CAD-free Adjoint Shape Optimization of Floating Vessels

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Peter Marvin Müller

aus
Solingen

2024

Vorsitzender des Prüfungsausschusses:
Prof. Dr.-Ing. Jürgen Grabe

Gutachter:
1. Prof. Dr.-Ing. Thomas Rung
2. Prof. Dr. Michael Hinze

Tag der mündlichen Prüfung:
8. Dezember 2023

https://www.orcid.org/0000-0003-0369-1196

DOI: https://doi.org/10.15480/882.9233

Abstract

This thesis aims to improve fluid-dynamic shape optimization methods for non-parameterized (CAD-free) shapes. The focus is computing shape transformations for shapes with Lipschitz boundary and approximating the direction of the steepest descent in Banach spaces with the help of shape sensitivity information, which is calculated using an adjoint-based method. Further, considering geometric constraints and their algorithmic treatment is a central aspect of the thesis. The work investigates a p-Laplace relaxation to approximate Lipschitz transformations with the help of $W^{1,p}$ functions. Further attempts are made to improve the approximation of the steepest descent direction in $W^{1,\infty}$ using an approach based on the ADMM algorithm. The suggested approaches are applied in multiple numerical experiments, which successively deform a domain by point-wise movement of the computational grids.

Acknowledgements

This work was supported by the Deutsche Forschungs Gemeinschaft (DFG) within the Research Training Group GRK 2583 "Modeling, Simulation and Optimization of Fluid Dynamic Applications."

This dissertation results from my time at the Institute for Fluid Dynamics and Ship Theory at the Hamburg University of Technology. First, I want to thank my advisor, Prof. Dr.-Ing Thomas Rung, who introduced me to the beauty of fluid dynamics.
Thank you for the excellent supervision and broad support; I immensely enjoyed the intensive, engaging, and motivating cooperation. Further, I thank Prof. Dr. Michael Hinze, Martin Siebenborn, and Philip Herbert for helping me understand several advanced topics in mathematics; thank you for the patient explanations and enlightening discussion.
I also want to thank my colleagues at Hamburg University of Technology and the research training group at Universität Hamburg, which I gratefully was a part of. I want to thank Niklas Kühl for his broad support in my research's early and later stages. Also, I want to thank George Bletsos, Henrik Wyschka, and Malte Loft for all the discussions on shared aspects of our subjects, the comprehensive joint work, and your sense of humor.
Finally, I want to thank my family for their tremendous support. Thank you, Yvonne, especially for your patience, support, and backing me in many ways.

For Jonathan Emil.

Contents

Nomenclature

Operators

$e_y(y,\cdot)^*$	Adjoint operator of $e_y(y,\cdot)$
$A : B$	Double contraction of matrices A, B
$\langle\cdot,\cdot\rangle$	Dual pairing
$\det(DF)$	Functional determinant of the transformation F
∇V	Gradient of V
id	Identity operator
$(\cdot,\cdot)$	Inner product
DF	Jacobian matrix of a vector-valued function F
ΔV	Laplace operator applied to V
$\lvert DV\rvert$	Operator (spectral) norm of DV
$u \otimes v$	Outer (or tensor-) product of two vectors u, v
$e_V(\cdot,\Omega)V$	Partial derivative of $e(\cdot,\Omega_V)$ with respect to V, and evaluated at $V = 0$
$e_y(y,\cdot)\delta_y$	Partial derivative of $e(y,\cdot)$ with respect to y in direction of δ_y
$u \cdot v$	Standard inner product in $\mathbb{R}^d$
$\lvert\Gamma_{ij}\rvert$	Surface area of the facet shared by the i-th and j-th control volumes
$d(\Omega_p, \Omega_q)$	Symmetric difference of the two shapes Ω_p and Ω_q
$\nabla_\Gamma V$	Tanget gradient of V
$\det_\Gamma(V)$	Tanget divergence of V
$\operatorname{tr}(DV)$	Trace of the matrix DV
$DF^\intercal, DF^{-\intercal}$	Transposed of the Jacobian DF and its inverse
$\lvert\Omega_i\rvert$	Volume the contro i-th control volume

Function Spaces and Sets

$C^{0,1}(\Omega,\mathbb{R}^d)$	Lipschitz continuous functions
$H^m(\Omega,\mathbb{R}^d)$	m-times weakly differentiable and square-integrable vector-valued functions
H	Subspace of $H^m(\Omega)$ incorporating constraints, e.g., Dirichlet boundary conditions
$L^2(\Gamma)$	Square-integrable functions defined over Γ
$\mathcal{S}$	Collection of admissible geometries (shapes)

W, W^p	Subspace of $W^{1,\infty}(\Omega, \mathbb{R}^d)$ or $W^{1,p}(\Omega, \mathbb{R}^d)$ incorporating constraints, e.g., Dirichlet boundary conditions
$W^{1,p}(\Omega, \mathbb{R}^d)$	Weakly differentiable and p-integrable vector-valued functions with compact support Ω
$W^{1,\infty}(\Omega, \mathbb{R}^d)$	Weakly differentiable and essentially bound vector-valued functions
Y	Function space of the generic state variable y

Latin Letters

BC(Ω), Vol(Ω)	Barycenter and volume of the domain Ω
c_Ω	State variable for concentration used a two-phase flow problem associated with the domain Ω
d	Dimension of the physical space
e_1	First unit vector
$e(y_\Omega, \Omega)$	PDE constraints defining the state y_Ω associated with the domain Ω
F, F^{-1}	Pull back/push forward transformation
$g(\Omega)$	Geometric constraints
h_Ω	Adjoint state/Lagrange multiplier associated with c_Ω
$J(\Omega)$	Cost function of the unconstrained problem
$\mathcal{J}(y, \Omega)$	Cost function of the problem with state constraints
$J'(\Omega)V$	Shape derivative of J at Ω in the direction V
k_Ω	State variable for turbulent kinetic energy of the k-ω turbulence model, associated with the domain Ω
$L(\cdot, \cdot, \cdot)$	Lagrange function
n	Unit normal vector field on Γ
$o(V)$	Landau notation "little-O of V"
p	Order of integrability
p_{max}, p_{inc}	Maximum and increment for solving the sequence of p-Laplace problems
q	Auxiliary variable used in the ADMM algorithm
Re	Reynolds number
S	Schur complement operator
T	Final simulation time in a transient flow problem
T_p	Computing time of one augmented Lagrange iteration in the p-harmonic approach
t	Physical time variable, step size of the shape update, or local tangent direction
t_τ	Tangent projection of the shear force
V	Deformation vector field in $W^{1,\infty}(\Omega, \mathbb{R}^d)$
V_p	Deformation vector field in $W^{1,p}(\Omega, \mathbb{R}^d)$
v_∞	Far-field or characteristic velocity

v_Ω, p_Ω, $\mathfrak{p}_\Omega$	State variables for velocity and pressure associated with the domain Ω
w_Ω, q_Ω	Adjoint state/Lagrange multipliers associated with v_Ω and p_Ω
x	Point in the d-dimensional physical space
y_Ω	Generic state variable associated with the domain Ω
$y_{\Omega_V} \circ F$	Composition of the function y_{Ω_V} and F

Greek Letters

α	Step size
β, β^*	Angle between face normal and cell center connecting line, and coefficient for turbulence modeling
Γ	Boundary of Ω
Γ_{obs}	Boundary of the obstacle Ω_{obs}
Γ_{in}, Γ_{out}, Γ_{wall}, Γ_{des}	Subsets of the boundary Γ
γ	Coefficient for turbulence modeling
Δx, Δy	Grid spacing in x- and y-direction
δ_y, δ_v, δ_p, δ_c	Test functions or directions used for directional derivatives
ϵ, ϵ_1, ϵ_2	Small numbers used in convergence criteria
η	Generic variable diffusivity in the Steklov-Poincaré method or local coordinate
Θ	Heaviside function
λ, λ_{BC}, λ_{Vol}	Lagrange multiplier
μ	Dynamic viscosity($\mu = \nu\rho$)
μ_y, μ_v, μ_p	Second test functions used in Newton's method
ν	Kinematic viscosity
ξ	Local coordinate
π	The number π
ρ	Fluid density
$\varrho, \varrho_{\text{BC}}, \varrho_{\text{Vol}}$	Penalty parameter
σ	Surface sensitivity distribution within the boundary form of $J'(\Omega)V$
τ	Hydrodynamic stress tensor, or Step size in the ADMM algorithm
τ^*	Adjoint stress tensor
Φ	Domain extension of velocity boundary values
ϕ	Solution of a generic PDE
χ, $\chi_{p,q}$	Indicator function
Ω	Open measurable domain of $\mathbb{R}^d$ with boundary Γ
$\hat{\Omega}$	Optimal domain/shape
Ω_V	Domain transformed by the vector field V
Ω_{obs}	Subset of a domain associated with an obstacle
ω	Specific turbulent dissipation rate of the k-ω turbulence model

Abbreviations

ADMM	Alternating Direction Method of Multipliers
CAD	Computer-Aided Design
FVM	Finite Volume Method
PDE	Partial Differential Equation
RANS	Reynolds Averaged Navier-Stokes
SP	Steklov-Poincaré
SP+p	Steklov-Poincaré and p-Laplace

Sub- and Superscripts

BC	Barycenter
i, j, k, l	Vector components, control volume and facet indices, and iteration counters
in, out, $wall$, obs, des	Boundary parts for inlet, outlet, wall, obstacle, and design surface
int, max	Initial and maximum value
$3ref$, $4ref$	3ed and 4th refinement level
Vol	Volume
Ω	Indicator for the state associated with the reference domain Ω
Ω_V	Indicator for the state associated with the transformed domain Ω_V

Chapter 1

Introduction

The ever-growing advances in computational sciences have made simulation-based design and optimization an indispensable tool for many engineering industries dealing with applications of, for example, fluid [SLY04; SL04; Oth08] or structural mechanics [AJT04; All+05; USD21]. The present thesis is motivated by shape optimization problems with fluid dynamics applications. The main focus is methods for the numerical approximation of descent directions and shape transformations for non-parameterized shapes with Lipschitz boundaries in fluid dynamic shape optimization studies.
This introduction begins with a brief overview of the current state of research on related subtopics of crucial interest. Further, essential engineering and applied mathematics references will be discussed, and links to the present thesis will be established.

Engineering Design Optimization

A typical optimization task in engineering is to improve the technical performance of a design. An example of structural mechanics is the ability of a building to withstand extreme environmental conditions while minimizing the use of materials. In aviation, for example, one might be interested in increasing the payload of an aircraft while reducing fuel consumption. This thesis, however, is mainly motivated by reducing the operational cost and needed resources for operating merchant vessels. In 2019, the world merchant fleet compromised about 55,655 vessels [Kel20]. The transport sector imitates about 25 % (8,258 Mt in 2018) of total CO_2 emissions worldwide. Even though the share from ships is only approximately 2 % to 3 % regarding the transport sector, the amount of 218 Mt is significant [Age20; IMO20]. Efforts in all areas of the maritime industry are necessary to meet the goal of the *International Maritime Organization* (IMO), namely, the reduction of greenhouse gas emissions by 50 % compared to 2008, while the demand is increasing. Shape optimization can contribute to reducing hydrodynamic drag for newly built vessels. The hydrodynamic drag force has a direct influence on the fuel consumption of a ship, which in turn accounts for about 50 % of the operational cost. Hence, reducing drag also plays an important role in economic aspects when building new ships. Therefore, one of the main tasks of a ship designer is to find a shape that minimizes the drag (or power requirement) while still meeting the needs for dead weight carriage and seaworthiness/maneuverability legal provisions.
A crucial aspect of the design is often the shape of the investigated device. To this end, various shape optimization methods have been developed to identify optimal – or rather improved – shapes, which minimize an objective functional J to a set of prescribed conditions. Such methods range from stochastic (or global) [Bäc96; GPK11; JS11] to deterministic (or local) [Jam88] procedures, with the latter requiring a descent direction for reducing the value of the objective function. Global approaches for finding the best possible design are challenging and may require expert knowledge to infer potentially better design candidates from the results. Moreover, suppose the shape, which

is supposed to be optimized, is not described by only a few parameters; global methods are deemed too time-consuming for an industrial process due to the nonlinear relationship of the potentially great number of industrial design variables and the computational effort.

In the context of local methods, calculating the descent direction is crucial regarding both the effort and the quality of the optimization process. In terms of the effort, it is interesting to reformulate the research question and ask not about the performance of a given design variant but rather what the design looks like at a given (desired) performance. The idea of adjoint-based local shape optimization is to provide an answer to this question, with which this introductory discussion continues.

Adjoint Shape Optimization

The adjoint approach, discussed in detail in Sections 2.2 and 2.3, allows the efficient computation of derivative information for an objective functional J, with respect to a general control. When attention is given to shape optimization, the control corresponds to the shape of a domain Ω, and J represents a physical quantity of the shape, e.g., the drag force experienced by an obstacle. The aim is to find a domain shape, which minimizes J. In addition to the control, the objective depends on a physical state variable, which also depends on the control variable. The resulting problem is an optimization problem with partial differential equation (PDE) constraints. In fluid dynamics, the latter typically governs the physics, e.g., the conservation of mass, momentum, and energy. The appeal of any adjoint method for such optimal control problems is that the computation of the derivatives is independent of the number of design or control variables of the problem, as this method does not require computing the derivative of the state variable with respect to the control. Adjoint-based local optimization has been matured toward an industrially applied strategy, e.g., [Oth08; PG16; Küh+22]. There are different approaches to developing the adjoint simulation model, the continuous-adjoint and the discrete-adjoint method [GP97; GP00; PD10].

A challenge of adjoint-based shape optimizations in fluid dynamics applications is the often very stiff, densely coupled governing equations and optima that may be fairly sensitive to apparently small changes of characteristic similarity parameters (e.g., the Reynolds number) [Stü12]. An example problem refers to the *NASA 2D Bump-in-channel*, which is a typical verification case for turbulence modeling of high Reynolds number flow [Lan]. Depending on the Reynolds number, the flow topology can substantially change in response to subtle geometry changes. For example, when considering free surface flows with marine applications, small changes in the geometry of a ship's hull may cause significant changes in the objective [Krö16; Küh21]. Another methodical example is the adjoint equations, which correspond to turbulent flow models that involve auxiliary modeling equations. To reduce the complexity – especially in the context of continuous-adjoint methods – simplified adjoint approaches are often used in technical applications, which do not take into account the direct influence of turbulence variables on the gradient of the target function [SL04; DB06a; Oth08; KMR21].

Disregarding the above-discussed issues the adjoint-based local optimization is deemed most efficient, and the (continuous-adjoint) approach is the basis of this dissertation. The method successively updates the domain in an iterative optimization process to minimize the objective J. Here, the domain is understood as a set, or more precisely, the subset of the two- or three-dimensional Euclidean space. Another possible shape description is, e.g., an a priori parametrization, as it is used in CAD frameworks with a finite number of design variables. In this thesis, however, the former approach is considered. An advantage of the non-parameterized (CAD-free) approach is that it allows an almost free choice of shapes with minimal restrictions. The following section discusses the optimization procedure by point-wise deformation of the domain.

Descent Direction

A crucial part of an efficient shape optimization procedure is choosing a "good" descent direction. In this work, the actual design variable refers to the underlying geometry Ω, which is a subset of the d-dimensional Euclidean space $\mathbb{R}^d$, the domain's boundary $\Gamma = \partial\Omega$, or parts of the boundary $\Gamma_{\text{des}} \subset \Gamma$. Considered domains are Lipschitz domains, that is, domains with a Lipschitz boundary. The descent direction corresponds to a deformation vector field V providing the shape update rule of the form $\mathrm{id} + V$, where id is the identity operator. The perturbation of the identity $\mathrm{id} + V$ transforms a shape by point-wise moving the domain along the trajectory of V[1].

In many engineering applications, the objective J is of integral form; that is, J is an integral over the domain Ω, or the boundary Γ. With the help of the shape calculus concept, see [SZ92; DZ11] for a mathematical perspective or [SR13; Küh+19] for an engineering perspective, the shape derivative of J often can be formulated as integral over the design boundary, cf. Section 2.2. Thus, the shape derivative of J is of the form $J'(\Omega)V = \int_\Gamma \sigma(V \cdot n)\, ds$, where σ is called the local shape sensitivity. Because computing the shape derivative of a complex optimization problem is rather involved, Appendix A shows a formal derivation of $J'(\Omega)V$ for the drag functional and the Navier-Stokes equations, which is frequently used in this thesis.

First attempts in shape optimization used the directional derivative $J'(\Omega)V$, also called the shape derivative, to formulate a shape update rule [SZ92; DZ11; Kaw+00]. In [Pir73], for example, the shape is updated using the local shape sensitivity σ by deforming the boundary Γ in the direction of the unit normal vector n. Using the descent direction $V = -\sigma n$, which yields $\sigma(V \cdot n) = -\sigma^2(n \cdot n) = -\sigma^2$, and $J'(\Omega)V < 0$, is a popular approach for reducing the objective functional value, cf. [Jam+06; Jam04]. From an engineering perspective, the attempt is limited since it yields shapes with rough/noisy boundaries [SR11; KR15] and distorted near-wall meshes, which in turn hamper the preservation of numerical accuracy during the optimization procedure [HSB14; SHB14]. Mathematically speaking, the deformation $V = -\sigma n$ is irregular, and the update is performed using the derivative rather than the gradient, which is not a valid operation[2].

In fluid dynamics applications, it is favorable to morph the whole computational grid, rather than generating a new one after each optimization step. The motivation for this is the far superior efficiency when using restart features to obtain a good initial guess for solving the fluid dynamics problem. Consequently, one is interested in a high-quality mesh morphing strategy, which heavily depends on the shape update procedure. One approach is based on defining a shape gradient ∇J by some inner product and the shape derivative J'. In such a Hilbert space setting (a vector space equipped with an inner product) the shape gradient is the Riesz representation of the directional derivative J'. However, because the order of integrability and differentiability depend on the spatial dimension d [Hin+09, Theorem 1.14], which restricts the choice of valid Hilbert space setting when considering Lipschitz transformations, see also [ADJ21].

In this regard, different descent directions are associated with different shape descriptions and transformation approaches. Several techniques have been proposed to increase the regularity of shape updates:

a) CAD-related shape definitions connect the node-based shape derivatives to the CAD parameterization using the chain rule of differentiation, cf. [SL04] and [Rob+12]. The procedure couples the various local derivatives and thereby ensures smooth shapes. However, the rigid finite dimensional initial CAD parameterization limits the attainable shapes, and different CAD models may result in different optimal shapes.

b) A coupling of mesh node updates using either local shape functions, e.g., FE-type functions [SL02; SLY04], or global shape functions, e.g., Hicks-Henne approaches [HH].

[1] It may be mentioned upfront that, in this thesis, the deformation vector field V is an element of the Banach space $W^{1,\infty}(\Omega, \mathbb{R}^d)$. See Chapter 6 for details.

[2] When omitting the fact that in $L^2(\Gamma)$ is not a subspace of $W^{1,\infty}(\Omega, \mathbb{R}^d)$, one could argue that $V = -\sigma n$ is the shape gradient associated with J', defined by the $L^2(\Gamma)$ standard inner product.

Both approaches (a,b) are not in the scope of the parameter-free shape optimization and are not pursued here. On the other hand, when the control corresponds to the (non-parameterized) shape itself, an additional step is required to identify admissible descent directions and sequentially applicable shape updates.

c) A more rigorous approach of [VGJ05; Jam+06] applies an implicit, continuous smoothing operator to either the shape derivative J' or the deformation field V, based on the definition of an inner product, frequently labeled "Sobolev-gradient". For example, formally using the $H^1(\Gamma)$ standard inner product leads to a diffusion problem on the surface, which involves the Laplace-Beltrami operator [ADJ21]. For computational reasons, the practice can be performed in an explicit manner, cf. [HSB14]. The explicitly filtered local shape sensitivity, e.g., using consistent kernel functions [KR15], marks a first-order approximation to the implicit Sobolev-gradient [SR11].

All above mentioned Hilbert space invoking methods display room for improvement concerning the obtained shapes and the mesh quality after large deformations, and in particular, when the optimal shape features sharp corners, cf. [DHH22]. This is an important starting point and motivation for this dissertation. Moreover, though all strategies (a-c) couple node updates and obtain smooth shape updates, they all leave the subject of the subsequent domain mesh open, i.e., the shape gradient of the design surface needs to be extended into the domain.

Extension to the Domain

The habitat of the shape gradient depends on the domain of definition and can be surface or volume-based. Prominent examples refer to the above-mentioned Laplace-Beltrami (LB) or the Steklov-Poincaré (SP) approach, cf. [SS16]. The LB approach exclusively operates in the tangent space of the design surface and requires solving a PDE on the surface of the domain. The SP strategy leads to a deformation of the design surface by mapping the shape sensitivity, entering as Neumann data, to Dirichlet values of the shape deformation field V [SS16; HSU21; ADJ21]. A similar approach has been used in [AW96; AFA13], where the descent direction is obtained by solving a linear elastic-like problem with the negative shape derivative as Neumann data. The approach is referred to as H^1 gradient method or *traction method* because the Neumann condition can be interpreted as fictional traction acting on the surface of the deformed surface. The volume-based SP approach and traction method are particularly attractive for optimization procedures that prefer mesh morphing over re-meshing strategies, as is customary for engineering simulations. While re-meshing can be automated, the lack of fair restart capabilities becomes prohibitively expensive in practical applications. Moreover, using standardized, High-Performance-Computing (HPC) capable solution routines supplied by the flow solver (assembling, solving, etc.) represents another significant benefit of the volume-based approaches.
Another crucial aspect refers to the preservation of the mesh quality during the optimization, which secures the convergence of the optimization procedure and the reliability of the sensitivity guidance. Therefore, the present thesis aims to harmonize innovative shape update rules with a volume-based habitat of the shape deformation.

1.1 Contributions of Thesis

Although smooth shapes may be desirable for different reasons, they are not necessarily optimal. The above strategies obtain smooth shape updates and display difficulties in capturing optima, which feature kinks and corners. One may obtain shapes, almost resolving such features with a high resolution of the computational mesh. In particular, when not considering an adaptive grid refinement strategy, the initial mesh must be refined in regions where sharp corners and kinks may develop. Thus, one requires expert knowledge about the shape, which will be obtained by

the optimization procedure. This, however, is not always the case. The same applies to the counterpart situations, that is, to transform an initially kinked shape into a curved optimum. Though this might be possible, the respective convergence is often relatively slow.

The principal contribution of the thesis is developing a viable concept for shape transformations in a CAD-free/node-based context. The proposed method is related to approximating the direction of the steepest descent in the $W^{1,\infty}$-topology [DHH22] and characterized by the fact that the domain topology is retained, but corners and edges should be able to appear or disappear if this supports the optimization. As outlined in Chapter 3, the p-Laplace relaxation approach leads to a volume-based formulation, like the above-mentioned SP approach. In addition, a first implementation is used for handling geometric constraints. Related aspects are published in [Mül+21]. In an attempt to reduce the computational cost while retaining the benefits, the subsequent Chapter 4 presents a hybridization of the p-Laplace and the Steklov-Poincaré method which shares content with another recent publication [MBR23].

A second contribution of engineering relevance refers to the procedural consideration of technical constraints. These ensure the practical applicability of the optimization results by avoiding the convergence to unusable, trivial solutions, such as moving the obstacle out of the domain or reducing its displacement. The efficient incorporation of related strategies into the procedure for computing the descent direction is discussed in Chapter 5, and aims to reduce the computational effort and at the same time increase the robustness of the optimization algorithm. Related aspects are published in a recent paper by the author and his colleagues [Mül+23], which outlines parallel computing aspects.

A third contribution is concerned with the efficiency of the computing process, which is of vital interest for engineering applications, as well as algorithmic means to approach large p-values ($p \to \infty$) without escalating the computational effort. Strictly speaking, the efficiency problems fall into two main areas, namely a) increasing iterative effort for solving the nonlinear p-Laplace problem for large values of p, b) and the significant increase in computational effort associated with geometric constraints. In addition to handling constraints outlined in Chapter 5, a novel solution strategy is used in Chapter 6, which is formulated based on the alternating direction method of multipliers (ADMM), first suggested in [DHH23] for shape optimization problems with Lipschitz domains. The algorithm solves a hard-to-solve non-linear optimization problem by breaking it into a sequence of easier-to-solve sub-problems.

A fourth contribution deals with validating the industrial feasibility of the implemented concepts using unstructured mesh configurations for internal and external flows. Applications relate to 2D and 3D fluid dynamic shape optimizations ranging from laminar to turbulent flows. The most demanding application is discussed in Chapter 7 and refers to the drag reduction of a container vessel while keeping the center of buoyancy and the displacement of the ship constant. The application follows the material presented in [MSR23].

1.2 Notation

This thesis uses common mathematical notation. However, a few details are pointed out for clarification. As mentioned above, the symbol $\Omega \subset \mathbb{R}^d$ denotes a domain in the d-dimensional Euclidean space, $d \in \{1, 2, 3\}$. The boundary of the domain Ω is denoted by Γ, and parts of the boundary include a subscript. The symbol J is the objective function of the shape optimization problem, and L is the corresponding Lagrange function. Here the gradient of a scalar field $f : \mathbb{R}^d \to \mathbb{R};\ x \mapsto f(x)$ is considered a column vector: $\nabla f(x) := (f_{x_1}, \ldots, f_{x_d})^\intercal$. The partial derivative of

$f(x) = f(x_1, \dots, x_d)$ with respect to the i-th component is denoted by f_{x_i} or $\frac{\partial f}{\partial x_i}$. For better readability, the argument x of a function $f(x)$ is neglected if there is no risk of confusion. The Jacobian of a vector field $V : \mathbb{R}^d \to \mathbb{R}^d$ is given by the matrix

$$DV := \begin{pmatrix} \frac{\partial V_1}{\partial x_1} & \cdots & \frac{\partial V_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial V_d}{\partial x_1} & \cdots & \frac{\partial V_d}{\partial x_d} \end{pmatrix} = \begin{pmatrix} (\nabla V_1)^\intercal \\ \vdots \\ (\nabla V_d)^\intercal \end{pmatrix}.$$

The symbol $\circ$ denotes the composition of functions, that is, for the functions $x \mapsto f(x)$ and $y \mapsto x = g(y)$, the composition of f and g reads $(f \circ g)(y) = f(g(y))$. Most of this work uses a compact notation for vector-vector, matrix-vector, and matrix-matrix operations. However, the Einstein summation convention is employed when it is more appropriate. The convention implies a sum over small Latin indices occurring twice within an expression. For example, the inner product of two vectors u, v is denoted as $u \cdot v = u_i v_i = \sum_{i=1}^d u_i v_i$, and the double contraction of two matrices is denoted by $A : B = a_{ij} b_{ij} = \sum_{i=1}^d \sum_{j=1}^d a_{ij} b_{ij}$. The outer (or tensor-) product of two vectors u, v is denoted as $u \otimes v = uv^\intercal = u_i v_j = \sum_{i,j=1}^d u_i v_j e_i e_j$. Here e_i and e_j are the vectors where the i-th and j-th components are 1 and the others are 0. Furthermore, for some linear functional $L : X \to \mathbb{R}$, where X is a Banach space, $\langle L, x \rangle := L(x)$ denotes the dual pairing of X, its dual space X^*, see [Hin+09, Definition 1.5]. One may read $\langle L, x \rangle$ as the linear operator $L \in X^*$ is applied to $x \in X$.

1.3 Structure of Thesis

The dissertation is structured into eight chapters. Following this introduction, Chapter 2 outlines the mathematical framework and the rationale of the shape optimization problem. The chapter briefly discusses possible shape descriptions and how sets are made variable. Moreover, it outlines first-order optimality conditions and how shape sensitivities are computed using the adjoint approach. In Chapter 3, a p-Laplace relaxation of the direction of steepest descent is introduced and scrutinized for geometrically constrained 2D and 3D drag minimization problems of fully submerged objects at a range of Reynolds numbers. Emphasis is given to the convergence rate, the attainable quality of the pointed shape, and the preservation of the grid quality. The subsequent Chapter 4 describes a hybrid SP+p-Laplace method and verifies the preservation of grid-quality aspects in 2D duct flows at a low Reynolds number using unstructured, locally refined grids. A novel, more efficient, and robust approach, based on the formally introduced p-Laplace relaxation, for handling geometric constraints is discussed in Chapter 5. Section 5.4 applies the method in a two- and three-dimensional experiment using a multi-grid approach and studies the scalability of the suggested algorithm. Chapter 6 discusses a method for approximating the direction of steepest descent in $W^{1,\infty}$, based on the ADMM algorithm. The chapter compares the discrete solution obtained from the p-Laplace relaxation and the ADMM-based approximations in a one- and 2D model problem. The approximated direction of the steepest descent is applied in a 2D shape optimization problem, considering a back-tracking line search approach to determine a feasible step size. A maritime two-phase flow application is addressed in Chapter 7, and the manuscript closes with a summary and conclusions in Chapter 8.

Chapter 2

General Shape Optimization Problem

This chapter introduces the basic shape optimization concept and the notation used in this thesis. Section 2.1 briefly recalls the domain properties and employed transformations. A central aspect of this thesis is optimizing Lipschitz domains, discussed below in more detail.
This section considers the general shape optimization problem of the form

$$\min \; J(\Omega), \qquad \Omega \in \mathcal{S}, \tag{2.1}$$

where Ω is a bounded domain with boundary Γ, and $\mathcal{S}$ is a collection of admissible geometries (i.e. subsets of the d-dimensional space $\mathbb{R}^d$) [SZ92]. Note that the objective functional J, later in this work, depends not only on the domain but also on a physical state, which is the solution of a PDE. The set $\mathcal{S}$ depends on the particular optimization problem, i.e., in most cases, $\mathcal{S}$ has to adhere to technical constraints, such as fixed boundaries, fixed fluid displacement, or fixed barycenter locations, cf. Chapter 5.
For computing sensitivities, the domain has to be made variable. This thesis follows the approach that is described in [SZ92; DZ11; ADJ21], where a reference domain Ω is transformed in the sense of the *perturbation of the identity* $(\mathrm{id} + V)$. For this purpose, the domain is parameterized with the displacement field $V : \Omega \to \mathbb{R}^d$, which provides the transformed domain Ω_V, viz.

$$\Omega_V = (\mathrm{id} + V)(\Omega) := \{x + V(x) : x \in \Omega\}\,. \tag{2.2}$$

It is assumed that the displacement field V conserves the topological properties of the reference domain. Mathematically, this is called a homeomorphism, which, roughly speaking, neither creates holes nor closes existing ones when applying the transformation $\mathrm{id} + V$. To this end, the set of admissible domains $\mathcal{S}$ is linked to a set of admissible transformations, and the domain Ω_V is obtained by moving the points of the reference domain Ω along the trajectory of the vector field V. For better readability, this thesis uses the abbreviation

$$F = \mathrm{id} + V \tag{2.3}$$

for the perturbation of the identity and

$$DF = I + DV \tag{2.4}$$

for its Jacobian, where I is the $d \times d$ identity matrix.

2.1 Domain and Transformation Properties

This section briefly explains the term regularity in the context of shapes and recalls how to describe a domain mathematically. Further, some essential properties of the shape deformation field V are summarized.
The aim is to solve optimization problems constrained by PDEs. However, the solution to a boundary value problem usually depends on the regularity of boundary Γ. The classical ways to describe a domain and its properties are summarized in [DZ11, Chapter 2]. One possible interpretation is transforming a "flat" reference plane (or interval in the 2D case) onto the boundary. The properties of the transformation then reflect the local properties of the domain. A local description means that the boundary is characterized piece-wise, and the contained information is valid within a neighborhood at some point on the boundary. However, one can also think of the boundary as the level set of a function. Unlike the transformation approach, the level set interpretation has the advantage of a relatively straightforward way of defining the normal vector field. A third viewpoint is the local boundary description as the graph of a function, where the smoothness of the boundary is given by the smoothness of the describing function. With the help of the latter interpretation, the regularity and smoothness of a domain are brought to the local properties function. The latter allows an illustrative interpretation of the regularity and smoothness of domains. Figure 2.1 illustrates a 2D domain ($\Omega \subset \mathbb{R}^2$) with boundary Γ and a global x, y-coordinate system. As sketched, the boundary Γ is the graph of a function in the local

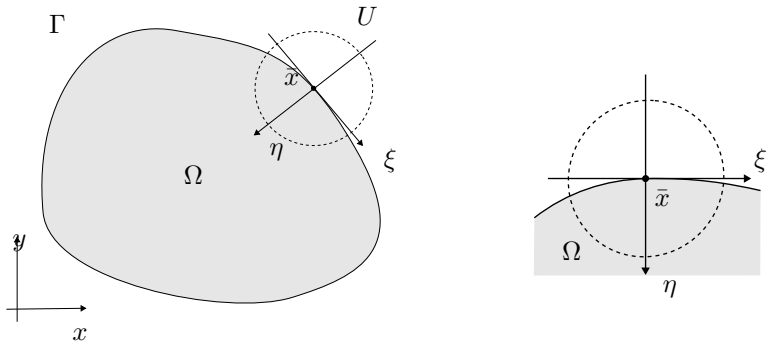


Figure 2.1: A domain Ω with boundary Γ. In a neighborhood U of the point $\bar{x} \in \Gamma$, the boundary is the graph of a function in the local ξ, η-coordinate system and the domain Ω is located on one side of this graph.

