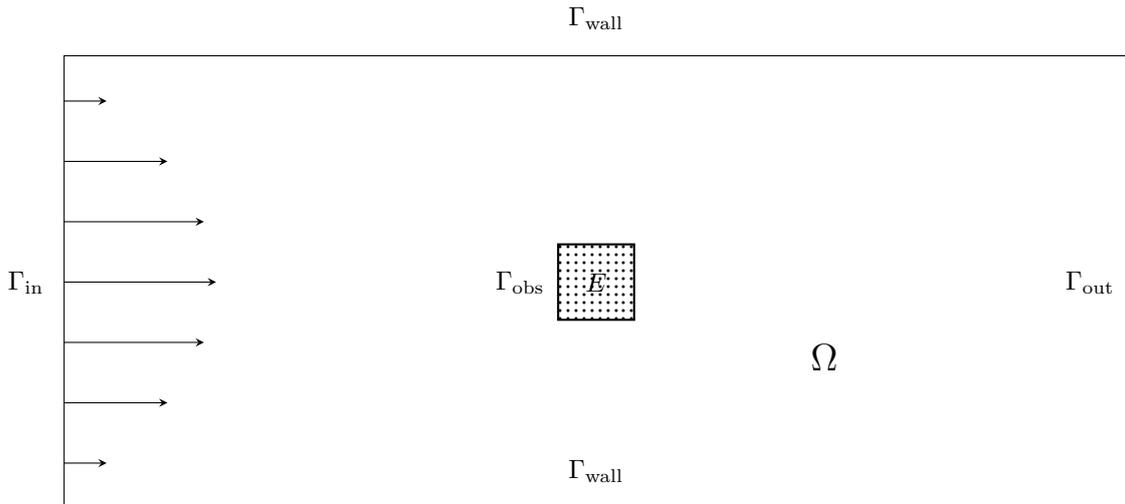


Figure 1: A 2d view of the used domain.

where  $\nu > 0$  is given and, for a given prescribed inflow  $v_\infty$ , the velocity  $v$  weakly satisfies the incompressible stationary Navier–Stokes equations

$$\begin{aligned}
 -\nu\Delta v + (v \cdot \nabla)v + \nabla \mathbf{p} &= 0 \text{ in } \Omega \\
 \operatorname{div} v &= 0 \text{ in } \Omega \\
 v &= 0 \text{ on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \\
 v &= v_\infty \text{ on } \Gamma_{\text{in}} \\
 \nu Dv \cdot n &= \mathbf{p}n \text{ on } \Gamma_{\text{out}}.
 \end{aligned} \tag{4}$$

The regions  $\Gamma_{\text{obs}}$ ,  $\Gamma_{\text{wall}}$ ,  $\Gamma_{\text{in}}$ , and  $\Gamma_{\text{out}}$  are to be described shortly. In this setting,  $\nu$  may be referred to as the kinematic viscosity, and  $v$ , the velocity. We will assume that the solution pair  $(v, \mathbf{p})$  is unique, up to an additive constant for the pressure  $\mathbf{p}$ .

We are interested in the case of an open and bounded domain  $D \subset \mathbb{R}^d$ , where again  $d \in \{2, 3\}$ , which will take the physical interpretation of a flow tunnel. Inside this flow tunnel we consider an obstacle, represented by an open and simply connected set  $E \subset D$ . The obstacle  $E$  should satisfy  $\partial E \cap \partial D = \emptyset$ , i.e. the obstacle should not touch the boundary of the flow tunnel. We are then interested in optimizing the domain  $\Omega := D \setminus \bar{E}$ . For convenience we denote  $\Gamma_{\text{obs}} := \partial E$ , the boundary of the obstacle. This has, in words, described the collection  $\mathcal{S}_{ad}$ . The full details of the necessary regularity of the domains is beyond the scope of this work. A sketch of the flow tunnel appears in Figure 1.

Let us note that the way in which we describe the domain  $\Omega$  means that one has that  $\partial\Omega = \partial D \cup \Gamma_{\text{obs}}$ , where we are not to perturb the flow tunnel  $D$ , only the obstacle i.e.  $\Gamma_{\text{obs}}$ .

Let us now discuss the physical motivation for the constraints which we consider. It is expected that reducing the volume of  $E$  would lead to a lower value for (3). For instance,  $E$  could vanish, leading to a degenerate minimizer. Furthermore, if the obstacle were to move to an area which is less impactful on the overall energy, e.g. closer to  $\Gamma_{\text{wall}}$ , where the velocity is vanishing, or downstream within the flow tunnel, one may again expect a reduction in value. To prevent these cases, geometric constraints are imposed on  $\Omega$ . These constraints are on the (normalized) barycenter and volume, and are given by

$$g_i(\Omega(u)) := \int_{\Omega} (F_i \det(DF) - \text{id}) \, dx = 0 \text{ for } i = 1, \dots, d, \tag{5}$$

$$g_{d+1}(\Omega(u)) := \int_{\Omega} (\det(DF) - 1) \, dx = 0, \tag{6}$$

which represent the difference between the reference configuration, described via the method of mappings, and the deformed domain. These expressions are used under the assumption that the barycenter is set to the origin, and that the volume remains constant.

### 3 Optimization Method

In this section, the algorithmic approach to finding a minimizing shape is described. The algorithm we are using is relatively straightforward besides finding descent directions in  $W^{1,\infty}$ . The majority of the section is dedicated to the description of how we choose to find this steepest descent. We explain the methodology used to approximate a solution to the optimization problem formulated in (1), which relies on the computation of the shape derivative  $J'(\Omega)u$  to formulate a descent-like method [1]. As mentioned before, the proposed methodology uses the ADMM, as in [6], to approximate a deformation field which preserves the mesh topology between shape iterates. Nevertheless, it is worth noting that the following optimization scheme is in no way limited to the use of the ADMM, but could be adapted to other approaches to find  $u$  in  $W^{1,\infty}$ . The task of finding a potentially better-suited solver is not only interesting, but an ongoing topic of research. In the following, the implemented routines are described, starting with the descent-like method and followed by the ADMM in the context of an iterative shape optimization scheme.

Henceforth, we will deal only with discrete quantities, that is to say, our shape is now represented by the moving mesh with triangulation  $\mathcal{T}_h$ . On that mesh we approximate the solution of stationary Navier–Stokes equation by means of a finite element discretization with lowest order Taylor-Hood elements, and the energy is computed using this finite element solution.

#### 3.1 Shape optimization algorithm

The main routine is outlined in Algorithm 1, where a descent-like method is used to obtain an optimized geometry from a given arbitrarily shaped initial guess,  $\Omega^0$ . Within the outer loop of Algorithm 1, a series of shape iterates is generated by computing a deformation vector  $u$  and applying it over the nodes of the current discrete domain  $\Omega^k$  to propose a domain  $\hat{\Omega}$ , as described in Section 2. The resulting geometry is carried on to the next step if  $J(\hat{\Omega}) < J(\Omega^k)$ , which requires solving the state equation and computing the objective function. On the contrary, if the aforementioned condition is not fulfilled then  $\sigma$  is reduced as indicated in line 13.

---

#### Algorithm 1 Shape Optimization-Descent method

---

**Require:** Initial shape  $\Omega^0, \epsilon_1, \epsilon_2, \sigma, N$

- 1:  $k = 0$
- 2:  $u^k \leftarrow 0$
- 3:  $y^k \leftarrow$  Solve state (4)
- 4: Compute  $J(\Omega)^k$
- 5: **for**  $k=0,1,\dots,M$  **do**
- 6:   Compute the shape derivative  $J'(\Omega^k)$  using the adjoint method
- 7:   Modification of  $J'(\Omega^k)$  as in Section 3.2
- 8:    $u^k \leftarrow$  ADMM( $J'(\Omega), \sigma, \epsilon_2, \epsilon_3, N$ ) as in Section 3.3
- 9:   Temporary geometry  $\hat{\Omega} = \Omega^k(u^k)$
- 10:    $y \leftarrow$  Solve state equation  $e$  on  $\hat{\Omega}$
- 11:   Compute objective function  $J(\hat{\Omega})$
- 12:   **if**  $J(\Omega^k) \leq J(\hat{\Omega})$  **then**
- 13:      $\sigma \leftarrow \sigma/2$
- 14:   **else**
- 15:      $y^{k+1} \leftarrow y$
- 16:      $\Omega^{k+1} \leftarrow \hat{\Omega}$
- 17:      $k = k + 1$
- 18:   **end if**
- 19: **end for**

---

The core of Algorithm 1 lies on finding the descent direction  $u$ , shown in line 8. This requires the solutions to the state equation (4) and its adjoint, for the computation of the shape derivative  $J'(\Omega)$ . The solution  $y$  is used to compute the objective function on the temporary geometry, which determines if a descent direction has been found.

The deformation field constitutes a descent direction such that, for the solution of the equation system (15),  $J'(\Omega)u < 0$  on every step. If one were to have unlimited computational time, one might choose to use the descent as the convergence criterion. Where we recall that  $J'(\Omega)u$ , when rescaled, approximates the directional derivative. In such a situation, for some  $\epsilon_1 > 0$ , one might choose to run the outer loop until

$$| -J'(\Omega^k)u^k | < \sigma\epsilon_1, \quad (7)$$

which would, for  $\sigma$  small, roughly correspond to  $\|J'(\Omega^k)\| < \epsilon_1$ .

### 3.2 Modification of the shape derivative in the interior of the mesh

It is well known by the Hadamard structure theorem that for sufficiently regular domains, the shape derivative  $J'(\Omega)$  should be supported only at the boundary, e.g. [7, Section 3.4, Theorem 3.6], moreover that it should be a measure. However, it is not the case that the shape derivative evaluated on the solution of the discrete state equation is supported at the boundary only. This may be attributed to the finite element approximation errors - see [16] who discuss the error due to different forms of the shape derivative. It is suggested in [34] that one can remove the values which have contributions only in the interior. Such an approach has been noted to produce qualitatively better results [31]. Computationally, this corresponds to only assembling (integrating) the shape derivative over cells which have a vertex which intersects the boundary  $\Gamma_{\text{obs}}$ . This is performed in line 7 of Algorithm 1.

Mathematically speaking, since we are expecting the shape derivative to be a measure, we may represent this by the discrete functional which is a sum of Dirac deltas over the vertices of the mesh. The aforementioned expert knowledge then says that the relevant nodes are those which are at the boundary of the obstacle. This recovers the method of [34].

### 3.3 The ADMM for the solution of the steepest descent

The minimization problem for the direction of steepest descent in (1) is highly non-trivial, both in terms of the constraint on the Jacobian, but also the geometric constraints. To handle the Jacobian condition, inspired by [2], [6] utilized the ADMM to find directions of steepest descent for PDE-constrained shape optimization problems. This approach is here extended to include geometric constraints, building on the scheme presented in [21]. In there, a methodology to incorporate the geometric constraints was proposed with the intention of avoiding an augmented Lagrangian.

The function which we will consider for the problem in (1) is given as

$$L(u, \psi_g, q, \lambda) := J'(\Omega)u + \sum_{i=1}^{d+1} \psi_{g,i} g_i(\Omega(u)) + \frac{\tau}{2} \|Du - q\|_{L^2}^2 + \tau \|Du - q, \lambda\|_{L^2}, \quad (8)$$

where the last two terms appeared in [6] as part of the ADMM augmented Lagrangian, cf. [3, 2]. Here,  $\psi_{g,i}$  are components of the finite dimensional vector  $\psi_g \in \mathbb{R}^{d+1}$  of Lagrange multipliers associated to the geometric constraints (6) and (5). In addition to the expected deformation and Lagrange multipliers,  $u$  and  $\psi_g$ , we also have the appearance of  $q$  and  $\lambda$  in  $L$ . Here  $q$  takes the role of a slack variable, and  $\lambda$  of a Lagrange multiplier which pushes  $q$  towards  $Du$ .