ξ, η coordinates within a neighborhood U at each point $\bar{x} \in \Gamma$. The domain Ω lies on one side of the graph (e.g., at $\eta > 0$). The regularity of a domain refers to the smoothness of the boundary and, thus, the smoothness of the function graph. A Lipschitz domain is characterized by boundaries that refer to the graph of Lipschitz functions; for a definition, see [Hin+09, Definition 1.13]. Grossly speaking, Lipschitz functions are functions with a bounded gradient. Thus, Lipschitz functions are continuous and cannot feature jumps. Most domains of engineering applications are Lipschitz domains. Figure 2.2 illustrates examples of domains, which are not Lipschitz domains. The 2D domain on the left features a cusp and is not a boundary of a Lipschitz function. The circle in the center features a horizontal slit of infinitesimal thickness. The right sketch refers to Maz'ya's two-brick domain, where the boundary in the neighborhood of P fails to be the graph of any function [HMT07].
Demanding a Lipschitz boundary of the domain may not be the minimum required regularity for shape optimization problems in general, but considering Lipschitz domains is sufficient for most problems. This thesis focuses on optimizing Lipschitz domains for at least the two following practical reasons: First, the initial or optimal shape shall be sufficiently smooth for a PDE solution

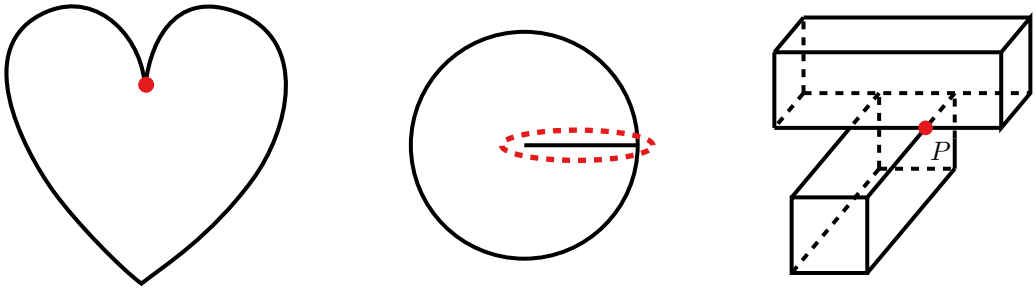


Figure 2.2: Examples for non-Lipschitz domains. The $2D$ domain on the left has a cusp at the top. The circle in the center features a horizontal slit, and on the right is Maz'ya's so-called two-brick domain [HMT07, Section 4.1].

and simultaneously feature sharp edges and pointy tips. Second, the existing literature provides a sound theoretical basis for PDE solutions in such domains; see, e.g., [DZ11; Hin+09; GRS07] for a broad overview and second-order elliptic equations in particular, and [Tem10] for Stokes and Navier-Stokes equations.

2.2 Optimality Condition

Solving the optimization problem (2.15) requires some condition, which characterizes the specific shape where J is minimal. For the finite-dimensional case $J : \mathbb{R}^n \to \mathbb{R}$, it is known that if $\hat{x} \in \mathbb{R}^n$ is a minimizer, it holds that $\nabla J(\hat{x})^\intercal (x - \hat{x}) \geq 0$ for all $x \in \mathbb{R}^n$. One method for finding a critical point (a point where ∇J vanishes) is solving the equation $\nabla J(x) = 0$ with $x \in \mathbb{R}^n$ using, for instance, Newton's method (if J is two times differentiable) or, e.g., a gradient descent method. However, when the solution to an optimization problem is a function y that is an element of a Banach space Y instead of the $\mathbb{R}^n$, the condition must be more general. That is, in the case of $J : Y \to \mathbb{R}$, and $\hat{y} \in Y$ is a function, which minimizes $J(y)$, then the variational form $J'(\hat{y})(y - \hat{y}) \geq 0$ holds for all admissible directions $y - \hat{y} \in Y$, where J' is the Gâteaux or Frechét derivative J, [De 15, Section 3.2]. These conditions can be transferred to problems where the variable is a shape with the help of variable domains from the former section and the following definition of the derivative with respect to a shape.
Following [ADJ21, Definition 4.1], J is called *shape differentiable* at Ω if the mapping $V \mapsto J(\Omega_V)$ is Fréchect-differentiable at $V = 0$. The shape derivative of J in direction V, evaluated at Ω is denoted by $J'(\Omega)V$ and fulfills the approximation condition

$$J(\Omega_V) = J(\Omega) + J'(\Omega)V + o(V) \quad \text{with} \quad \|V\|_{W^{1,\infty}(\Omega,\mathbb{R}^d)} \to 0. \tag{2.5}$$

An optimal (local) solution of the shape optimization problem (2.1) is characterized by the following optimality condition. Suppose that $\hat{\Omega} \in \mathcal{S}$ is a local optimum of J. Then the relation

$$J\big(F(\hat{\Omega})\big) \geq J(\hat{\Omega}) \tag{2.6}$$

holds for all sufficiently small directions $V \in W^{1,\infty}(\Omega, \mathbb{R}^d)$. And by linearity of the Frechéchet derivative (2.5) follows that

$$J'(\hat{\Omega})V \geq 0 \tag{2.7}$$

holds for all admissible directions $V \in W^{1,\infty}(\Omega, \mathbb{R}^d)$.
In many applications, the objective is given in integral form; that is, J is an integral over the domain Ω or the boundary Γ. An essential part of computing the shape derivative is the change

of variables formula for such objectives. That is, for a bounded domain with Lipschitz boundary $\Omega \subset \mathbb{R}^d$, a transformation (2.3), and the function $f : \Omega_V \to \mathbb{R}$, with $\Omega_V = F(\Omega)$, it holds that [DZ11, Chapter 9, Section 4.1]

$$\int_{\Omega_V} f \, dx = \int_{\Omega} (f \circ F) |\det(DF)| \, dx. \tag{2.8}$$

Note that from (A.1) follows that the functional determinant is positive, i.e., $|\det(DF)| = \det(DF)$ if V is small enough. The change of variables formula for surface integrals defined over $\Gamma_V = F(\Gamma)$ can be found in, e.g., [DZ11, Chapter 9, Section 4.2]. However, this thesis does not require the formula for surface integrals.
The following example outlines the derivation of the shape derivative of a prototype functional before turning to PDE-constrained shape optimization problems.

Example 1. *Considering the shape functional*

$$J(\Omega) = \int_{\Omega} f \, dx. \tag{2.9}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a given differentiable function. The expression

$$\begin{aligned} J(\Omega_V) - J(\Omega) &= \int_{\Omega_V} f \, dx - \int_{\Omega} f \, dx \\ &= \int_{\Omega} \Big[(f \circ F) \det(DF) - f \Big] \, dx \end{aligned} \tag{2.10}$$

is obtained utilizing (2.8) and (2.5). Inserting the first order expansion for $f(x)$ at $V = 0$: $(f \circ F)(x) = f(F(x)) = f(x + V(x)) = f(x) + \nabla f(x) \cdot V(x) + o(V(x))$; and for the determinant $\det(DF(x)) = \det(I + DV(x)) = 1 + \operatorname{div}(V(x)) + o(V(x))$ and reordering the terms leads to

$$\begin{aligned} J(\Omega_V) - J(\Omega) &= \int_{\Omega_V} \Big[(f + \nabla f \cdot V + o(V))(1 + \operatorname{div}(V) + o(V)) \Big] \, dx \\ &= \int_{\Omega} \Big[\nabla f \cdot V + f \operatorname{div}(V) \Big] \, dx + o(V) \\ &= J'(\Omega) V + o(V), \end{aligned} \tag{2.11}$$

and, thus, the shape derivative reads

$$J'(\Omega) V = \int_{\Omega} \Big[\nabla f \cdot V + f \operatorname{div}(V) \Big] \, dx. \tag{2.12}$$

Note that if $f(x)$ is constantly equal to 1, the expression for the shape derivative simplifies to

$$J'(\Omega) V = \int_{\Omega} \operatorname{div}(V) \, dx. \tag{2.13}$$

Furthermore, if the boundary Γ is sufficiently smooth (such that $f(x)$ exists on the boundary), there is an equivalent surface expression

$$J'(\Omega) V = \int_{\Gamma} f (V \cdot n) \, ds \tag{2.14}$$

that is obtained with integration by parts of (2.12). Furthermore, the above example is a specific version of the *Reynolds transport theorem*, which generalizes the *Leibnitz integral rule* for the differentiation under the integral sign with parameter-dependent integration bounds.

For the shape derivative of integrals over the boundary Γ see, e.g., [DZ11, Chapter 9, Secton 4.2] and [HP18, Section 5.2]. The derivation of a general formula involves tangential calculus and the change of variable formula for boundary integrals. However, as stated in [HP18, Proposition 5.4.4.], it is possible to derive rather general formulas for sufficient regular boundaries with the help of a domain extension and by applying (2.12). For example, [PS10] uses the latter approach for the shape derivative of the drag functional in gas dynamics.
The objective J in (2.1) only depends on the domain Ω. In engineering applications, however, the considered objective often has the form $\mathcal{J}(y, \Omega)$, with a physical state y that is the solution of a boundary value problem. To distinguish the role of the objective, the symbol $\mathcal{J}$ denotes an objective function, which depends on y and Ω, where J only depends on Ω. The general shape optimization problem with state constraints has the form

$$\min_{\Omega \in \mathcal{S}} \mathcal{J}(y, \Omega) \quad \text{subject to} \quad y \in Y, \quad e(y, \Omega) = 0, \tag{2.15}$$

where $e(y, \Omega)$ denotes the PDE constraint, which defines the state variable y, and Y is a suitable function space where the state y lives. For many applications the objective function J in (2.15) has the form of an integral over the domain Ω

$$\mathcal{J}(y, \Omega) = \int_\Omega j(\cdot, y)\, dx \tag{2.16}$$

or over the boundary Γ

$$\mathcal{J}(y, \Omega) = \int_\Gamma j(\cdot, y)\, ds, \tag{2.17}$$

where $j(x, y)$ is a suitable function, which depends on the state y and may also explicitly depend on the point $x \in \Omega$ or $x \in \Gamma$, respectively.

Example 2. *The prototype example of PDE-constrained optimization is the minimization of a tracking-type functional*

$$\mathcal{J}(y, \Omega) := \frac{1}{2} \int_\Omega (y - y_d)^2\, dx \tag{2.18}$$

where the state is given by the solution of the boundary value problem

$$\begin{cases} -\Delta y = f, & \text{in} \quad \Omega, \\ \quad y = 0, & \text{on} \quad \Gamma, \end{cases} \tag{2.19}$$

with the given functions y_d and f.

Problems of the structure of (2.15), where the solution (y, Ω) consists of the state y and control variable (here the domain Ω) is referred to as *optimal control problems* because y depends on Ω. Such problems occur not only in shape optimization, though typical for optimization problems with PDE constraints [Hin+09; De 15], e.g., optimal flow control and optimal heating. It is assumed that the state equation $e(\cdot, \Omega) = 0$ has a corresponding locally unique solution y_Ω for all $\Omega \in \mathcal{S}$ such that $e(y_\Omega, \Omega) = 0$ holds. That means $e(\cdot, \Omega) = 0$ defines a mapping $\Omega \mapsto y_\Omega$, which refers to the so-called control-to-state operator. In the following, the symbol y_{Ω_V} denotes the state on the transformed domain Ω_V, that is, y_{Ω_V} satisfies $e(y_{\Omega_V}, \Omega_V) = 0$. This itself gives the mapping $V \mapsto y_{\Omega_V}$, and one finds that

$$V \mapsto J(\Omega_V) = \mathcal{J}(y_{\Omega_V}, \Omega_V), \tag{2.20}$$

which takes the role of a reduced functional. Computing the shape derivative of objective (2.20) also requires the shape derivative of the state. A straightforward approach is applying the implicit function theorem to $e(y_\Omega, \Omega) = 0$ at $V = 0$. Differentiating the equality $V \mapsto e(y_{\Omega_V}, \Omega_V) = 0$ in the sense of (2.5) yields

$$e_y(y_\Omega, \Omega)\, y' V + e_V(y_\Omega, \Omega) V = 0 \tag{2.21}$$

where $y = y_{\Omega_V} \circ (\mathrm{id} + V)$ is the solution on Ω_V transported back to Ω. Assuming that $e_y(y_\Omega, \Omega)$ is locally invertible, the solution of the equation system

$$e_y(y_\Omega, \Omega)\, \delta_y = -e_V(y_\Omega, \Omega)V, \tag{2.22}$$

gives shape sensitivity of the state $\delta_y = y'V$ at $V = 0$, and the shape derivative of the reduced functional (2.20) is expressed by

$$J'(\Omega)V = \langle \mathcal{J}_y(y_\Omega, \Omega), \delta_y \rangle + \langle \mathcal{J}_V(y_\Omega, \Omega), V \rangle. \tag{2.23}$$

Here the symbol $\langle \cdot, \cdot \rangle$ denotes the dual pairing, see [Hin+09, Definition 1.5]. An illustrative example and a detailed sketch of the proof that shows the structure of (2.22) using a state constraint (2.19) can be found in [ADJ21, Section 4.4].
However, solving (2.22) is known to be numerically expensive for high dimensional control variables. The following so-called adjoint approach is independent of the number of control variables [Hin+09; De 15]. Introducing the so-called adjoint state z_Ω that is the solution to

$$e_y(y_\Omega, \Omega)^* z_\Omega = -\mathcal{J}_y(y_\Omega, \Omega), \tag{2.24}$$

where $e_y(y_\Omega, \Omega)^*$ is the adjoint operator of $e_y(y_\Omega, \Omega)$. With the adjoint state, for the shape derivative of the reduced functional (2.20) holds that

$$J'(\Omega)V = \langle \mathcal{J}_V(y_\Omega, \Omega), V \rangle + \langle e_V(y_\Omega, \Omega)^* z_\Omega, V \rangle. \tag{2.25}$$

The optimal shape $\hat{\Omega} \in \mathcal{S}$ and state $\hat{y} \in Y$ ($\hat{y}$ is an abbreviation for $y_{\hat{\Omega}}$) is a local solution to (2.15) and there exists an adjoint state $\hat{z} = z_{\hat{\Omega}}$ such that the following optimality system is satisfied:

$$\begin{aligned} e(\hat{y}, \hat{\Omega}) &= 0, \\ e_y(\hat{y}, \hat{\Omega})^* \hat{z} &= -\mathcal{J}_y(\hat{y}, \hat{\Omega}), \\ \langle \mathcal{J}_V(\hat{y}, \hat{\Omega}) + e_V(\hat{y}, \hat{\Omega})^* \hat{z}, V \rangle &\geq 0 \end{aligned} \tag{2.26}$$

for all $V \in W^{1,\infty}(\Omega, \mathbb{R}^d)$.
Note that typically, one considers the weak form of the state equation $e(y, \Omega) = 0$. Then the solution y is an element of a Sobolev space $Y = W^{k,p}(\Omega)$, $k \geq 1$, which depends on the domain Ω and y_{Ω_V} lives in the $W^{k,p}(\Omega_V)$. Thus, the function space depends on the transformed domain Ω_V, which itself depends on V. For dealing with such difficulty, there are two techniques suggested in [DZ11, Chapter-10]. This work follows the so-called *Function space parameterization*, where the state y_{Ω_V} is pulled back from the perturbed domain Ω_V onto the reference domain Ω, as done in Example 1.

2.3 Lagrangian Calculus

Deriving the optimality system (2.26), particularly for shape optimization problems, can be complex. What was presented in the previous section can be expressed in the following way. A convenient approach is formally using the method of Lagrange multipliers, which also allows writing the optimality system in a more compact form, and it turns out that the Lagrange multipliers precisely are the adjoint state. The Lagrange function, corresponding to (2.15), is defined by

$$L(y, V, z) := \mathcal{J}(y, \Omega_V) + \langle z, e(y, \Omega_V) \rangle, \tag{2.27}$$

and, when choosing the feasible state $y = y_{\Omega_V}$, which solves the state equation $e(y_{\Omega_V}, \Omega_V) = 0$, the identity

$$J(\Omega_V) = L(y_{\Omega_V}, V, z) \tag{2.28}$$

holds for arbitrary multipliers z. Differentiating the Lagrange function and evaluating at $V = 0$ gives

$$J'(\Omega)V = \langle L_y(y_\Omega, 0, z), y'V\rangle + \langle L_V(y_\Omega, 0, z), V\rangle, \tag{2.29}$$

where $y = y_{\Omega_V} \circ (\mathrm{id} + V)$ is the solution on Ω_V but transported back to Ω. Choosing $z = z_\Omega$ such that

$$\langle L_y(y_\Omega, 0, z_\Omega), \delta_y\rangle = 0, \tag{2.30}$$

for all $\delta_y \in Y$, which is exactly the adjoint equation (2.24), namely

$$\langle L_y(y_\Omega, 0, z_\Omega), \delta_y\rangle = \langle \mathcal{J}_y(y_\Omega, \Omega) + e_y(y_\Omega, \Omega)^* z_\Omega, \delta_y\rangle. \tag{2.31}$$

With $y = y_\Omega$ and $z = z_\Omega$, it follows from (2.29) and (2.30) that

$$J'(\Omega)V = \langle L_V(y_\Omega, 0, z_\Omega), V\rangle = \langle \mathcal{J}_V(y_\Omega, \Omega) + e_V(y_\Omega, \Omega)^* z_\Omega, V\rangle. \tag{2.32}$$

With the Lagrange multiplier formulation, the above optimality system can be written in the variational form

$$\begin{aligned} &\langle L_z(\hat{y}, 0, \hat{z}), \delta_z\rangle = \langle \delta_z, e(\hat{y}, \hat{\Omega})\rangle = 0 \quad \forall \delta_z, \\ &\langle L_y(\hat{y}, 0, \hat{z}), \delta_y\rangle = 0 \quad \forall \delta_y, \\ &\langle L_V(\hat{y}, 0, \hat{z}), V\rangle \geq 0 \quad \forall V \in W^{1,\infty}(\Omega, \mathbb{R}^d). \end{aligned} \tag{2.33}$$

The following two examples demonstrate how to derive the adjoint equations for a constrained optimization problem of the form (2.15) in one spatial dimension and a tracking type functional. In Example 3, the state is the solution to a second-order ordinary differential equation (ODE), and in Example 4, the state is the solution to the viscous Burgers equation. A third example demonstrates the derivation of (2.32).

Example 3. *The aim is to find the left and right boundaries of the open interval $\Omega = (x_\mathrm{L}, x_\mathrm{R})$ and a corresponding state $y : (x_\mathrm{L}, x_\mathrm{R}) \to \mathbb{R}$, which minimizes the objective*

$$\mathcal{J}(y_\Omega, (x_\mathrm{L}, x_\mathrm{R})) = \frac{1}{2}\int_{x_\mathrm{L}}^{x_\mathrm{R}} (y_\Omega - y_d)^2\, dx, \tag{2.34}$$

where y_Ω is the solution to the boundary value problem

$$\begin{cases} -\dfrac{\partial^2 y_\Omega}{\partial x^2} = f \quad \text{in } (x_\mathrm{L}, x_\mathrm{R}) \\ y_\Omega(x_\mathrm{L}) = 0, \\ y_\Omega(x_\mathrm{R}) = 0, \end{cases} \tag{2.35}$$

with the given function y_d and f. Recall the weak formulation of (2.35): Find $y = y_\Omega$ such that

$$\int_{x_\mathrm{L}}^{x_\mathrm{R}} \left[\frac{\partial y_\Omega}{\partial x}\frac{\partial \varphi}{\partial x} - f\varphi\right] dx = 0 \quad \forall \varphi \in C_0^\infty((x_\mathrm{L}, x_\mathrm{R})). \tag{2.36}$$

Following (2.27), the Lagrange function associated with the above optimization problem is defined by

$$L(y, V, z) := \frac{1}{2}\int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} (y - y_d)^2\, dx + \int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} \left[\frac{\partial y}{\partial x}\frac{\partial z}{\partial x} - fz\right] dx, \tag{2.37}$$

Differentiating the Lagrange function with respect to z in the direction of the test function δ_z, inserting $\delta_z = \varphi$, and evaluation at $V = 0$ recovers the weak formulation of the state equation. The derivative of L with respect to y and evaluated at $y = y_\Omega$ and $V = 0$ gives

$$\langle L_y(y_\Omega, 0, z_\Omega), \delta_y\rangle = \int_{x_\mathrm{L}}^{x_\mathrm{R}} (y_\Omega - y_d)\delta_y\, dx + \int_{x_\mathrm{L}}^{x_\mathrm{R}} \frac{\partial \delta_y}{\partial x}\frac{\partial z_\Omega}{\partial x}\, dx = 0 \tag{2.38}$$

for all test functions δ_y. With integration by parts of the second term, one obtains

$$\begin{cases} -\dfrac{\partial^2 z_\Omega}{\partial x^2} = -(y_\Omega - y_d) \quad in\,(x_\mathrm{L}, x_\mathrm{R}), \\ z_\Omega(x_\mathrm{L}) = 0, \\ z_\Omega(x_\mathrm{R}) = 0, \end{cases} \tag{2.39}$$

in a weak sense.

The following example illustrates how to derive the adjoint equation for the viscous Burgers' equation. The equation is, in particular, interesting because it is a simplification of the Navier-Stokes momentum equation.

Example 4. *This example considers the objective*

$$\mathcal{J}(y_\Omega, (x_\mathrm{L}, x_\mathrm{R})) := \frac{1}{2}\int_0^T \int_{x_\mathrm{L}}^{x_\mathrm{R}} (y_\Omega - y_d)^2 \, dx \, dt, \tag{2.40}$$

which is the time-dependent variant for a fixed time interval $[0, T]$ of the objective (2.34) from the previous example, and the state $y = y_\Omega$ satisfies

$$\begin{cases} \dfrac{\partial y_\Omega}{\partial t} + y_\Omega \dfrac{\partial y_\Omega}{\partial x} - \nu \dfrac{\partial^2 y_\Omega}{\partial x^2} = 0 \quad in\,(0, T] \times (x_\mathrm{L}, x_\mathrm{R}), \\ y_\Omega(t, x_\mathrm{L}) = 0 \qquad\qquad t \in [0, T] \\ y_\Omega(t, x_\mathrm{R}) = 0 \qquad\qquad t \in [0, T], \\ y_\Omega(0, x) = 0 \qquad\qquad x \in (x_\mathrm{L}, x_\mathrm{R}). \end{cases} \tag{2.41}$$

and the Lagrange function is defined by

$$\begin{aligned} L(y, V, z) &:= \frac{1}{2}\int_0^T \int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} (y - y_d)^2 \, dx \, dt \\ &+ \int_0^T \int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} \left[\left(\frac{\partial y}{\partial t} + y \frac{\partial y}{\partial x} \right) z - \nu \frac{\partial y}{\partial x} \frac{\partial z}{\partial x} \right] dx \, dt \end{aligned} \tag{2.42}$$

Differentiating L with respect to y in direction δ_y and evaluating at $V = 0$ leads to the expression

$$\begin{aligned} \langle L_y(y_\Omega, 0, z_\Omega), \delta_y \rangle &= \int_0^T \int_{x_\mathrm{L}}^{x_\mathrm{R}} (y_\Omega - y_d)\delta_y \, dx dt \\ &+ \int_0^T \int_{x_\mathrm{L}}^{x_\mathrm{R}} \left[\left(\frac{\partial \delta_y}{\partial t} + \delta_y \frac{\partial y_\Omega}{\partial x} + y_\Omega \frac{\partial \delta_y}{\partial x} \right) z_\Omega - \nu \frac{\partial \delta_y}{\partial x} \frac{\partial z_\Omega}{\partial x} \right] dx \, dt \end{aligned} \tag{2.43}$$

which characterizes the adjoint state z_Ω. With integration by parts of the terms $\frac{\partial \delta_y}{\partial t} z_\Omega$, $y_\Omega \frac{\partial \delta_y}{\partial x} z_\Omega$ and $-\nu \frac{\partial \delta_y}{\partial x} \frac{\partial z_\Omega}{\partial x}$, and using that $y_\Omega(0, x) = 0$ and $y_\Omega = 0$ at x_L and x_R, one obtains

$$\begin{cases} -\dfrac{\partial z_\Omega}{\partial t} - y_\Omega \dfrac{\partial z_\Omega}{\partial x} + \dfrac{\partial y_\Omega}{\partial x} z_\Omega - \nu \dfrac{\partial^2 z_\Omega}{\partial x^2} = -(y_\Omega - y_d) \quad in\,[0, T) \times (x_\mathrm{L}, x_\mathrm{R}), \\ z_\Omega(t, x_\mathrm{L}) = 0 \qquad\qquad t \in [0, T], \\ z_\Omega(t, x_\mathrm{R}) = 0 \qquad\qquad t \in [0, T], \\ z_\Omega(T, x) = 0 \qquad\qquad x \in (x_\mathrm{L}, x_\mathrm{R}), \end{cases} \tag{2.44}$$

which should be fulfilled in a weak sense.

Note that, in the second example, with the time-dependent state $y_\Omega(t,x)$ and adjoint state $z_\Omega(t,x)$, the sign of the term containing the time derivative and the convective term changes due to the integration by parts. Further, the initial value for the adjoint state is set at the end time $t = T$. Thus, the time integration is in the opposite direction, from $t = T$ to $t = 0$.

Example 5. *This example considers the minimization problem from Example 3. To compute the partial derivative of the Lagrangian (2.37) with respect to V, the integral is rewritten as integral over the reference domain $\Omega_0 = (x_\mathrm{L}, x_\mathrm{R})$. In the one-dimensional case, this is done with integration by substitution, a special case of the change of variables formula. In the following, x_0 denotes a point in the reference domain Ω_0 and x in the transformed domain $(\mathrm{id} + V)(\Omega) = (x_\mathrm{L} + V_L, x_\mathrm{R} + V_R)$. With (2.3) the transformed point reads $x = F(x_0)$, and with $dx = \frac{\partial F}{\partial x}(x_0)dx_0$ one obtains*

$$\begin{aligned} L(y_{\Omega_V}, V, z_{\Omega_V}) &= \frac{1}{2}\int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} \left((y_{\Omega_V} \circ F) - (y_d \circ F)\right)^2 \frac{\partial F}{\partial x}\, dx_0 \\ + \int_{x_\mathrm{L}+V_L}^{x_\mathrm{R}+V_R} &\left[\left((\frac{\partial y_{\Omega_V}}{\partial x} \circ F)(\frac{\partial z_{\Omega_V}}{\partial x} \circ F) - (f \circ F)(z_{\Omega_V} \circ F) \right) \frac{\partial F}{\partial x}\right] dx_0. \end{aligned} \tag{2.45}$$

The symbol $\cdot \circ \cdot$ denotes the composition of two functions. That is, $(y \circ F)_{\Omega_V}(x_0) = y_{\Omega_V}(F(x_0))$ is the composition of y_{Ω_V} and F, and $\frac{\partial y_{\Omega_V}}{\partial x} \circ F = \frac{\partial y_{\Omega_V}}{\partial x}(F(x_0))$ is the composition of the derivative $\frac{\partial y_{\Omega_V}}{\partial x}$ with the F. With $\frac{\partial F}{\partial x}(x_0) \neq 0$ for all $x_0 \in (x_\mathrm{L}, x_\mathrm{R})$, and $\frac{\partial y_{\Omega_V}}{\partial x} \circ F = \frac{\partial F}{\partial x}^{-1} \frac{\partial (y_{\Omega_V} \circ F)}{\partial x}$ following from the chain rule, the Lagrangian can be reformulated as follows:

$$\begin{aligned} L((y_{\Omega_V} \circ F), 0, (z_{\Omega_V} \circ F)) &= \frac{1}{2}\int_{x_\mathrm{L}}^{x_\mathrm{R}} \left[\left((y_{\Omega_V} \circ F) - (y_d \circ F)\right)^2 \frac{\partial F}{\partial x}\right] dx_0 \\ + \int_{x_\mathrm{L}}^{x_\mathrm{R}} &\left[\left(\frac{\partial F}{\partial x}^{-1} \frac{\partial (y_{\Omega_V} \circ F)}{\partial x} \frac{\partial F}{\partial x}^{-1} \frac{(z_{\Omega_V} \circ F)}{\partial x} - (f \circ F)(z_{\Omega_V} \circ F)\right) \frac{\partial F}{\partial x}\right] dx_0. \end{aligned} \tag{2.46}$$

with $\frac{\partial F}{\partial x} = \frac{\partial}{\partial x}(x + V(x)) = 1 + \frac{\partial V}{\partial x}$. Due to the adjoint approach, the shape derivative of J is decoupled from the shape derivative of y, and the expression in (2.32) only requires the partial derivative of L with respect to V. Thus, with the fixed variables y and z in the above expression and computing $L_V(y, \cdot, z)V$, evaluated at Ω, finally leads to

$$\begin{aligned} \langle L_V(y, 0, z), V\rangle &= \int_{x_\mathrm{L}}^{x_\mathrm{R}} \left[\frac{1}{2}(y - y_d)^2 \frac{\partial V}{\partial x}\right] dx \\ + \int_{x_\mathrm{L}}^{x_\mathrm{R}} &\left[(-2\frac{\partial y}{\partial x}\frac{\partial z}{\partial x}\frac{\partial V}{\partial x} - \frac{\partial f}{\partial x}Vz) + (\frac{\partial y}{\partial x}\frac{\partial z}{\partial x} - fz)\frac{\partial V}{\partial x}\right] dx \\ &- \int_{x_\mathrm{L}}^{x_\mathrm{R}} (y - y_d)\frac{\partial y_d}{\partial x} V\, dx. \end{aligned} \tag{2.47}$$

By inserting the specific state and adjoint variables $y = y_\Omega$ and $z = z_\Omega$ in the above expression, the shape derivative at $V = 0$ is $J'(\Omega)V = \langle L_V(y_\Omega, 0, z_\Omega), V\rangle$.

The objective functionals considered in this thesis are the hydrodynamic drag (in Chapters 3 and 7), power loss (in Chapter 4), and energy dissipation (in Chapter 5). The derivation of the respective shape derivatives follows the same procedure as the examples above illustrate. The difference is the spatial dimension, which implies higher dimensional derivative operators, integrals over a d-dimensional domain, and state variables containing scalar- and vector-valued functions. A formal derivation of the shape derivative of the hydrodynamic drag functional is supplied in Appendix A.

Chapter 3

A p-Harmonic Descent Approach

The chapter is devoted to using the p-Laplacian operator to optimize initially circular (2D) or spherical (3D) shapes with respect to the fluid dynamic drag force in laminar and turbulent flow conditions. The procedure suggests optimal shapes for the scenario illustrated in Figure 3.1 featuring point tips, especially for low Reynolds numbers. These are challenging to obtain, and the degree of the pointed shape formation usually depends strongly on the local density of the initial mesh at the location of the two tips. The central aim of this chapter is to develop a strategy to compute a descent direction that is 1. able to convert the initially smooth shape into pointed objects without excessively refining the initial grid prior at the forward and the aft tips of the object and 2. does not significantly deteriorate the grid quality. Computing the descent direction of the shape functional in the $W^{1,\infty}$-topology [ADJ21; DHH22] is considered a much better strategy than the conventional Hilbert space methods to assess a larger set of optimal shapes. However, computing a descent direction in $W^{1,\infty}$ requires the solution of a highly non-linear problem, and the numerical approximation may be computationally demanding. The strategy in this chapter uses a p-Laplace relaxation of the minimization problem, which defines the direction of the steepest descent in $W^{1,\infty}$. The goal is to compute a deformation vector field that lives in $W^{1,p}$ for p as large as possible as the $W^{1,p}$ solution tends to a $W^{1,\infty}$ solution for $p \to \infty$ [IL05]; the numerical experiments carried out in Chapter 6 and [Her23] underline this statement. Note that the solution to the limiting problem is not necessarily unique [DHH23]. It may be mentioned that, for $p = 2$, the p-Laplace relaxation recovers the Hilbert space approach suggested in, e.g., [ADJ21, Section 5.2]. The p-Laplace approach thus offers an elegant way to combine both methods and assess blended strategies. As suggested in [SS16], the domain is anchored in the domain by its barycenter, and the volume is preserved using an augmented Lagrange approach to exclude trivial solutions. To improve the efficiency, this part of the procedure will be examined in more detail and improved in Chapter 5.

3.1 p-Laplace Relaxation

The section outlines the basic idea of the p-harmonic approach. In the current setting, the set $\mathcal{S}$ is associated with the domain sketched in Figure 3.1, where $B \subset \mathbb{R}^d$ is a flow domain containing an obstacle Ω_{obs} with boundary Γ_{obs}, and the domain $\Omega = B \setminus \Omega_{\text{obs}}$ is fully occupied by a fluid. The obstacle's boundary Γ_{obs} is free for deformation while the outer boundary $\partial B = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{slip}}$ remains fixed. The intention for the algorithmic minimization of J, for a given domain $\Omega \in \mathcal{S}$, is to specify descent vector fields $V : \Omega \to \mathbb{R}^d$ such that $J'(\Omega)V < 0$ holds, where $J'(\Omega)V$ is the shape derivative of J in the sense of (2.5). However, descent in this context requires specifying an appropriate topology. Moreover, it frequently is required that Lipschitz domains Ω are mapped to Lipschitz domains $\tilde{\Omega}$. Common practical approaches use Hilbert Space methods, i.e., seek descent

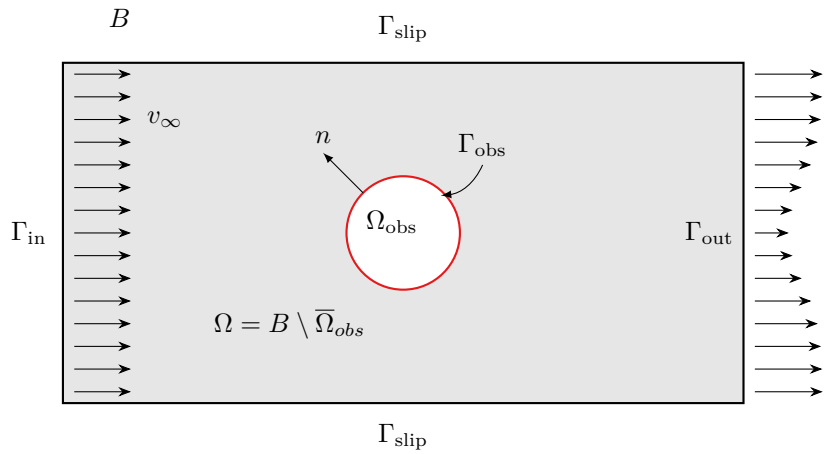


Figure 3.1: Sketch of the geometric configuration, where a fluid occupies the shaded region. The region of the obstacle is empty void space, i.e., a cutout not belonging to the flow channel. The red line denotes the obstacles boundary Γ_{obs}, which is deformed within the optimization procedure.

vector fields V determined with the help of the shape derivative by

$$a(V, W) = J'(\Omega)W \text{ for all } W \in H, \tag{3.1}$$

where $(H, a(\cdot,\cdot))$ denotes an appropriate Hilbert space, cf. [ADJ21, Section 5-9] for an extensive discussion of approaches related to the Hilbert space setting. This requires computing the Riesz representative of the functional $J'(\Omega;\cdot)$. A typical choice of H is $H^m(\Omega, \mathbb{R}^d)$, where, however, $m \in \mathbb{N}$ has to be chosen large enough to obtain a Lipschitz transformation. A way around this would be to directly choose $V \in W^{1,\infty}(\Omega, \mathbb{R}^d)$ as a direction of steepest descent for J at Ω, where $W^{1,\infty}(\Omega, \mathbb{R}^d)$ denotes the set of Lipschitz transformations from Ω to $\mathbb{R}^d$. The latter leads to the minimization problem

$$\min_{V \in W^{1,\infty}(\Omega,\mathbb{R}^d),\, |DV| \leq 1} \quad J'(\Omega)V, \tag{3.2}$$

where $|DV|$ is the operator (spectral) norm of the Jacobian DV. To obtain the direction of steepest descent, which is defined by (3.2), is challenging both from the mathematical and the numerical perspective since it represents a variational problem in the non-reflexive Banach space $W^{1,\infty}(\Omega, \mathbb{R}^d)$. Variational problems of this kind are studied in e.g. [IL05], where it is proposed to approach solutions $\hat{V}$ of problem (3.2) by a sequence of solutions $V_p \in W^{1,p}(\Omega, \mathbb{R}^d)$ of the variational problem

$$\min_{V_p \in W^{1,p}(\Omega,\mathbb{R}^d)} \frac{1}{p} \int_\Omega (\nabla V_p \colon \nabla V_p)^{\frac{p}{2}}\, dx + J'(\Omega)V_p \tag{3.3}$$

for $p > 2$, compare [IL05, Proposition 5.2, 5.3] for the mathematical analysis of the limit process $p \to \infty$. Note that for $p > 1$, this defines a strictly convex functional in V_p, and the particular case $p = 2$ recovers a Hilbert space setting as described above. The problem (3.3) is referred to as p-Laplace relaxation of problem (3.2). In addition, the descent direction is assumed to live in the space

$$W^p := \left\{ V_p \in W^{1,p}(\Omega, \mathbb{R}^d) : V_p = 0 \,\text{a.e. on}\, \Gamma \setminus \Gamma_{\text{obs}} \right\} \tag{3.4}$$

rather that the full $W^{1,p}(\Omega, \mathbb{R}^d)$, to ensure that the outer boundary, i.e., $\Gamma \setminus \Gamma_{\text{obs}}$, is not deformed. In the following section, the variational form (3.3) is adapted to the fluid dynamic problem, and its solution is used as descent directions to advance the shape optimization problem.

3.2 Optimization Problem

This section introduces the mathematical setting for our hydrodynamic shape optimization problem. Figure 3.1 shows the geometrical setting and outlines the notation used for parts of the domain and its boundary. The governing equations here are the stationary Navier-Stokes equations for incompressible fluids and laminar flows. However, at high Reynolds numbers, the flow is described by the Reynolds averaged Navier-Stokes (RANS) equations, which consider turbulence. Note that the latter case requires turbulence modeling. The aim is to find the shape of an obstacle $\Omega_{\text{obs}} \subset B$ with a Lipschitz boundary Γ_{obs}, located within the flow channel $B \subset \mathbb{R}^d$ and has minimal drag. The considered flow domain is given by $\Omega = B \setminus \Omega_{\text{obs}}$, as shown in Figure 3.1. The state then is given by the velocity $v_\Omega : \Omega \to \mathbb{R}^d$ and pressure $p_\Omega : \Omega \to \mathbb{R}$, which are assumed to be unique on Ω. The symbols v_Ω and p_Ω denote the velocity and pressure defined on Ω. The domain Ω is filled with a fluid that flows from the left inlet boundary Γ_{in} with a prescribed velocity $v_\infty \in \mathbb{R}^d$ into the domain towards the outflow boundary Γ_{out} on the right. The top and bottom boundaries Γ_{slip} are considered frictionless slip wall boundaries. For all test cases, the boundary Γ_{obs} of the obstacle Ω_{obs} controls the total force that acts on the obstacle Ω_{obs}. The drag is the force component pointing towards the approaching flow $-v_\infty/\|v_\infty\|$, and, thus, the drag functional reads

$$\mathcal{J}((v_\Omega, p_\Omega), \Omega) = -\frac{v_\infty}{\|v_\infty\|} \cdot \int_{\Gamma_{\text{obs}}} \left(\mu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right) - I p_\Omega\right) \cdot n \, ds, \tag{3.5}$$

where I is the identity tensor, and n is the (from Ω outward pointing) unit normal vector. The state $y_\Omega = (v_\Omega, p_\Omega)$ is the solution to the boundary value problem

$$\begin{cases} -\operatorname{div}(v_\Omega) = 0 & \text{in } \Omega, \\ \rho(v_\Omega \cdot \nabla)v_\Omega - \operatorname{div}\left(\mu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right)\right) + \nabla p_\Omega = 0 & \text{in } \Omega, \\ v_\Omega = 0 & \text{on } \Gamma_{\text{obs}}, \\ v_\Omega = v_\infty & \text{on } \Gamma_{\text{in}}, \\ v_\Omega \cdot n = 0, \quad \text{and} \quad n \cdot \tau \cdot t = 0 & \text{on } \Gamma_{\text{slip}}, \\ \mu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right) \cdot n = p_\Omega n & \text{on } \Gamma_{\text{out}}, \end{cases} \tag{3.6}$$

where the local tangent direction is defined by $t = t_\tau/\|t_\tau\|_2$ with the tangent projection of the shear force $t_\tau := \tau \cdot n - (n^T \tau n)n$ and $\tau = \mu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right) - I p_\Omega$, and $\rho, \mu \in \mathbb{R}_+$ are the density and molecular viscosity of the fluid.
In addition to the PDE constraint, which describes the fluid flow, one must consider additional geometric constraints

$$\text{BC}(\Omega) := \frac{\int_\Omega x \, dx}{\int_\Omega 1 \, dx} - \text{BC}_0 = 0, \quad \text{and} \quad \text{Vol}(\Omega) := \int_\Omega 1 \, dx - \text{Vol}_0 = 0. \tag{3.7}$$

The problem formulation in this chapter uses a prescribed location of the barycenter $\text{BC}_0 \in \mathbb{R}^d$ and volume $\text{Vol}_0 > 0$. These are necessary because, on the one hand, the geometry must stay in place to prevent the body from moving during optimization. On the other hand, the geometry would shrink to a single point if the volume is not preserved.
For the augmented Lagrange approach [NW06; And+08; SS16; ADJ21], the objective (3.5) is extended with quadratic penalty terms for the geometric constraints. With sufficient large penalty factors $\varrho_{\text{BC}}, \varrho_{\text{Vol}} \in \mathbb{R}_+$ the penalized objective reads

$$\mathcal{J}_\varrho((v, p), \Omega) := \mathcal{J}((v, p), \Omega) + \frac{\varrho_{\text{BC}}}{2}\|\text{BC}(\Omega)\|^2 + \frac{\varrho_{\text{Vol}}}{2}\text{Vol}(\Omega)^2. \tag{3.8}$$

It is clear that the $\mathcal{J}_\rho = \mathcal{J}$ if the geometry fulfills the postulated geometric constraints for arbitrary penalty factors $\varrho_{\text{BC}}, \varrho_v > 0$. The Lagrange function, associated with the above problem, is defined

as

$$
\begin{aligned}
L((v,p),V,(w,q),\lambda,\lambda_{\mathrm{BC}},\lambda_{\mathrm{Vol}}) := \mathcal{J}((v,p),\Omega_V) + \frac{\varrho_{\mathrm{BC}}}{2}\|\mathrm{BC}(\Omega_V)\|^2 + \frac{\varrho_{\mathrm{Vol}}}{2}\,\mathrm{Vol}(\Omega_V)^2 \\
+ \int_{\Omega_V} w \cdot \Big(\rho(v\cdot\nabla)v - \mathrm{div}\big(\mu\,(\nabla v + \nabla v^{\intercal})\big) + \nabla p\Big)\,dx \\
- \int_{\Omega_V} q(\nabla\cdot v)\,dx \\
+ \int_{\Gamma_V} \lambda\cdot v\,ds \\
+ \lambda_{\mathrm{BC}}\cdot \mathrm{BC}(\Omega_V) + \lambda_{\mathrm{Vol}}\,\mathrm{Vol}(\Omega_V),
\end{aligned}
\tag{3.9}
$$

where (w,q) are the Lagrange multipliers for the PDE constraints (3.6), λ is a Lagrange multiplier for the Dirichlet boundary condition $v_\Omega|_{\Gamma_{\mathrm{obs}}} = 0$, and $\lambda_{\mathrm{BC}} \in \mathbb{R}^d$ and $\lambda_{\mathrm{Vol}} \in \mathbb{R}$ are the multipliers corresponding to the geometric constraints. Not that additional PDEs and corresponding multipliers are required when using a turbulence model for the RANS equations. However, the influence of turbulence effects is assumed to be small enough to justify the frequently employed frozen turbulence assumption, and additional equations can be neglected, see [DB06a; Oth08; SR13]. Following (2.3), differentiation of (3.9) with respect to the state $y = (v,p)$ and integration by parts leads to the adjoint equation systems

$$
\begin{cases}
-\mathrm{div}\big(\mu\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\big) - \rho(v\cdot\nabla)w_\Omega + \rho\nabla v^T\, w_\Omega = -\nabla q_\Omega & \text{in } \Omega,\\
\nabla\cdot w_\Omega = 0 & \text{in } \Omega,\\
w_\Omega + \dfrac{v_\infty}{\|v_\infty\|} = 0 & \text{on } \Gamma,\\
w_\Omega = 0 & \text{on } \Gamma_{\mathrm{in}}\\
\rho(v_\Omega\cdot n)w_\Omega + \mu\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\cdot n = q_\Omega n & \text{on } \Gamma_{\mathrm{out}}\\
w_\Omega\cdot n = 0,\ n\cdot\tau^*\cdot t = 0 & \text{on } \Gamma_{\mathrm{slip}},
\end{cases}
\tag{3.10}
$$

where $\tau^* = \mu\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal}) - I q_\Omega$, and the multiplier associated with the Dirichlet boundary conditions

$$
\lambda = q_\Omega n - \mu\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\cdot n. \tag{3.11}
$$

Recall from (2.32) that the shape derivative of (3.5) reads

$$
J'(\Omega)V_p = L_V((v_\Omega,p_\Omega),0,(w_\Omega,q_\Omega),\lambda,\lambda_{\mathrm{BC}},\lambda_{\mathrm{Vol}})V_p. \tag{3.12}
$$

Together with the state (v_Ω,p_Ω) and the adjoint state (w_Ω,q_Ω), which satisfy (3.6) and (3.10), and assuming that $(\nabla v + \nabla v^{\intercal}) = \nabla v$, one can derive the surface formulation

$$
J'(\Omega)V_p = \int_\Gamma \sigma V_p\cdot n\,ds, \tag{3.13}
$$

with the surface sensitivity

$$
\sigma = -\mu\left(\frac{\partial w_\Omega}{\partial n}\cdot\frac{\partial v_\Omega}{\partial n}\right) + \lambda_{\mathrm{BC}}\cdot\frac{x-\beta}{\int_\Omega 1\,dx} + \lambda_{\mathrm{Vol}} + \varrho_{\mathrm{BC}}\frac{(\beta-\mathrm{BC}_0)(x-\beta)}{\int_\Omega 1\,dx} + \varrho_{\mathrm{Vol}}\Big(\int_\Omega 1\,dx - \mathrm{Vol}_0\Big), \tag{3.14}
$$

where

$$
\beta = \frac{\int_\Omega x\,dx}{\int_\Omega 1\,dx}. \tag{3.15}
$$

The first order optimality condition for the minimization problem (3.3), and replacing the space $W^{1,p}(\Omega, \mathbb{R}^d)$ with W^p, defined in (3.4), leads to the p-Laplace problem

$$\begin{cases} -\operatorname{div}\left((\nabla V_p \colon \nabla V_p)^{\frac{p-2}{2}} \nabla V_p\right) = 0 \text{ in } \Omega, \\ (\nabla V_p \colon \nabla V_p)^{\frac{p-2}{2}} \dfrac{\partial V_p}{\partial n} = -\sigma n \text{ on } \Gamma_{\text{obs}}, \\ V_p = 0 \text{ on } \Gamma \setminus \Gamma_{\text{obs}}. \end{cases} \tag{3.16}$$

The approach can also be interpreted as a generalization of the Steklov-Poincaré operator approach proposed in [SS16], where a linear boundary value problem is solved that realizes the Neumann-to-Dirichlet map applied to σn. The boundary value problem used in [SS16] is inspired by the linear elasticity problem from structural mechanics. Furthermore, in [AJT04; SS16], the descent direction is obtained by solving an elliptic boundary value problem similar to

$$\begin{cases} -\Delta V = 0 \text{ in } \Omega \\ \dfrac{\partial V}{\partial n} = -\sigma n \text{ on } \Gamma \\ V = 0 \text{ on } \partial\Omega \backslash \Gamma \end{cases} \tag{3.17}$$

which is the Euler-Lagrange equation of the minimization problem

$$\min_{V \in W^2} \quad \frac{1}{2} \int_\Omega \nabla V \colon \nabla V \, dx - \int_\Gamma \sigma n \cdot V \, ds, \tag{3.18}$$

with $W^2 = \{V \in H^1(\Omega, \mathbb{R}^d) : V = 0 \,\text{a.e. on}\, \Gamma \backslash \Gamma_{\text{obs}}\}$. The latter is a special case of (3.3) with $p = 2$ and serves as a reference for the numerical experiments discussed in Section 3.4. The abovementioned concept is enhanced in [OS21], using a non-linear extension operator. Observing a strong distortion of the discrete grid in the main deformation direction motivated adding a non-linear advection term to the PDE in (3.17). The additional advective character allows for large deformations with still reasonable mesh quality and no need for remeshing. However, as our numerical results show, the p-Laplace relaxation of problem (3.2) provides a systematic approach to fluid dynamic shape optimization and guarantees meshes of high quality even after large deformations. Moreover, the p-harmonic approach allows the development of sharp corners for the optimal shape.

3.3 Optimization Algorithm

Algorithm 1 outlines the minimization algorithm. A similar method was used before in, e.g., [SS16], to determine the multipliers λ_{BC} and λ_{Vol} for the barycenter and volume constraints for a shape optimization problem. Other than in [SS16], updating the penalty factors $\varrho_{\text{BC}}, \varrho_{\text{Vol}}$ and multipliers $\lambda_{\text{BC}}, \lambda_{\text{Vol}}$ are separated, which is motivated by the big difference in the order of magnitude ($\varrho_{\text{BC}} \propto 10^8$ and $\varrho_{\text{Vol}} \propto 10^2$) for the numerical experiments in this chapter. However, the critical step in Algorithm 1 is the shape optimization problem in Line 3, which is outlined in Algorithm 2, in more detail. To compute the multipliers λ_{BC} and λ_{Vol}, Algorithm 2 is performed with decreasing convergence tolerance ϵ_1 to tighten the feasibility of the shape. This also keeps the computation numerically stable and circumvents unfeasible shapes during the optimization.

The Computational Method used, is a conventional, pressure-based, second-order accurate finite-volume scheme (*FreSCo*$^+$) for a cell-centered variable arrangement is employed to discretize the partial differential equations of the primal (3.6) and adjoint systems (3.10), cf. [Run+09; SR13; KMR21]. The basic discretization concept is outlined in Appendix B. The existing infrastructure

Algorithm 1 Augmented Lagrange procedure

Require: Ω_0, $\mathrm{BC}_0 \in \mathbb{R}^d$, $\mathrm{Vol}_0 \in \mathbb{R}_+$, $\lambda_{\mathrm{BC}} \in \mathbb{R}^d$, $\lambda_{\mathrm{Vol}} \in \mathbb{R}$, $\varrho_{\mathrm{BC}}, \varrho_{\mathrm{Vol}}, \varrho^{\mathrm{inc}}, \mathrm{tol}_{\mathrm{BC}}, \mathrm{tol}_{\mathrm{Vol}}, \epsilon_1, \epsilon_2 \in \mathbb{R}_+$

$k \leftarrow 0$
repeat
  $\Omega_{k+1} = \textsc{MinimizeDrag}(\Omega_k, \mathrm{BC}_0, \mathrm{Vol}_0, \lambda_{\mathrm{BC}}, \lambda_{\mathrm{Vol}}, \varrho_{\mathrm{BC}}, \varrho_{\mathrm{Vol}}, \epsilon_1)$
  $b \leftarrow \int_{\Omega_k} x\,dx / \int_{\Omega} 1\,dx - \mathrm{BC}_0$
  $c \leftarrow \int_{\Omega_k} 1\,dx - \mathrm{Vol}_0$
  if $\|b\| > \mathrm{tol}_{\mathrm{BC}}$ **then**
    $\varrho_{\mathrm{BC}} \leftarrow \varrho^{\mathrm{inc}} \varrho_{\mathrm{BC}}$
  else
    $\lambda_{\mathrm{BC}} \leftarrow \lambda_{\mathrm{BC}} + \varrho_{\mathrm{BC}}\, b$
  end if
  if $|c| > \mathrm{tol}_{\mathrm{Vol}}$ **then**
    $\varrho_{\mathrm{Vol}} \leftarrow \varrho^{\mathrm{inc}} \varrho_{\mathrm{Vol}}$
  else
    $\lambda_{\mathrm{Vol}} \leftarrow \lambda_{\mathrm{Vol}} + \varrho_{\mathrm{Vol}}\, c$
  end if
  $k \leftarrow k + 1$
  $\epsilon_1 \leftarrow \epsilon_1 / 2$
until $(\|b\|^2 + c^2)^{1/2} \leq \epsilon_2$

Algorithm 2 Shape optimization procedure

function $\textsc{MinimizeDrag}(\Omega_0, \mathrm{BC}_0, \mathrm{Vol}_0, \lambda_{\mathrm{BC}}, \lambda_{\mathrm{Vol}}, \varrho_{\mathrm{BC}}, \varrho_{\mathrm{Vol}}, \epsilon)$
  $j \leftarrow 0$
  repeat
    Compute $(v_{\Omega_j}, p_{\Omega_j})$ according to (3.6)
    Compute $(w_{\Omega_j}, q_{\Omega_j})$ according to (3.10)
    Compute σ according to (3.14)
    Compute V_p^{j+1} by solving the p-Laplace problem (3.16)
    Determine a sep size t_{j+1} such that $|J((\mathrm{id} + t_{j+1} V_p^{j+1})(\Omega_j))| \leq |J(\Omega_j)|$
    Update geometry $\Omega_{j+1} \leftarrow (\mathrm{id} + t_{j+1} V_p^{j+1})(\Omega_j)$
    $j \leftarrow j + 1$
    Compute $J'(\Omega_{j+1}) V_p^{j+1}$
  until $|J'(\Omega_{j+1}) V_p^{j+1}| \leq \epsilon |J'(\Omega_1) V_p^1|$
  return Ω_{j+1}
end function

and generic subroutines of the fluid solver can be re-used with limited effort for implementation. In the case of $p = 2$, the initial guess with $V_p = 0$ in Ω converges. However, to solve the problem for $p > 2$, a sequence of problems with growing values for $p > 2$ is solved to obtain an initial guess. That is, for the finite sequence $(p_1, \ldots, p_k)$, with $p_1 := 2$ and $p_k := p$, the solution of the p_l-Laplacian problem is used as initialization for solving the p_{l+1}-Laplace problem $(0 \leq l < k)$.

3.4 Applications

Three fluid dynamic applications are discussed to investigate the performance of the p-Laplace approach (3.3) in shape optimization. All cases are concerned with drag minimization at a steady state and subjected to conserve the wetted volume and its barycenter, cf. Figure 3.1. The first case starts from (3.16) with $p = 2$ and analysis the influence of increasing p for a frequently referenced $2D$ Stokes flow example that features a pointy oval optimum. Emphasis is given to (a) the convergence of the optimization, (b) the final shape and the attainable drag reduction as well as (c) the quality of the mesh updates . The second case demonstrates the applicability of the approach for an analog $3D$ configuration and analyses the same aspects (a-c). Since both initial applications refer to low Reynolds (low Re) number flows, a third example is supplemented to scrutinize the performance in a $2D$ turbulent flow at a high Reynolds number.

3.4.1 Drag Optimization in $2D$ Low Reynolds Number Flow

The first case studies the drag minimization of a $2D$ circular cylinder exposed to low Reynolds number flow for $p = 2, 3, \ldots, 6$. The computational domain is illustrated in Figure 3.1. The case is related to the setting initially described by Pironneau [Pir73]. Instead of the Stokes flow considered in [Pir73], a low Re Navier-Stokes formulation of the boundary value problem is employed.
The initial cylinder features a unit diameter ($D = 1[m]$) and is centered in a channel of length $50[m]$ and height $10[m]$. The fluid is characterized by a unit density and dynamic viscosity, i.e., $\rho = 1[kg/m^3]$, $\mu = 1[Ps \cdot s]$. Dirichlet conditions are imposed at the inflow Γ_{in} with $v_\infty = (1, 0)^T[m/s]$, which yields a unit Reynolds number $Re_D = 1$. Slip wall boundary conditions are applied to the top and bottom boundaries of the channel. Outflow boundary conditions are used along Γ_{out}. The initial grid is displayed in Figure 3.2 and features 927 evenly distributed cells

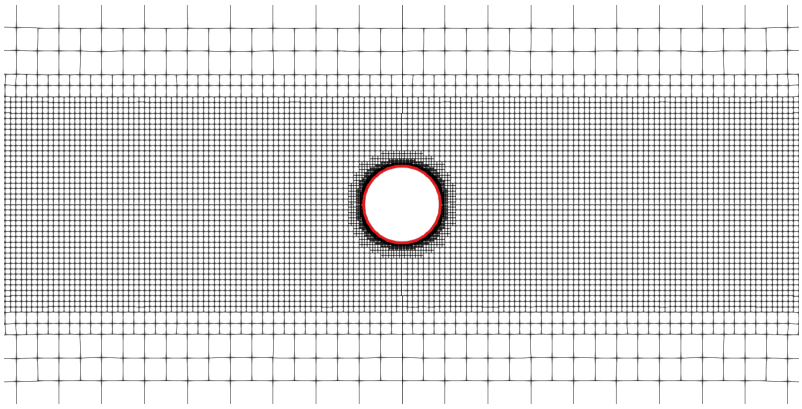


Figure 3.2: Initial configuration with the shape indicated by the red line, and the mesh of the $2D$ low Reynolds number problem.

along the circumference of the cylinder boundary Γ_{obs}. The CAD-free optimization procedure employs an unstructured, locally refined mesh with approximately 14.5k control volumes, which is deformed along with the surface using (2.2). Mind that the optimal shape is expected to reveal

pointy tips at the front and the aft, which shall deliberately not be preempted by the initial discretization of the boundary Γ. The target barycentre and wetted volume are set to $b_0 = (0,0)^T$ and $c_0 = 50 \cdot 10 - \pi/4[m^2]$ respectively. All parameters employed to initialize the Algorithms 1 and 2 are displayed in Table 3.1. The multipliers for the barycenter and volume are initialized with $\lambda_{\mathrm{BC}} = (0,0)^T$ and $\lambda_{\mathrm{Vol}} = 0$, respectively. The sequence of tolerances applied to the convergence criteria of the shape optimization problem is given by $\epsilon = 10^{-1}, 10^{-2}, \ldots, 10^{-7}$. However, it turned out that the multipliers λ_{BC} and λ_{Vol} converge very fast and after four augmented Lagrange steps the multipliers are determined sufficiently exact in each computation with different values for p. The field values for the fixed point iteration for $p = 2$ can be initialized with $V_p = 0$ for all discrete points. Larger p-values were initialized by solutions obtained from the previous smaller value, each to a suitable tolerance to provide an initial guess for the next following p-Laplace problem. Although the theory suggests driving $p \to \infty$, lower p-values are of interest for large-scale applications due to the more exhausting computational effort. The numerical effort for solving the p-Laplace problem is investigated in [Loi20] for different values for p, which show that it is of polynomial complexity but depends on p and the number of unknowns. The related experience from this investigation reveals an increase of computing time T_p for one iteration of the shape optimization Algorithm 2 by $T_{p=4}/T_{p=3} \approx 2.8$ and $T_{p=5}/T_{p=4} \approx 1.6$. In addition, the representable floating point arithmetic of the machine limits the value for p.

Parameter	$2D$ low Re	$3D$ low Re	$2D$ high Re
ϱ_{BC}	$5 \cdot 10^7$	$1 \cdot 10^3$	$1.2 \cdot 10^8$
ϱ_{Vol}	$1 \cdot 10^2$	50	$4 \cdot 10^2$
ϱ^{inc}	2	2	1.2
τ_b	$1 \cdot 10^{-6}$	$1 \cdot 10^{-1}$	$5 \cdot 10^{-5}$
τ_c	$2 \cdot 10^{-2}$	$1 \cdot 10^{-4}$	$2 \cdot 10^{-2}$
t	$2 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$1 \cdot 10^{-3}$

Table 3.1: Initial values for the parameters of the augmented Lagrange procedure.

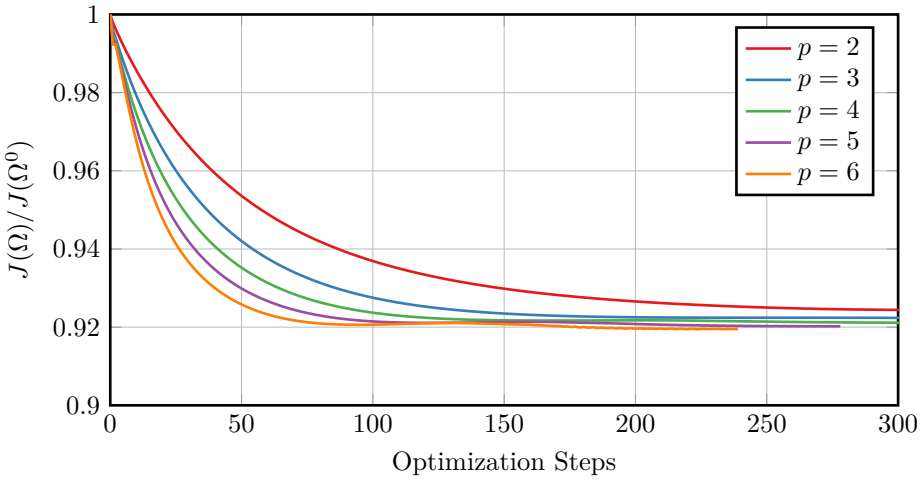


Figure 3.3: Influence of p-value on the evolution of the normalized objective functional (2D low Re case).

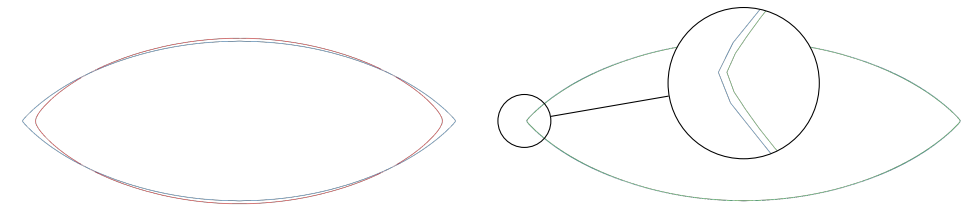
Optimal Shapes and Convergence

Figure 3.3 shows the normalized evolution of the drag objective (3.8) over a selected number of optimization steps for the five investigated values of p. The baseline solution refers to $p = 2$, which is also related to the investigations in [SS16]. Table 3.2 outlines a comparison of performance

p	Tip angle [°]	$a/b_{(it=50)}$	J/J^0	conv. step
2	163.8	1.4	0.9243	/
3	152.4	1.6	0.9222	415
4	140.4	1.8	0.9211	337
5	129.4	2.0	0.9202	278
6	116.4	2.1	0.9195	239

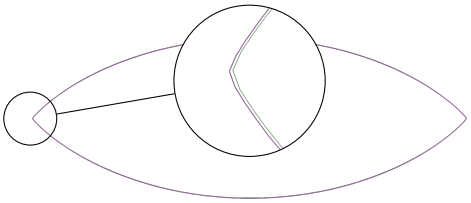
Table 3.2: Performance indicators obtained with different p-values for 2D low Re case.

indicators obtained for the five investigated values of p. The last column refers to the maximum number of design steps that are needed to reach a sufficiently converging objective function (cf. Algorithm 2). The table reveals that the convergence improves and fewer optimization steps are needed for larger values of p. For $p = 2$ the optimization could not reach convergence but terminated after 356 steps due to grid quality issues, which are discussed below. At this step, the convergence criterion was about two orders of magnitude above the threshold. However, from a practical point of view, the convergence criteria employed in this study might appear rather strict, and practical applications would also reach sufficient optima for $p = 2$. Table 3.2 also displays the final objective function values, which again reveal improvements for increasing p-values. Drag reductions refer to about 7.6% for the Laplacian approach with $p = 2$ and increase to approximately 8.1% for $p = 6$. Improvements seen for the objective function are attributed to the more extreme deformations obtained from the p-Laplacian problem (3.3) with $p > 2$. In order to judge the final shape, the opening angle at the upstream tip may also serve as a measure. The interior opening angles listed in Table 3.2 decrease with greater values of p. Hence, increasing p clearly yields more pointy tips, as also indicated by the comparison of tip shapes in Figure 3.4. The Stokes flow problem investigated in [Pir73] reported an opening angle of 120°. The present results rapidly approach the reported opening angle from above. However, for $p = 6$ the opening angle falls below the reference value, which is attributed to the use of Navier-Stokes instead of the Stokes flow model [Pir73]. When attention is directed to the convergence speed, the half-axis ratio at the 50th design step, i.e., $a/b_{(it=50)}$, mentioned in Table 3.2 may be considered as a measure to assess the convergence speed. As all simulations start with $a/b = 1$, the tabulated data renders the influence of p-values on the ability of the p-Laplace approach to rapidly adjust the shape. It is also observed that large deformations take place at an early stage of the optimization. For example, the final half-axis ratio for $p = 4$ refers to approximately 2.7, and already 2/3 of this ratio is reached after 50 design steps. A closer inspection of Figure 3.3 reveals that the value of the objective function slightly increases after a quick descent for $p = 5$ (step 120-160) and $p = 6$ (step 90-140). This is possible due to the absence of a step size control and occurs because the multipliers that control the barycenter and the displacement, strictly speaking, only hold for a preceding iteration. Moreover, the p-Laplace problem (3.3) is not solved exactly in every iteration of Algorithm 2 and preceding results are used as the initial guess for a subsequent optimization step. A similar phenomenon can be seen in results reported by Allaire et al. [AJT04], who used a similar augmented Lagrange procedure. The contours of the final shapes are depicted in Figure 3.4, where optimal shapes for two consecutive values of p are compared with each other, i.e., for $p = 2$ with $p = 3$, $p = 3$ with $p = 4$ and so forth. The shape contours in Figure 3.4a show significant differences, in particular at the tips of the resulting geometry. The computation with $p = 2$ clearly

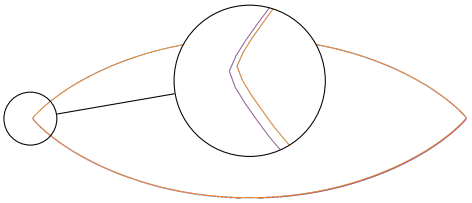


(a) $p = 2$ (red) and $p = 3$ (blue).

(b) $p = 3$ (blue) and $p = 4$ (green).



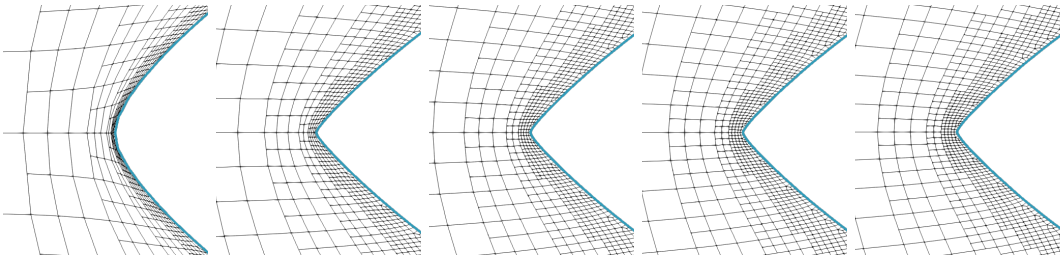
(c) $p = 4$ (green) and $p = 5$ (purple).



(d) $p = 5$ (purple) and $p = 6$ (orange).