The ADMM strategy we consider is provided in Algorithm 2. It operates by alternately minimizing  $L$  over  $q \in Q_h$  and  $u \in V_h$  such that the geometric constraints are fulfilled, then a simple update is used for the multiplier  $\lambda$ . The iterative process is repeated until the convergence criteria, line 11 are fulfilled. The condition  $\|\Delta_\lambda\|_{L^2}^2 + \|\Delta u\|_{L^2}^2 < \epsilon_2$  is similar to that used in [2, 6], and represents the residual of the updates. To ensure that  $\|Du\|_{L^\infty}$  is as close as possible to  $\sigma$ , we introduce an additional condition on the convergence. The loop stops when  $\|Du\|_{L^\infty}$  is also close to the given  $\sigma$ , up to a small value  $\epsilon_3 > 0$ , which may depend on  $\sigma$  itself.

Let us discuss the ad-hoc modification which has been made in lines 12-14 of Algorithm 2. When  $\|J'(\Omega)\|$  is not equal to zero, we expect the deformation field  $u$  to satisfy  $\|Du\|_{L^\infty}$  is close to  $\sigma$  - in the setting without the geometric constraints, these quantities are equal. In order to avoid degenerate solutions, the vector which stores the shape derivative  $J'(\Omega)$  has its components doubled. This could be compared to the variable step-size ADMM presented in [3], where the value

$\tau$  is determined as part of the algorithm. Here however, this has not been undertaken to speed up the convergence, but to ensure an appropriate deformation field is found.

---

**Algorithm 2** Descent Direction in  $W^{1,\infty}$

---

**Require:**  $J'(\Omega), \sigma, \epsilon_2, \epsilon_3, N$

```

1:  $\tau \leftarrow 1$ 
2:  $u_0 \leftarrow 0$ 
3: for  $i = 0, 1, \dots, N$  do
4:   Find  $q \leftarrow \arg \min\{L(u, q, \lambda) : q \in Q_h, |q| \leq \sigma\}$  as in Section 3.3.1
5:   Find  $u \leftarrow \arg \min\{L(u, q, \lambda) : u \in V_h, g(\Omega(u)) = 0\}$  as in Section 3.3.2
6:    $\Delta_\lambda \leftarrow \tau(Du - q)$ 
7:    $\lambda \leftarrow \lambda + \Delta_\lambda$ 
8:    $\Delta_\sigma \leftarrow \sigma - \max(|Du|_{L^\infty})$ 
9:    $\Delta u \leftarrow u - u_0$ 
10:   $u_0 \leftarrow u$ 
11:  if  $(\|\Delta_\lambda\|_{L^2}^2 + \|\Delta u\|_{L^2}^2 < \epsilon_2)$  and  $\Delta_\sigma > -\epsilon_3$  then
12:    if  $\Delta_\sigma > \epsilon_3$  then
13:       $J'(\Omega) \leftarrow 2J'(\Omega)$ 
14:    else
15:      break
16:    end if
17:  end if
18: end for

```

---

Within every iteration of Algorithm 2, Newton's method is called to solve the nonlinear optimality system, as shown in line 5. This is described in Section 3.3.2. The strategy is based on that which is described in [21, Section 3], where the Schur complement operator is computed by using  $d + 1$  numerical solves. The computational cost is reduced by using preconditioners with grid-independent convergence, such as the geometric multigrid.

Let us expand further on the minimization with respect to  $q$  and the saddle point for  $(u, \psi_g)$  corresponding to minimization of  $u$  under the geometric constraints which are utilized for the ADMM algorithm.

### 3.3.1 Optimality for $q$

For the system involving  $q$ , it is useful to denote the discrete space

$$Q_h := \{q \in L^2(\Omega; \mathbb{R}^{d \times d}) : q|_T \in \mathbb{P}^0(T; \mathbb{R}^{d \times d})\}, \quad (9)$$

where we note that this is not the only possible discrete space one could choose for  $q$ , however it retains a level of simplicity. The optimality system for  $q$  is given by

$$\begin{aligned} \frac{\partial}{\partial q} L(u, \psi_g, q, \lambda)(\delta_q - q) &\geq 0 \quad \forall \delta_q \in Q_h : |\delta_q| \leq \sigma \\ |q| &\leq \sigma \end{aligned} \quad (10)$$

which is a convex minimization problem. For convenience, we note that

$$\frac{\partial}{\partial q} L(u, \psi_g, q, \lambda) \delta_q = \tau \int_{\Omega} (q - Du - \lambda) : \delta_q \, dx. \quad (11)$$

This may be decomposed into two steps, solving

$$\frac{\partial}{\partial q} L(u, \psi_g, q, \lambda)(\delta_q - \tilde{q}) = 0 \quad \forall \delta_q \in Q_h \quad (12)$$

which, observing (11), is seen to correspond to a mass solve, and a pointwise projection

$$\tilde{q} \mapsto \frac{\tilde{q}}{\max\left(1, \frac{|\tilde{q}|}{\sigma}\right)} =: q. \quad (13)$$

### 3.3.2 Optimality for $(u, \psi_g)$

For the optimality system involving  $(u, \psi_g)$ , it is convenient to introduce

$$V_h := \{u \in C^0(\bar{\Omega}; \mathbb{R}^d) : u|_T \in \mathbb{P}^1(T; \mathbb{R}^d), T \in \mathcal{T}_h, u|_{\partial D} = 0\} \quad (14)$$

which is the natural space for deforming a mesh which has flat triangles. The optimality system for the saddle point for  $(u, \psi_g)$  is

$$\begin{aligned} \frac{\partial}{\partial u} L(u, \psi_g, q, \lambda) \delta_u &= 0 & \forall \delta_u \in V_h \\ \frac{\partial}{\partial \psi_g} L(u, \psi_g, q, \lambda) \delta_{\psi_g} &= 0 & \forall \delta_{\psi_g} \in \mathbb{R}^{d+1}, \end{aligned} \quad (15)$$

which corresponds to a (discrete) Poisson problem with non-linear constraints. Following the strategy proposed in [21], the system is solved via its linearization in the context of Newton's method. A particular advantage in our setting is that the second derivative of  $L$  with respect to  $u$  is more convenient to handle:

$$\frac{\partial^2}{\partial u^2} L(u, \psi_g, q, \lambda) (\delta_u, \mu_u) = \tau \int_{\Omega} D\delta_u : D\mu_u + \sum_{i=1}^{d+1} \langle g_i''(F(\Omega)) \delta_u, \mu_u \rangle. \quad (16)$$

The advantage becomes evident by contrasting (16) to the first integral term of the same derivative presented in [21], the second variation of the  $p$ -Laplace energy, which is a degenerate elliptic operator. In the  $p$ -Laplace case,  $L_{uu}$  was modified to prevent divide-by-zero operations. Here, this is not necessary.

To solve the saddle point system which appears in (15), the increments due to the Newton method may be expressed as

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

where, for readability purposes, we write

$$\begin{aligned} A\delta_u &:= \frac{\partial^2}{\partial u^2} L(u^k, \psi_g^k, q^k, \lambda^k)(\cdot, \delta_u) \\ B\delta_{\psi_g} &:= \frac{\partial}{\partial u} \frac{\partial}{\partial \psi_g} L(u^k, \psi_g^k, q^k, \lambda^k)(\cdot, \delta_{\psi_g}) \\ B^T \delta_u &:= \frac{\partial}{\partial \psi_g} \frac{\partial}{\partial u} L(u^k, \psi_g^k, q^k, \lambda^k)(\delta_u, \cdot) \\ r_u &:= -\frac{\partial}{\partial u} L(u^k, \psi_g^k, q^k, \lambda^k) \\ r_{\psi_g} &:= -\frac{\partial}{\partial \psi_g} L(u^k, \psi_g^k, q^k, \lambda^k). \end{aligned} \quad (17)$$

Let us comment that the Schur complement operator,  $S := -B^T A^{-1} B$ , is related to the computation of the  $\psi_g$  Lagrange multipliers.

## 4 Results

We present simulation results for 2d and 3d fluid dynamics case studies, where the domain is as described in Figure 1. The initial obstacle  $E$  is set as a square, or a box, in 2d and 3d, respectively. This showcases the successful creation and removal of geometric singularities through the optimization process. Results are shown to highlight the large deformations present, particularly at the initial steps, in 2d. This can be compared to the results presented in similar studies, e.g. [22, 12, 21], to mention some. Mainly, emphasis is placed on the preservation of mesh quality even under the aforementioned large deformations within a single optimization step.

These results are generated using UG4 [36], a simulation framework tailored for distributed-memory systems [37, 38]. Most importantly we make use of its geometric multigrid preconditioner [10] for the iterative methods used within this work. Our models were implemented using UG4's plugin functionality, and the performance was measured with its builtin profiler. The parallel communication is MPI-based [8]. The computational grids were created with GMSH [9], using triangular (2d) and tetrahedral (3d) elements. ParMetis [18] was used for grid partitioning.

We recall that the state equation is described by the weak form of the incompressible Navier-Stokes equations and is discretized via the mixed Taylor-Hood finite element, with the lowest order elements  $P_2 - P_1$ . The adjoint equation is similarly discretized with this. The viscosity is set to  $\nu = 0.02$  for all simulations. The dimensions of the flow tunnel, the holdall domain, for 2d and 3d are

$$D_{2d} = (-7, 7) \times (-3, 3) \text{ and } D_{3d} = (-10, 10) \times (-3, 3) \times (-3, 3),$$

respectively. The obstacles are given by

$$E_{2d} = (-0.5, 0.5) \times (-0.5, 0.5) \text{ and } E_{3d} = (-0.5, 0.5) \times (-0.5, 0.5) \times (-0.5, 0.5)$$

and are not triangulated, since we are interested primarily on the optimization of its outer surface  $\Gamma_{\text{obs}}$ . At  $\Gamma_{\text{in}}$  the inflow profile is described by

$$v_\infty(x) = e_1 \prod_{i=2}^d \cos\left(\frac{\pi}{3} x_i\right) \in \mathbb{R}^d$$

where  $e_1 := (1, 0, \dots) \in \mathbb{R}^d$ . Note that  $v_\infty \cdot e_1 \in [0, 1]$  on  $\Gamma_{\text{in}}$ .

We recall that (15) is approximated with vector-valued  $P_1$  finite elements. Following the methodology described in [21], the Lagrange multipliers of the geometric constraints,  $\psi_g$ , are not associated to a finite element discretization. As described in Section 3, their solution is obtained via a direct solver of an  $m \times m$  system of equations, i.e. the Schur complement system  $S$ .

The code used for the simulation can be found in [23].

## 4.1 Simulations in 2d

In these a studies, a computational mesh with 282 624 triangular elements was used. The surface of the obstacle consists of 512 edges. We utilize the spectral norm for (1), in this setting. In order to reduce the dissipated energy, the deformations must remove the preexisting geometric singularities, which in this case are the corners of  $\Gamma_{\text{obs}}$ . At the same time, the necessary tips must be created parallel to the direction of the flow.