Figure 3.4: Influence of the p-value on the predicted optimal (final) shapes for the 2D low Re case.

leads to a more rounded shape and a larger vertical extent than the other investigated p values. Shapes returned by $3 \leq p \leq 6$ are displayed Figures 3.4b to 3.4d. Remarkably, a general difference between the respective contours is hard to identify for $p \geq 3$. Thus, close-ups are used to assess the tip region. While a rounded tip region is still observed for $p = 3$, the tip becomes more pointy for $p \geq 4$. Figures 3.4c and 3.4d only display small differences between the shapes obtained with $p = 4, 5, 6$. Thus, one can assume that the predicted optimal shapes converge with p large enough, e.g., $p \geq 4$. The convergence behavior in p is estimated using the symmetric difference of the



(a) $p = 2$ (b) $p = 3$ (c) $p = 4$ (d) $p = 5$ (e) $p = 6$

Figure 3.5: Influence of the p-value on the final grids in the vicinity of the upstream tip (2D low-Re case).

optimized shape obtained using the p-Laplace method with $p \geq 2$ and the initial circular shape. In this work, the symmetric difference is the mapping $d\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, defined by

$$d(\Omega_p, \Omega_q) \coloneqq \int_{\mathbb{R}^d} \chi_{p,q}(x)\, dx \tag{3.19}$$

with $\Omega_i, \Omega_j \in \mathcal{S}$ and the indicator function

$$\chi_{p,q}(x) \coloneqq \begin{cases} 1 & : x \in (\Omega_p \setminus \Omega_q) \cup (\Omega_q \setminus \Omega_p), \\ 0 & : \text{else.} \end{cases} \tag{3.20}$$

Figure 3.6 illustrates the region where the indicator function (3.20) is not zero. Figure 3.7 shows

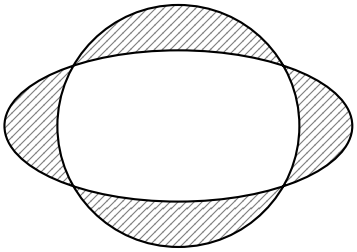


Figure 3.6: Illustration of the symmetric difference between an optimized shape Ω_p and the initial shape Ω_0. The shaded region ▨ marks where the indicator function (3.20) is not zero.

the difference, in the sense of (3.19), between the initial and the optimal shapes obtained for $p \in [2, 6]$. The symmetric difference between the initial and the optimal shape seems to converge with increasing $p \geq 3$. This observation shall not be taken as a proof but may show the relation of our approach to the investigations in [IL05]. Note that the reference case $p = 2$ is referred to

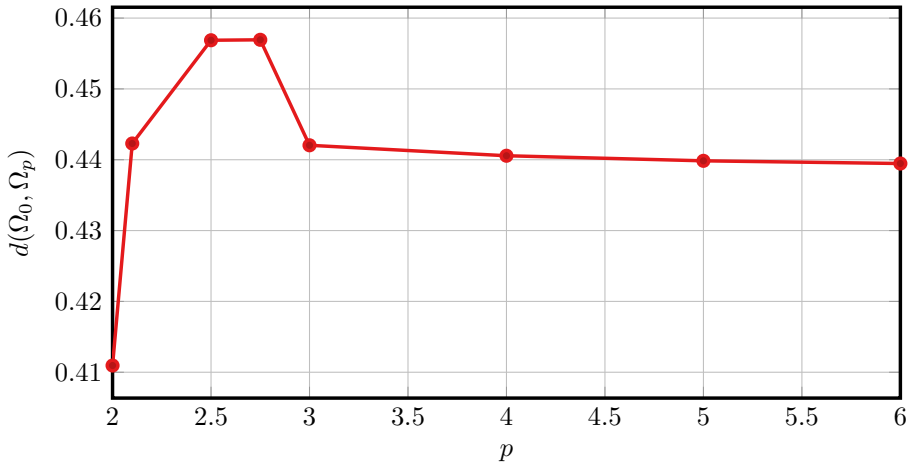


Figure 3.7: The symmetric difference between the optimized shape Ω_p and the initial shape Ω_0 for the $2D$ low-Re number case; the values are computed for $p = 2, 2.1, 2.5, 2.75, 3, 4, 5, 6$.

as a Hilbert space method and does not belong to $\{V_p\}_{p>2}$ with $V_p \in W^{1,p}(\Omega, \mathbb{R}^d)$. Therefore, the transition regime between $p = 2$ and $p = 3$ is assessed in greater detail. As indicated by Figure 3.7, an increase of p above the reference value $p = 2$ results in a rapid response of the symmetric difference between Ω_p and Ω_0.

Grid Deformation

Besides the convergence, the attainable optimal shapes, and the reduction of the objective function value, another major aspect refers to the quality of the mesh updates. Maintaining the grid quality

during the optimization process is crucial to the success of the CAD-free optimization procedure. The focus of this study is on 1. the grid orthogonality near the boundaries as well as 2. the cell aspect ratio in the vicinity of the walls. It is seen, that using the p-Laplacian problem (3.3) with $p > 2$ significantly improves the quality of the mesh updates in comparison with updates obtained from $p = 2$. Figures 3.5a and 3.5e display the final grids in the upstream tip region for the five investigated values of p. A reasonable grid quality is generally observed for large values of p, even after substantial cumulative deformations due to several hundred optimization steps. On the contrary, the aspect ratio of the near wall cells in Figure 3.5a increases significantly for $p = 2$. The cells become stretched and tend to buckle in the normal direction, which hampers the iterative convergence of the primal and adjoint flow solver. Therefore, the procedure terminated after 354 optimization steps for $p = 2$. In line with the change of the shape characteristics, a huge change of the mesh characteristics is observed when p is increased from $p = 2$ to $p = 3$, cf. Figures 3.5a and 3.5b. The grid is less compressed in the vicinity of the upstream tip for $p = 3$, despite the larger deviations from the initial grid indicated in Figure 3.4a. As outlined by Figures 3.5c and 3.5e, the grid does further improve when p is augmented to $p = 4, 5, 6$. A detailed comparison of the grids

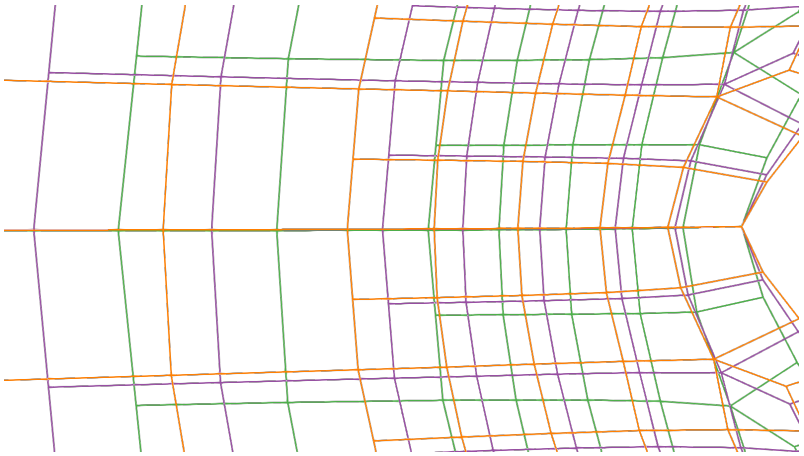


Figure 3.8: Superposition of upstream tip grids for $p = 4$ (green), $p = 5$ (purple) and $p = 6$ (orange) for the 2D low Re case.

at the upstream tip of the final shape follows from Figure 3.8 for $p = 4$ (green), $p = 5$ (blue) and for $p = 6$ (red). The post-processed grids are aligned at the tip to support the comparison. The stretching of the grid increases with increasing p, which becomes obvious when observing the spacing along the horizontal center line. Figure 3.9 depicts contour plots of the cell aspect ratio for the final grids obtained with $p = 2$ (left) and $p = 3$ (right). These p-values are particularly illustrative since they denote a threshold for the characteristics of the mesh and the shape. The aspect ratio (AR) of the initial grid is generally close to $AR = 1$. Only cells within the first two layers next to the cylinder boundary initially reach aspect ratios of up to $AR \leq 1.5$. If one focuses on the final meshes, computations using $p = 2$ lead to a substantial amount of cells where the aspect ratio exceeds values of $AR = 2.5$ and beyond, particularly in the vicinity of the tip. The maximum values approximately read $AR_{max} \leq 11$. A significant improvement is achieved in conjunction with $p = 3$. Here, only a few cells of the final mesh, located within a small area around the upstream tip, display aspect ratios greater than $AR > 1.5$, and the peak values of the aspect ratio are limited to $AR_{max} \leq 3$. The orthogonality of the cells within a mesh is of particular interest for the approximation of the boundary layer flow around the obstacle. Figure 3.10 displays contour plots for the minimum interior angle of the final meshes using $p = 2$ (left) and $p = 3$ (right). Displayed angles reach from a less favorable value of 45° (red) to a preferred value of 90° (blue). Skewed cells occur close to the tip and in regions where the shape tends towards a straight edge.

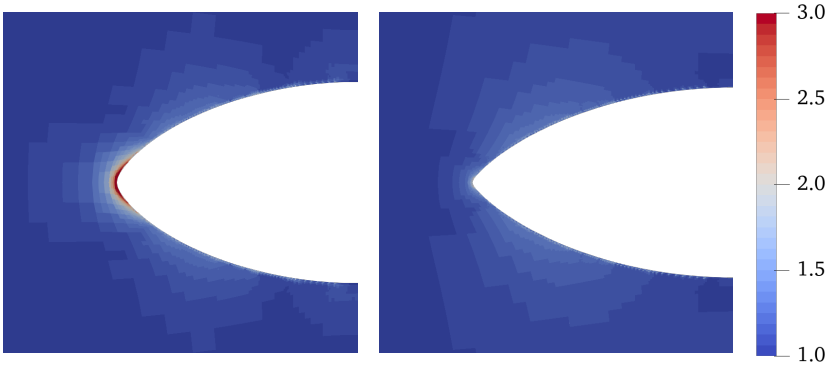


Figure 3.9: Cell aspect ratio of the final shapes obtained with $p = 2$ (left) and $p = 3$ (right) for the 2D low Re case.

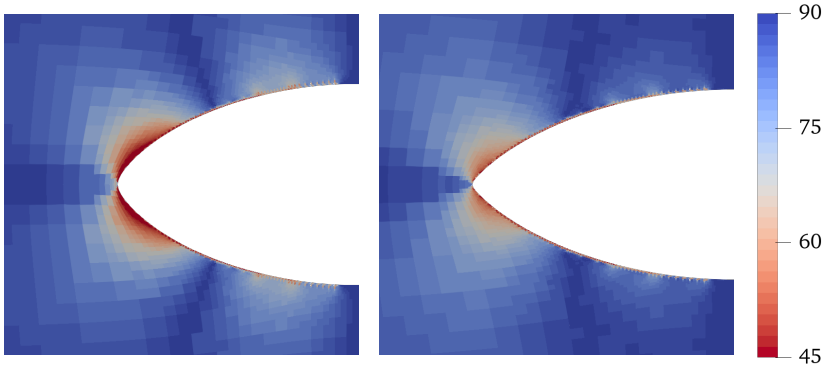


Figure 3.10: Minimum interior cell angle for the final meshes obtained with $p = 2$ (left) and $p = 3$ (right) for the 2D low Re case.

This effect is much more pronounced for $p = 2$ (left) and significantly less obvious for $p = 3$ (right). As illustrated by Figure 3.8, the characteristic features of the mesh do not substantially change for higher p-values.

3.4.2 Drag Optimization in $3D$ Low Reynolds Number Flow

This section is devoted to the drag optimization of a $3D$ unit-diameter sphere exposed to low Reynolds number flow. Reported results are limited to the baseline case $p = 2$ and a single augmented p-level of $p = 4$ – which did already display substantial benefits in the $2D$ study. The computational domain, the position of the obstacle's barycenter, and the boundary conditions agree with the $2D$ case outlined in Section 3.4.1. The $2D$ domain is supplemented in the lateral direction by $10[m]$, and slip wall conditions are imposed along the lateral boundaries. The Reynolds number compiled with the diameter and the approaching flow again reads $Re = 1$. The boundary

p	Tip angle [°]	$\frac{1}{2}(\frac{a}{b} + \frac{a}{c})_{(opt)}$	J/J^0	conv. step
2	176.4	1.61	0.9507	/
4	130.2	2.13	0.9360	164

Table 3.3: Performance indicators obtained with different p-values for the 3D low-Re case.

Γ of the sphere is resolved by approximately 113k wall adjacent cells and deforms under control. The total grid features about 1050k control volumes. The wetted volume is restricted to conserve $c_0 = 5000 - \pi/6[m^3]$ and the initial parameters of the augmented Lagrange Algorithm 1 are again denoted in Table 3.1. The performance observed with the two investigated p-values is summarized in Table 3.3. For $p = 4$ the optimization did converge after 164 design steps and yields 6.4% drag reduction. Similar to the 2D case, the lower quality of the volume grid update restricted the number of design steps for $p = 2$. Figure 3.11 depicts the last shape obtained from computations with $p = 2$ which referred to 94 design steps and 4.93% drag reduction. Figure 3.11a reveals that this shape is clearly characterized by round tips at the upstream and downstream ends.

(a) Side view.

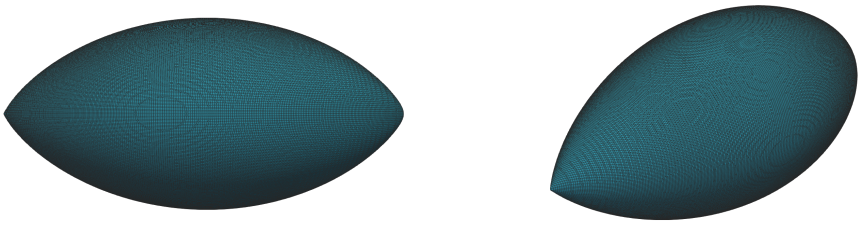
(b) Perspective view onto the upstream tip.

Figure 3.11: Drag optimization of a sphere at $Re = 1$; final shape obtained from $p = 2$ after 94 steps.

The optimal shape returned by $p = 4$ is shown in Figure 3.12. As also indicated by the data listed in Table 3.3, pointy upstream and downstream ends are seen for $p = 4$, which also results in a significantly larger average half-axis ratio.

3.4.3 Drag Optimization in $2D$ Turbulent Flow

Supplementary to the two low Reynolds number cases described above, this section reports the results obtained for a turbulent $2D$ drag optimization of an initial $a/b = 4/1$ ellipses where the longer half axis has a unit length of $a = 1[m]$. The incompressible fluid is characterized by a unit



(a) Side view. (b) Perspective view onto the upstream tip.

Figure 3.12: Drag optimization of a sphere at $Re = 1$; optimal shape obtained from $p = 4$ after 164 steps.

density and a dynamic viscosity of $\mu = 6.\overline{66} \cdot 10^{-7}$[Pa·s]. The governing equations refer to RANS equations using a standard k-ω eddy-viscosity turbulence model [Wil98] in combination with a wall-function approach. The Reynolds number is based on the longer axis of the ellipses, and the approach flow reads $Re = 3 \cdot 10^6$. The computational domain and the boundary conditions agree with the information already used in the first example, see Section 3.4.1. Results are again restricted to the baseline case $p = 2$ and $p = 4$. The wetted volume and the target barycenter are identical to those applied to the low Reynolds number experiment in Section 3.4.1. The initial values for the Algorithm 1 are denoted within the third column of Table 3.1. The initial grid

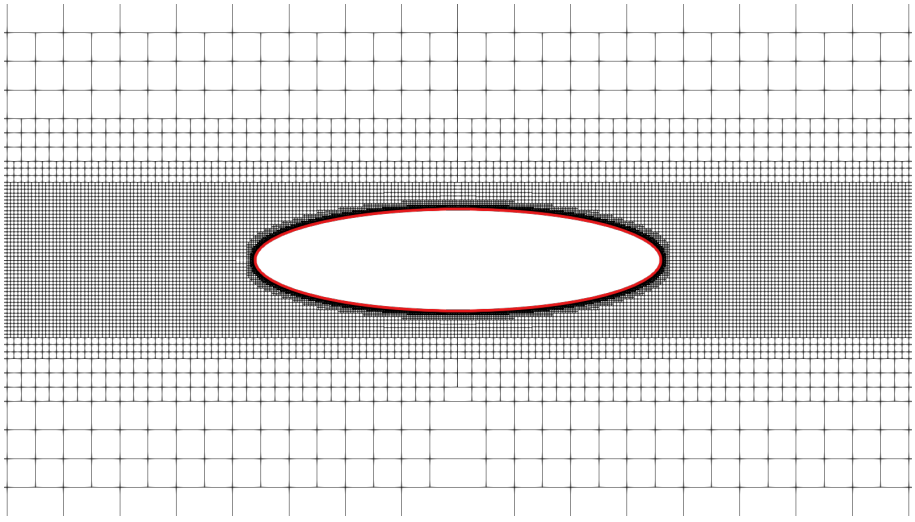


Figure 3.13: Initial mesh for the drag optimization of a $2D$ ellipses exposed to horizontal approach flow at $Re = 3 \cdot 10^6$.

again features local grid refinement near the boundary of the design surface to ensure an adequate resolution of the high curvature region and is depicted in Figure 3.13. The design surface Γ is discretized by approximately 2300 cells of equal size, and the volume grid features 46k cells. The final shape contours obtained for $p = 2$ (blue) and $p = 4$ (red) are outlined in Figure 3.14. Data listed in Table 3.4 reveals a significant difference in the shape and the drag reduction experienced with the different p-values. Similar to the $3D$ study, pointy upstream and downstream ends are seen for $p = 4$. The larger p-value results in a significant increase of the initial half-axis ratio, while a/b hardly increases for the smaller p-value. As in the previous studies, the shape optimization terminated before an optimal shape could be reached due to the severe distortion of the grid for

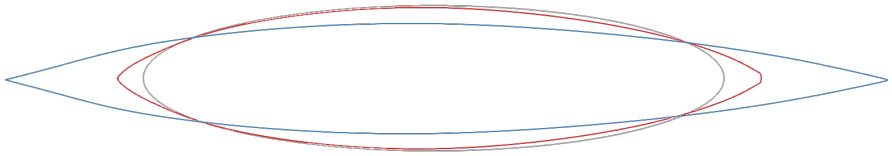
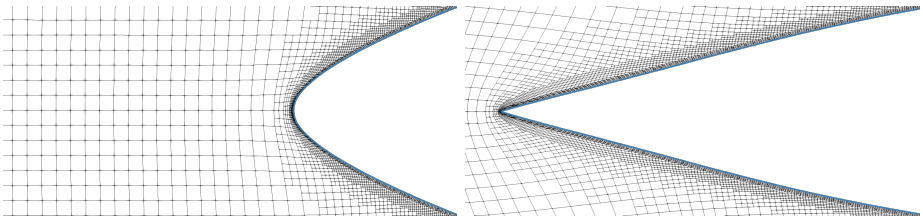


Figure 3.14: Drag optimization of a $2D$ elliptical cylinder exposed to horizontal approach flow at $Re = 3 \cdot 10^6$; Comparison of the final shapes obtained for $p = 2$ (red) and $p = 4$ (blue) against the initial shape (gray).

the baseline value $p = 2$. For the baseline approach $p = 2$, the drag force of the body is reduced by 10.57% within 540 optimization steps. However, for $p = 4$ the drag force is reduced by 32.6% when reaching the convergence criterion.

p	Tip angle [°]	$a/b_{(opt)}$	J/J^0	conv. step
2	149.5	4.55	0.894	/
4	29.4	8.02	0.674	589

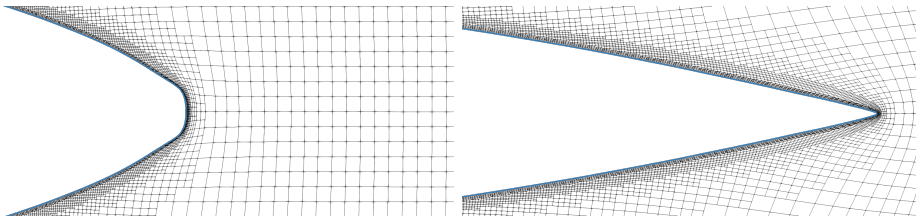
Table 3.4: Performance indicators obtained with different p-values for the $2D$ high Re case.



(a) $p = 2$ (b) $p = 4$

Figure 3.15: Upstream tip of final shapes for the $2D$ high Reynolds case.

Figure 3.15 compares the final grids in the vicinity of the upstream tip for $p = 2$ (left) and $p = 4$ (right). As indicated by Figure 3.15a, the aspect ratio deteriorates for $p = 2$ since the near wall cells stretch in the tangential direction. Moreover, cells (again) cluster at the tip. In contrast, the grid for $p = 4$ depicted in Figure 3.15b features evenly distributed cells along the design surface. The predicted shape displays a pointy upstream tip, which is a much better approximation of the solution to the optimization problem. A similar conclusion follows from Figure 3.16 which describes the situation at the downstream end. It is observed, that the pointy ends develop at



(a) $p = 2$ (b) $p = 4$

Figure 3.16: Downstream tip of final shapes for the 2D high Reynolds case.

later stages of the optimization process, particularly in the rear. Thus, the (almost) pointy rear is hardly reached for $p = 2$. In combination with $p = 4$, the cell distribution at the rear remains evenly distributed. Mind that a small round tip can still observed in Figure 3.16b. To receive a sharp pointy tip at the downstream end, a step size control would be beneficial.

3.5 Concluding Remarks on the Suggested p-Laplace Descent

This chapter reported on a novel approach for shape optimization by approximating Lipschitz continuous transformations based on the relaxation of the definition of the steepest descent direction. The main goal was to improve the shape optimization algorithm by considering a descent direction as the solution to the p-Laplace problem and investigating the influence of increasing p on the convergence of the shape optimization procedure as well as the obtained shapes and the related updates of the volume grid. An important aspect refers to the behavior of the relative differences when p is increased since this might guide towards sufficiently high p values associated with appreciated lower computational effort.
Results show that directions obtained from p-harmonic solutions improve the convergence with increasing p. At the same time, the optimal shapes improve regarding the value of the objective function. A remarkable feature is related to the ability of the p-Laplace approach to yield shapes with edges or pointy shapes, even when the initial shape does not contain such features. Furthermore, the quality of the computational grid is virtually preserved even when large deformations of the initial shape occur and no specific grid adjustment is required. Results of the present study suggest that $p = 4$ seems a sufficiently large p-value to gather the benefits of the p-harmonic approach.
While the convergence of the optimization improves with increasing p-values, the computational effort for the iterative solver increases substantially. Thus Chapter 5 aims to improve handling geometric constraints more efficiently and Chapter 6 investigates an alternative for approximating $W^{1,\infty}$ functions. Moreover, the application to a more complex 3D two-phase problem is investigated in Chapter 7. Aiming to reduce the computational cost, the following Chapter 4 outlines an attempt to hybridize the p-Laplace and the Steklov-Poincaré. The approach is compared to using the p-Laplace and Steklov-Poincaré approach only for a 2D internal flow.

Chapter 4

Shape Transformation Approaches

A critical aspect of CAD-free shape optimization is the mesh quality of the discrete mesh, which is used for evaluating the fluid dynamic variables. So far, it is common practice to generate a new computational mesh after performing a couple of shape updates, [Bla21; MBF23]. The mesh is usually deformed until the mesh quality is decreased, so computing the state becomes unreliable or even impossible. The ability to maintain the mesh quality means that one can restart from the previous design using the previous results as an initial guess for the primal and dual state. This, however, requires a high-quality morphing strategy, which maintains a "good" mesh quality (in the sense of Section 3.4.1), and ideally, the strategy updates the surface and volume mesh at the same time. The p-Laplace approach discussed in the previous chapter provides an excellent platform to achieve these goals. Especially the numerical results in Section 3.4.1 show that the above goals are achieved better and better for increasing p-values. At the same time, the method is computationally expensive, which motivates the investigation of alternatives.

To this end, the current chapter discusses

a) a Hilbert space method, implemented as a variant of the Steklov-Poincaré approach presented in [SS16] using a variable diffusivity coefficient [Küh+22],

b) the p-Laplace (Banach space) method outlined in Chapter 3,

c) a hybrid version.

While a) is deemed more efficient, the p-Laplace method b) preserves the mesh quality much better and can resolve pointed objects but requires solving a costly non-linear PDE. Because of a viable compromise between the above-stated goals and the required computational effort, the hybrid approach combines elements of both strategies to reduce the computing cost. The hybrid method utilizes the Steklov-Poincaré approach to efficiently compute the displacement vector field on the boundary. Then, it employs a p-harmonic domain extension by solving a p-Laplace Dirichlet problem to compute the displacement on the internal mesh nodes. The hybrid concept shares ideas with the extension method recently suggested in [OS21].

Within this chapter, emphasis is given to mesh quality aspects and computational efficiency. The presented application refers to power-loss optimization for a 2D S-bent duct flow. The unstructured grid discretization involves locally refined quadrilateral control volumes. The optimized shapes are assessed in terms of convergence of the objective, the obtained shape, and the mesh quality for the final configuration by the measures used in Section 3.4.1.

4.1 Methodology

Figure 4.1 shows a sketch of the geometric configuration and the naming scheme for the internal flow domain used in this chapter. Unlike the minimal drag problem from the previous chapter,

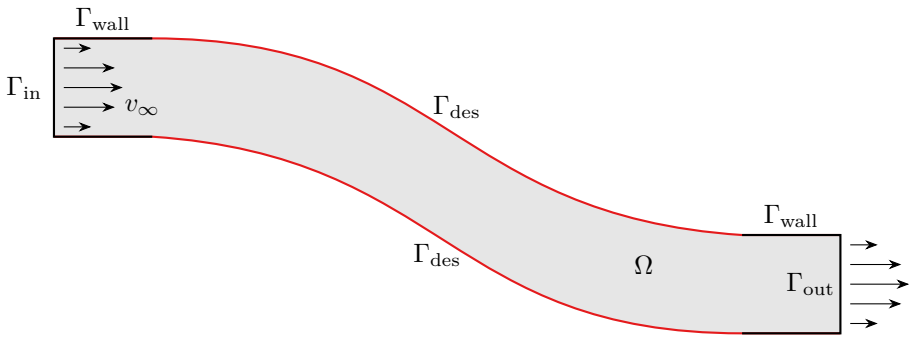


Figure 4.1: Schematic view of the geometrical configuration of the 2D S-bent duct.

the domain Ω is optimized by deforming the outer design boundary $\Gamma_{\text{des}} \subset \Gamma$, the parts of the boundary Γ_{wall}, Γ_{in}, and Γ_{out} remain fixed. Recall the general shape optimization problem (2.15) and the shape derivative (2.5) of a shape functional (2.16) or (2.17).
For a given shape derivative J', the Steklov-Poincaré method, introduced in [SS16], uses a linear elasticity-like problem from structural mechanics of the form (3.1). In the elasticity-like problem, the shape derivative $J'(\Omega)W$ enters as a right-hand side, which can be interpreted as a forcing term. The method is similar to the *Hilbertian extension and regularization* from [Gou06] and [ADJ21, Section 5.2]. Recall from Section 3.1 that for the Hilbert space H, with the inner product $a(\cdot,\cdot) : H \times H \to \mathbb{R}$, and $J'(\Omega)W < 0$, the vector field $V \in H$ is a descent direction solving (3.1). The space H, considered a function space over Ω, incorporates Dirichlet boundaries on the fixed boundary $\Gamma_{\text{in}}, \Gamma_{\text{out}}, \Gamma_{\text{wall}}$ and considers natural boundary conditions on the deformed part Γ_{des}. Depending on the choice of $a(\cdot,\cdot)$, the Hilbert space approach leads to regular vector fields, but (3.1) may not give "good" deformations of the internal mesh of the domain; that is, V does not sufficiently preserve mesh quality. The latter is only a technical issue, but it significantly influences the implementation and algorithmic realization of the approach. Instead of using V, obtained from (3.1), to simultaneously deform the surface and internal mesh in Ω, the suggested hybrid method only uses the boundary values of V. A domain extension $\tilde{V}$ is computed by solving a p-Laplace problem where $V\big|_\Gamma$ enters as Dirichlet boundary data. This second approach builds on the idea proposed in [OS21], where the extension of the boundary deformation is extended via a non-linear convection-diffusion problem into the domain. Thus, the shape deformation is computed in a two-step process. First, the descent direction V is given by the solution of (3.1) using the bi-linear form

$$a(V,W) = \int_\Omega \eta \nabla V : \nabla W \, dx \tag{4.1}$$

with the diffusivity [Küh+22]

$$\eta(x) = \frac{1}{\frac{1}{\eta_{max}} + \min\limits_{\tilde{x}\in\Gamma} \|x - \tilde{x}\|_2}. \tag{4.2}$$

Second, the movement of the discrete nodes within the domain is given by the solution $\tilde{V}$ of the Dirichlet problem

$$\begin{cases} -\operatorname{div}\left((\nabla\tilde{V} : \nabla\tilde{V})^{\frac{p-2}{2}} \nabla\tilde{V}\right) = 0 & \text{in } \Omega, \quad p > 2 \\ \tilde{V} = V & \text{on } \Gamma \end{cases} \tag{4.3}$$

in a weak sense. Because the solution of (3.1) enters as Dirichlet data, the shape itself still is deformed by $V \in H$, and only the internal nodes of the mesh are affected. Note that the fact that the Hilbert space $H(\Omega)$ is not a subspace of $W^{1,\infty}(\Omega, \mathbb{R}^d)$ when considering the inner product (4.1) is neglected here. However, the obtained solutions give regular deformations and are applicable for practical use. In [SS16], this method is associated with applying the Dirichlet-to-Neumann map or Steklov-Poincaré operator for the case in which $J'(\Omega)V$ has a boundary formulation of the form

$$J'(\Omega)V = \int_{\Gamma_{\text{des}}} \sigma \, (V \cdot n) \, ds \tag{4.4}$$

where $\sigma : \Gamma \to \mathbb{R}$ depends on the state and the adjoint state. Note that the formulation in (3.1) is rather general and does not necessarily require the boundary formulation of the shape derivative. The Application described in the following section uses the procedure outlined in Algorithm 2, where the boundary formulation (4.4) is used to extract the descent direction V. The process to determine the modified shape deformation with p-Laplace extension is summarized in Algorithm 3. A full non-linear approach is to solve the p-Laplace problem, which resides from p-Laplace relax-

Algorithm 3 Steklov-Poincaré with p-Laplace domain extension (SP+p)

Require: p_{max}, p_{inc}, ϵ_1, ϵ_2

1: SOLVE the boundary deformation u according to (3.1)
2: $\tilde{V}_0 \leftarrow V$ ▷ Use the solution from (3.1) as initial guess.
3: $p \leftarrow 2$
4: **repeat**
5: **if** $p < p_{max}$ **then**
6: $\epsilon \leftarrow \epsilon_1$
7: **else**
8: $\epsilon \leftarrow \epsilon_2$ ▷ Where $\epsilon_2 < \epsilon_1$
9: **end if**
10: SOLVE the p-Laplace domain extension problem: Find $\tilde{V} : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\operatorname{div}\left((\nabla \tilde{V} : \nabla \tilde{V})^{\frac{p-2}{2}} \nabla \tilde{V} \right) = 0 & \text{in } \Omega, \\ \tilde{V} = V & \text{on } \Gamma. \end{cases} \tag{4.5}$$

for $j = 1, \ldots, d$ iteratively with initial guess $\tilde{V}_0$ and tolerance ϵ or until maximum number of iterations is reached.
11: $p \leftarrow p + p_{inc}$
12: $\tilde{V}_0 \leftarrow \tilde{V}$ ▷ Set the initial guess for the next p.
13: **until** $p > p_{max}$
14: $V \leftarrow \tilde{V}$ ▷ Set the deformation for the whole domain.

ation of the steepest descent direction in $W^{1,\infty}$-topology [DHH23; Mül+21]. Algorithm 4 schematically illustrates the realization of this approach. A rigorous comparison of the descent direction influence on the shape optimization procedure would require the determination of an optimal step size for each direction. However, the identification of the optimal step size is computationally demanding when the state is defined by the solution of a PDE, and thus this might be unfeasible. Nevertheless, the sequence of successive shape updates k is controlled by applying

$$t_k = \frac{\alpha_k}{\max\limits_{x \in \Omega} \|V^k(x)\|_2}. \tag{4.7}$$

where α_k is chosen such that the *Armijo condition* is fulfilled.

Algorithm 4 p-Laplace relaxed steepest descent direction

Require: p_{max}, p_{inc}, ϵ_1, ϵ_2

$V_0 \leftarrow 0$
$p \leftarrow 2$
repeat
  if $p < p_{max}$ **then**
    $\epsilon \leftarrow \epsilon_1$
  else
    $\epsilon \leftarrow \epsilon_2$ ▷ Where $\epsilon_2 < \epsilon_1$
  end if
  SOLVE the p-Laplace descent direction problem: Find $V : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\operatorname{div}\left((\nabla V : \nabla V)^{\frac{p-2}{2}} \nabla V\right) = 0 & \text{in } \Omega, \\ V = 0 & \text{on } \Gamma \setminus \Gamma_{\text{des}}, \\ (\nabla V : \nabla V)^{\frac{p-2}{2}} \dfrac{\partial V}{\partial n} = -\sigma n & \text{on } \Gamma_{\text{des}}. \end{cases} \tag{4.6}$$

  for $j = 1, \ldots, d$ iteratively with an iterative method with initial guess V_0 and tolerance ϵ or until the maximum number of iterations is reached.
  $p \leftarrow p + p_{inc}$
  $V_0 \leftarrow V$ ▷ Set the initial guess for the next p.
until $p > p_{max}$

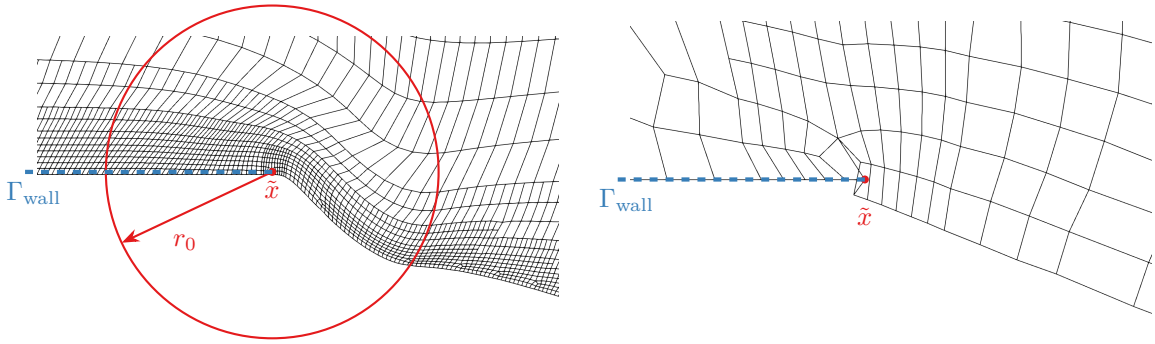
Furthermore, an additional issue that one might face in CAD-free shape optimization relates to the construction of the problem. Not in general but most frequently, shape optimization for internal flow problems, like the one studied in this thesis, the objective is optimizing certain wall sections, namely $\Gamma_{\text{des}} \subset \Gamma_{\text{wall}}$. A common problem that might arise is the sudden change of boundary types leads to distorted computational grids, as shown in Figure 4.2(b). Therefore, the following filter is used to smoothen the transformation field in the vicinity of the join of Γ_{des} and Γ_{bnd}:

$$V_f = \begin{cases} V \frac{1}{2}\left(1 - \cos\left(\pi \frac{r}{r_0}\right)\right) & : \text{if } r \leq r_0, \\ V & : \text{else,} \end{cases}, \quad \text{with} \quad r = \|x - \tilde{x}\|_2, \tag{4.8}$$

where r_0 controls the filtering radius with. For the application studied herein, $\tilde{x} = (\tilde{x}_1, \tilde{x}_2)^\intercal$ corresponds to the position vector of a node connecting Γ_{des} and Γ_{bnd}. Figure 4.2 schematically shows the impact of the filter for the same 2D case. As shown in Figure 4.2(a), the optimizer updated the shape while maintaining its grid quality. In contrast, when the solution u is directly applied, the grid quality is rapidly deteriorated, leading to even intersecting faces as shown in Figure 4.2(b), making the numerical solution of the PDE constraints unfeasible.