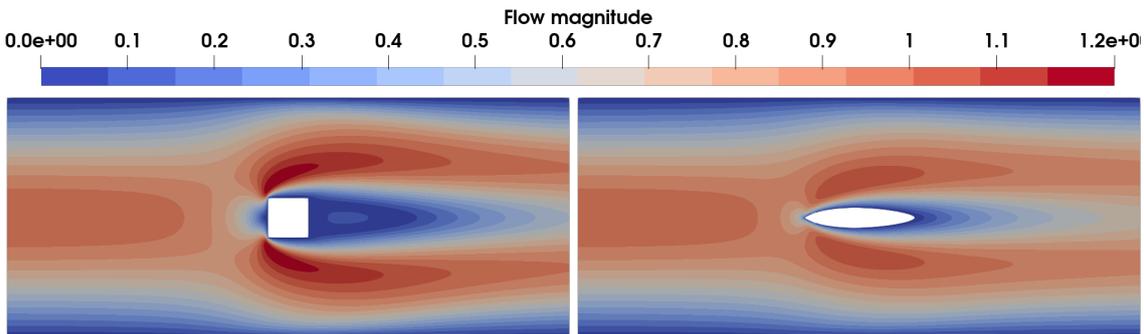


Figure 2: 2d simulation results for a 282 624 element grid. The streamlines are shown over the initial (top) and optimized (bottom)  $\Gamma_{\text{obs}}$  configurations.

The optimization process of Algorithm 1 is illustrated by the results given in Figure 2, where the removal and creation of grid singularities is evident. This optimized object surface corresponds to 50 steps of the outer loop of the shape optimization method of Section 3. The resulting grid for these simulations appears later in Section 5 and is shown in the upper row of Figure 9. It shows the

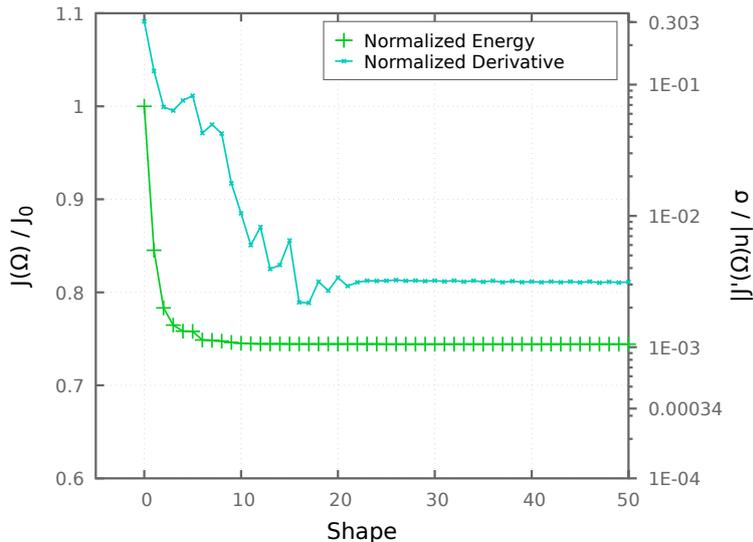


Figure 3: The objective function  $J(\Omega)$  divided by the initial value -  $J_0$  - over 50 steps is shown, together with the convergence criterion (7).

regions of the surface  $\Gamma_{\text{obs}}$  where singularities were created and removed. In the figure, the front tip is featured, together with the area corresponding to the upper left corner of the obstacle in the reference configuration. As previously mentioned, tips must be generated parallel to the direction of the flow and the corners must be smoothed out. These show that, although the obstacle's surface experiences very large deformations already on the initial deformation steps, the elements retain their shape and, since the  $W^{1,\infty}$  deformation is assured to yield a bi-Lipschitz map, no overlapping between elements occurs.

The plot in Figure 3 shows the energy dissipation divided by the value calculated on the initial configuration,  $J_0 := J(\Omega_0)$ . Together with the objective function, a scaling of the absolute value of the directional derivative is plotted. Let us recall the discussion after (7) which comments that this rescaled directional derivative can potentially be related to the norm of the derivative. The observed monotonic decrease of  $J(\Omega)$  is related to the condition in Algorithm 1, where it is checked whether the new geometry  $\hat{\Omega}$  obtained from the calculated descent direction reduces  $J(\Omega)$ . During the first steps, a very large reduction of the objective function and the derivative can be observed. The energy dissipation is reduced in the range of 25% from the initial value, while the directional derivative is reduced several orders of magnitude, which is to be expected.

We provide in Figure 4 a few of the shape iterates. These demonstrate the large deformations which the optimization scheme provides in the first steps. In the first column, shape 5 already shows the removal of the preexisting corners of  $\Gamma_{\text{obs}}$ , as well as how the obstacle stretches across the direction of the flow. This can be observed in detail in the next columns, where the elements across the upper left corner and the front tip are shown. It is observed that large deformations are possible without provoking a failure of the numerical solver due to highly degenerate elements. Moreover we wish to emphasize, besides solver tolerances, there are no quantities to tune, especially quantities which are designed to preserve mesh quality.

## 4.2 Simulations in 3d

For the 3d case a grid with 622 592 tetrahedral elements was used. The surface of the obstacle consists of 12 288 triangular elements. The derivative  $J'$  is scaled by a factor of  $t = 0.1$  for the computation of descent direction. Moreover, for computational reasons, we utilize the Frobenius norm for (1) in this setting. The outer loop in Algorithm 1 is again set to stop after 50 iterations.

The initial and final step of a 3d simulation are shown in Figure 5. The streamlines show the regions on the initial grid where there is a disruption in the flow, leading to a higher energy dissipation. As in the 2d case, the edges and corners of the box must be removed. This is shown

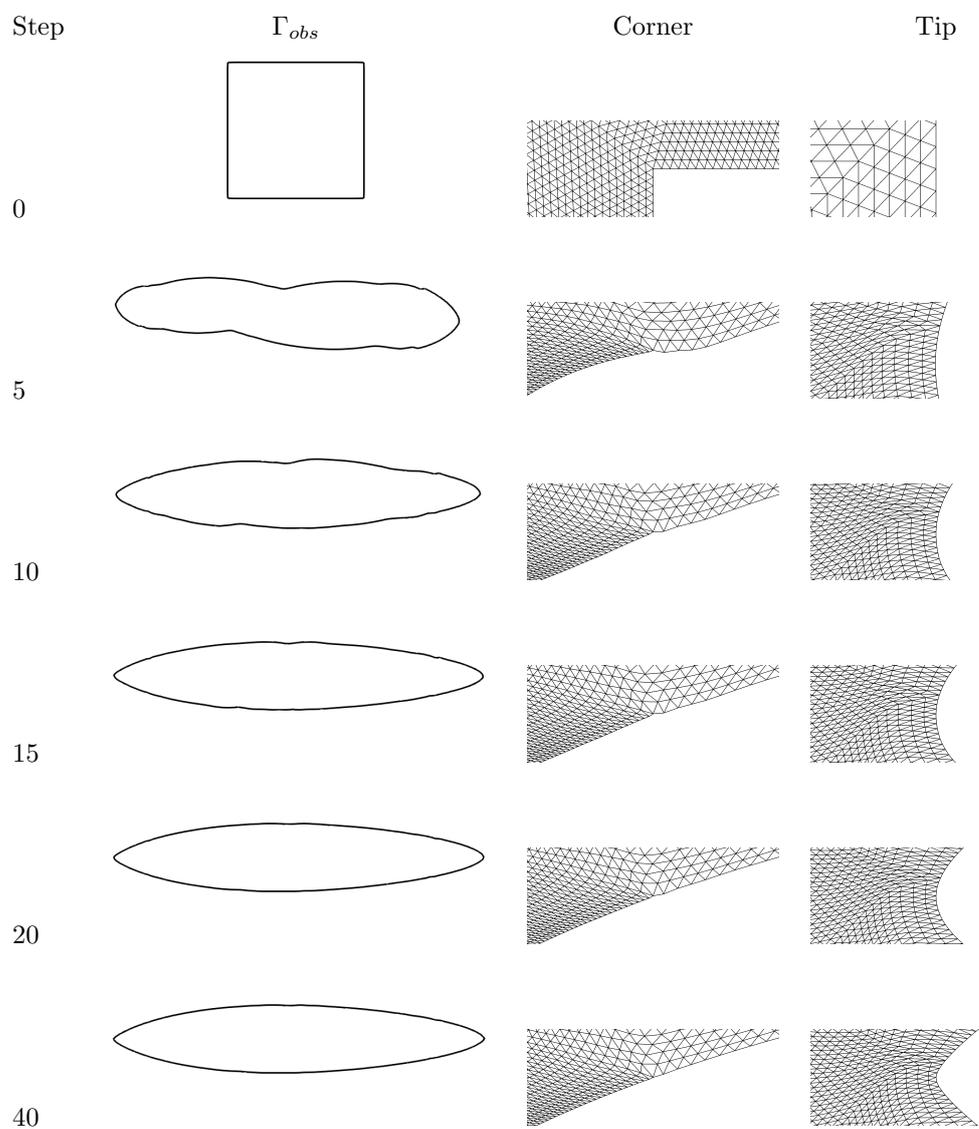


Figure 4: Selected steps  $\{0,5,10,15,20,40\}$  of the optimization process show the large deformations occurring during the initial steps. The upper left corner and front tip are featured together with the profile of  $\Gamma_{obs}$ .

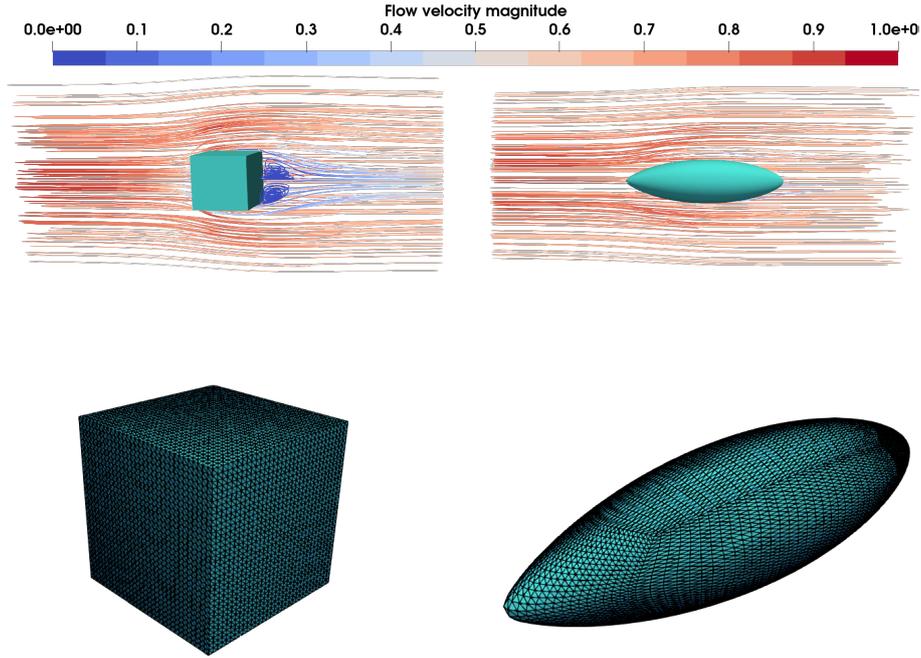


Figure 5: 3d results for a grid with 622 592 tetrahedrons. The optimized obstacle is shown next to the reference. The streamlines around the object are shown (top), together with the grids (bottom).

in the lower row of Figure 5, where the optimized geometry can be seen. The triangular elements are shown over the surface of the object, which allows one to see the large deformations necessary for the removal of the geometric singularities. Additionally, the creation of tips leads to a more optimal flow across  $\Gamma_{\text{obs}}$ , as can be seen on the streamlines of the optimized geometry.

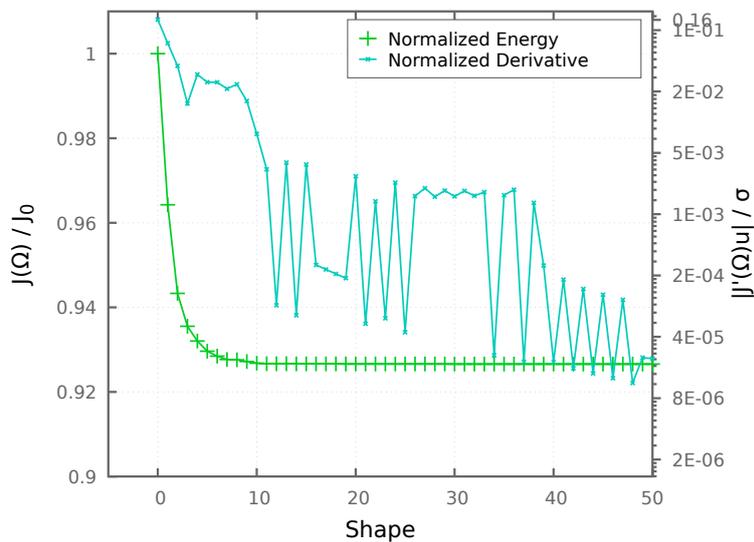


Figure 6: The objective function  $J(\Omega)$ , for a 3d simulation, divided by the initial value -  $J_0$  - over 50 shapes is shown, together with the convergence criterion (7).

The results in Figure 5 can be linked to the plots given in Figure 6. As the preexisting singularities, i.e. the edges and vertices, are removed, the dissipated energy decreases pronouncedly. By step 10, an 8% reduction has been achieved. This is similar to the sequence shown in Figure 4,

where in the early steps the obstacle is elongated parallel to the flow direction. Together with the streamlines for the optimized geometry in Figure 5, it can be seen how the more uniform flow around the surface results in a lower energy dissipation.

## 5 Comparison of descent directions in $W^{1,\infty}$ and $W^{1,p}$

In this section a comparison between the approach given in [21] for vector fields in  $W^{1,p}$  and  $W^{1,\infty}$  is performed. The so-called  $p$ -Laplace relaxation scheme was proposed in [5] in a shape optimization context and first applied in [20]. It is based on solving a relaxed problem in  $W^{1,p}$  which is meant to approximate a steepest descent in  $W^{1,\infty}$ . The work [14] discusses some of these approximations. This relaxed formulation was inspired by [17] which considers scalar functions. Even moderate values of  $p$  are found to be useful, however high values of  $p$  are necessary to yield a good approximation to  $W^{1,\infty}$ .

As in [21], let the Lagrangian

$$L^p(u, \lambda) := J'(\Omega) u + \frac{1}{p} \int_{\Omega} (Du : Du)^{p/2} dx + \sum_{i=1}^{d+1} \psi_{g,i} g_i(\Omega(u)), \quad (18)$$

be used to obtain a highly nonlinear optimality system, which is solved for  $u$ . For moderate values of  $p$ , the solution is found by using increasing values of  $p$ , which compute the initial guess for the next increment, up to a given  $p_{\max}$ . The latter is a caveat on itself, given the high computational requirements of successively solving a problem, which depends on an arbitrary increment to  $p$ . Additionally, as described in [21, Sec.2.1], care must be taken to prevent numerical problems associated to the second derivative of the Lagrangian, (18). The algorithm proposed in [21] is tailored in such a way that the largest possible deformations are allowed per step, without loss of solver convergence, which is a limiting factor for this  $p$ -Laplace algorithmic approach. An in-depth discussion and high-performance computing results are presented in [21].

In order to compare both optimization schemes, the same energy dissipation problem (3) is used. The viscosity continues to be given by  $\nu = 0.02$ . A 2d computational mesh with four refinement levels and 70 656 triangular elements is used for these simulations. As in [21], for the  $p$ -Laplace algorithm the maximum value of  $p$  is set to  $p_{\max} = 4.8$ . The ADMM-based optimization is configured as described in Section 4. Similarly, the initial configuration features a box-shaped obstacle, thus it is necessary to remove and create geometric singularities.

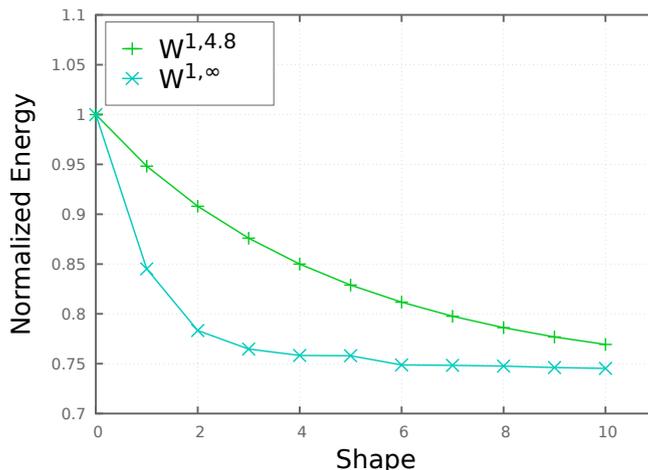


Figure 7: The dissipated energy for each approach, divided by their corresponding initial value -  $J_0$ - for the first 10 shapes is given.

Given that a highly viscous model is used, a large reduction of the dissipated energy is not necessarily related to the appearance of sharp geometric singularities over the surface of obstacle,

$\Gamma_{\text{obs}}$ , but to the deformation of the obstacle into the well-known prolate spheroid [27, 26]. As shown in Figure 4, for the ADMM this occurs within the initial 10 steps of Algorithm 1.

The reduction of the objective function for the first 10 steps is illustrated in Figure 7. The values of  $J$  are divided by the initial value  $J_0$ , with the purpose of allowing for a correct visualization of the differences between the two methods. The ADMM-based algorithm with  $u \in W^{1,\infty}$  allows for a 15% objective function reduction within the first shape optimization step. On the contrary, the  $p$ -Laplacian approach reaches this reduction level only after about four shape iterates. Around step 6, the  $W^{1,\infty}$  approach allows for a 25% decrease on the initial dissipated energy. The relaxed approach required more than 10 steps to achieve this level of reduction, as seen in the figure.

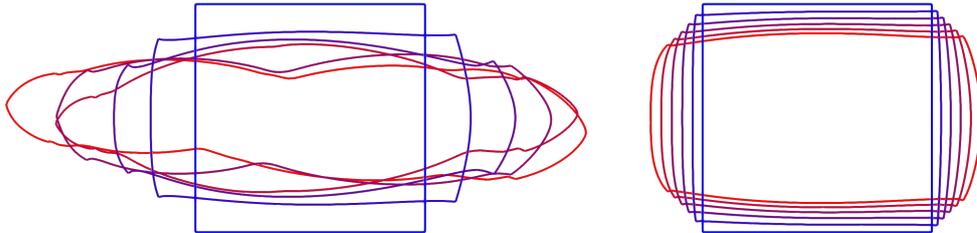


Figure 8: The initial and first five shape iterates for the  $W^{1,\infty}$  (left) and the  $W^{1,p}$  (right) shape optimization schemes. The initial configuration  $\Omega_0$  (blue) is deformed by applying the computed deformation field until  $\Omega_5$  (red) is obtained.

When observing the shapes rather than the energy, one might attribute the large differences of the energies between the methods to the aggressive deformations which appear from the  $W^{1,\infty}$  method. A comparison of the first five shapes for the  $W^{1,p}$  and  $W^{1,\infty}$  are presented in Figure 8. It may be seen that, while we expect that they will eventually provide the same shape, their paths to becoming an optimized shape are rather different. The deformation fields with the  $W^{1,p}$  method gently deform  $\Gamma_{\text{obs}}$ , slowly stretching it out. As the geometry is updated, the edges perpendicular to the flow are elongated and acquire a rounded profile. Using  $W^{1,\infty}$  for the descent it immediately generate this oblong and elongated shape which is seen as the fifth shape for the  $p$ -Laplace approach. The appearance of the tips can already be observed in step 2, together with a large elongation parallel to the flow direction. By steps 4 and 5, the tips have been created and the geometric singularities, i.e. the corners of the box, have mostly been removed.

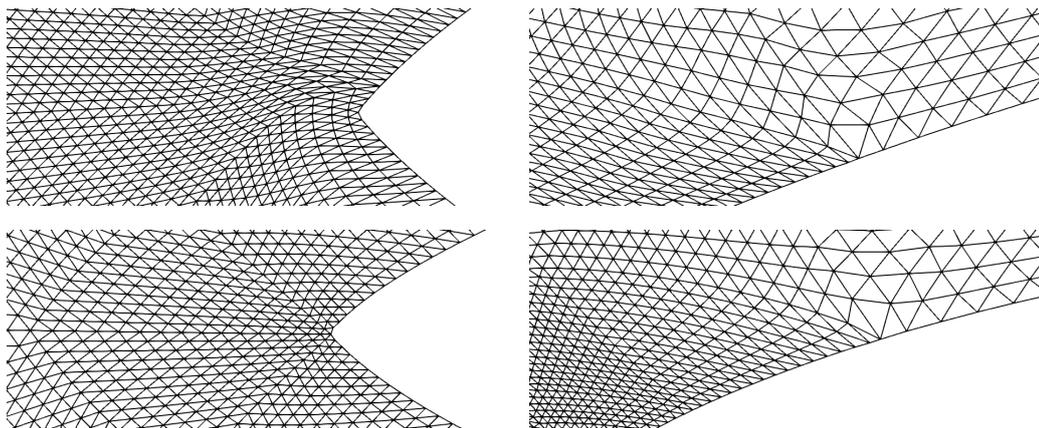


Figure 9: Computational meshes, corresponding to the optimized shapes after 50 steps, for a  $W^{1,\infty}$  (top) and  $W^{1,p}$  (bottom) optimization schemes. The front tips (left) and upper left corners (right) are shown.

However, these large deformations during the early stages of the optimization process may have a negative impact on mesh quality. In the context of the finite element discretization, it is

of interest that the optimized shape retains mesh quality, so that remeshing can be avoided. For instance, the computational grid must allow for the efficient and accurate solution of the state equation and the computation of the objective function. A particular concern is the overlapping of the grid, which is prevented by the  $W^{1,\infty}$  method, and degeneracy of triangles (in 2d) which is not guaranteed.

The grids of the optimized shape are compared in Figure 9. Two regions around  $\Gamma_{\text{obs}}$  are studied, the tip frontal to the flow direction and the section of the grid where the upper left corner of the box has been smoothed out. Given that the  $W^{1,\infty}$  vector fields largely deform these regions in the initial steps, the effects can be seen in the triangles over the surface of the obstacle. Compared to the  $p$ -Laplace descent directions, the elements have moved around the  $\Gamma_{\text{obs}}$ . The surrounding elements are more evenly distributed in the relaxed case, compared to the uneven pattern at the front tip of the  $W^{1,\infty}$ . Nevertheless, the large deformations enabled by the ADMM do not result in degenerate elements, particularly on the critical areas shown.

Table 1: Edge quality data for different refinement levels. The edge length ratio is used as a quality metric, i.e. the ratio of the longest to the shortest edge over  $\Gamma_{\text{obs}}$ .

	3-refs		4-refs		5-refs	
Triangles in $\Omega$	17 664		70 656		282 624	
Edges in $\Gamma_{\text{obs}}$	128		256		512	
Step	ADMM	PLAP	ADMM	PLAP	ADMM	PLAP
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.89	1.20	1.91	1.23	1.96	1.28
2	3.62	1.42	3.67	1.49	3.61	1.59
3	5.58	1.64	5.27	1.76	6.45	1.91
4	10.54	1.87	10.57	2.03	11.01	2.24
5	10.52	2.09	10.53	2.29	9.71	2.56
⋮						
47	13.49	8.33	14.96	9.56	13.69	8.86
48	13.50	8.51	15.05	9.89	13.64	8.98
49	13.50	8.54	15.03	9.73	13.64	9.09
50	13.50	8.56	15.12	10.17	13.60	9.30

The differences in the resulting grids, across the whole optimization process, can be quantified by the effect of the deformations on the elements that conform the obstacle surface  $\Gamma_{\text{obs}}$ . In the 2d computational grid, the surface of the obstacle is formed by the edges of the adjacent triangles. This is the region of the domain where the geometric singularities are to be created and removed, thus it is one of the regions that undergoes large deformations. In this sense, it is of interest to study the effect that these deformations have on the quality of the surface  $\Gamma_{\text{obs}}$ .