4.2 Problem Formulation

Consider the domain $\Omega \subset \mathbb{R}^2$ and the state given by the velocity $v_\Omega : \Omega \to \mathbb{R}^d$ and the density-specific pressure $p_\Omega : \Omega \to \mathbb{R}$, which satisfy the stationary Navier-Stokes equations of an incom-



(a) (b)

Figure 4.2: Two-dimensional optimization case: Detail of the computational unstructured grid of intermediate updated shapes around the connection of Γ_{des} and Γ_{wall} using (a) V_f and (b) V.

pressible, Newtonian fluid, viz.

$$\begin{cases} -\operatorname{div}(v_\Omega) = 0 & \text{in } \Omega, \\ \operatorname{div}\big(v_\Omega \otimes v_\Omega - \nu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right)\big) + \nabla p_\Omega = 0 & \text{in } \Omega, \\ v_\Omega = 0 & \text{on } \Gamma_{\text{des}} \cup \Gamma_{\text{wall}}, \\ v_\Omega = v_{in} & \text{on } \Gamma_{\text{in}}, \\ \nu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right) n = p_\Omega n & \text{on } \Gamma_{\text{out}} \end{cases} \tag{4.9}$$

where $\nu > 0$ is the kinematic viscosity and $\rho > 0$ denotes the density of the fluid. The aim of the present shape optimization procedure is to minimize the power loss within the flow domain, which is described by the objective function

$$\mathcal{J}((v,p),\Omega) = -\int_\Gamma \rho\left(p + \frac{1}{2}\|v\|^2\right)(v\cdot n)\,ds. \tag{4.10}$$

The corresponding Lagrange function is defined by

$$\begin{aligned} L((v,p),V,(w,q)) &:= -\int_\Gamma \rho\left(p + \frac{1}{2}\|v\|^2\right)(v\cdot n)\,ds \\ &+ \int_\Omega \nu\left(\nabla v + \nabla v^\intercal\right) : \nabla w + \operatorname{div}(v\otimes v)\cdot w - p\operatorname{div}(w) - \operatorname{div}(v) q\,dx \end{aligned} \tag{4.11}$$

with the multipliers (w,q). The Lagrange multipliers are identified with the adjoint state, which satisfies the system of adjoint equations. The adjoint equation system on Ω, for the optimization problem in this chapter, reads

$$\begin{cases} -\operatorname{div}(w_\Omega) = 0 & \text{in } \Omega, \\ -\operatorname{div}\big(\nu\left(\nabla w_\Omega + \nabla w_\Omega{}^\intercal\right)\big) - \left(\nabla w_\Omega + \nabla w_\Omega{}^\intercal\right)\cdot v - \nabla q_\Omega = 0 & \text{in } \Omega, \\ w_\Omega = 0 & \text{on } \Gamma_{\text{wall}}, \\ w_\Omega \cdot n = v_\Omega \cdot n & \text{on } \Gamma_{\text{in}}, \\ q_\Omega = \|v_\Omega\cdot n\|\,\|w_\Omega\cdot n\| - \|v_\Omega\cdot n\|^2 - \frac{1}{2}\|v_\Omega\|^2 & \text{on } \Gamma_{\text{out}}. \end{cases} \tag{4.12}$$

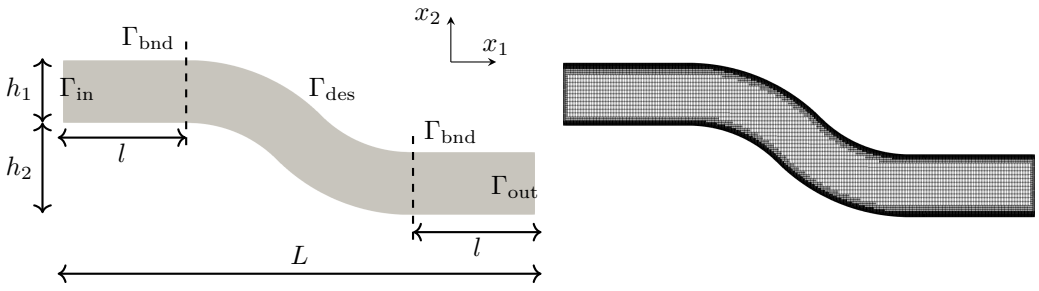
With thte velocity v_Ω and the adjoint velocity w_Ω, the shape derivative of (4.10) has the form of (3.13) with

$$\sigma = -\mu\left(\frac{\partial w_\Omega}{\partial n}\cdot\frac{\partial v_\Omega}{\partial n}\right) \tag{4.13}$$

The computational method is similar to the studies outlined in Chapter 3, the numerical procedure for the solution of the primal (4.9) and adjoint system (4.12) in this study is also based upon the Finite Volume Method (FVM) *FreSCo*$^+$. The implicit numerical approximation is second-order accurate in space and time and supports arbitrary polyhedral cells as well as local grid refinement. The segregated algorithm uses a cell-centered, co-located storage arrangement for all transport properties. The basic discretization concept is given in Appendix B and details of the primal method are outlined in [Run+09]. The pressure-velocity coupling utilizes a pressure-correction scheme, and parallelization is realized by means of a domain decomposition approach [Yak+13; YMR15]. A detailed derivation of the hybrid adjoint approach can be found in [KKR18; Küh+22; SR13].

4.3 Application

The application considered refers to a 2D S-bent duct. Results compare three different concurrent mesh & shape update approaches, i.e., a Steklov-Poincaré (SP) method, a Steklov-Poincaré with a subsequent p-Laplace extension (SP+p), and a p-Laplace approach. Both p-Laplace methods refer to a value of $p_{max} = 4.1$. This value is slightly above the threshold value $p = 4$ that is deemed to by large enough as described in Section 4.1. The SP approach essentially follows from (3.1) with the inner product (4.1) and the SP+p method is described by Algorithm 3. The respective regularization parameter value for the diffusion coefficient refers to $\eta_{max}^{-1} = 10^{-3}$ and agrees for both mesh update approaches (SP, SP+p). The p-Laplace mesh & shape update approach corresponds to the original p-Laplace method outlined in Algorithm 4. Emphasis is put on assessing the evolution of the objective function, the final shape, the computational cost, and the preservation of the grid orthogonality and the aspect ratio.
The application considers the optimization of a 2D S-bent duct. A sketch of the initial shape and the employed unstructured grid is shown in Figure 4.3. The total length of the computational domain is L while $h_1 + h_2$ is its height. The wall sections from the in- and outlet are of length l, where the shape remains in its initial configuration. The following non-dimensional geometric ratios hold: $l/h_1 = 2$, $h_2/h_1 = 1.5$, $L/h_1 = 7.5$. The unstructured-grid discretization employs



(a) (b)

Figure 4.3: Illustration of the S-bent 2D duct initial shape with geometric/boundary annotations (a) and employed unstructured, locally refined grid (b). The lower wall of the duct is split in Γ_{bnd} and Γ_{des} in accordance with the split of the upper wall.

approximately 31,600 quadrilateral control volumes. As indicated by Figure 4.3 (b), the grid is progressively refined towards the wall boundaries. It features almost orthogonal control volumes with an approximately unity cell aspect ratio in the boundary-adjacent cell layer. The filtering approach described in Section 4.1 is applied around the four points, connecting Γ_{bnd} and Γ_{des}

boundaries, with a normalized filtering radius of $r_0/h_1 = 0.2$. The inlet velocity has a parabolic profile

$$v_{1,\text{in}}(x_2) = 2\, v_{ref} \left(1 - \left(\frac{2x_2}{h_1}\right)^2\right), \quad v_{2,\text{in}}(x_2) = 0, \tag{4.14}$$

where the coordinate origin aligns with the midpoint of Γ_{in}. The laminar flow is characterized by a Reynolds number of $\text{Re} = (v_{ref}\, h_1)/\nu = 500$.

Shape and Objective Functional Evolution

Figure 4.4 displays the computed reduction of the normalized power-loss objective functional by about 18 % from its initial value. Figure 4.5 depicts the corresponding evolution of the shape difference from the initial shape. It is observed that all approaches converge remarkably fast toward the final shape. An increase of the regularization parameter η_{max} in (4.2) to a unit value of reveals a substantial change of deformation and thus the final shape, cf. Figure 4.4. Figure 4.6 displays

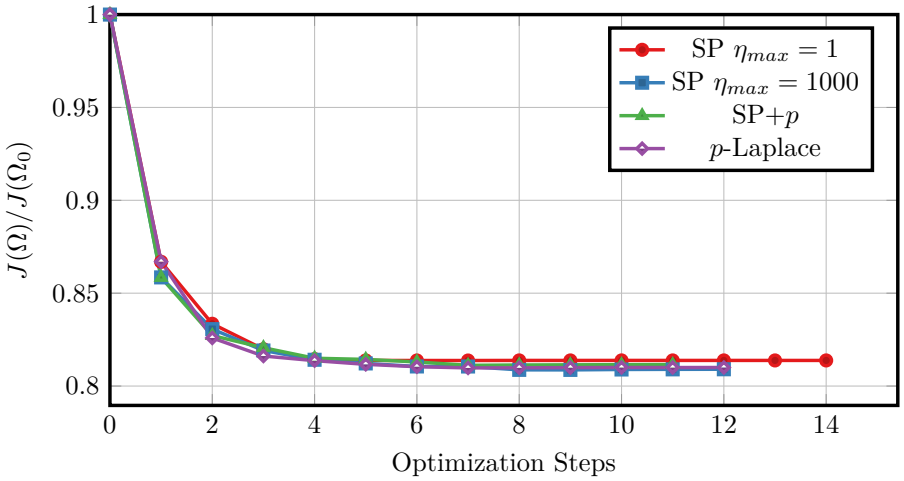


Figure 4.4: Evolution of the normalized objective functional, relative to the initial value, compared using proposed descent directions.

the final shapes predicted by the four optimizations. Minor differences between the respective shapes returned by the p-Laplace method and the SP+p, SP ($\eta_{max} = 1000$) methods are observed. Except for the lower left transition region, the SP shape obtained by modifying the regularization parameter η_{max} to a unit value is globally smoother and deviates less from the initial shape. The latter also causes a more rapid convergence of the optimization.
Figure 4.7 displays the distribution of the objective function along the inlet and outlet for the initial and optimized shape (left) and the respective drop of the absolute values from initial to optimized (right). The computed values follow from $j_\Gamma(\Gamma_{\text{in}}) = j_\Gamma(0, x_2)$ and $j_\Gamma(\Gamma_{\text{out}}) = j_\Gamma(L, x_2 - h_2)$ where $j_\Gamma(\Gamma)$ denotes the integrand of (4.10). Recall that $(x_1, x_2) = (0, 0)$ lies at the midpoint of Γ_{in}. Along the inlet, where the velocity values are fixed, a parabolic drop of the absolute objective value is observed, which is due to the homogeneous decrease of the pressure value, as illustrated in Figure 4.6. Along the outlet, the pressure is fixed, and the velocities of the optimized shape are more homogeneous as compared to the initial shape. Smaller core flow velocities are confirmed for the optimal shape in Figure 4.7 and yield a reduction of the power loss, cf. Figure 4.7.
Table 4.1 provides information on the computational efforts. It is seen that the severe non-linearity inherent to both p-Laplace approaches significantly increases the computational cost. Compared to the SP approaches, the SP+p method increases the effort approximately by a factor of 2.5,

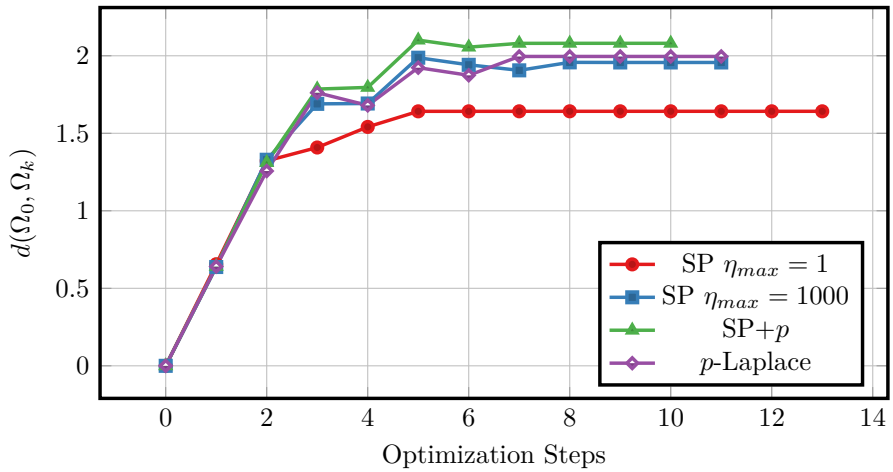


Figure 4.5: The Evolution of the symmetric shape difference defined in (3.20), using the different descent directions.

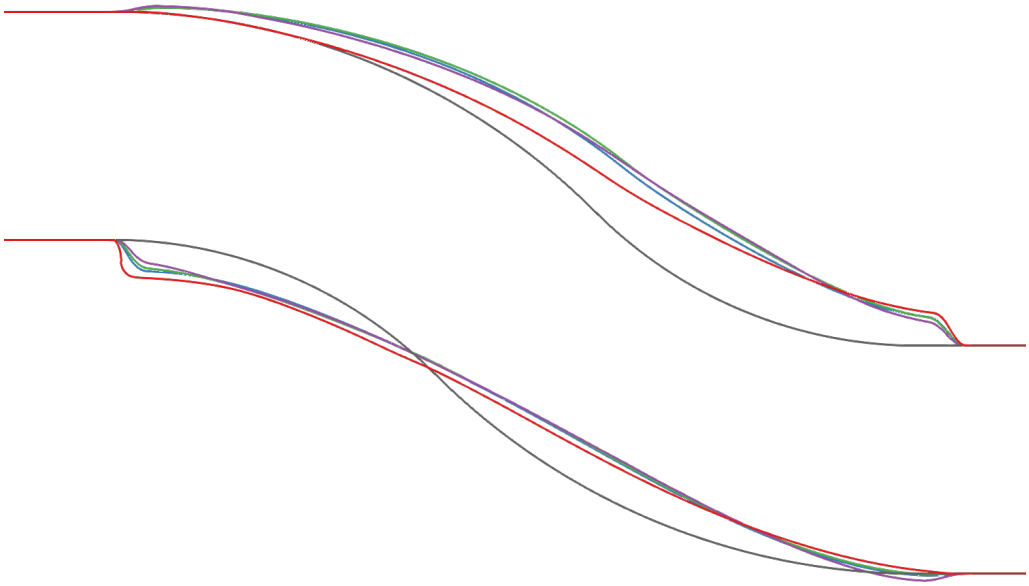


Figure 4.6: Contours of obtained the shapes for the 2D S-bent optimization compared with the Initial shape ___ Steklov-Poincaré ___ with $\eta_{max} = 1$ and ___ with $\eta_{max} = 1000$, the Steklov-Poincaré with p-Laplace extension ___ and the p-Laplace ___ methods.

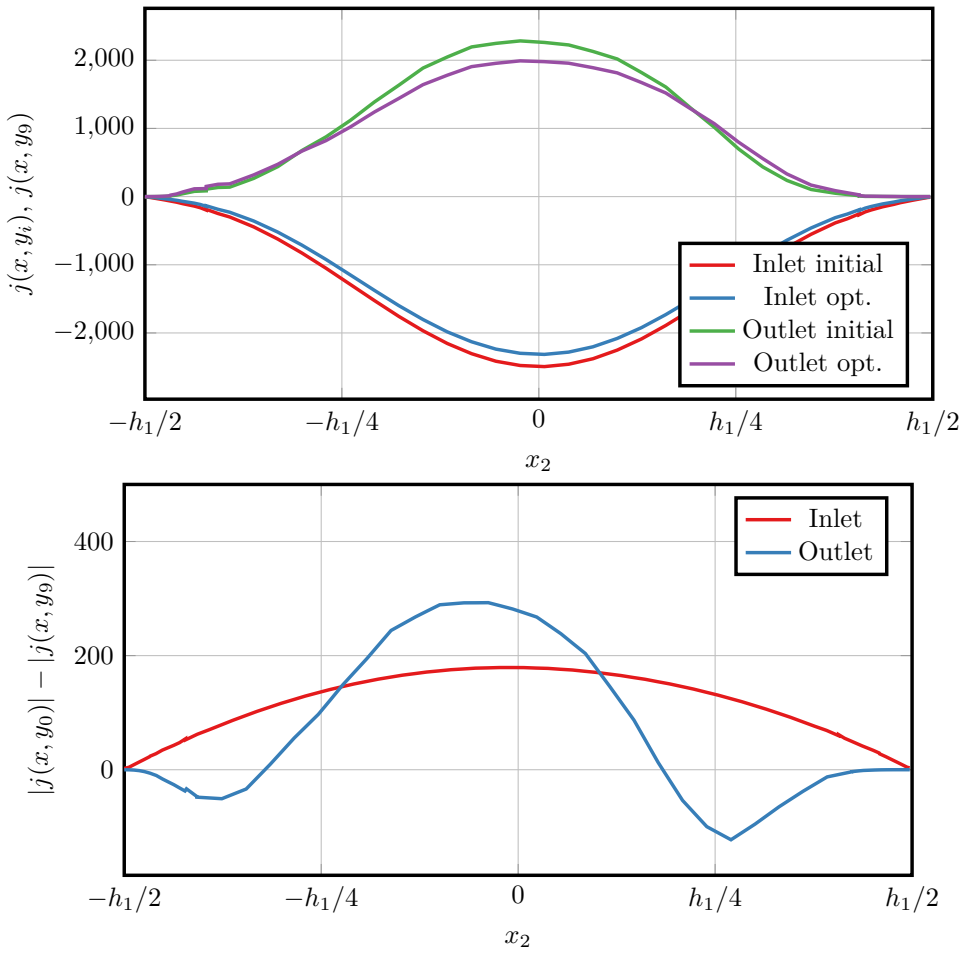


Figure 4.7: Upper figure: The integrant $j(\cdot, (v, p))$ of the objective function (4.10) along the inlet (continuous lines) and outlet (dashed lines) boundaries for the initial and the optimized shape of the 2D S-bent, using the p-Laplace relaxed approach. Here, $-h_1/2$ refers to the bottom corner of each boundary. Bottom figure: Distribution of the difference of the absolute objective function from initial (j_Γ^{init}) to optimized (j_Γ^{opt}) shape along the inlet and outlet.

$2D$ S-bend	Steps	normalized time per step	normalized convergence effort
SP ($\eta_{max} = 1000$)	11	1	1
SP ($\eta_{max} = 1$)	5	2.4	1.1
SP+p ($\eta_{max} = 1000$, $p = 4.1$)	10	2.8	2.6
p-Laplace ($p = 4.1$)	9	4.3	3.6

Table 4.1: The total number of optimization steps, normalized average wall-clock time per design step required to compute the descent direction, and normalized convergence effort. All normalizations with results of the SP approach and all p-Laplace approaches employed $p = 4.1$.

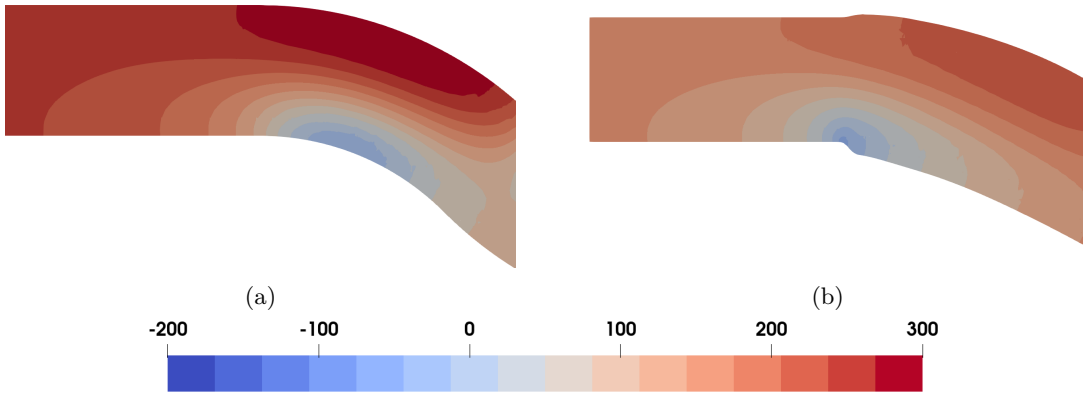


Figure 4.8: Pressure contours in [Pa] near the inlet for the initial (left) and optimized (right; p-Laplace) shapes of the 2D S.bent. The exit pressure is assigned to zero in both cases.

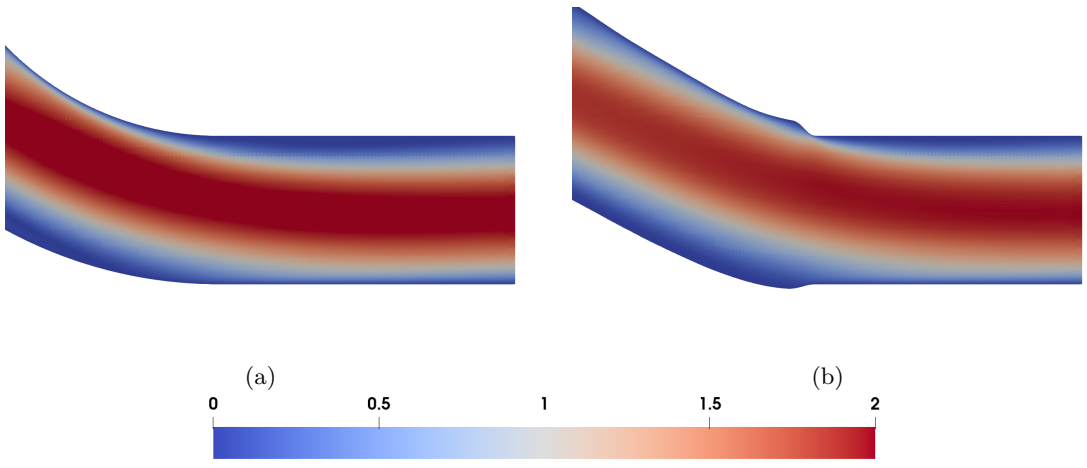


Figure 4.9: Contours of the velocity magnitude in [m/s] near the outlet for the initial (left) and optimized (p-Laplace) (right) shapes of the 2D S-bent.

whereas the p-Laplace method is afflicted with roughly 3.5 times the costs of the SP approach. Since the optimization convergence varies, these average costs per design step were normalized by the convergence ratio. These costs will usually be compared to the flow simulation efforts, which are rapidly increasing for more complex flows and can substantially benefit from an improved mesh quality. Moreover, the credibility of the optimization result is an issue that is usually assessed by comparing the objective functional for the optimized geometry obtained in conjunction with the morphed and a new mesh. In this regard, an improved mesh quality might speed up the convergence and the feasibility of a morphing-based optimization.

Mesh Quality

With attention directed to the mesh quality, Figure 4.6 also indicates that all methods display the most severe deformation where the design wall meets the non-design wall, which might also challenge the attainable mesh quality in this region. Figures 4.10 and 4.11 and Table 4.2 reveal that the SP+p method can lift the cell minimum angle in comparison to the SP method ($\eta_{max} = 1000$) while also reducing the number of inferior cells by one order of magnitude, and thereby improve the mesh quality. However, the p-Laplace method displays even more localized changes and smaller deviations from the ideal grid arrangement, cf. Figure 4.12. The critical lowest value refers to approximately 35°, and the total amount of critical cells is reduced by approximately one[two] order[s] of magnitude in comparison to the SP+p[SP] method, cf. Figure 4.11. Interestingly, a reduction of the regularization parameter $\eta_{max} = 1000$ to an exemplary alternative value of $\eta_{max} = 1$ results in a considerable deterioration of the cell shapes, as indicated by the critical cells denoted in Table 4.2 and the histogram in Figure 4.11. The latter is also seen by an observation of the minimum orthogonality depicted in Figure 4.11, where a continuous deterioration of cell quality is observed with increasing deformation when the regularization parameter is reduced. Moreover, the benefits of the p-Laplace and the hybrid SP+p approach are outlined.

Method/Initial	Minimum angle in degrees
Initial	50
SP ($\eta_{max} = 1$)	17
SP ($\eta_{max} = 1000$)	27
SP+p ($\eta_{max} = 1000$, $p = 4.1$)	30
p-Laplace ($p = 4.1$)	35

Table 4.2: Most critical grid-orthogonality values were observed for a morphed mesh of the final design of the $2D$ S-bent. Values indicate a minimum of $90° - \beta$, where β refers to the angle between a face normal and the connecting line between the adjacent cell centers.

Figure 4.13 displays the cell aspect ratio near the transition region for the final shapes of all three methods. The deterioration of the aspect ratio is moderate and most pronounced by the SP method. Again, a p-Laplace extension helps to preserve the mesh quality, as depicted by the SP+p results in the center. Aspect ratios obtained from the p-Laplace method are in fair qualitative agreement with the SP+p grids, though reported p-Laplace values are again superior.

4.4 Concluding Remarks on Shape Transformation Approaches

This chapter investigated approaches that simultaneously compute the shape and mesh update in the context of parameter-free, adjoint-assisted shape optimization. In addition to the Steklov-

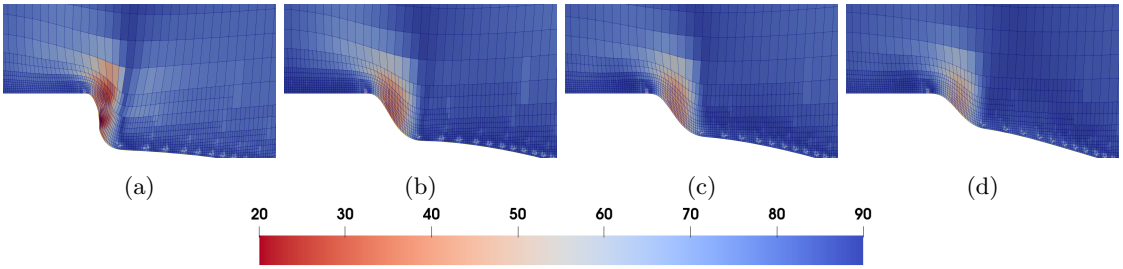


Figure 4.10: Grid orthogonality of the final design in the lower downstream transition region of the 2D S-bent. The displayed property is $90° - \beta$, where β refers to the angle between a face normal and the connecting line between the adjacent cell centers, and higher (blue) is better. The results are obtained with SP Figure 4.10a with $\eta_{max} = 1$ and Figure 4.10b with $\eta_{max} = 1000$, SP+p Figure 4.10c and p-Laplace Figure 4.10d approach.

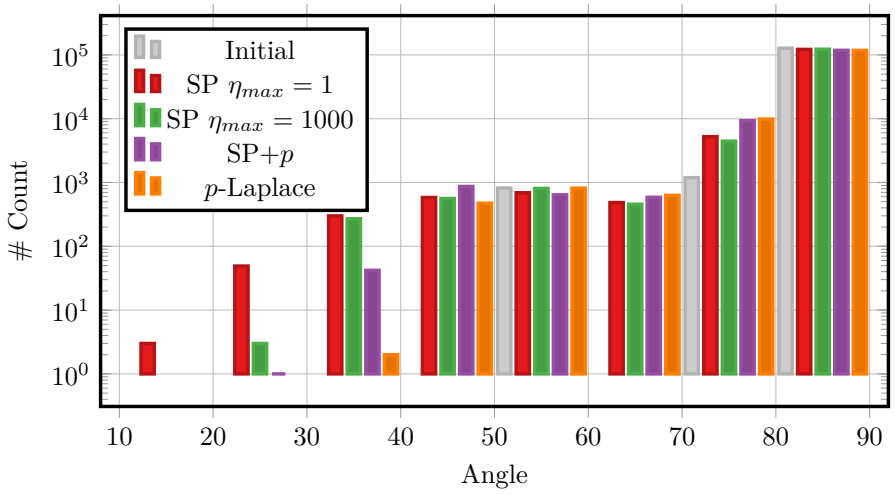


Figure 4.11: Grid-orthogonality observed for the morphed mesh of the final design of the 2D S-bent. Distribution of $90° - \beta$, where β refers to the angle between a face normal and the connecting line between the adjacent cell centers.

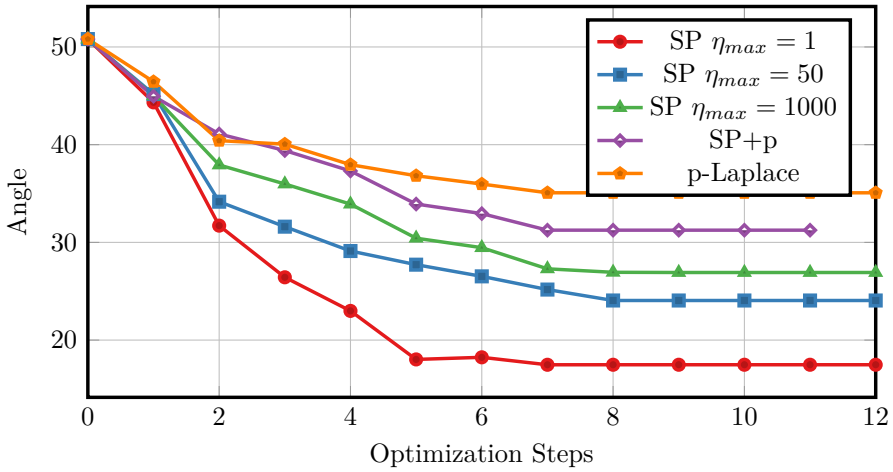


Figure 4.12: Evolution of the minimum orthogonality for the initial and deformed meshes during the optimization.

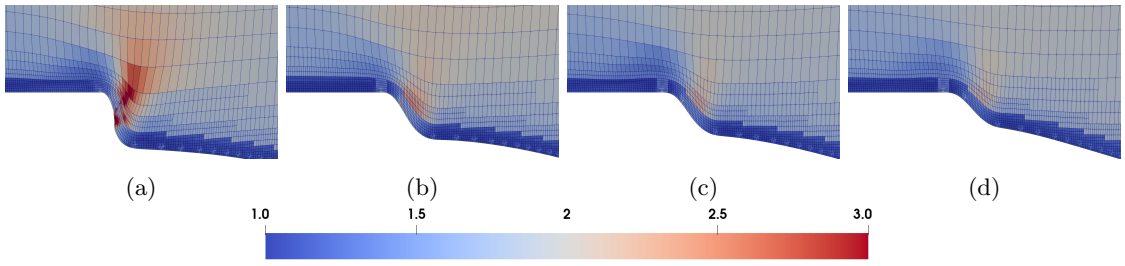


Figure 4.13: Cell aspect ratio at the lower channel for the final shapes each obtained with the SP Figure 4.13a with $\eta_{max} = 1$ and Figure 4.13b with $\eta_{max} = 1000$, the SP+p Figure 4.13c and with the p-Laplace method Figure 4.13d for the optimized shapes of the 2D bent in the lower downstream transition region (range from 1 to 3 where lower is better).

Poincaré (SP) and p-Laplace methods, a hybrid approach is proposed, in which the shape update follows from the former while the extension to the domain is realized based on the latter. The motivation of the hybrid method (SP+p) was to extract the positive properties of each, being the lower computational cost of the SP method and the mesh quality benefits of the p-Laplace approach.
All three approaches were scrutinized for the power loss optimization of a $2D$ laminar ducted flow, where they all managed to locate an optimal solution after 10 shape updates. However, as regards the necessary computational effort and mesh quality metrics, the observations verified the initial hypothesis. The SP+p method performed better than a pure SP approach in terms of the quality of the optimized mesh while also requiring approximately half the computational effort of the p-Laplace approach.
Although the hybrid formulation could reduce the costs, they are still considerable. This is particularly true if geometrical constraints occur. Geometric constraints can significantly increase the time to solution. The approach outlined in Algorithm 1, in Chapter 3, for example, repetitively solves the shape optimization problem to approximate the Lagrange multipliers corresponding to the barycenter and volume constraints. To overcome this issue, Chapter 5 investigates an efficient way of handling constraints for admissible geometries using the p-Laplace method with a related implementation on a massively parallel high-performance computer. Furthermore, Chapter 6 solves the highly non-linear minimization problem (3.3) to directly determine the direction of steepest descent in $W^{1,\infty}$ instead of an approximation by $V_p \in W^{1,p}$ with $p > 2$.

Chapter 5

A Scalable Algorithm with Geometric Constraints in Banach Spaces

For large-scale problems with time-dependent flow simulations, the solution of the computational effort for solving the p-Laplace problem (3.3) is minor compared to the solution of the Navier-Stokes equations. In connection with geometric constraints, however, the effort increases drastically so that conceptual changes appear necessary.

In [MP04], geometrical constraints are taken into account via penalization of the objective function. Even though this approach allows for more general shape deformations, the step size for updating the shape must be small to keep the procedure numerically stable, which deteriorates the convergence. Besides the pure penalty method, an augmented Lagrange method can be used to improve the procedure and reduce the number of shape optimization steps. Algorithm 1 implements an augmented Lagrange approach to approximate the multipliers associated with constraints that are of very different orders of magnitude [AJT04; SS16; ADJ21]. However, this method has some disadvantages. In the algorithm, the whole shape optimization problem is inside an augmented Lagrange loop and thus must be solved several times to achieve a sufficient approximation of the multipliers. However, a good choice for the initial values of several problem-dependent parameters can significantly influence the convergence of the method but may require expert knowledge and experience of the user. Furthermore, the constraints first have to be violated in order to determine the desired multipliers and the whole shape optimization problem has to be solved repetitively until the multipliers converge. This can lead to unfeasible shapes throughout the optimization procedure, as was previously mentioned. The central idea of the scheme suggested in this chapter, which was recently published in a joint paper [Mül+23], is that the geometric constraints are introduced while computing the descent vector field V and thereby do not -as usual- depend on the physical state variables, cf. Section 5.1.2. In conclusion, the optimality system can be solved sequentially starting with the state, then the adjoints to the state, and finally the descent direction, cf. Algorithm 5. This provides the ability to handle problems with very large degrees of freedom while fulfilling the geometric constraints.