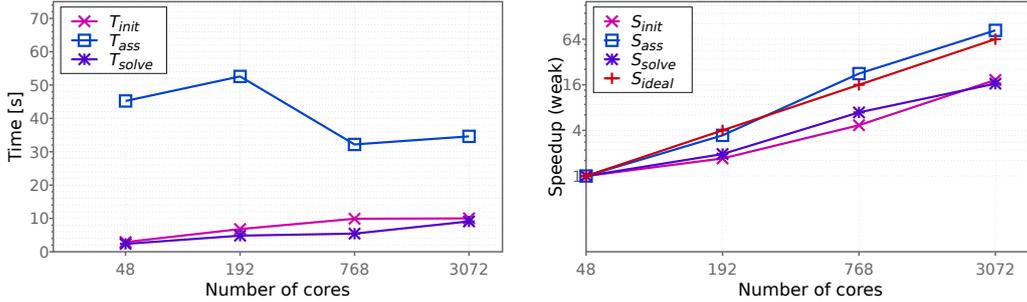
In Table 1, tests were carried out for several levels of refinements, up to 282 624 elements in  $\Omega$  and 512 edges in  $\Gamma_{\text{obs}}$ . The simulations were configured to run for a total of 50 steps. The edge length ratio, that is ratio between the longest and the shortest edge of  $\Gamma_{\text{obs}}$ , is shown for the first 5 and last 4 shape iterates. A comparison is made between the grids resulting from simulations based on  $W^{1,\infty}$  and  $W^{1,p}$  descent directions.

The evenly distributed grid allows for the edge length ratio to be equal to a unit for the initial grid  $\Omega_0$ . In line with the previously mentioned aggressive objective function reduction and with the observed large deformations in the initial steps, the mesh quality data in Table 1 shows how the ratio increases faster for the  $W^{1,\infty}$  than for the  $p$ -Laplace method. This trend is repeated across all refinement levels. At the end of the simulation the quality metric for the edges is higher for the  $W^{1,\infty}$  case. Nonetheless, the ADMM algorithm allows for larger deformations without provoking negative effects on the iterative solvers used for the system of equations. While such large deformations would most likely not be reachable by the  $p$ -Laplace approach or would have a negative impact on the convergence of the solvers used to compute  $u$ , using the ADMM it is possible to promote the preservation of mesh quality while allowing for a very aggressive shape

optimization.

## 6 High-performance computing results

In this section we present a parallel scalability study for weak scaling in 2d simulations. The results were computed using the supercomputer Lise at HLRN. This distributed-memory system consists of a total of 1270 compute nodes, each with a dual-socket architecture of 48 cores per CPU at a maximum frequency of 2.30GHz and 384GBs of RAM. The initial core count is set to 48 cores, all assigned within one socket. This study was carried out for up to 3072 cores, distributed across 32 nodes at full usage.



(a) Weak scalability: time measurements and speedup relative to 48 cores.

Procs	Refs	Number of Elements	DoFs	AVG ADMM Its. per Shape	AVG Newton Its. per Shape	AVG Lin.Its. per Shape	AVG Time per Shape[s]
48	3	17 664	18 016	31	166	5267	59
192	4	70 656	71 360	37	197	6825	76
768	5	282 624	284 032	25	139	5068	56
3072	6	1 130 496	1 133 312	25	141	5463	65

(b) Average iteration counts for the weak scaling results

Figure 10: Parallel scalability results for the first five optimization steps in a 2d simulation. Accumulated time and gained speedup relative to 48 cores are shown for the linearization solved within the minimization with respect to  $u$ . Average iteration counts are given for the ADMM routine and Newton’s method used therein.

A 2d mesh with 17 664 triangular elements was used as starting point for these measurements. The number of elements is increased by a factor of four with each level of refinement. Accumulated wallclock times and speedup relative to 48 cores are provided for up to a million elements for the first five optimization steps. Measurements of the time for the assembly, ( $T_{ass}$ ), of the linearized system; the initialization of the preconditioner, ( $T_{init}$ ); and the application of the linear solver, ( $T_{solve}$ ), for each step within Newton’s method are presented. Specifically, line 5 of Algorithm 2 is measured, because it is the most computationally relevant aspect of the routine. It is where the majority of the computational effort is spent. Since it is not the focus of this article, the time required for the solution of the state and adjoint equation systems are excluded from the displayed measurements.

In addition to the mentioned wallclock times, average iteration counts of the ADMM routine are analyzed, together with an estimate of the average time per shape of the whole outer loop where shape iterates are created. For the ADMM, an iteration represents one full cycle of the loop in lines 3-18 of Algorithm 2. From this loop, the average calls to Newton’s method are given, including the linear iterations required to find  $u$  in line 5 of Algorithm 2. The time-per-shape measurement is comprised of all the calls to the routine Algorithm 2 within an iteration of the outer loop in Algorithm 1, excepting for the previously mentioned fluid solver quantities.

Further to the algorithmic details given in Section 3, we now describe the implemented linear algebra solvers. The linearization within (15) is solved using a BiCGStab method preconditioned by

a geometric multigrid. The preconditioner uses 3 pre- and post-smoothing Gauss-Seidel steps with a V-cycle configuration. The linear solver is configured to reach a relative and absolute reduction of  $10^{-10}$  and  $10^{-12}$ , respectively. Moreover, the coarsest grid is solved in a single core via an LU factorization. For the solution of (12), a BiCGStab method is used with a Jacobi preconditioner. Given the structure of this problem, the system can be solved in a single iteration.

The weak scalability study is presented in Figure 10, where scalable results are seen for up to 3072 cores. The accumulated times show a slight performance drop for a 192 core distribution in the assembly phase. Also note that there is a coinciding increase in the average calls to the ADMM routine. Correspondingly, there is an increase in the average time required to generate the shape iterates. For the latter core counts, the average computational cost remains roughly equal, as observed in the iteration counts table. The number of degrees of freedom (DoFs) refers to the size of the system used to compute  $u$ , which is equal to  $d$  times the number of vertices of the mesh. This equation system increases in size by two orders of magnitude at six levels of refinement. Nevertheless, the measurements in Figure 10 show that the computational load is bound throughout all refinement levels.

## 7 Conclusion

In this paper, we presented a shape optimization methodology that uses the ADMM to obtain descent directions in  $W^{1,\infty}$ . The advantages of this approach were illustrated through 2d and 3d results, which show how this methodology allows for large deformations in the early steps of the optimization process. Our formulation introduced nonlinear geometric constraints, necessary to avoid trivial solutions. The method to handle these constraints was taken from [21] and it allows for fulfillment of the geometry constraints on each generated shape.

Weak parallel scalability results for up to 3072 processes in a distributed-memory system were presented. The use of appropriate preconditioners, with mesh-independent convergence properties, were utilized to achieve an scalable implementation. This is accentuated within the large number of linear iterations required for one shape optimization step. In our studies it was seen that the geometric multigrid method bounds the average computational workload, even under large increments of the number of DoFs. Additionally, a comparison of the early optimization steps was performed between the approach here proposed and the  $p$ -Laplace method as formulated in [21]. This showed that larger deformations are encouraged with a  $W^{1,\infty}$  optimization scheme. Per shape, it is seen that the objective function decreases faster for the  $W^{1,\infty}$  method. A mesh quality study shows that the  $W^{1,\infty}$  approach results in a not-meaningful decrease in quality of the line elements which compose the surface of the obstacle, as compared to the  $W^{1,p}$  algorithm. Taking into account that large deformations are needed to take the initial geometry to the optimized domain, a decrease in mesh quality is expected. However, in industrial applications, by expert-knowledge, the initial mesh is likely to be closer to the sought optimizer.

In this work a fixed step-size ADMM is used, unlike in [6] where the variable step-size version is implemented. It is worth mentioning that the methodology here presented is not bound to the ADMM, i.e. another method could be used. However, the area of shape optimization in  $W^{1,\infty}$ , and the algorithms required to compute adequate descent directions, is an ongoing field of research.

We suggest that our proposed technique could be considered for more complex geometries. The topic of mesh quality, in terms of its impact on the state variable, is to be explored in further work to widen our understanding of the advantages and disadvantages of different shape optimization methodologies.

## 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.

Computing time on the national supercomputer Lise at the North German Supercomputing Alliance (HLRN) under the project hhm00006 (An optimal shape matters) is gratefully acknowledged.

## References

- [1] 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](https://doi.org/10.1016/bs.hna.2020.10.004).
- [2] S. Bartels. “Nonconforming discretizations of convex minimization problems and precise relations to mixed methods”. In: *Computers & Mathematics with Applications* 93 (2021), pp. 214–229. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2021.04.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0898122121001541>.
- [3] S. Bartels and M. Milicevic. “Efficient iterative solution of finite element discretized nonsmooth minimization problems”. In: *Computers & Mathematics with Applications* 80.5 (2020), pp. 588–603. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2020.04.026>. URL: <https://www.sciencedirect.com/science/article/pii/S0898122120301735>.
- [4] S. Blauth. “Nonlinear Conjugate Gradient Methods for PDE Constrained Shape Optimization Based on Steklov–Poincaré-Type Metrics”. In: *SIAM Journal on Optimization* 31.3 (2021), pp. 1658–1689. DOI: [10.1137/20M1367738](https://doi.org/10.1137/20M1367738).
- [5] K. Deckelnick, P. J. Herbert, and M. Hinze. “A novel  $W^{1,\infty}$  approach to shape optimisation with Lipschitz domains”. In: *ESAIM: COCV* 28 (2022), p. 2. DOI: [10.1051/cocv/2021108](https://doi.org/10.1051/cocv/2021108). URL: <https://doi.org/10.1051/cocv/2021108>.
- [6] K. Deckelnick, P. J. Herbert, and M. Hinze. “Shape optimisation in the  $W^{1,\infty}$  topology with the ADMM algorithm”. In: *arXiv preprint arXiv:2301.08690* (2023).
- [7] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. SIAM, 2011. DOI: [10.1137/1.9780898719826](https://doi.org/10.1137/1.9780898719826).
- [8] M. P. I. Forum. *MPI: A Message-Passing Interface Standard Version 4.0*. 2021. URL: <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf>.
- [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](https://doi.org/10.1002/nme.2579).
- [10] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. 1st ed. Springer, 1994.
- [11] J. Haslinger and R. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation*. Vol. 7. Advances in Design and Control. SIAM, 2003.
- [12] 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](https://doi.org/10.1137/20M1332050).
- [13] A. Henrot and M. Pierre. *Shape Variation and Optimization: a geometrical analysis*. First. European Mathematical Society, 2018. ISBN: 9783037191781. DOI: <https://doi.org/10.4171/178>.
- [14] P. J. Herbert. “Shape Optimisation with  $W^{1,\infty}$ : A connection between the steepest descent and Optimal Transport”. In: *arXiv preprint arXiv:2301.07994* (2023).
- [15] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Vol. 23. Mathematical Modelling: Theory and Applications. Springer, 2009.
- [16] R. Hiptmair, A. Paganini, and S. Sargheini. “Comparison of approximate shape gradients”. In: *BIT Numerical Mathematics* 55 (June 2014). DOI: [10.1007/s10543-014-0515-z](https://doi.org/10.1007/s10543-014-0515-z).
- [17] 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](https://doi.org/10.1137/S0036141004432827).
- [18] 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).

- [19] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2009. DOI: 10.1093/acprof:oso/9780199546909.001.0001.
- [20] P. M. Müller, N. Kühn, M. Siebenborn, K. Deckelnick, M. Hinze, and T. Rung. “A novel p-harmonic descent approach applied to fluid dynamic shape optimization”. In: *Struct Multidisc Optim* (2021). DOI: 10.1007/s00158-021-03030-x.
- [21] P. M. Müller, J. Pinzón, T. Rung, and M. Siebenborn. “A Scalable Algorithm for Shape Optimization with Geometric Constraints in Banach Spaces”. In: *SIAM Journal on Scientific Computing* 45.2 (2023), B231–B251. DOI: 10.1137/22M1494609. eprint: <https://doi.org/10.1137/22M1494609>. URL: <https://doi.org/10.1137/22M1494609>.
- [22] S. Onyshkevych and M. Siebenborn. “Mesh Quality Preserving Shape Optimization Using Nonlinear Extension Operators”. In: *Journal of Optimization Theory and Applications* 189 (2020), pp. 291–316. DOI: 10.1007/s10957-021-01837-8.
- [23] J. Pinzon and M. Siebenborn. *ADMMOptim*. <http://www.github.com/multigridshapeopt>. Version 1.0. 2023.
- [24] J. Pinzon and M. Siebenborn. “Fluid Dynamic Shape Optimization using Self-adapting Non-linear Extension Operators with Multigrid Preconditioners”. In: *Optimization and Engineering* (2022). DOI: 10.1007/s11081-022-09721-8. eprint: [arXiv:2108.07788](https://arxiv.org/abs/2108.07788).
- [25] J. Pinzon, M. Siebenborn, and A. Vogel. “Parallel 3d shape optimization for cellular composites on large distributed-memory clusters”. In: *Journal of Advanced Simulation in Science and Engineering* 7.1 (2020), pp. 117–135. DOI: 10.15748/jasse.7.117. eprint: [arXiv:2003.09683](https://arxiv.org/abs/2003.09683).
- [26] O. Pironneau. “On optimum design in fluid mechanics”. In: *Journal of Fluid Mechanics* 64.1 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.
- [27] O. Pironneau. “On optimum profiles in Stokes flow”. In: *Journal of Fluid Mechanics* 59.1 (1973), pp. 117–128. DOI: 10.1017/S002211207300145X.
- [28] S. Schmidt, C. Ilic, V. Schulz, and N. R. Gauger. “Three-dimensional large-scale aerodynamic shape optimization based on shape calculus”. In: *AIAA journal* 51.11 (2013), pp. 2615–2627.
- [29] S. Schmidt, E. Wadbro, and M. Berggren. “Large-scale three-dimensional acoustic horn optimization”. In: *SIAM Journal on Scientific Computing* 38.6 (2016), B917–B940.
- [30] V. Schulz and M. Siebenborn. “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.
- [31] V. H. Schulz, M. Siebenborn, and K. Welker. “Efficient PDE Constrained Shape Optimization Based on Steklov–Poincaré-Type Metrics”. In: *SIAM Journal on Optimization* 26.4 (2016), pp. 2800–2819. DOI: 10.1137/15M1029369.
- [32] M. Siebenborn. “A Shape Optimization Algorithm for Interface Identification Allowing Topological Changes”. In: *Journal of Optimization Theory and Applications* 177(2) (2018), pp. 306–328. DOI: 10.1007/s10957-018-1279-4.
- [33] M. Siebenborn and A. Vogel. “A shape optimization algorithm for cellular composites”. In: *PINT Comput. Visual Sci.* (2021). DOI: <https://www.doi.org/10.51375/IJCUSE.2021.1.1.5>.
- [34] M. Siebenborn and K. Welker. “Algorithmic Aspects of Multigrid Methods for Optimization in Shape Spaces”. In: *SIAM Journal on Scientific Computing* 39.6 (2017), B1156–B1177.
- [35] 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.
- [36] A. Vogel, S. Reiter, M. Rupp, A. Nägel, and G. Wittum. “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.
- [37] A. Vogel et al. “10,000 Performance Models per Minute – Scalability of the UG4 Simulation Framework”. In: *in: Euro-Par 2015: Parallel Processing, J. L. Träff et al. (eds), Springer* (2015), pp. 519–531.

- [38] A. Vogel et al. “Automated performance Modeling UG4 Simulation Framework”. In: *in: Software for Exascale Computing — SPPEXA 2013-2015, H.-J. Bungartz et al. (eds), Springer (2016)*.

# Chapter 4

## Conclusion and Outlook

### 4.1 Concluding remarks

In this thesis the topic of PDE-constrained shape optimization was addressed. The main concern being algorithmic implementations used to infer a descent direction using the shape derivative  $J'$  within a descent-like method. In turn, these are used as deformation fields to iteratively optimize the given domain, resulting in an optimized domain with respect to a shape functional. The different strategies had in focus the preservation of mesh quality throughout the optimization process -for instance by avoiding overlapping of elements-, the removal and creation of geometric singularities, and the implementation and benchmarking of algorithms in an HPC setting.

A fluid dynamics application was used to benchmark the proposed optimization methods. In it, a domain represents a flow tunnel with an object located at its center. The surface of the object is to be optimized, and the flow is described by Navier–Stokes equations. As explained in Section 1.3, geometrical constraints are needed to avoid trivial solutions. Consequently, the optimization problem must be formulated in such a way that these constraints are fulfilled. However, implementing this is numerically and computationally challenging. This was detailed in Chapter 2.

For instance, the nonlinear extension approach [55] in Section 2.1, formulated the problem via an augmented Lagrangian function. It was discussed how the associated Lagrange multipliers, and their convergence, can have an impact on the space of admissible shapes. E.g., non-converged values of these multipliers can lead to shapes which do not necessarily fulfill the imposed constraints. This can be detrimental to the convergence of iterative methods, as well as posing difficulties for the algorithmic implementation. On the other hand, the methodology proposed in [51], where the geometric constraints are incorporated to the descent direction, which was featured in Sections 2.2 and 2.3, leads to a nonlinear system of equations. In this way, only shapes which fulfill the geometric constraints to machine precision can be obtained on every step. The disadvantages being the computation of the Schur complement operator, which requires several numerical solves of the aforementioned system of nonlinear equations; as well as the use of incrementing values of  $p$  to approximate  $W^{1,\infty}$ . On the other hand, this also simplified the formulation of the Lagrangian function, because no penalization was required to enforce the geometric constraints. This also removed the need for a heuristically determined step-size for the update of the multipliers, i.e. step sizes. In computational terms, this reduced the required time needed to determine the parameters used for the augmented Lagrangian. In the context of a distributed-memory system, this helped reduce the expense of computing budget spent on test runs, which are used to determine a correct set of parameters to configure a simulation.

The simulations in this thesis involved a very large number of DoFs. Both with the purpose of correctly representing the geometric constraints, as to be a starting

point for more complex, industrial applications. As previously mentioned, this made it necessary to use distributed-memory systems with high computational power. Therefore, the shape optimization algorithms were also tested in terms of their parallel feasibility. In order to reduce the computational workload, the iterative solvers were preconditioned with a geometric multigrid method, due to its mesh-independent convergence properties. The generation of results was limited by the availability of the computational resources at the used computing centers. Nevertheless, the methods here proposed were tested for their parallel computing capabilities in several thousand cores. As explained in Chapter 2, the weak scalability and grid independence of the algorithms were studied within this work.

It was shown that applying successive deformation fields to the initial discretized domain, can have a negative impact on the mesh quality. This is particularly evident when the initial object has features, such as edges and corners, that have to be smoothed out. This highlights the importance of developing strategies that decrease the loss of mesh quality, i.e. preventing the elements from becoming highly degenerate. The latter can also result in the domain overlapping itself. The importance of this, for instance, can be related to the convergence of iterative solvers being dependent on the quality of the computational mesh. Throughout this work, several strategies were investigated. For instance, constraining the deformation gradient by imposing a constraint as part of the optimization. However, determining the correct set of corresponding parameters for the augmented Lagrangian proved challenging, because the set of admissible shapes was affected by the heuristic determination of these parameters. This increased the difficulty of the formulation and solution of the optimization problem. This is the reason why a simpler approach was pursued in this work, i.e. finding descent directions in Banach spaces that preserve the Lipschitz quality of the transformed domain, even under large deformations. This can be understood as the formulations in  $W^{1,p}$  and  $W^{1,\infty}$  Banach spaces described Sections 2.2 and 2.3, as opposed to the Hilbertian setting in Section 2.1. One of the caveats, was the increase in computational work needed to compute the descent direction. As an example, the solution of the optimality system in [51] for increments of  $p$ , together with the computation of the Schur complement operator, resulted in a computationally expensive optimization methodology. Similarly, the use of the ADMM in a  $W^{1,\infty}$  descent method, required a higher number of calls to linear solvers compared to other approaches. As a concluding remark, it could be said that descent directions in  $W^{1,\infty}$  or  $W^{1,p}$  promote the preservation of the Lipschitz property, which can lead to a better mesh quality. However, based on the methods proposed and studied in this thesis, this comes at a high computational cost. Therefore, finding descent directions in different Banach spaces presents several trade-offs which have to be taken into account and can be the subject of further studies.

## 4.2 Outlook

The solution of the state equation was carried out using the default iterative methods included with UG4. Given that a precise value of the state variable was needed for the computation of the shape derivative,  $J'$ , the use of a stabilization term was avoided where possible. This required the use of a  $P_2 - P_1$  stable finite element discretization for the Navier-Stokes equation. However, the assembly of the discrete

linearized system results in a saddle point problem, which usually requires specialized preconditioning techniques. An efficient solution scheme for this problem is proposed in [18] and further expanded in [35]. These can be a good starting point for further parallel studies, where the computational effort and time required to solve the state equation could be reduced.

Further work in a parallel setting could be performed by increasing Reynolds number, which for these studies remained large. A higher Reynolds number would require a more complex model for the description of the flow than the one used in this work, for instance as implemented in [50]. Moreover, further studies could implement a time-dependent model, although this would increase the memory requirements and computational costs. Additionally, the flow could account for other phenomena proper of high Reynolds numbers [17].

The shape optimization scheme used here was limited to descent-like methods, based on the first shape derivative  $J'(\Omega)$ . However, a second order optimization scheme can be proposed, which would require the computation of the second shape derivative  $J''(\Omega)$ . Within the context of this thesis, this was not implemented. Therefore, it can be explored in future work. As previously mentioned, a high computational effort is needed to solve the state equation and the adjoint, particularly for very large number of DoFs. This effort increases for the computation of  $J''(\Omega)$ , as described in [36]. Based on [68, 15], further studies could be carried out in this area.

Mesh quality is a recurring topic in this work. This work relied on [72] for a description of many mesh quality metrics used in finite elements. Generally, 2 dimensional simulations were used to visualize the grid and analyze the quality of individual elements. With these metrics the different methodologies were compared to each other [34]. However, an in-depth study of how mesh quality is preserved across several techniques and the impact these have, is yet to be done. Further studies would require a unified metric to determine the advantages and disadvantages of the large deformations generated by a shape optimization algorithm. The latter is directly related to determining which shape optimization method is better for a given application. Taking into account that a shape functional is used as a performance metric for the domain, and that this functional depends on the state variable, it is evident that the solution of the state equation is a key aspect in PDE-constrained shape optimization. As mentioned before, the accuracy of the approximate solution of the state equation is directly related to the quality of the mesh. Therefore, it makes sense to use an a-posteriori error estimator as a performance metric of different shape optimization schemes. The general overview in [27] and the Navier-Stokes error estimators given in [53] could be used as a starting point for a study of this kind.