Results will again be compiled for deeply immersed 2D and 3D shapes at low Reynolds number ($Re = 20$), that were already investigated in Chapter 3. Though the optimal shapes are similar to the one investigated in Chapter 3, the initial geometries are different, and do not refer to circular/spherical shapes but sharped-edged 2D & 3D boxes.

5.1 Model Equations

Recall the geometric setting sketched in Figure 3.1. In this chapter, the situation is similar to Chapter 3. However, the boundaries on the top and bottom of the flow domain Ω are fixed walls, i.e. Γ_{wall} instead of Γ_{slip}, in this chapter. In addition, the initial shape of the obstacle used in the numerical experiments in Section 5.4 is a different one. This chapter proposes an optimization methodology for PDE constraint shape optimization problems, which is of the form of (2.15). In addition to the PDE constraint $e(y_\Omega, \Omega) = 0$, the geometry must fulfill a finite number of geometric constraints. For some $m \in \mathbb{N}$, the geometric constraint is given as $g : \mathcal{S} \to \mathbb{R}^m$. Therewith, the abstract optimization problem reads

$$\min_{\Omega \in \mathcal{S}} \mathcal{J}(y, \Omega) \quad \text{s.t.} \quad y \in Y, \quad e(y, \Omega) = 0, \quad \text{and} \quad g(\Omega) = 0. \tag{5.1}$$

With the transformation (2.2), and the parameterization V, the above optimization problem can be interpreted as a problem in some function space $W \subset W^{1,\infty}(\Omega, \mathbb{R}^d)$. In this chapter, W is the admissible set of displacements V defining the transformation $F = \mathrm{id} + V$, such that the geometric constraints are inherently fulfilled, that is, if $g(\Omega) = 0$, then also $F(\Omega) = \Omega_V \mapsto g(\Omega_V) = 0$ holds. The crucial aspect of the present method is to separate the geometric constraints (5.1) from the remaining PDE-constrained shape optimization problem (2.15), and move it to the admissible set of descent directions W_{ad}. In contrast to other popular approaches, where admissibility is only guaranteed in the optimal configuration, the latter ensures that (5.1) is fulfilled in each optimization step. This is possible because, here, the geometric constraints do not depend on the state y and rely on the properties of the shape only.

The derivative of the reduced functional J then can be expressed with the help of the adjoint state, see [Bel+97]. The problem in this chapter is minimizing the dissipation of energy of the fluid, which is caused mainly by an obstacle Ω_{obs} located in a laminar flow. Therefore, the objective functional

$$\mathcal{J}(v, \Omega) = \frac{\nu}{2} \int_\Omega \nabla v : \nabla v \, dx \tag{5.2}$$

is to be minimized. In the following, v denotes the velocity, p is the density-specific, ν is the kinematic viscosity, and v_∞ is a prescribed inflow velocity. The PDE constraint $e(y_\Omega, \Omega) = 0$, with state variable $y_\Omega = (v_\Omega, p_\Omega)$, is given by the stationary, incompressible Navier-Stokes equations:

$$\begin{cases} -\mathrm{div} v_\Omega = 0 & \text{in } \Omega \\ -\nu \Delta v_\Omega + (v_\Omega \cdot \nabla) v_\Omega + \nabla p_\Omega = 0 & \text{in } \Omega \\ v_\Omega = 0 & \text{on } \Gamma_{\text{obs}} \cup \Gamma_{\text{wall}} \\ v_\Omega = v_\infty & \text{on } \Gamma_{\text{in}} \\ \nu \nabla v_\Omega \cdot n = p_\Omega n & \text{on } \Gamma_{\text{out}}, \end{cases} \tag{5.3}$$

in a weak sense. The weak formulation can be found, e.g., in [ESW14, Chapter 8.2], and [Bra+09]. For details on the adjoint Navier-Stokes equations, see e.g. [Hin+09; Ulb03; OS21; PS23], and for details on the shape derivative $J'(\Omega)V$ of the objective function in (5.2), see e.g. [MP09; OS21].

5.1.1 Descent Direction

Following [ADJ21, Proposition 4.1], the aim is to approximate the direction of the steepest descent V, which is a function of the space

$$W := \left\{ V \in W^{1,\infty}(\Omega, \mathbb{R}^d) : |DV| \leq 1, \text{ and } V = 0 \text{ a.e. on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \right\}, \tag{5.4}$$

with the corresponding minimization problem

$$\min_{V \in W} \quad J'(\Omega)\, V \quad \text{s.t.} \quad g(F(\Omega)) = 0, \quad F = \mathrm{id} + V. \tag{5.5}$$

Note that the above problem does not necessarily have a unique solution. (5.5) is relaxed using a p-Laplace relaxation with $p > 2$ inspired by [IL05; DHH22]. Hence, let

$$W^p := \left\{ V \in W^{1,p}(\Omega, \mathbb{R}^d) : V = 0 \,\text{a.e. on}\, \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \right\} \tag{5.6}$$

and consider

$$\min_{V \in W^p} \quad \frac{1}{p} \int_\Omega (\nabla V : \nabla V)^{p/2}\, dx + J'(\Omega)V \quad \text{s.t.} \quad g(F(\Omega)) = 0, \quad F = \mathrm{id} + V \tag{5.7}$$

where it is assumed that $g : W^p \to \mathbb{R}^m$, $V \mapsto g(F(\Omega))$, $m \geq 1$. Thus, the admissible set $\mathcal{S}$ is locally parameterized by W^p-deformations of Ω. In this chapter, $m = d + 1$ and g refers to the non-linear barycenter and volume constraints

$$\int_\Omega F \, \det(DF)\, dx = 0, \quad \text{and} \tag{5.8}$$

$$\int_\Omega \Big[\det(DF) - 1 \Big]\, dx = 0. \tag{5.9}$$

Without loss of generality, it is assumed that the barycenter of the initial domain Ω is located at the origin $0 \in \mathbb{R}^d$ of the domain, cf. [PS23], and $\det(DF) > 0$ for V small enough [SZ92]. Different to (3.7), the formulation here omits the division by the reference volume (5.8) because due to (5.9) the volume of Ω is constant.

5.1.2 Optimality System

The corresponding Lagrangian to (5.7) is defined as

$$\begin{aligned} L(V, \lambda) :=& \frac{1}{p} \int_\Omega (\nabla V : \nabla V)^{p/2}\, dx + J'(\Omega)V \\ &+ \sum_{i=1}^{d} \lambda_i \int_\Omega (x_i + V_i(x)) \det(DF)\, dx + \lambda_{d+1} \int_\Omega \Big[\det(DF) - 1 \Big]\, dx \end{aligned} \tag{5.10}$$

with $\lambda = (\lambda_1, \ldots, \lambda_d, \lambda_{d+1})^T$, where $\lambda_1, \ldots, \lambda_d$ are associated with the barycenter (5.8) and λ_{d+1} with the volume constraint (5.9). To recall some rules of differentiation, let $\delta_V, \mu_V : \Omega \to \mathbb{R}^d$ be generic differentiable vector fields, $B : \Omega \to \mathbb{R}^{d \times d}$ and

$$DF = \left(\frac{\partial}{\partial x_j} F_i \right)_{1 \leq i,j \leq d} = I + DV \tag{5.11}$$

the Jacobian of F. The following useful formulae are specified by applying the product and chain rule:

$$\begin{aligned}
\frac{\partial}{\partial V} DF\ \delta_V &= D\delta_V, \\
\frac{\partial}{\partial V} \det(DF)\ \delta_V &= \operatorname{tr}(DF^{-1} D\delta_V) \det(DF), \\
\frac{\partial}{\partial V} \left(\operatorname{tr}(DF\, B)\right)\ \delta_V &= B^T : \left(\frac{d}{du} DF\, \delta_V\right) = B^T : D\delta_V, \\
\frac{\partial}{\partial V} \left(DF^{-1}\right)\ \delta_V &= -DF^{-1} D\delta_V DF^{-1}, \\
\frac{\partial}{\partial V} \left(\operatorname{tr}(DF^{-1} D\delta_V)\right)\ \mu_V &= -D\delta_V{}^T : DF^{-1} D\mu_V DF^{-1} \\
&= \operatorname{tr}(-D\delta_V DF^{-1} D\mu_V DF^{-1}) \\
&= \operatorname{tr}(-DF^{-1} D\mu_V DF^{-1} D\delta_V).
\end{aligned} \tag{5.12}$$

The derivatives of the Lagrangian (5.10) with respect to V in the direction $\mu_V \in W^p$ is obtained using the rules above and that $\nabla(\cdot) = D(\cdot)^\intercal$:

$$\begin{aligned}
\frac{\partial}{\partial V} L(V, \lambda)\mu_V = &\int_\Omega (\nabla V : \nabla V)^{\frac{p-2}{2}} (\nabla V : \nabla \mu_V)\ dx + J'(\Omega)\mu_V \\
&+ (\lambda_1, \ldots, \lambda_d)^T \cdot \int_\Omega \left[\mu_V\ \det(DF) + (x + V(x)) \operatorname{tr}(DF^{-1} D\mu_V) \det(DF)\right] dx \\
&+ \lambda_{d+1} \int_\Omega \operatorname{tr}(DF^{-1} D\mu_V) \det(DF)\ dx.
\end{aligned} \tag{5.13}$$

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

$$\begin{aligned}
\frac{\partial}{\partial V} L(V, \lambda)\mu_V = 0 \qquad &\forall\ \mu_V \in W^p \\
\frac{\partial}{\partial \lambda} L(V, \lambda)\mu_\lambda = 0 \qquad &\forall\ \mu_\lambda \in \mathbb{R}^{d+1}.
\end{aligned} \tag{5.14}$$

As the derivatives with respect to λ can directly be taken from (5.10), the details are omitted here. Solving the non-linear system (5.14) requires the linearization

$$\frac{\partial^2}{\partial V^2} L(V^k, \lambda^k)(\mu_V, \delta_V) + \frac{\partial}{\partial \lambda\, \partial V} L(V^k, \lambda^k)(\mu_V, \delta_\lambda) = -\frac{\partial}{\partial V} L(V^k, \lambda^k)\, \mu_V \quad \forall \delta_V, \mu_V \in W^p \tag{5.15}$$

$$\frac{\partial}{\partial V\, \partial \lambda} L(V^k, \lambda^k)(\delta_V, \mu_\lambda) = -\frac{\partial}{\partial \lambda} L(V^k, \lambda^k)\, \mu_\lambda \qquad \forall \delta_\lambda, \mu_\lambda \in \mathbb{R}^m \tag{5.16}$$

and the updates

$$V^{k+1} = V^k + \delta_V, \quad \lambda^{k+1} = \lambda^k + \delta_\lambda \tag{5.17}$$

where

$$\begin{aligned}
&\frac{\partial^2}{\partial V^2} L(V,\lambda)(\delta_V,\mu_V) = \\
&\int_\Omega (p-2)(\nabla V:\nabla V)^{\frac{p-4}{2}}(\nabla V:\nabla\delta_V)(\nabla V:\nabla\mu_V) + (\nabla V:\nabla V)^{\frac{p-2}{2}}(\nabla\delta_V:\nabla\mu_V)\,dx \\
&+(\lambda_1,\dots,\lambda_d)^T\cdot\int_\Omega \Big(\delta_V \operatorname{tr}(DF^{-1}D\mu_V) + \mu_V \operatorname{tr}(DF^{-1}D\delta_V) \\
&+(x+V(x))\left(\operatorname{tr}(-DF^{-1}D\mu_V DF^{-1}D\delta_V)\right. \\
&\left.+\operatorname{tr}(DF^{-1}D\delta_V)\operatorname{tr}(DF^{-1}D\mu_V)\right)\Big)\det(DF)\,dx \\
&+\lambda_{d+1}\int_\Omega \Big(\operatorname{tr}(-DF^{-1}D\mu_V DF^{-1}D\delta_V) \\
&+\operatorname{tr}(DF^{-1}D\delta_V)\operatorname{tr}(DF^{-1}D\mu_V)\Big)\det(DF)\,dx.
\end{aligned} \tag{5.18}$$

Reviewing the first integral in (5.18), one observes that these terms do not exist for $p \leq 4$, where $\nabla V : \nabla V = 0$ holds on a set of non-zero measures. However, this issue does not appear in the defect equation (5.13), since all exponents are non-negative. Thus, the first integral in (5.18) is modified:

$$\begin{aligned}
\int_\Omega \Big[&(p-2)(\nabla V:\nabla V+\epsilon\Theta(4-p))^{\frac{p-4}{2}}(\nabla\mu_V:\nabla V)(\nabla\delta_V:\nabla V) \\
&+(\nabla V:\nabla V+\epsilon)^{\frac{p-2}{2}}(\nabla\mu_V:\nabla\delta_V)\Big]\,dx,
\end{aligned} \tag{5.19}$$

where Θ denotes the Heaviside function and $\epsilon > 0$ a sufficiently small constant. Notice that, within Newton's method in (5.15) and (5.16), this modification only affects the linearization and not the defect. Thus, solutions of the original problem (5.7) are still obtained upon convergence. Adding ϵ in (5.19) serves two purposes. On the one hand, guarantees invertibility, and on the other, it prevents divide-by-zero operations in the first term.

5.2 Optimization Algorithm

This section describes the algorithm for approximating the solution of (5.7). 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 on 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 the initial values of the multipliers λ, the penalty factors, and the penalty increment values. Thus, the practical appeal of the approach outlined here is that there are fewer heuristic and problem-dependent quantities to be adjusted. The user only has to provide the convergence criteria, the parameters of the step size control, and the values corresponding to the sequence of p, i.e. $p_{\max}$ and p_{inc}.
From a mathematical point of view, the computational price one has to pay is the following: The set of admissible descent directions is not convex anymore, but the solution manifold of the nonlinear equation $g(F(\Omega)) = 0$. For example, having computed an admissible step V_p does not

imply that $\frac{1}{2}V_p$ is also permitted. Thus, a step size control is expensive because the geodesics on the solution manifold are not straight lines due to the nonlinearity of the problem. In Algorithm 5 the step size control is thus handled by scaling the shape sensitivity $\alpha J'(\Omega)V$ with a decreasing sequence $\alpha = 1, \frac{1}{2}, \frac{1}{4}, \dots$. The latter is possible because multiplying the objective function in (5.5) with $\alpha > 0$ does not change the descent direction but the maximum displacement, it is sufficient to search for a minimizing function for $\alpha J'(\Omega)V$.
As a note, for numerical reasons, it might prove profitable to multiply equation (5.14) with $1/\alpha$.

Algorithm 5 Shape Optimization Relaxed Steepest Descent Method

Require: Ω, $p_{\max}$
$y \leftarrow$ Solve primal problem
$y_0 \leftarrow y$
Compute objective $\Phi_0 = J(\Omega)$
repeat
  $z \leftarrow$ Solve adj. problem
  $\alpha \leftarrow 1$
  while True **do**
    $p \leftarrow 2$
    $\bar{V} \leftarrow 0$
    while $p \leq p_{\max}$ **do**
      $(V_p, \lambda) \leftarrow$ NewtonSolver$(\bar{V}, \alpha, y, z)$
      $\bar{V} \leftarrow V_p$
      Increase p
    end while
    Update geometry Ω with $V_{p_{\max}}$
    $y \leftarrow$ Solve primal problem
    Compute objective $\Phi = J(\Omega)$
    if $\Phi \geq \Phi_0$ **then**
      Update geometry Ω with $-V_{p_{\max}}$
      $\alpha \leftarrow \alpha/2$
      $y \leftarrow y_0$
    else
      $\Phi_0 \leftarrow \Phi$
      $y_0 \leftarrow y$
      break
    end if
  end while
until $\|V_{p_{\max}}\|_{W^{1,p}(\Omega)} < \epsilon_1$

For better readability, the linearized optimality system (5.15) and (5.16) is abbreviated using the symbols

$$\begin{aligned}
A\delta_V &:= \frac{\partial^2}{\partial V^2} L(V^k, \lambda^k)(\mu_V, \delta_V) && \forall\, \mu_V \in W_h^p \\
B\delta_\lambda &:= \frac{\partial}{\partial V\, d\lambda} L(V^k, \lambda^k)(\mu_V, \delta_\lambda) && \forall\, \mu_V \in W_h^p \\
B^T\delta_V &:= \frac{\partial}{\partial \lambda\, \partial V} L(V^k, \lambda^k)(\delta_V, \mu_\lambda) && \forall\, \mu_\lambda \in \mathbb{R}^d \\
r_V &:= -\frac{\partial}{\partial V} L(V^k, \lambda^k)\, \mu_V && \forall\, \mu_V \in W_h^p \\
r_\lambda &:= -\frac{\partial}{\partial \lambda} L(V^k, \lambda^k)\, \mu_\lambda && \forall\, \mu_\lambda \in \mathbb{R}^d.
\end{aligned} \tag{5.20}$$

Algorithm 6 Newton's Method for p-Laplacian Problem

1: **function** NEWTONSOLVER(V_p, α, y, z)
2: $\lambda \leftarrow 0$
3: **repeat**
4: (A,B,r_V,r_λ) $\leftarrow$ Assemble($V_p, \lambda, y, z, \alpha$) according to (5.20)
5: $(\delta_{V_p}, \delta_\lambda) \leftarrow$ SCHURSOLVER($A, B, r_V, r_\lambda, \delta_{V_p}, \delta_\lambda$)
6: $V_p \leftarrow V_p + \delta_{V_p}$
7: $\lambda \leftarrow \lambda + \delta_\lambda$
8: **until** $\|\delta_{V_p}\|_{W^{1,p}(\Omega)} + \|\delta_\lambda\|_2 < \epsilon_2$
9: **return** (V_p, λ)
10: **end function**

Algorithm 7 Schur Complement Product

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

With W_h^p a discrete approximation of W^p used for a finite element discretization of (5.15) and (5.16), which then leads to the saddle point problem.

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \delta_V \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_V \\ r_\lambda \end{pmatrix} \tag{5.21}$$

where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. By applying one block wise Gauss elimination step to solve for the increments δ_V and δ_λ, one obtains

$$\begin{pmatrix} A & B \\ 0 & -B^T A^{-1} B \end{pmatrix} \begin{pmatrix} \delta_V \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} r_V \\ r_\lambda - B^T A^{-1} r_V \end{pmatrix} \tag{5.22}$$

where $S := -B^T A^{-1} B$ is the so-called Schur complement operator. In order to not explicitly compute A^{-1} a equation system with A is solved instead. In general, the optimality system (5.14) of problem (5.7) is highly nonlinear. Especially with increasing values of p, the solution process becomes more challenging unless a good initial guess V_p^0 is provided. Considering a finite sequence $p_k := p_{\text{init}} + k p_{\text{inc}}$ where $p_{\text{init}} := 2$ helps to overcome this issue and reduce computational effort. First, the solution for p_{init} with initial $V_{p_{\text{init}}} = 0$ and $\lambda = 0$ is computed. Thereafter the solution of the constraint p_k-Laplacian problem (5.7) is used as an initial guess for the p_{k+1}-Laplacian problem, cf. [Mül+21]. The approximation quality of the Lipschitz transformation is adjuced with the choice of $p_{\max}$.
The overall optimization procedure is outlined in Algorithm 5. The steepest descent method is the loop from Lines 4 to 28, where the necessary optimality condition is checked to determine convergence. Here, y again denotes the state variable of the PDE constraint e, which is referred to as the primal problem. Nested within this loop, a step-size control operates in lines Lines 7 to 27. It checks whether the proposed next shape $F(\Omega)$ improves the objective function in terms of the

displacement field V_p. If not, then the parameter α is reduced. Thereafter, the descent direction V_p must be recomputed, in contrast to the classical backtracking line search spaces, which only requires evaluation of the objective functional. This is because by reducing the step length, one can not follow a straight line towards $0 \in W^p$ but some path to stay within the solution manifold of the non-linear geometric constraints $g(F(\Omega)) = 0$. In Line 5 the adjoint PDE is solved, which yields the adjoint state z. In Line 11, the nonlinear solver for the steepest descent problem (5.7) is called.
Details on the nonlinear solver are outlined in Algorithm 6. The key part of this solver is the solution to the saddle point problem (5.21) in the Schur complement form (5.22). This could be realized with a variety of iterative solvers, which are not further specified here. Popular approaches for this kind of problem are the Uzawa iteration and the Arrow-Hurwicz algorithm. For this purpose, Algorithm 7 outlines the computational steps for a matrix-vector product with S.

5.3 Numerical Method

Results of the present study are not obtained from a finite volume method but from the open-source finite element toolbox UG4 [Vog+13]. Since the focus of this chapter is not on the simulation of technical flows but on the efficient integration of geometric constraints, the flow solver is not that crucial. This simulation framework has MPI-based parallelization and features a geometrical multigrid preconditioner [Hac85]. The grid partitioning and load balancing scheme is based on ParMetis [KSK13].
Stable $P_2 - P_1$ finite elements were used to discretize the governing nonlinear Navier-Stokes equations (5.3) and their linearization, therefore no additional stabilization is required. Moreover, the viscosity is $\nu = 0.02$ in all cases. The same setting was used to discretize the linear adjoint problem, cf. [PS23; OS21], and the references therein. As regards the p-Laplace relaxation problem, for which the optimality system is described in (5.22), P_1 Lagrange shape functions were employed. Computational grids consist of triangular (2D) and tetrahedral elements (3D). They were generated using GMSH [GR09].
The simulations followed the workflow proposed in Algorithm 5. At the beginning of each optimization step, the steady, incompressible flow, described by Navier-Stokes equations (5.3), was computed followed by the solution of the corresponding adjoint system. The p-Laplacian descent Algorithm 5 initially employed $p_{\text{init}} = 2.0$ and incremented p by $p_{\text{inc}} = 0.19$. The given maximum values of p read $p_{\max} = 4.8$ [$p_{\max} = 4.1$] for the computed 2d [3d] test cases. Termination criteria of Algorithm 5 and Algorithm 6 were set as $\epsilon_1 = 1 \cdot 10^{-5}$ and $\epsilon_2 = 1 \cdot 10^{-8}$, respectively. The modification term introduced in (5.19) reads $\epsilon = 1 \cdot 10^{n-8}$ for all cases.
As a practical note, care must be taken to correctly interpolate the values of v, p, and their respective adjoints. These are involved in the assembly of J', which is present in (5.22). The geometrical constraints are part of this system of equations. It has been described that they lead to an $m \times m$ system of equations, so their discretization is not within a finite element space but in $\mathbb{R}^m$. For the investigated case cases, $m \in \{3, 4\}$ in 2D and 3D, it is appropriate to use a direct solver for the Schur complement system.
The corresponding codes used for these results can be found in the online repository [PS21].

5.4 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 $Re = \frac{v_\infty H}{\nu} = 20$, where H refers to the length of the initial edges and v_∞ as defined below. The employed box-domain is sketched in Figure 3.1. It spans 20 units in length and $\gamma = 6$ units in height (2D, 3D) and depth (3D), respectively, and the flow enters the domain through the left

vertical boundary. The inflow profile on Γ_{in} features a peak unit value in the center of the inlet plane and is described by

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

where δ corresponds to the inlet height.
The central aspect of this chapter 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 5. A crucial aspect is the evolution of the mesh quality during an optimization. The 2D studies are used to compare the mesh quality of the optimal and the initial design by means of the ratio ρ between the radii of the circumcircle and incircle and report the extreme values of the interior angles of the triangulation. Moreover, the behavior of the proposed algorithm is described for two different 3D configurations, where the surface of the obstacle is highly resolved.
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 6, therefore there is no need to provide results for their fulfillment per step. Solving the nonlinear system of (5.14) implies solving the geometrical constraints (5.8) and (5.9) to the error reduction tolerance set for Newton's method. The major portion of the computational effort in Algorithm 5 is spent on solving the p-Laplace relaxed problem via the scheme described in Section 5.2. Particularly, Lines 10 to 14 of Algorithm 5 are computationally expensive, as will be explained here and in Section 5.4.3.

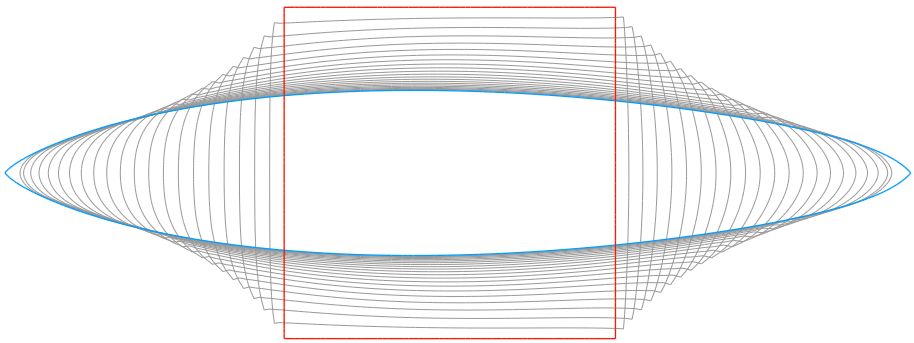


Figure 5.1: Superposition of the deformation sequence for a 2D configuration. The obstacle's initial shape red is presented superimposed to the sequence of generated shapes gray until an optimal shape blue is obtained upon convergence.

5.4.1 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 5.1 compares the initial (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 5.1, 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 [PS23] 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 5.2 magnifies, the geometry and the mesh in the upper-left corner of the obstacle. The initial and final shapes are presented on the top and bottom, respectively, for different grid refinement levels from left to right. The figure displays that the smoothing occurs similarly on all grids, and the elements around the initial singularity are not dramatically degenerated during the optimization. Towards the last step, no indication of the initial geometric singularity is visible on the obstacle's surface.

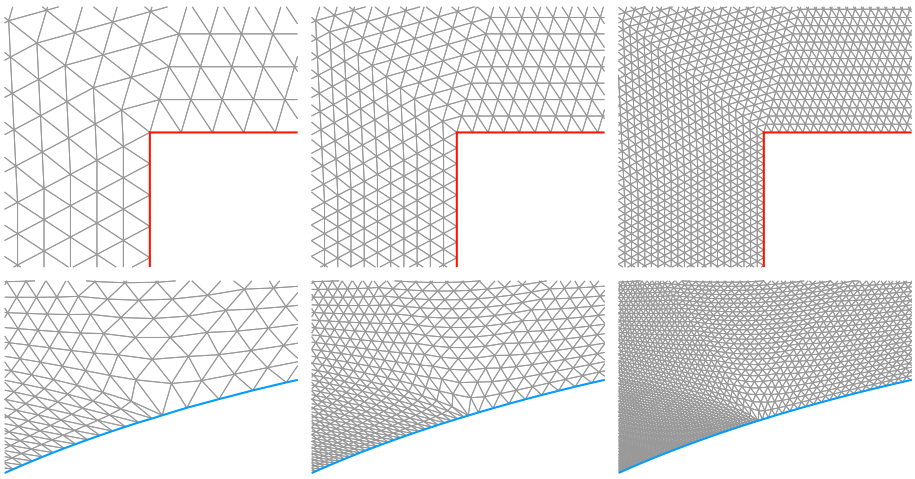


Figure 5.2: Removal of the geometrical singularity in the obstacle's initial configuration across several levels of refinement. For 4, 5, and 6 refinements the upper left corner of the box is smoothed via updating the geometry Ω iteratively, as stated in Line 15 in Algorithm 5.

The mesh quality is investigated for the final step using 4, 5, and 6 levels of refinement. As outlined in Section 5.2, a series of shape iterates are obtained throughout the optimization. The geometric multigrid preconditioner, which is used to allow for numerical scalability, requires the generation of a grid hierarchy. Therefore, a coarse mesh (base level) is provided to the solver, and the simulations are based on a predetermined mesh quality. While the optimization proposed in Algorithm 5 aims at preserving grid quality, it does not contemplate improving it with respect to the initial geometry. Table 5.1 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 angle, and the largest radius ratio. In addition, the radius ratio is compared between the first and last configuration. 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 (5.4) and (5.6), 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 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 (5.18) as an exponent. The latter fact turned out to be a limiting factor in our numerical simulations. However, the measurement of the worst minimum and maximum angles express that the triangles, which have undergone the largest deformation, are still not close to being critical.

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

Table 5.1: Assessment of mesh quality evolution for several refinement levels observed in 2D. Displayed data for minimum and maximum interior angles supplemented by the largest radius ratio of the triangulation extracted for the last optimization step, where an optimal shape is reached. The last column compares the largest radius ratio of the optimal shape (subscript ∞) and the initial configuration (subscript 0, $\rho_0 = 1.468$).

5.4.2 Three-Dimensional Studies

Results for the 3D simulations refer to 4 levels of grid refinement. The computational grid has a total of 4,980,736 tetrahedrons, and 49,152 triangles discretize the surface of the obstacle, Γ_{obs}, on the highest refinement level. Our optimization scheme generates a series of deformation fields V_p that, applied to the domain Ω, results in an optimal shape with respect to the energy dissipation (5.2).

Figure 5.3 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 opposite the main flow direction. Since this effect contributes to energy dissipation, it vanishes during the optimization at an early stage. One can quantify this phenomenon by observing the shear stress acting on the surface of the obstacle $\tau \cdot e_1 = \big(\nu(Dv + Dv^T) \cdot n\big) \cdot e_1$. Here, e_1 is the first unit vector describing the main flow direction. For the 3d (cf. Section 5.4.2) 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

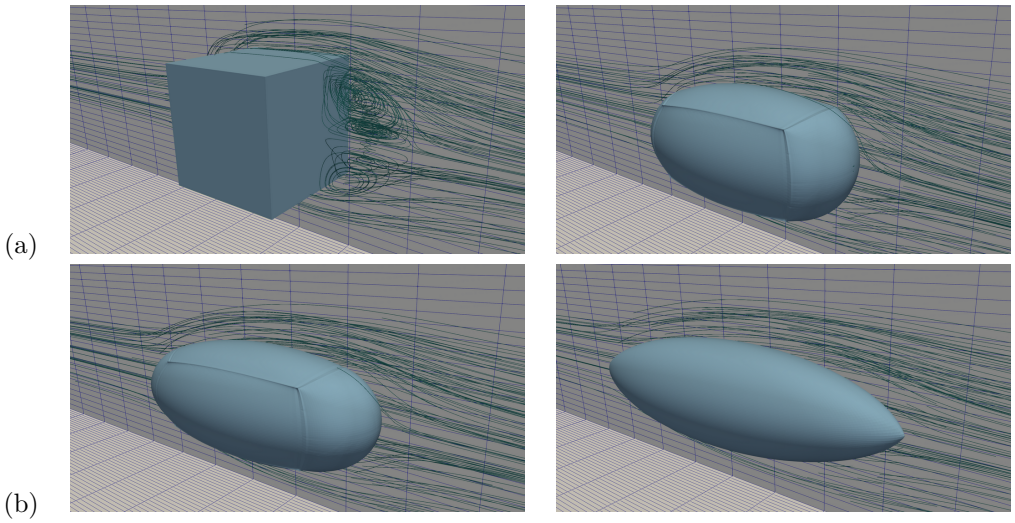


Figure 5.3: Streamlines for the rear of the obstacle located in the wind tunnel.

geometry are gradually removed as part of the optimization process, cf. Figure 5.4. 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 5.3.

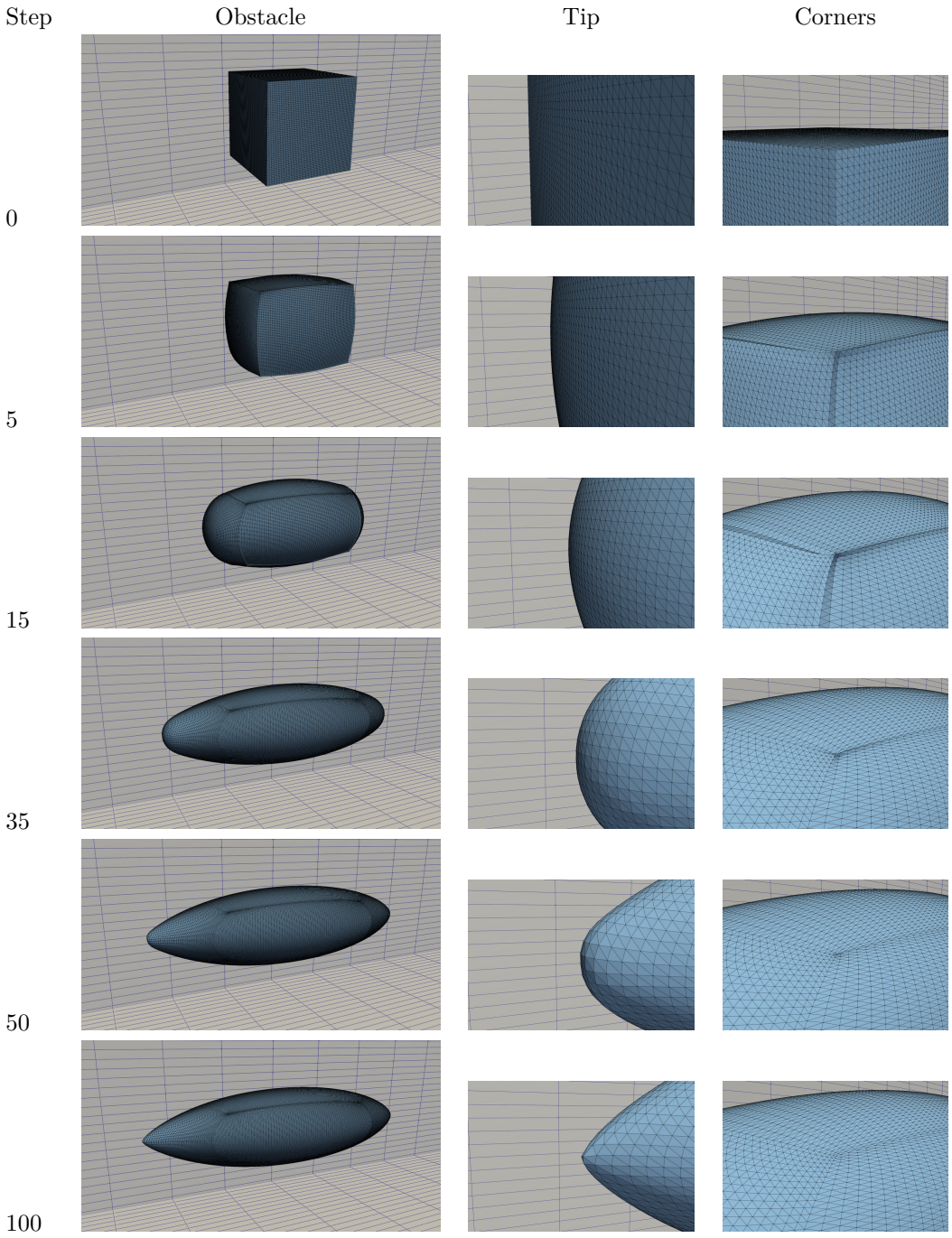


Figure 5.4: Deformation sequence for optimization steps {0, 5, 15, 35, 50, 100}. The complete obstacle, together with a detailed view of the geometrical singularity removal and generation process, is presented.