# Chapter 5

## Declaration of personal contribution

The doctoral candidate specifies the following contributions to each chapter of the dissertation:

- Chapter 1: the content of Chapter 1 is due to the candidate with no contribution from other people. No other sources were used besides the referenced citations.
- Chapter 2: the written content of Chapter 2 is due to the candidate, no other sources were used besides the referenced citations. In this chapter, the publications included in this cumulative dissertation are discussed in the style of a synthesis essay. With this purpose, the figures and tables presented in this chapter have been taken from the publications [55, 51, 34] and referenced accordingly. For these figures and tables, the data collection, generation and visualization was carried out by the candidate.
- Chapter 3: a detailed description for each of the three publications included in this cumulative dissertation is provided. They are subdivided as in the contents table.

**Paper I** : the paper [55] is co-authored by Dr. Martin Siebenborn and the candidate. The writing of the preprint was mostly done by the candidate (approximately 85% of the workload). The corrections required by journal reviewers were mostly performed by the candidate (an estimated 90% of the workload), Dr. Siebenborn was available for verification and consultation. The content of section 1 was written mostly by the candidate, this included a literature review as well as the abstract. The content and analysis of section 2 was written and developed mostly by the candidate. The adjoint work consisted of the verification of formulas and concepts by the co-author in a supervisory capacity. The algorithmic development in section 3 is jointly due to the two co-authors. The candidate carried out most of the investigative work of this section (an estimated 90% of the workload) guided by Dr. Martin Siebenborn, who provided insight and guidelines. The results in section 4 and its text were done by the candidate, although no other co-authors contributed directly to the presented text and figures, these were done in consultation with all the co-authors. The parallel studies in section 5 were carried out by the candidate. And the conclusion in section 6 was written with equal contributions by all co-authors. The software was written by the candidate (approximately 85% of the workload), and Dr. Martin Siebenborn reviewed and debugged when needed. The data generation and collection was performed mostly by the candidate (90%), as well as the preparation of the figures (90%). Dr. Martin Siebenborn contributed to the figures via consultation and suggestions on design.

**Paper II**: the publication [51] was written in collaboration with Dr. Martin

Siebenborn, Dr. Peter Marvin Müller, and Prof. Dr. Thomas Rung. The underlying idea was investigated previously by the other co-authors (excluding the candidate) in [50], and our work built partly on those studies. The writing of the preprint was carried out by the candidate and Dr. Peter Marvin Müller. The candidate wrote most of the text (approximately 75% of the text). Before submission, the paper was reviewed by all co-authors. Dr. Martin Siebenborn and Prof. Dr. Thomas Rung were available for consultation, as well as for providing suggestions related to the presentation of the results. The corrections requested by the journal reviewers were done mostly by the candidate (approximately 75% of the text), and Dr. Müller provided insights, suggestions, and corrections when needed. The content of section 1 was mostly written by the candidate. The text in section 2 was a joint work of the candidate with Dr. Peter Marvin Müller, with equal contributions from both parts. The mathematical formulations presented in section 2 were carried out by the candidate and verified individually by all co-authors. The algorithmic developments of section 3 were carried out by the candidate, Dr. Martin Siebenborn, and Dr. Peter Marvin Müller in equal contributions. The content of section 3 was written by the candidate, and verified by Dr. Martin Siebenborn and Dr. Peter Marvin Müller, who provided corrections and guidelines where necessary. The content in section 4 was written by the candidate. The results presented in section 5 were generated by the candidate. Of this section, Figure 6 was partly prepared by Dr. Martin Siebenborn using the boolean filter in Paraview. The parallel studies in section 5 were carried out, written, and presented by the candidate with no contributions from other co-authors. The text of section 5 was mostly written by the candidate. The software was written by the candidate (approximately 90% of the workload), the co-authors helped to find bugs through testing. The data generation and collection was performed by the candidate (100%), the preparation of the figures was mostly done by the candidate (85%).

**Paper III:** paper [34] was written together with Dr. Martin Siebenborn and Dr. Philip Herbert. This work was based on previous work from one of the co-authors [15] (Dr. Philip Herbert), and our work builds on it. Dr. Martin Siebenborn carried out the initial testing of the methodology. The preprint was mostly written by the candidate (an estimate of 75%). The co-authors carried out corrections and suggested changes where needed. The content of section 1 was mostly written by Dr. Philip Herbert, the candidate provided the initial version. Section 2 was mostly written by the candidate, and corrected by Dr. Herbert. The formulations presented in section 2 were verified by Dr. Philip Herbert. The algorithmic developments of section 3 were equally contributed by all authors. Section 3 was mostly written by the candidate, and Dr. Philip Herbert contributed with corrections and verification of the formulations presented therein. The results in section 4 and 5 were generated by the candidate with no contributions from other authors. The text in both sections was mostly written by the candidate, the other co-authors contributed through corrections and suggestions. The parallel studies in section 6 were performed by the candidate with no contribution from other authors, this includes the presentation of results. The conclusion in section 7 was done with equal contributions by the candidate and Dr. Philip Herbert. The software was written by

the candidate (approximately 90% of the workload), the co-authors helped to find bugs through testing. The data generation and collection was performed by the candidate (100%). Although the preparation of the figures was mostly done by the candidate (90%), Dr. Herbert contributed through suggestions, corrections, and design observations where needed.

- Chapter 4: the concluding remarks were written solely by the candidate, no other sources were used other than the mentioned citations.

# Bibliography

- [1] J. Ahrens et al. “ParaView: An end-user tool for large-data visualization”. In: *The visualization handbook* (2005).
- [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 G. Michailidis. “Thickness control in structural optimization via a level set method”. In: *Structural and Multidisciplinary Optimization* 53.6 (2016), pp. 1349–1382. DOI: 10.1007/s00158-016-1453-y.
- [4] 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.
- [5] J. S. Arora. *Introduction to Optimum Design*. 4th ed. Boston: Academic Press, 2017, pp. 919–928. DOI: 10.1016/B978-0-12-800806-5.00021-4.
- [6] S. Bartels. “Nonconforming discretizations of convex minimization problems and precise relations to mixed methods”. In: *Computers & Mathematics with Applications* 93 (2021), pp. 214–229. DOI: 10.1016/j.camwa.2021.04.014.
- [7] S. Bartels and M. Milicevic. “Efficient iterative solution of finite element discretized nonsmooth minimization problems”. In: *Computers & Mathematics with Applications* 80.5 (2020), pp. 588–603. DOI: 10.1016/j.camwa.2020.04.026.
- [8] J. A. Bello et al. “The Differentiability of the Drag with Respect to the Variations of a Lipschitz Domain in a Navier–Stokes Flow”. In: *SIAM Journal on Control and Optimization* 35.2 (1997), pp. 626–640. DOI: 10.1137/S0363012994278213.
- [9] M. Berzins. “Mesh quality: a function of geometry, error estimates or both?” In: *Engineering with Computers* 15 (1999), pp. 236–247. DOI: 10.1007/s003660050019.
- [10] S. Blauth. “Nonlinear Conjugate Gradient Methods for PDE Constrained Shape Optimization Based on Steklov–Poincaré-Type Metrics”. In: *SIAM Journal on Optimization* 31.3 (2021), pp. 1658–1689. DOI: 10.1137/20M1367738.
- [11] C. Brandenburg et al. “A Continuous Adjoint Approach to Shape Optimization for Navier Stokes Flow”. In: *Optimal Control of Coupled Systems of Partial Differential Equations*. Birkhäuser Basel, 2009, pp. 35–56. DOI: 10.1007/978-3-7643-8923-9\_2,.
- [12] C. Brandenburg et al. “Advanced Numerical Methods for PDE Constrained Optimization with Application to Optimal Design in Navier Stokes Flow”. In: *Constrained Optimization and Optimal Control for Partial Differential Equations*. Ed. by G. Leugering et al. Basel: Springer Basel, 2012, pp. 257–275. DOI: 10.1007/978-3-0348-0133-1\_14.
- [13] R. De Borst et al. *Nonlinear finite element analysis of solids and structures*. John Wiley & Sons, 2012. DOI: 10.1002/9781118375938.
- [14] K. Deckelnick, P. J. Herbert, and M. Hinze. “A novel  $W^{1,\infty}$  approach to shape optimisation with Lipschitz domains”. In: *ESAIM: COCV* 28 (2022), p. 2. DOI: 10.1051/cocv/2021108.
- [15] K. Deckelnick, P. J. Herbert, and M. Hinze. “Shape optimisation in the  $W^{1,\infty}$  topology with the ADMM algorithm”. In: *arXiv preprint arXiv:2301.08690* (2023). DOI: 10.48550/arXiv.2301.08690.
- [16] M. C. Delfour and J. P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Second. Society for Industrial and Applied Mathematics, 2011. DOI: 10.1137/1.9780898719826.
- [17] R. P. Dwight and J. Brezillon. “Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization”. In: *AIAA Journal* 44.12 (2006), pp. 3022–3031. DOI: 10.2514/1.21744.

- [18] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. 2nd ed. Oxford University Press, June 2014. DOI: 10.1093/acprof:oso/9780199678792.001.0001.
- [19] K. Eppler and H. Harbrecht. “Shape Optimization for Free Boundary Problems – Analysis and Numerics”. In: *Constrained Optimization and Optimal Control for Partial Differential Equations*. Ed. by G. Leugering et al. Basel: Springer Basel, 2012, pp. 277–288. DOI: 10.1007/978-3-0348-0133-1\_15.
- [20] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. 1st ed. Applied Mathematical Sciences. Springer, 2010. DOI: 10.1007/978-1-4757-4355-5.
- [21] T. Etling and R. Herzog. “Optimum Experimental Design by Shape Optimization of Specimens in Linear Elasticity”. In: *SIAM Journal on Applied Mathematics* 78.3 (2018), pp. 1553–1576. DOI: 10.1137/17M1147743.
- [22] T. Etling et al. “First and Second Order Shape Optimization Based on Restricted Mesh Deformations”. In: *SIAM Journal on Scientific Computing* 42.2 (2020), A1200–A1225. DOI: 10.1137/19M1241465.
- [23] M. Fischer et al. “Fréchet Differentiability of Unsteady Incompressible Navier–Stokes Flow with Respect to Domain Variations of Low Regularity by Using a General Analytical Framework”. In: *SIAM Journal on Control and Optimization* 55.5 (2017), pp. 3226–3257. DOI: 10.1137/16M1089563.
- [24] P. Gangl et al. “Shape Optimization of an Electric Motor Subject to Nonlinear Magnetostatics”. In: *SIAM Journal on Scientific Computing* 37.6 (2015), B1002–B1025. DOI: 10.1137/15100477X.
- [25] 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. DOI: 10.1002/nme.2579.
- [26] M. B. Giles and N. A. Pierce. “An introduction to the adjoint approach to design”. In: *Flow, turbulence and combustion* 65.3-4 (2000), pp. 393–415. DOI: 10.1023/A:1011430410075.
- [27] T. Grätsch and K.-J. Bathe. “A posteriori error estimation techniques in practical finite element analysis”. In: *Computers & Structures* 83.4 (2005), pp. 235–265. DOI: <https://doi.org/10.1016/j.compstruc.2004.08.011>.
- [28] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. 2nd ed. Springer, 2016. DOI: 10.1007/978-3-319-28483-5.
- [29] G. Hager and G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 1st ed. CRC, 2010.
- [30] J. Haslinger, J. Málek, and J. Stebel. “Shape optimization in problems governed by generalised Navier-Stokes equations: Existence analysis”. In: *Control and cybernetics* 34 (Mar. 2005), pp. 283–303.
- [31] J. Haslinger and R. Mäkinen. *Introduction to shape optimization. Theory, approximation, and computation*. Advances in Design and Control. SIAM, Jan. 2003. DOI: 10.1137/1.9780898718690.
- [32] J. Haubner, M. Siebenborn, and M. Ulbrich. “A Continuous Perspective on Shape Optimization via Domain Transformations”. In: *SIAM Journal on Scientific Computing* 43.3 (2021), A1997–A2018. DOI: 10.1137/20M1332050.
- [33] A. Henrot and M. Pierre. *Shape Variation and Optimization: a geometrical analysis*. 1st ed. European Mathematical Society, 2018. DOI: 10.4171/178.
- [34] P. J. Herbert, J. A. P. Escobar, and M. Siebenborn. *Shape optimization in  $W^{1,\infty}$  with geometric constraints: a study in distributed-memory systems*. 2023. DOI: 10.48550/arXiv.2205.01912.
- [35] R. Herzog and K. M. Soodhalter. “A Modified Implementation of MINRES to Monitor Residual Subvector Norms for Block Systems”. In: *SIAM Journal on Scientific Computing* 39.6 (2017), A2645–A2663. DOI: 10.1137/16M1093021.