Figure 5.4 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 with 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 as seen in step 35, where the final tip locations also become more apparent. Recall that the mesh deformation corresponds to Line 15 in Algorithm 5. It may be emphasized that all shape iterates meet the volume and barycenter constraints, which are deemed crucial for the success of this optimization scheme. Footprints of initial corners and edges are still visible in the mesh at later stages of the optimization, e.g., step 50. However, they are completely smoothed out towards the end of the simulation and only the macro elements, resulting from the grid hierarchy, are visible. The front tip is shown for step 100, where the previously existing singularities have disappeared.
As mentioned in Section 5.1, this approach optimizes the obstacle's shape for the functional given in (5.2). Therefore, results are provided in Figure 5.5 for a 3D setting, which shows how the generated shape, after convergence of Algorithm 5, consists of an optimum with respect to the cost function.
Figure 5.5 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 (5.2) decreases monotonically is linked to Lines 15 to 27 of Algorithm 5, where a line search strategy is implemented. Once the deformation field is obtained for p_{max}, the geometry is updated. The state equation is solved, and the cost function is calculated to guarantee that the new shape iterate represents a descent direction. As seen in Lines 18 to 21, the deformation is withdrawn whenever the condition is false, and the step size control value is reduced to repeat the unsuccessful step with a scaled shape sensitivity J'. As a further indicator of the convergence, the distance between the iterated shapes Ω^k_{3ref} of the 3 refinements run to the optimal solution of the 4 refinements run $\Omega^\infty_{\text{4ref}}$ is approximated. Figure 5.5 shows the symmetric difference

$$d(\Omega^k_{\text{3ref}}, \Omega^\infty_{\text{4ref}}) := \left|\Omega^k_{\text{3ref}} \setminus \Omega^\infty_{\text{4ref}}\right| + \left|\Omega^\infty_{\text{4ref}} \setminus \Omega^k_{\text{3ref}}\right|. \tag{5.23}$$

The integration is carried out using the Boolean filters of the VTK library [Kit] 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, the volume can be found utilizing the divergence theorem.

5.4.3 Scalability Study

Weak scalability of the solution strategy for the p-Laplacian relaxed problem, from $p_{\text{init}} = 2.0$ up to p_{max} is presented here. This solution strategy, described in Section 5.2, is referred to in these results as the p-solver. This corresponds to Lines 10 to 14 of Algorithm 5. It was studied for up to 26,2144 cores in a 3D setting. The study was carried out with the supercomputer Hawk at HLRS. It features 5632 compute nodes, each with a dual-socket architecture and a total of 128 cores. Each core with a maximum frequency of 2.25GHz, and 256GB of RAM. The runs were carried out taking into account the hypercube topology of the system to maximize core usage and minimize parallel communication overhead. The grid partitioning and load balancing scheme is based on ParMetis [KSK13].
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 5.6. An eight-fold increase in the number of cores is performed for each level of refinement Results are presented for the solution of the nonlinear system of equations given in (5.14) via its linearization in (5.15) and (5.16). This system is solved using Newton's

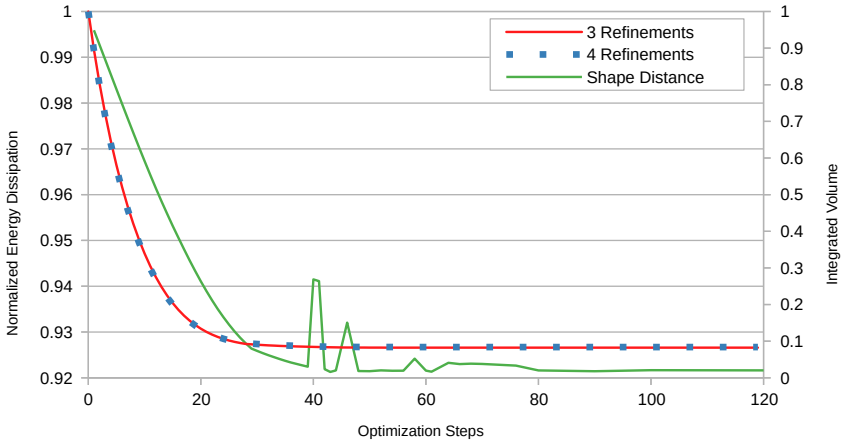


Figure 5.5: 3d results for 3 and 4 levels of refinement are compared. The energy dissipation, see (5.2), is plotted against the difference between shapes of each refinement level per step.

method with a BiCGStab as a solver for the underlying linearization. The linear solver is set to absolute and relative error reductions of $1 \cdot 10^{-10}$ and $1 \cdot 10^{-16}$, 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.

Procs	Refs	Number of Elements	DoFs	Newton Its.	Total Lin.Its	Lin.Its. (5.15)
512	2	77,824	44,730	68	2,080	394
4,096	3	622,592	334,158	68	2,458	472
32,768	4	4,980,736	2,581,014	68	2,606	509
262,144	5	39,845,888	20,283,942	68	2,912	577

Table 5.2: The iteration counts for one optimization step of the solver used to obtain V_p, Lines 10 and 14 in Algorithm 5, summarizing the number of Newton steps across several levels of refinement, and the linear solver iterations. Furthermore, the summary contains the number of elements per refinement level and the corresponding DoFs in the discretization of (5.21).

The accumulated times and iteration counts for the routines in Line 10 to Line 14 in Algorithm 5 are measured for one optimization step. The upper plot in Figure 5.6 exemplarily shows the measured time 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 solver. The procedure is carried out for each value of p starting at p_{init} up to $p_{\max}$ with p_{inc} intervals as explained in Section 5.2. The time measurement starts for every optimization at $V_{p_{\text{init}}}$ and ends when the corresponding Newton iteration converges for $V_{p_{\max}}$. The speedup in the lower plot of Figure 5.6 is presented relative to the base measurement with 512 cores. The iteration counts are shown to the number of elements and DoFs in Table 5.2. The column of the total linear iterations includes all the necessary calls to the linear solver used within the linearization. As shown in (5.22), for each solution of the linear system of equations, it is necessary to solve $m + 2$ times with A^{-1}. These

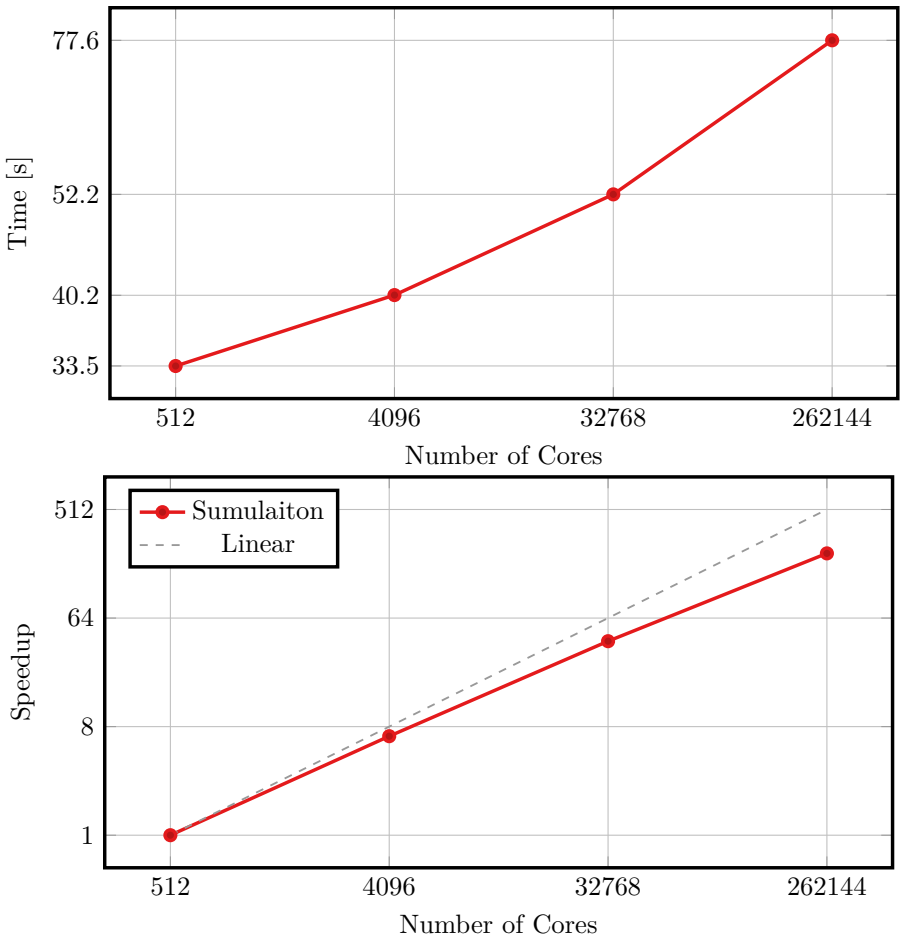


Figure 5.6: Weak Scaling: Results for the first optimization step. The upper figure shows the accumulated wallclock time for all values of p, and the lower shows the speedup relative to 512 cores.

include one time for the RHS in the second equation of (5.22), and m for the computation of S. Additionally, the first equation of (5.22) has to be solved for δ_u, whose iteration counts are shown, individually, in the rightmost column of Table 5.2.
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 in the number of DoFs. Altogether, the results show the need for using numerical solvers with grid-independent convergence. Recall that our target is to use the solution of the p-Laplace relaxed problem for the highest value of p, i.e. $p_{\max}$, as a deformation field to generate a series of shape iterates. Moreover, this is done by solving the same problem for lower values of p, to have a good initial guess when approaching the maximum p. The latter fact is necessary since with each increment of p_{inc}, our problem becomes more nonlinear, making it more difficult to solve, particularly without a good initial guess. For the given settings, $p_{\text{init}} = 2.0$ to $p_{\max} = 4.1$ and an increment of $p_{\text{inc}} = 0.19$, Algorithm 6 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 that 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 a grid hierarchy can represent, see [Rei+13], which implies that care must be taken during the generation of the grid. Nevertheless, it is a very effective approach towards solving for V_p with increments of the p value. The results in Figure 5.6 show that the p-relaxed problem becomes inexpensively solvable. Additionally, the benefits of the multigrid preconditioner are evident by noticing how Newton's method is perfectly scalable in the number of steps needed for all refinement levels and in the slight increase in linear solver iterations between the initial and final runs. The table shows that 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

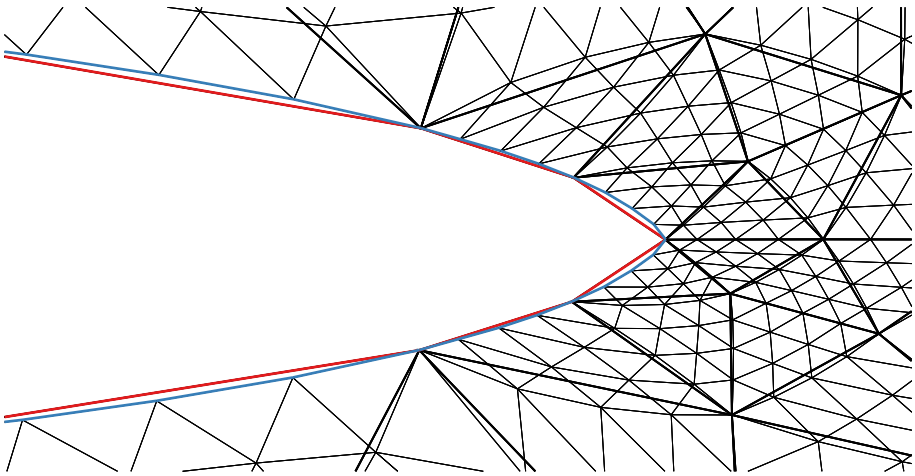


Figure 5.7: The base level and the second level of refinement are compared for a 2D simulation for the last step before convergence. The macro and refined triangular elements are shown in bold and thin black lines, respectively. Given that the deformation field is restricted and applied throughout the grid hierarchy, the coarsest grid is an interpolation of the finest.

across all optimization steps, it is necessary to apply the deformation field across the complete grid hierarchy. This is shown in Figure 5.7, where the base level is compared to the finest grid with two refinements. It is visible how V_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 into 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 in tetrahedrons and DoF, with a slight increase in the computational work of linear iterations. Although the performance dropped marginally, the wallclock times and speedup show that the proposed numerical scheme for the solution of the p-Laplace relaxed problem could be used for problems with large numbers of DoF, corresponding to real-world industrial applications.

5.5 Conclusions on Scalable Algorithms with Geometric Constraints in Banach Spaces

This chapter assessed the efficiency of the steepest descent method based on $W^{1,p}$ approximations of $W^{1,\infty}$-deformations for PDE-constraint shape optimization problems exposed to fixed-dimensional geometric constraints. Results were compiled for fully submerged 2D ($p = 4.8$) and 3D ($p = 4.1$) shapes and low Reynolds number flow ($Re = 20$), similar to the shapes investigated in Chapter 3. It was (again) demonstrated that the p-Laplace method and the related algorithms work for general Lipschitz shapes since deformations allow singularities in the surface to be smoothed from the initial design or newly generated for the final design. Different from the implementation in Chapter 3, the fixed-dimensional constraints are incorporated together with the PDE constraints into the optimization algorithm using a Schur Complement approach. Compared to approximate algorithms such as the penalty-based and augmented Lagrangian approaches, a significant gain of robustness in the treatment of geometric constraints over the optimization steps is observed. Additionally, line search schemes for the steepest descent direction in $W^{1,\infty}$ are addressed, which –in contrast to the aforementioned Hilbert space methods– is a nonlinear problem. Moreover, this problem lives on the solution manifold of the nonlinear geometric constraints posing a non-convex set in general. Solving the actual p-Laplace problem remains an aspect to be improved. Chapter 6 circumvents solving the non-linear p-Laplace problem. Instead, the aim is to approximate the direction of the steepest descent in $W^{1,\infty}$ rather than $W^{1,p}$, with reasonable computational effort and, at the same time, a reliable calculation procedure.

Chapter 6

A $W^{1,\infty}$ Steepest Descent Approach using ADMM

This chapter discusses a method for approximating the direction of steepest descent in $W^{1,\infty}$-topology for a shape optimization problem investigated in Chapters 3 and 5. In previous chapters, the p-Laplace and Steklov-Poincaré methods, partly hybridized, were used to determine the shape deformation field. When p increased, these methods were sometimes associated with extensive computational efforts, which motivated investigating alternatives. The forthcoming *Alternating Direction Method of Multipliers* (ADMM) strategy discussed in this chapter is an interesting route since the p-Laplace and Steklov-Poincaré methods can be interpreted as a relaxation of the $W^{1,\infty}$ problem. The numerical 1D and 2D examples in Sections 6.2 and 6.3 are computed using [Mül].

Recall Figure 3.1, which sketches the geometric setting, the fluid dynamic state defined in (5.3), and the objective is the fluid dynamic drag (3.5). With a given shape derivative J', additionally assuming that Ω is a bounded domain, the minimization problem

$$\min_{V \in W} J'(\Omega)V \qquad \text{s.t.} \qquad |DV| \leq 1, \text{ a.e. in } \Omega \qquad W \subset W^{1,\infty}(\Omega, \mathbb{R}^d) \tag{6.1}$$

is solved to obtain a shape deformation field V, where $|DV|$ denotes the spectral norm of the Jacobian matrix DV, and W is a function space specified below. Different from the geometric constraints (5.8) and (5.9) used in Chapter 5, here, the linear constraints

$$\int_\Omega \Big[V + \mathrm{id}\,\mathrm{div}(V)\Big]\, dx = 0, \tag{6.2}$$

$$\text{and} \int_\Omega \mathrm{div}(V)\, dx = 0 \tag{6.3}$$

are employed to preserve the volume and barycenter of Ω up to first-order. Furthermore, the vector field V must fulfill Dirichlet boundary conditions on the outer boundary, that is, $V = 0$ almost everywhere on $\Gamma \setminus \Gamma_{\text{obs}}$. Hence, the function space

$$W := \Big\{ V \in W^{1,\infty}(\Omega, \mathbb{R}^d) : V = 0 \text{ a.e. on } \Gamma \setminus \Gamma_{\text{obs}}, \int_\Omega \mathrm{div}(V)\, dx = 0, \int_\Omega \Big[V + \mathrm{id}\,\mathrm{div}(V)\Big]\, dx = 0 \Big\} \tag{6.4}$$

is considered for the direction V. This chapter approximates the solution of the highly non-linear problem (6.1) using an Alternating Direction Method of Multipliers.

The ADMM-based solution approach was initially suggested in [DHH23] for shape optimization model problems constrained by PDEs, including the Laplace and the Bi-Laplace operator. In this chapter, the method is applied to the minimal drag problem of an obstacle using the surface formulation of the shape derivative $J'(\Omega)V$ from (3.5). Before discussing the algorithmic approach, the following may be mentioned. The space of Lipschitz functions $C^{0,1}(\Omega, \mathbb{R}^d)$ – see, e.g., [AW96, Definition 1.29] – can be identified with $W^{1,\infty}(\Omega, \mathbb{R}^d)$ functions if the domain Ω is sufficiently smooth. If the domain is convex, one has even equality of the Lipschitz and $W^{1,\infty}$ semi-norms. In practice, this restricts the maximum step size that can be used for the shape update. Further, non-local information, which is known to the Lipschitz semi-norm, is lost, which may result in a false minimizing sequence, see [Her23, Section 2.2.1]. Recall that, a domain Ω is called convex, if for any two points $x, y \in \Omega$, the connecting straight line $\{ty + (1-t)x : t \in (0,1)\}$ is contained in Ω. Of course, this depends on the definition of a "straight line". In this context, concepts other than using an Euclidean distance would complicate the implementation (for example, the geodesic distance could be used for computations on the of a 3D object in the Euclidian space). The Lipschitz condition $|V(x) - V(y)| \leq |x - y|$ for a function V is a global property and holds for any two points $x, y \in \Omega$, but $|DV| \leq 1$ is a local property almost everywhere in Ω. However, the globally valid Lipschitz condition can be handled with $|DV| \leq 1$ almost everywhere in Ω for a convex domain [Her23, Section 2.4]. If Ω is not a convex domain, one way to circumnavigate this issue is suggested by Herbert in [Her23, Remark 2.15]. In fact, one could include a fictitious convex and bounded hold-all domain $B \subset \mathbb{R}^d$ such that $\Omega \subset B$, see Figure 6.1.

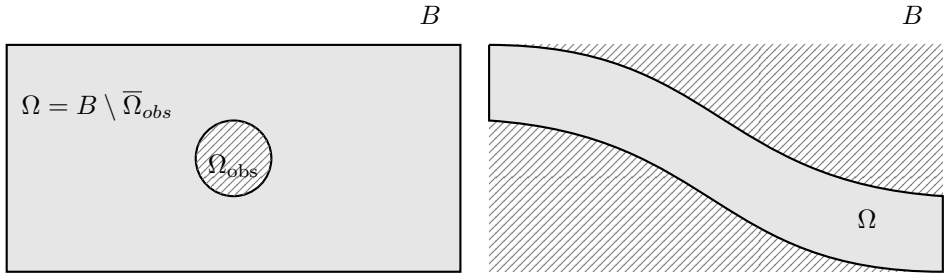


Figure 6.1: Domains converted to fictitious convex hold-all domains where the gray region belongs to the actual domain, and the shaded region is void space. The sketch on the left corresponds to the setting of the minimal drag problem used in Chapters 3, 5 and 7 and this chapter, created by filling the obstacle Ω_{obs}. The hold-all domain on the right, corresponding to the duct flow in Chapter 4, which is obtained by virtually extending the domain to an extensive box.

6.1 Algorithm

With a given shape derivative J' and by introducing the matrix-valued slack variable $q : \Omega \to \mathbb{R}^{d\times d}$ and multiplier $\lambda : \Omega \to \mathbb{R}^{d\times d}$ the Lagrange function is defined by

$$L(q, V, \lambda) := J'(\Omega)V + \int_\Omega \phi(q)\, dx + \frac{\tau}{2}\|q - DV\|_{L^2}^2 + (\lambda, q - DV)_{L^2}, \tag{6.5}$$

for a given $\tau > 0$ and where $(\cdot,\cdot)_{L^2}$ is the L^2-inner product and $\|\cdot\|_{L^2}$ the L^2-norm for matrix-valued variables over Ω, and

$$\phi(q) = \begin{cases} 0 & : \ |q| \leq 1, \\ +\infty & : \ \text{else.} \end{cases} \tag{6.6}$$

The ∞-ADMM approach is similar to the method of Lagrange multipliers, but the optimality system of the unconstrained problem is not solved at once. Instead, L is minimized regarding the arguments q and V individually, and an update to the multiplier λ is performed thereafter. Furthermore, the sequence of minimization problems regarding V leads to a variational form that has a solution in a Hilbert space

$$H := \Big\{ V \in H^1(\Omega, \mathbb{R}^d) : V = 0 \text{ a.e. on } \Gamma \setminus \Gamma_{\text{obs}}, \int_\Omega \operatorname{div}(V)\, dx = 0, \int_\Omega \Big[V + x \operatorname{div}(V) \Big]\, dx = 0 \Big\}, \tag{6.7}$$

rather than W. Algorithm 8 outlines the procedure in detail. Here, the step size $\tau > 0$ is fixed,

Algorithm 8 ADMM Algorithm

1: $j \leftarrow 0$
2: **repeat**
3: Minimize with respect to q with fixed V^j and λ^j:

$$\tilde{q} \leftarrow \arg\min_q L(V^j, q, \lambda^j) \tag{6.8}$$

4: Project element-/cell-wise

$$q^{j+1} \leftarrow \tilde{q} / \max(|q|, 1) \tag{6.9}$$

5: Minimize with respect to V with fixed q^{j+1} and λ^j:

$$V^{j+1} \leftarrow \arg\min_V \Big\{ L(q^{j+1}, V, \lambda^j) : V \in H \Big\} \tag{6.10}$$

6: Update multiplier:

$$\lambda^{j+1} \leftarrow \lambda^j + \tau(q^{j+1} - DV^{j+1}) \tag{6.11}$$

7: $j \leftarrow j + 1$
8: **until** $r^j := \big(\|\lambda^{j+1} - \lambda^j\|_{L^2}^2 + \|DV^{j+1} - DV^j\|_{L^2}^2 \big)^{1/2} \leq \text{tol}$

but more general ADMM formulations exist, e.g., in [BM20], which considers a variable step size τ. Note that Lines 3 and 4 are, in fact, a practical implementation of the convex minimization problem.
For the first step in Algorithm 8, a preliminary $\tilde{q}$ given at the point where $L_q(V^j, \tilde{q}, \lambda^j)\delta_q = 0$ for all test functions δ_q and fixed V^j, λ^j. Thus, the preliminary $\tilde{q}$ solves

$$\int_\Omega \tau(\tilde{q} - DV^j) : \delta_q\, dx + \int_\Omega \lambda^j : \delta_q\, dx = 0, \qquad \forall \delta_q, \tag{6.12}$$

which is fulfilled when setting

$$\tilde{q} = DV^j - \frac{1}{\tau}\lambda^j, \tag{6.13}$$

without solving a linear equation system. To ensure that the penalty term $\int_\Omega \phi(q)\, dx$ is zero for $q = q^{j+1}$, the preliminary $\tilde{q}$ is projected, using the largest singular value of the $d \times d$ matrix $\tilde{q}^\intercal q$. Therefore, $\sigma_1 := \sqrt{\max_i(\lambda_i(\tilde{q}^\intercal \tilde{q}))}$ is computed element-/cell-wise and $q^{j+1} = \tilde{q} / \max(1, \sigma_1)$. Because the resulting deformation vector field V^{j+1}, in the second step of Algorithm 8, must

fulfill the geometric constraints (6.2) and (6.3). Thus, the sub-minimization problem (6.10) is a constrained problem by itself. By introducing the Lagrange multipliers $\mu = (\mu_{bc}, \mu_v)^\intercal \in \mathbb{R}^{d+1}$, with $\mu_{bc} \in \mathbb{R}^d$ and $\mu_v \in \mathbb{R}$, the new Lagrange function reads

$$\mathcal{L}(V, \mu) := L(q^{j+1}, V, \lambda^j) + \mu_{bc} \cdot \int_\Omega \Big[V + x \operatorname{div}(V)\Big]\, dx + \mu_v \int_\Omega \operatorname{div}(V)\, dx \tag{6.14}$$

for fixed q^{j+1} and λ^j. The the deformation field V^{j+1} then is characterized by $\mathcal{L}_V(V^{j+1}, \mu)\delta_V = 0$ for all V in H, and soves the Poisson problem

$$\begin{aligned} J'(\Omega)\delta_V - \int_\Omega \tau(q^{j+1} - DV) : D\delta_V \, dx - \int_\Omega \lambda^j : D\delta_V \, dx \\ +\mu_{bc} \cdot \int_\Omega \Big[\delta_V + x \operatorname{div}(\delta_V)\Big]\, dx + \mu_v \int_\Omega \operatorname{div}(\delta_V)\, dx = 0, \quad \forall \delta_V \in H. \end{aligned} \tag{6.15}$$

To efficiently solve the optimality system of the sub minimization problem with (6.14), one could use, for example, an iterative method with the Schur complement product in Algorithm 7. However, the method in this chapter uses the iterative approach outlined in Algorithm 9, where V_0^{j+1} is an initial guess and α, tol > 0 is a user-prescribed step size and tolerance. The lower case index k of V_k^{j+1} denotes the k-th approximation for V^{j+1}, used in Algorithm 8. Algorithm 9 is an

Algorithm 9 Iterative solver for the constrained Poission

1: **function** ConstraintPoissonSolver(V_0^{j+1}, α, tol)
2: $k \leftarrow 0$, $\mu \leftarrow 0$
3: **repeat**
4: Solve (6.15) for V_{k+1}^{j+1} with μ_k and initial guess V_k^{j+1}
5: Update multipliers

$$\mu_{k+1} \leftarrow \mu_k - \alpha \begin{pmatrix} \int_\Omega \Big[V_k + x \operatorname{div}(V_{k+1}^{j+1})\Big]\, dx, \\ \int_\Omega \operatorname{div}(V_{k+1}^{j+1})\, dx \end{pmatrix} \tag{6.16}$$

6: $k \leftarrow k + 1$
7: **until** $\left(\|\mu_{k+1} - \mu_k\|_2^2 + \|V_{k+1}^{j+1} - V_k^{j+1}\|_{L^2}^2\right)^{1/2} \leq$ tol
8: **return** V_k^{j+1}
9: **end function**

alternative to the method used in Chapter 5 for handling the geometric constraints numerically. The advantage is that is much simpler to implement and allows
The multiplier λ in (6.5) is updated element-/cell-wise for the discrete system using the update formula (6.11). The procedure is converged if the change in the multipliers λ^j, λ^{j+1} and the Jacobians DV^j, DV^{j+1}, measured in the L^2-norm over Ω is smaller than a given tolerance tol.
The following two sections study the algorithm with simple one- and two-dimensional model problems before using the approach for computing the described direction shape optimization procedure for the minimal drag problem.

6.2 A One-Dimensional Model Problem

This section studies the solution procedure, outlined in Algorithm 8 using a one-dimensional model problem, which is a variant of the numerical experiment performed in [Her23, Section 5.1]. The solution is verified with the help of a series of p-Laplace relaxed problems (3.3). It is expected

that the approximated $W^{1,p}$ solutions tend to the $W^{1,\infty}$ solution computed using the ADMM algorithm, above. To verify the computations, a grid refinement study and an error analysis are carried out, using a Richardson extrapolation to estimate the exact solutions. The aim is to find a function $V : (0, 2\pi) \to \mathbb{R}$ that is a solution to the minimization problem

$$\min_{V \in W^{1,\infty}((0,2\pi))} \int_0^{2\pi} \left[fV + gV' \right] dx \quad \text{s.t.} \quad |V'| \leq 1, \text{ and } \int_0^{2\pi} V \, dx = 0 \tag{6.17}$$

for given functions f and g, and with $V(0) = 0$ and a natural boundary condition at the right boundary. The objective in (6.17) has the form of the volume formulation of $J'(\Omega)V$ for a simple objective functional $J(\Omega)$, see Example 1. The artificial "shape sensitivity" is, for this example, given by the functions

$$g(x) = -0.1|x - \pi| \quad \text{and} \quad f(x) = \begin{cases} 0.1 & : \ x < \pi, \\ -0.1 & : \ x \geq \pi. \end{cases} \tag{6.18}$$

The p-Laplace relaxed problem employs different values $p > 2$ and reads

$$\min_{V_p \in W^{1,p}((0,2\pi))} \int_0^{2\pi} \left[fV_p + gV_p' \right] dx + \frac{1}{p} \int_0^{2\pi} |V_p'|^p \, dx \quad \text{s.t.} \quad \int_0^{2\pi} V_p \, dx = 0, \tag{6.19}$$

with $V_p(0) = 0$, a natural boundary condition at the right boundary.
The problem is discretized using the finite element method with continuous piecewise linear elements for V and piecewise constant (discontinuous) values for the slack variable q and the multiplier λ. The constrained Poisson equation, which occurs in Algorithm 8, Line 4, is solved directly with LU-factorization. The LU-factorization is computed using the *LAPACK* routine *DGESV*, and the SVD is computed by *DGESVD* [Ten+; And+99]. The p-Laplace problem (6.19) is solved analogously to Section 5.3.
Figure 6.2 shows the numerical solutions of V and V_p for $p = 2, 4, 8$, and 16 on a grid with a spacing of $2\pi/\Delta x = 512$. It can be observed, that for increasing p the graph of V_p approaches the graph of V from below, which is the expected behavior according to the statements in [IL05].

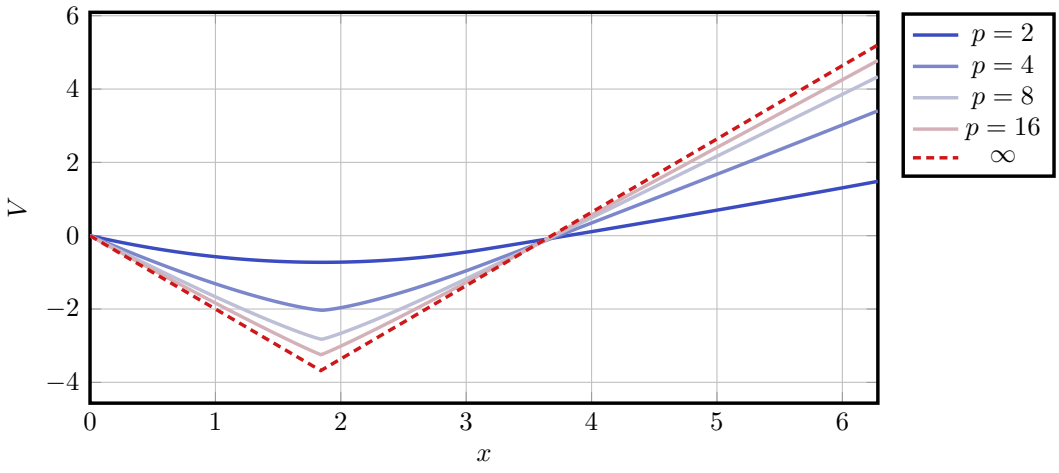


Figure 6.2: Comparison of solutions for the p-Laplace relaxed problem (straight lines) and the $W^{1,\infty}$-solution (dashed line) of the model problem (6.17) obtained on a grid with $2\pi/\Delta x = 512$).

Figure 6.3 shows the residual r^j from Algorithm 8, Line 8 for the computation with different grid refinement levels, starting from a spacing of $2\pi/\Delta x = 32$ down to $2\pi/\Delta x = 512$. The algorithm

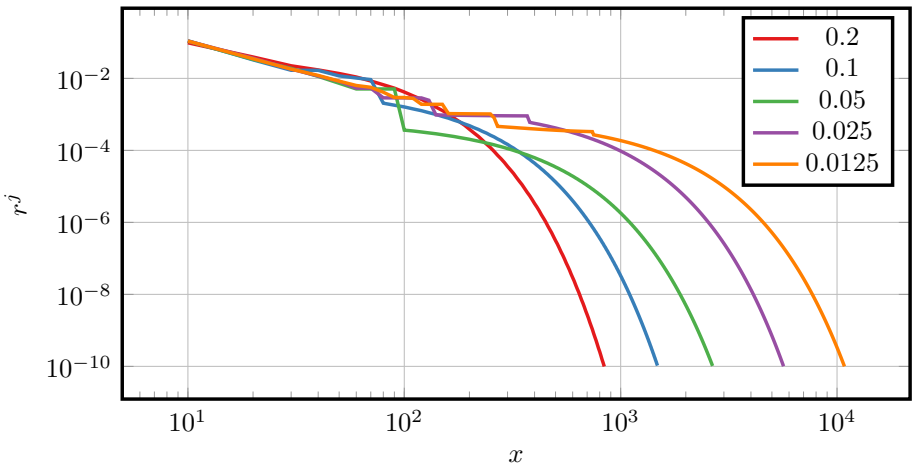


Figure 6.3: The evolution of the residuals extracted from Algorithm 8, Line 8 for different grid refinement levels $2\pi/\Delta x \in [32, 64, 128, 256, 512]$ (converg. criterion: tol $= 10^{-10}$).