- [36] M. Hinze et al. *Optimization with PDE Constraints*. 1st ed. Vol. 23. Mathematical Modelling: Theory and Applications. Springer, 2009. DOI: 10.1007/978-1-4020-8839-1.
- [37] R. Hiptmair, A. Paganini, and S. Sargheini. “Comparison of approximate shape gradients”. In: *BIT Numerical Mathematics* 55 (2015), pp. 459–485. DOI: 10.1007/s10543-014-0515-z.
- [38] J. A. Iglesias, K. Sturm, and F. Wechsung. “Two-Dimensional Shape Optimization with Nearly Conformal Transformations”. In: *SIAM Journal on Scientific Computing* 40.6 (2018), A3807–A3830. DOI: 10.1137/17m1152711.
- [39] 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.
- [40] A. Jameson. “Aerodynamic shape optimization using the adjoint method”. In: *Lectures at the Von Karman Institute, Brussels* (2003).
- [41] A. Jameson. “Advances in Aerodynamic Shape Optimization”. In: *Computational Fluid Dynamics*. Ed. by C. Groth and D. W. Zingg. Springer Berlin, Heidelberg, 2006, pp. 687–698. DOI: 10.1007/3-540-31801-1\_100.
- [42] A. Jung et al. “3D shape optimization of turbine blades”. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 85.12 (2005), pp. 878–895. DOI: 10.1002/zamm.200410215.
- [43] K. Kunisch and G. Peichl. “Numerical gradients for shape optimization based on embedding domain techniques”. In: *Computational Optimization and Applications* 18 (2001), pp. 95–114. DOI: 10.1023/A:1008779803348.
- [44] E. Laporte and P. Le Tallec. *Numerical Methods in Sensitivity Analysis and Shape Optimization*. 1st ed. Birkhauser Boston, MA, 2003. DOI: 10.1007/978-1-4612-0069-7.
- [45] D. Luft and V. Schulz. “Pre-shape calculus and its application to mesh quality optimization”. In: *Control and Cybernetics* 50.3 (2021), pp. 263–301. DOI: 10.2478/candc-2021-0019.
- [46] D. Luft and V. Schulz. “Simultaneous shape and mesh quality optimization using pre-shape calculus”. In: *Control and Cybernetics* 50.4 (2021), pp. 473–520. DOI: 10.2478/candc-2021-0028.
- [47] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 4.0*. June 2021.
- [48] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. 2nd ed. Oxford University Press, Sept. 2009. DOI: 10.1093/acprof:oso/9780199546909.001.0001.
- [49] 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.
- [50] P. M. Müller et al. “A novel p-harmonic descent approach applied to fluid dynamic shape optimization”. In: *Structural and Multidisciplinary Optimization* 64 (2021), pp. 3489–3503. DOI: 10.1007/s00158-021-03030-x.
- [51] P. M. Müller et al. “A Scalable Algorithm for Shape Optimization with Geometric Constraints in Banach Spaces”. In: *SIAM Journal on Scientific Computing* 45.2 (2023), B231–B251. DOI: 10.1137/22M1494609.
- [52] F. Murat and J. Simon. “Etude de problemes d’optimal design”. In: *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*. Ed. by J. Cea. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 54–62. DOI: 10.1007/3-540-07623-9\_279.
- [53] J. Oden, W. Wu, and M. Ainsworth. “An a posteriori error estimate for finite element approximations of the Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 111.1 (1994), pp. 185–202. DOI: 10.1016/0045-7825(94)90045-0.
- [54] S. Onyshkevych and M. Siebenborn. “Mesh Quality Preserving Shape Optimization Using Nonlinear Extension Operators”. In: *Journal of Optimization Theory and Applications* 189 (2020), pp. 291–316. DOI: 10.1007/s10957-021-01837-8.

- [55] J. Pinzon and M. Siebenborn. “Fluid Dynamic Shape Optimization using Self-adapting Non-linear Extension Operators with Multigrid Preconditioners”. In: *Optimization and Engineering* 24 (2022), pp. 1089–1113. DOI: 10.1007/s11081-022-09721-8.
- [56] J. Pinzon, M. Siebenborn, and A. Vogel. “Parallel 3d shape optimization for cellular composites on large distributed-memory clusters”. In: *Journal of Advanced Simulation in Science and Engineering* 7.1 (2020), pp. 117–135. DOI: 10.15748/jasse.7.117.
- [57] J. Pinzon, M. Siebenborn, and A. Vogel. “Geometric-Constrained Scalable Algorithm for PDE-Constrained Shape Optimization”. In: *High Performance Computing in Science and Engineering '22*. Ed. by W. E. Nagel, D. H. Kröner, and M. M. Resch. Springer International Publishing, 2023.
- [58] J. A. Pinzon Escobar and M. Siebenborn. “Shape Optimization Algorithms for Fluid Dynamics Applications”. In: *PAMM* 22.1 (2023), e202200279. DOI: <https://doi.org/10.1002/pamm.202200279>.
- [59] O. Pironneau. “On optimum design in fluid mechanics”. In: *Journal of Fluid Mechanics* 64.1 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.
- [60] O. Pironneau. “On optimum profiles in Stokes flow”. In: *Journal of Fluid Mechanics* 59.1 (1973), pp. 117–128. DOI: 10.1017/S002211207300145X.
- [61] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. 1st ed. Springer Berlin, Heidelberg, 1984. DOI: 10.1007/978-3-642-87722-3.
- [62] O. Pironneau. “Optimal Shape Design with Applications to Aerodynamics”. In: *Shape Optimization and Free Boundaries*. Ed. by M. C. Delfour and G. Sabidussi. Springer, 1992, pp. 211–251. DOI: 10.1007/978-94-011-2710-3\_6.
- [63] P. I. Plotnikov and J. Sokolowski. “Geometric Aspects of Shape Optimization”. In: *The Journal of Geometric Analysis* 33.7 (2023), p. 206. DOI: 10.1007/s12220-023-01252-7.
- [64] *ProMesh*. URL: <http://www.promesh3d.com>.
- [65] L. Radtke et al. “Parameter-Free Shape Optimization: Various Shape Updates for Engineering Applications”. In: *Aerospace* 10.9 (2023). DOI: 10.3390/aerospace10090751.
- [66] P. Rentrop, S.-O. Stoll, and U. Weyer. “Sensitivity Calculations for 2D-Optimization of Turbomachine Blading”. In: *Optimal Control of Complex Structures*. Ed. by K.-H. Hoffmann et al. Basel: Birkhäuser Basel, 2002, pp. 203–216. DOI: 10.1007/978-3-0348-8148-7\_17.
- [67] S. Schmidt et al. “Three-dimensional large-scale aerodynamic shape optimization based on shape calculus”. In: *AIAA journal* 51.11 (2013), pp. 2615–2627. DOI: 10.2514/1.J052245.
- [68] S. Schmidt and V. H. Schulz. “A Linear View on Shape Optimization”. In: *SIAM Journal on Control and Optimization* 61.4 (2023), pp. 2358–2378. DOI: 10.1137/22M1488910.
- [69] V. Schulz and S. 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.
- [70] V. Schulz, M. Siebenborn, and K. Welker. “Efficient PDE constrained shape optimization based on Steklov–Poincaré-Type metrics”. In: *SIAM Journal on Optimization* 26.4 (2016), pp. 2800–2819. DOI: 10.1137/15M1029369.
- [71] V. Schulz. “A Riemannian View on Shape Optimization”. In: *Foundations of Computational Mathematics* 14 (Mar. 2012). DOI: 10.1007/s10208-014-9200-5.
- [72] J. R. Shewchuk. “What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures”. In: *Proceedings of the 11th International Meshing Roundtable, IMR 2002, Ithaca, New York, USA, September 15-18, 2002*. Ed. by N. Chrisochoides. 2002, pp. 115–126.
- [73] M. Siebenborn and A. Vogel. “A shape optimization algorithm for cellular composites”. In: *arXiv preprint arXiv:1904.03860* (2019). DOI: 10.48550/arXiv.1904.03860.
- [74] J. Simon. “Differentiation with Respect to the Domain in Boundary Value Problems”. In: *Numerical Functional Analysis and Optimization* 2.7-8 (1980), pp. 649–687. DOI: 10.1080/01630563.1980.10120631.

- [75] T. Slawig. “Shape optimization for semi-linear elliptic equations based on an embedding domain method”. In: *Applied Mathematics and Optimization* 49 (2004), pp. 183–199. DOI: [doi.org/10.1007/s00245-003-0787-1](https://doi.org/10.1007/s00245-003-0787-1).
- [76] J. Sokolowski and J. Zolésio. *Introduction to Shape Optimization. Shape Sensitivity Analysis*. 1st ed. Springer, Berlin, Heidelberg, 1992. DOI: [10.1007/978-3-642-58106-9](https://doi.org/10.1007/978-3-642-58106-9).
- [77] K. Sturm. “Shape Differentiability Under Non-linear PDE Constraints”. In: *New Trends in Shape Optimization*. Ed. by A. Pratelli and G. Leugering. Springer, 2015, pp. 271–300. DOI: [10.1007/978-3-319-17563-8\\_12](https://doi.org/10.1007/978-3-319-17563-8_12).
- [78] T. Suchan, K. Welker, and W. Wollner. “A new shape optimization approach for fracture propagation”. In: *arXiv preprint arXiv:2210.05502* (2022). DOI: [10.48550/arXiv.2210.05502](https://doi.org/10.48550/arXiv.2210.05502).
- [79] R. Udawalpola and M. Berggren. “Optimization of an acoustic horn with respect to efficiency and directivity”. In: *International Journal for Numerical Methods in Engineering* 73.11 (2008), pp. 1571–1606. DOI: [10.1002/nme.2132](https://doi.org/10.1002/nme.2132).
- [80] A. Vogel et al. “10,000 Performance Models per Minute – Scalability of the UG4 Simulation Framework”. In: *in: Euro-Par 2015: Parallel Processing, J. L. Träff et al. (eds)*. Springer, 2015, pp. 519–531. DOI: [10.1007/978-3-662-48096-0\\_40](https://doi.org/10.1007/978-3-662-48096-0_40).
- [81] A. Vogel et al. “UG 4: A novel flexible software system for simulating PDE based models on high performance computers”. In: *Computing and Visualization in Science* 16.4 (2013), pp. 165–179. DOI: [10.1007/s00791-014-0232-9](https://doi.org/10.1007/s00791-014-0232-9).
- [82] D. N. Wilke, S. Kok, and A. A. Groenwold. “A quadratically convergent unstructured remeshing strategy for shape optimization”. In: *International Journal for Numerical Methods in Engineering* 65.1 (2006), pp. 1–17. DOI: <https://doi.org/10.1002/nme.1430>.