Grid size Δx	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p \to \infty$
$2\pi/32$	-0.8277	-2.881	-4.3018	-5.0925	-5.8724
$2\pi/64$	-0.815	-2.8567	-4.2746	-5.0644	-5.5401
$2\pi/128$	-0.8081	-2.8428	-4.2578	-5.0466	-5.3654
$2\pi/256$	-0.8046	-2.8355	-4.2491	-5.0374	-5.2767
$2\pi/512$	-0.8028	-2.8318	-4.2446	-5.0326	-5.2316
$\Delta x \to 0$	-0.801	-2.8279	-4.2398	-5.0272	-5.1851

Table 6.1: Minimal values of the 1D model problem's objective function computed for different grid levels. Comparison of relaxed problem solutions (6.19) for a selection of p-values and solutions of (6.17) in the final column. The bottom row refers to values obtained by a Richardson extrapolation.

terminates when the convergence criterion $r^j \leq \text{tol} = 10^{-10}$ is met. The convergence is reached for all grid levels, but finer grids require a larger number of iterations. The corresponding minimal values of the objective function observed in conjunction with (6.17) and the relaxed problem (6.19) are listed in Table 6.2. The bottom row displays the estimated exact solutions using the Richardson extrapolation, e.g., in [FPS20, Section 3.9]. Further insight into the behavior of the two solution strategies follows from Figure 6.4. The figure shows a plot of the estimated discretization error, which is calculated using the extrapolated exact solution obtained from the Richardson extrapolation and outlines that, for all computations, the error reduction is approximately of first order, as indicated by the gray dashed line.

6.3 A Two-Dimensional Scalar Model Problem

Before applying the $W^{1,\infty}$ approach to a shape optimization problem in the following section, this section investigates a 2D scalar-valued model problem. Here, the aim is to minimize a functional, which is the integral over the boundary of the unit square $\Omega = (0,1) \times (0,1)$. This is interesting because the shape optimization problem uses the boundary formulation of the shape derivative $J'(\Omega)V$. The task is to find a function $V : (0,1) \times (0,1) \to \mathbb{R}$, which solves the minimization

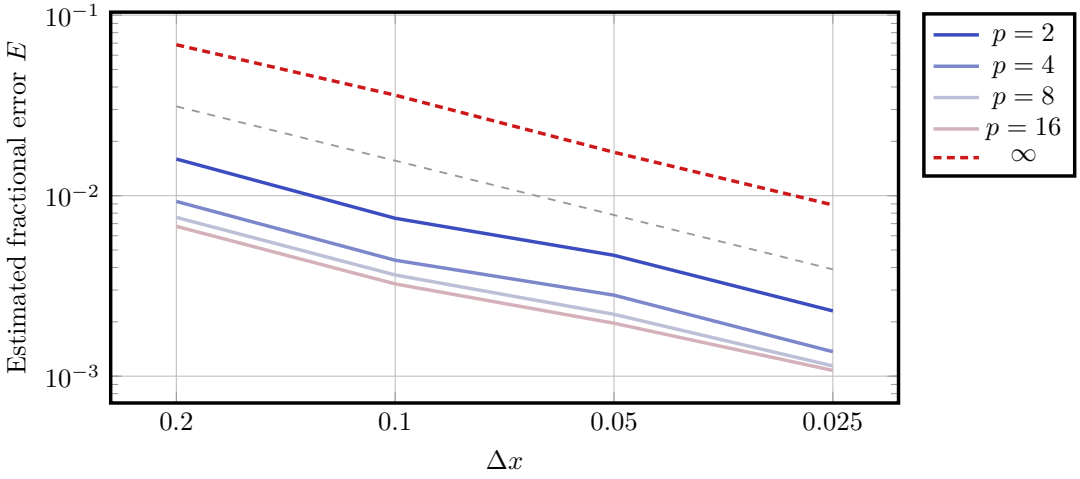


Figure 6.4: Error estimation of the objective function value tabulated in Table 6.1 using Richardson extrapolation. The gray dashed line indicates a linear regression rate.

problem

$$\min_{V \in W^{1,\infty}(\Omega)} \int_{\Gamma_N} gV \, ds \quad \text{s.t.} \quad \|\nabla V\| \leq 1 \tag{6.20}$$

with Dirichlet conditions $V(x, 0) = 0$, and $V(1, y) = 0$, and an artificial sensitivity

$$g(x, y) = \sin(2\pi(x - 0.1)) \sin(2\pi(y - 0.2)). \tag{6.21}$$

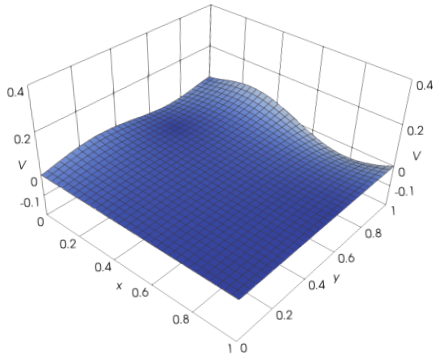
Since the problem is scalar-valued, the operator norm for the gradient condition, which is used in the vector-valued case, is replaced by the 2-norm, that is, $\|\nabla V\| = \sqrt{\nabla V \cdot \nabla V} \leq 1$. As above, the approximated $W^{1,\infty}$ solution is compared with $W^{1,p}$ solutions for a sequence of p-values, which solve the relaxed problem

$$\min_{V \in W^{1,p}(\Omega)} \int_{\Gamma_N} gV_p \, ds + \frac{1}{p} \int_{\Omega} (\nabla V_p \cdot \nabla V_p)^{\frac{p}{2}} \, dx \tag{6.22}$$

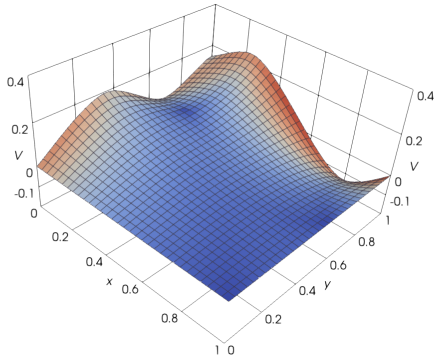
with the boundary conditions $V_p(x, 0) = 0$, $V_p(1, y) = 0$, for $x, y \in [0, 1]$ and g as above.

Similar to the previous experiment, the problem is discretized using the finite element method with piecewise linear elements for V and piecewise constant values for the slack variable q and multiplier λ. Integrals are approximated using the Gauß-Legendre quadrature rule with 25 quadrature points for the quadrilaterals and 5 for the one-dimensional boundary elements.

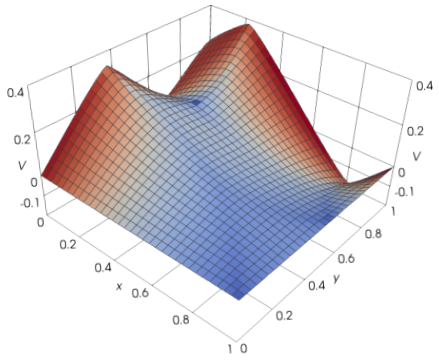
Results obtained on a regular equidistant grid with 32×32 quadrilaterals are visualized in Figure 6.5. The figure shows the graph of the resulting function for the p-Laplace relaxed problem with $p = 2, 4$ and 16 in comparison with the solution obtained from Algorithm 8. The ADMM solution displays pronounced ridges that reach from their roots at the extreme values of the boundary data into the domain. The $p = 16$ case also reveals such kinks along the boundaries, however, much less pronounced as for the ADMM method, and significant smoothing is observed in the interior. The color map indicates the gradient norm $\|\nabla V\|_2$ for the values between 0 and 1. For the particular choice of (6.21) the plots show that the $W^{1,p}$ solution tends to the $W^{1,\infty}$ approximation from below, that is, for larger p, the function values of V_p increase.



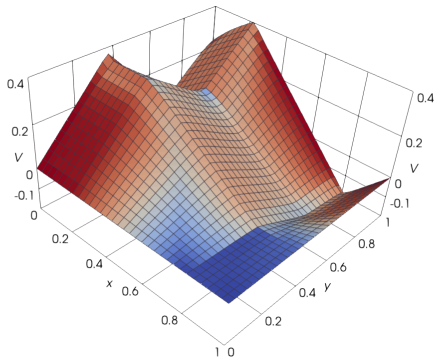
$p = 2$



$p = 4$



$p = 16$



∞-ADMM solution

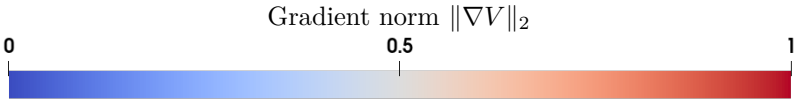


Figure 6.5: Comparison of scalar-valued solution graphs obtained from the p-Laplace relaxed problem for a sequence p-values and the $W^{1,\infty}$ approximation obtained with ADMM. The color map indicates the 2-norm of the gradient.

6.4 Drag Optimization in $2D$

This section discusses the application of the $W^{1,\infty}$ approach using ADMM applied to minimize the fluid dynamic drag. The performance of the optimization procedure, the resulting final shapes and the quality of the computational grids obtained are compared between the suggested $W^{1,\infty}$ approach, the descent method using a p-Laplace relaxation (p=4.1) and the Steklov-Poincaré approach described in Section 4.1.
Again the external flow around an initially circular shape at a Reynolds number of $Re_D = 10$ is considered. The initial computational mesh, which consists of approximately 14.5k control volumes, is identical to the one used in Section 3.4.1, shown in Figure 3.2. For the descent direction, the convergence tolerance in Algorithm 8 is set to $\text{tol} = 10^{-2}$ and the step size is fixed with $\tau = 1$.
The governing fluid dynamic equations and the Poisson problems (6.15) are both discretized using a second-order accurate cell-centered finite volume method, as outlined in Appendix B. The linear equation system in Algorithm 9, Line 4, resulting from the discretization of (6.15), is solved using the parallel conjugated gradient method outlined in Appendix D with block Jacobi and block-wise SSOR preconditioning with $\omega = 0.8$. Algorithm 9 is used to compute V^{j+1} and the multipliers for the geometric constraints with a constant step size $\alpha = 1$. The sequence of shapes obtained by the

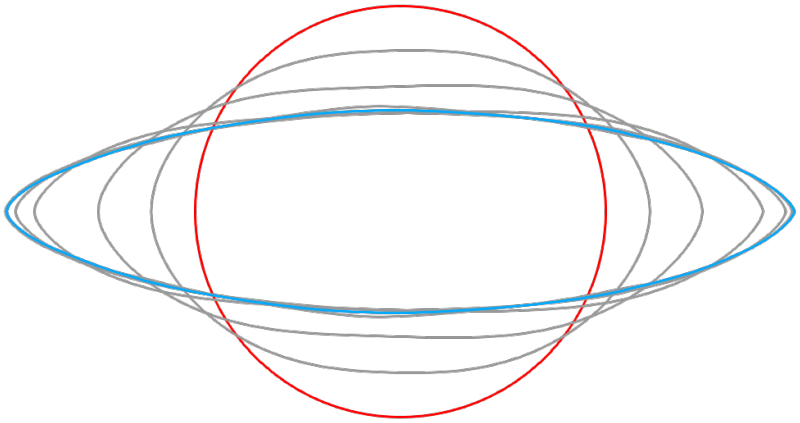


Figure 6.6: Superposition of the deformation sequence using the ∞-ADMM approach. The obstacle's initial shape (red) transforms through the sequence of generated shapes (gray) until a final optimized shape (blue) is obtained upon convergence.

∞-ADMM approach is visualized in Figure 6.6, where the red line is the initial and the blue line is the final shape.
For all three strategies, the deformation vector field V^k obtained for the k-the optimization step enters the shape and grid update by moving all grid nodes via $(\text{id} + t_k V^k)(x_i)$. However, this requires reconstructing the nodal values using the discrete solution obtained by the cell-centered finite volume method. Appendix C explains the interpolation strategy.
Mind that the step size t_k is controlled by a back-tracking line search approach. If the value of the objective functional (3.5) increases, the step size halves. An initial step size of $t_0 = 0.2$ was employed for the ∞-ADMM and p-Laplace approach. The Steklov-Poincaré method allows starting with a larger initial step size; here, the initial step size is set to $t_0 = 0.5$. In every step, it is checked if the objective functional (3.5) decreases. The optimization proceeds if $J((\text{id}+t_k V^k)(\Omega_k)) < J(\Omega_k)$ and the step size halves, otherwise. If the step fails, the domain is deformed again using the smaller step size, and the fluid dynamics problem is recomputed.
Figure 6.7 shows the evolution of the objective functionals (3.5) and the step sizes for all three methods. The functional value is normalized with the value at the initial shape $J(\Omega_0)$. A stopping criterion for the shape optimization problem is if the step becomes too small. Here, the procedure

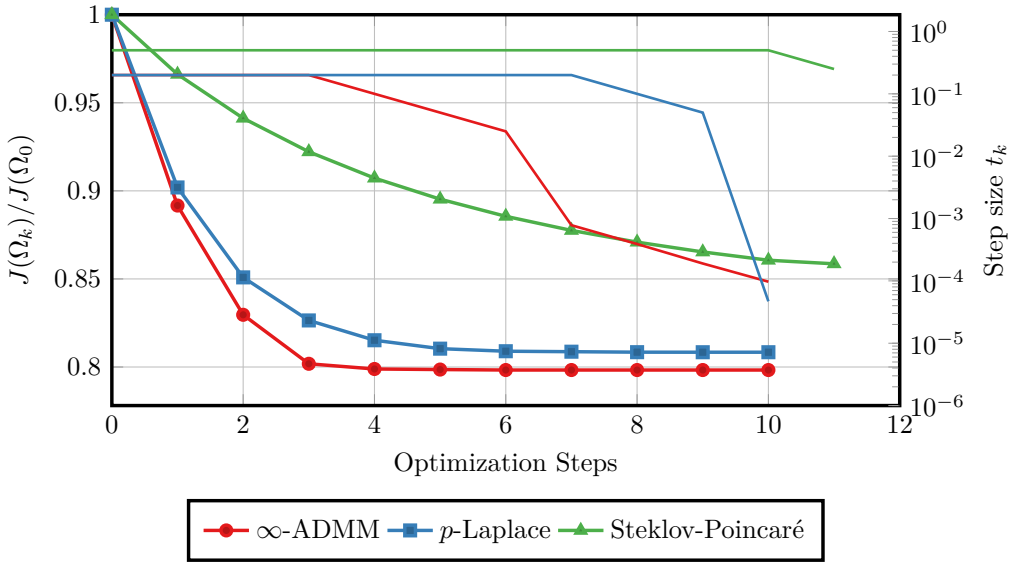


Figure 6.7: Comparison of the ∞-ADMM (red), the p-Laplace approach with $p = 4.1$ (blue) and the Steklov-Poincaré method (green) for the 2D drag reduction of an initial circle at $Re_D = 10$. Lines with symbols refer to the evolution of the normalized objective function. Solid lines without symbols depict the corresponding step size evolution.

stops if $t_k \leq 1 \cdot 10^{-4}$. The ∞-ADMM and the p-Laplace approach require 11 additional Navier-Stokes solves to reach this criterion. As the experiments in Chapters 3 and 4 also show, the Steklov-Poincaré approach, which here implements a Hilbert space method, fails due to grid quality issues before reaching the optimal shape. The issue is discussed below in more detail. Note that this work intentionally does not consider generating a new mesh for the shape when the computation breaks down. Besides the step size, the directional derivative $J'(\Omega_k)V^k$ and the symmetric shape difference $d(\Omega_0, \Omega_k)$, with $d(\cdot,\cdot)$ defined in (3.19) and (3.20), are reasonable measures to identify convergence of the shape optimization problem. Figure 6.8 shows the directional derivatives for the different descent approaches. The value of $J'(\Omega_k)V^k$ of the ∞-ADMM and p-Laplace solution are significantly smaller than for the Steklov-Poincaré approach. The derivative of the ∞-ADMM method is slightly smaller than the p-Laplace for the first steps. Likewise, the symmetric shape difference, previously defined in (3.19), (3.20) and (5.23), may be used to identify up to which step the changes of the shapes are significant. The shape difference in Figure 6.8 converges within narrow limits using the ∞-ADMM and p-Laplace descent directions, although the ∞-ADMM solution displays a slightly improved initial convergence. The optimization strategy should affect the mesh quality as little as possible. The influence of the descent direction on the grid quality is compared by using the cell aspect ratio and the minimal interior cell angle. Figure 6.10 depicts the cell aspect rations of the initial and final mesh obtained using the ∞-ADMM and p-Laplace descent directions. Both strategies result in grids, which only feature cells with high aspect ratios close to the obstacle boundary. Respective deviations from unity are mainly inherited from the wall-normal refinement of the initial mesh and differences between the two displayed methods are negligible. Figure 6.11 shows the corresponding comparison for the minimum internal angle of a control volume (grid orthogonality). Ideally, the internal angle of the quadrilaterals should be 90 degrees, and values below 20 degrees should be avoided. Distorted cells are mainly clustered in the tip regions, where local deformations are most pronounced. It is seen that the ∞-ADMM features less spreading of unfavorable internal angles from the deformed boundary into the domain. A closer inspection of the mesh section displayed in Figure 6.12 confirms these observations. Table 6.2

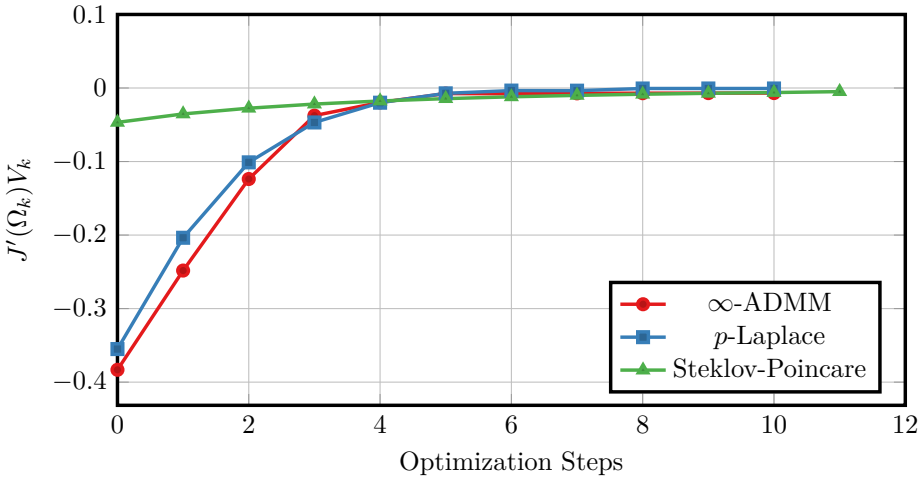


Figure 6.8: Comparison of directional derivatives $J'(\Omega_k)V^k$ for the 2D drag optimization of an initial circle at $Re_D = 10$ using the ∞-ADMM, p-Laplace with $p = 4.1$ and Steklov-Poincaré approach.

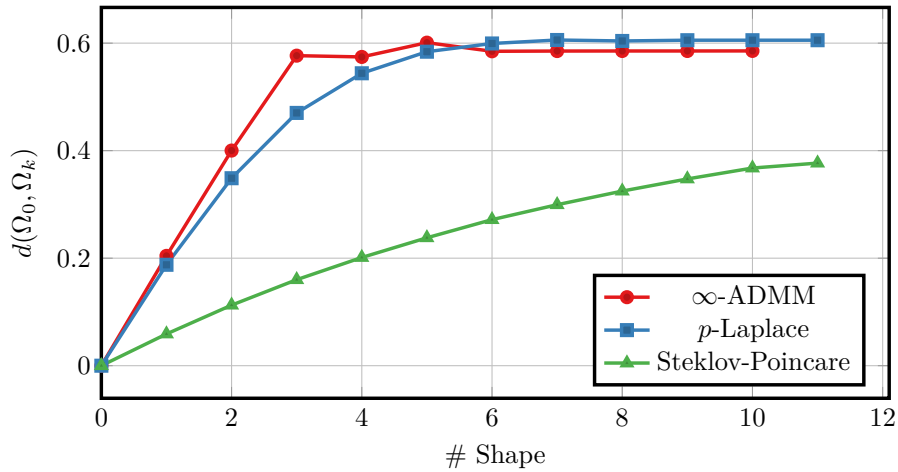


Figure 6.9: Comparison of the symmetric shape distance for the 2D drag optimization of an initial circle at $Re_D = 10$ using the ∞-ADMM, p-Laplace with $p = 4.1$ and Steklov-Poincaré approach.

Method	Optimization steps	Total time for computing the descent direction	Average time per descent direction
∞-ADMM	10	17 h	1.7 h
p-Laplace with $p = 4.1$	10	6.3 h	38 m
Steklov-Poincaré	11	6 m	33 s

Table 6.2: Comparison of the computational effort associated with the different descent approaches. The number of steps refers to the number of shape updates, which are significant for the ∞-ADMM and the p-Laplace descent direction. For the Steklov-Poincaré approach, the number of steps is the number of shape updates before the procedure broke down due to mesh quality issues.

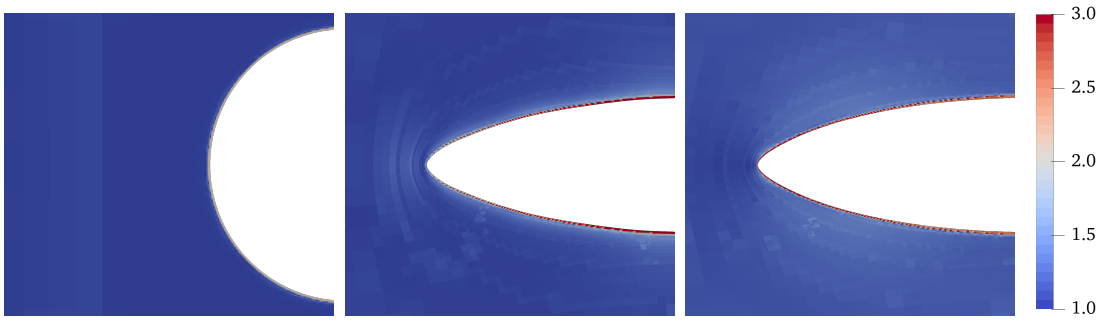


Figure 6.10: Cell aspect ratio of the upstream half of the initial and final shape obtained using the ∞-ADMM and p-Laplace approach with $p = 4.1$ for the 2D drag optimization of an initial circle at $Re_D = 10$.

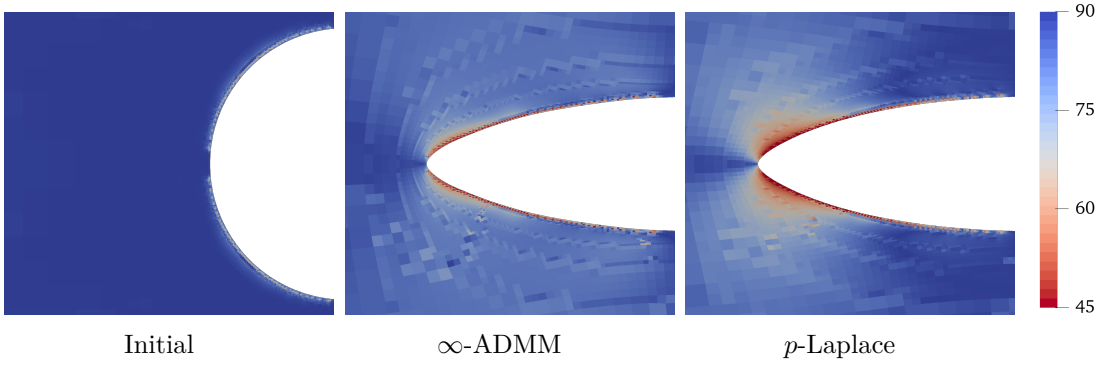


Figure 6.11: Grid orthogonality indicated by the minimum interior cell angle of the initial and final shapes obtained using the ∞-ADMM and p-Laplace approach with $p = 4.1$ for the 2D drag optimization of an initial circle at $Re_D = 10$.

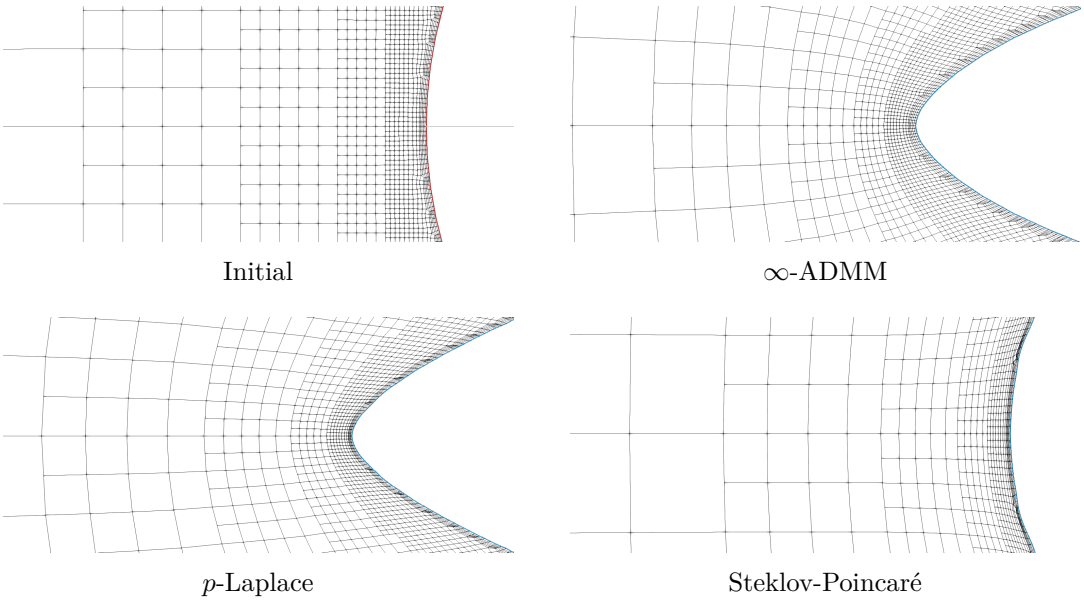


Figure 6.12: Comparison of the front tip and nearby meshes obtained for the 2D drag optimization of an initial circle at $Re_D = 10$. Upper left: Initial configuration. Upper right: Final shape after eight steps using ∞-ADMM for the $W^{1,\infty}$ direction of steepest descent. Lower left: Final shape after 11 steps using the p-Laplace relaxation with $p = 4.1$ as in Chapters 3 and 5. Lower right: Best shape after 11 steps using the Steklov-Poincaré approach as in Chapter 4.

reports the computational effort associated with the three investigated descent approaches. Despite the convergence advantages, the current implementation of the ∞-ADMM method does not lead to a reduction in the overall computational time.

6.5 Conclusions on the ADMM Approximations of the Direction of the Steepest Descent in $W^{1,\infty}$

This chapter discusses a possible approach approximating the direction of the steepest descent in $W^{1,\infty}$-topology using ADMM. The performance of the algorithm is examined for 1D and 2D model problems and compared with the results of the p-Laplace relaxation for a selection of increasing values for $p \geq 2$.
It is observed, that the p-Laplace solutions tend to the $W^{1,\infty}$ solution for increasing p, which is encouraging. Such comparison must be evaluated with caution, since solutions of the p-Laplace problem incorporate evaluations of highly non-linear terms (e.g. terms which are taken to the power of $\frac{p-2}{2}$, and $\frac{p-4}{2}$, cf. (5.13) and (5.18)), and, thus, rounding errors will lead to unreliable numerical solutions for large values of p. At the same time, this limits the range of p-values for the p-Laplace approach and suggests using the ∞-ADMM method instead of large values of p.
The strategy was applied to a 2D fluid dynamic shape optimization problem, which aimed to minimize the drag of an initially circular obstacle at $Re_D = 10$. Unlike the results obtained in Chapters 3 and 5, the optimal shape returned by the ∞-ADMM method did not feature pointy tips at the front and aft. A possible explanation for the lack of pointy tips might be the backtracking line search approach to determine a feasible step size in combination with linearized geometric constraints. When carefully comparing the objectives in Figure 3.3 and Figure 6.7, it is noticeable that the objective, using a fixed step size, slightly increases before decreasing again. Using a step

size approach, which is reduced if the objective does not decrease, as done in this chapter, the optimization stops before the algorithm develops pointy tips. Figure 5.5, however, does not show such behavior, even though a backtracking line search approach for determining the step size is used, but in combination with the non-linear geometric constraints. Thus, the subject of future research may be improving the shape optimization procedure by using ∞-ADMM in combination with the non-linear geometric constraints and the Schur complement method proposed in Chapter 5. In this context, one may further consider the second-order ADMM approach, suggested in [DHH23], when the approach already requires a non-linear solver, such as Newton's method.
The investigation of the ADMM algorithm showed the potential to reduce the number of optimization steps compared to the p-Laplace method and improve the grid quality. However, the computational efforts for the implementation of Algorithm 8, which is used in Section 6.4, is relatively high compared to using the p-Laplace method for moderate p-values. The method can potentially be improved using a different discretization method and, as mentioned in chapter Chapter 5, considering the Schur complement method to handle the geometric constraints.

Chapter 7

Drag Reduction for a Free-Floating Container Vessel in Two-Phase Flow

The final chapter concerns the minimal drag problem in shape optimization of a container ship exposed to turbulent two-phase flows. Attention is directed to the solution of Reynolds-Averaged Navier-Stokes (RANS) equations using the primal/adjoint finite volume method [Run+09; SR11; Küh+22] already employed in Chapters 3, 4 and 6. Central aspects are the use of a p-Laplace relaxation of the direction of the steepest descent (3.16) with $p = 2.6$, in combination with geometric constraints on the center of buoyancy and the water displacement to minimize the hydrodynamic drag of the Kriso container ship (KCS) at $Re = 1.4 \cdot 10^7$ and $Fn = 0.26$.
The considered application contains geometric constraints depending on the geometry Ω and the state variable y. Thus, the approach is advanced towards a more realistic scenario. In Chapters 5 and 6, the geometric constraints do not depend on the state, which allows separating the constraints from the remaining PDE-constrained shape optimization problem.
Results of the presented approach aim to display the general applicability of the optimization procedure illustrated in Chapter 3. They were compiled in parallel with the advancements on constraint handling outlined in Chapter 5 and on solutions of the p-Laplace problem in Chapter 6 and therefore do include them.

7.1 Mathematical Model

The problem considered here has the general form (5.1). Here, however, the geometric constraints g also depend on the state y, that is, with $(y, \Omega) \mapsto g : Y \times \mathcal{S} \to \mathbb{R}^m$, and the domain Ω is admissible if $g(y, \Omega) = 0$.
The hydrodynamic problem for the state constraint $e(y, \Omega) = 0$ is described in greater detail in Section 7.2. Recall the associated Lagrange function

$$L(y, V, z, \lambda) := \mathcal{J}(y, \Omega_V) + \langle z, e(y, \Omega_V)\rangle + \lambda^\intercal g(y, \Omega_V), \tag{7.1}$$

with the multipliers z and $\lambda = (\lambda_1, \dots, \lambda_m)^\intercal \in \mathbb{R}^m$. The classical augmented Lagrange approach for handling geometric constraints would contain two nested loops, such as Algorithm 1. The shape optimization problem is solved several times with constant values for λ in each iteration, and the update of the multiplier λ is performed after the shape optimization loop converges. Algorithm 10 outlines a modified shape optimization procedure based on the Lagrange multiplier rule. To reduce the related effort in practical applications, (7.2) uses an approximation for λ,

Algorithm 10 Shape optimization procedure

1: **repeat**
2: Compute the state $y = y_\Omega$ with $e(y_\Omega, \Omega) = 0$.
3: Compute the adjoint state $z = z_\Omega$, which satisfies

$$\langle e_y(y_\Omega, \Omega)^* \, z_\Omega, \delta_y \rangle = -\langle \mathcal{J}_y(y_\Omega, \Omega) + \lambda^T g_y(y_\Omega, \Omega), \, \delta_y \rangle \qquad \forall \delta_y \tag{7.2}$$

4: Find a descent direction V by solving the minimization problem

$$\min_{V \in W^{1,\infty}, \, \|DV\| \leq 1} J'(\Omega)V = \min_{V \in W^{1,\infty}, \, \|DV\| \leq 1} \langle \mathcal{J}_V(y_\Omega, \Omega) + e_V(y_\Omega, \Omega)^* z + \lambda^T g_V(y_\Omega, \Omega), \, V \rangle \tag{7.3}$$

with an iterative scheme while successively updating $\lambda \leftarrow \lambda + \varrho \langle g_V(y_\Omega, \Omega), V \rangle$ with a suitable step size $\varrho > 0$.
5: Choose a sufficient step size $\alpha > 0$ such that $J((\mathrm{id} + \alpha V)(\Omega)) \leq J(\Omega)$ holds.
6: Update the shape by applying the transformation $\Omega \leftarrow (\mathrm{id} + \alpha V)(\Omega)$
7: **until** converged

computed in the previous iteration. The update of λ is performed within the sub-optimization problem in (7.3). This allows solving the shape optimization problem only once. However, the algorithm is not guaranteed to converge, but the procedure is stable for the numerical experiments discussed in Section 7.3, and the objective function is reduced. At the same time, the geometric constraints meet within a prescribed tolerance.

Computing the deformation field V in Algorithm 10, Line 4 itself is demanding. This chapter follows the approach, suggested in [DHH23; Mül+21] and used in Chapters 3 and 5, to determine the descent direction by finding $\hat{V} \in W^p := \{V \in W^{1,p}(\Omega, \mathbb{R}^d) : V = 0 \text{ a.e. on } \Gamma \setminus \Gamma_{\mathrm{obs}}\}$, which is a local minimizer of the p-Laplace relaxed problem

$$\min_{V \in W^p} \frac{1}{p} \int_\Omega (\nabla V : \nabla V)^{\frac{p}{2}} \, dx + \langle \mathcal{J}_V(y_\Omega, \Omega) + e_V(y_\Omega, \Omega)^* z + \lambda^T g_V(y_\Omega, \Omega), \, V \rangle. \tag{7.4}$$

7.2 Computational Model

Figure 7.1 shows a sketch of the flow domain with an internal and an outer boundary. The domain is occupied by two immiscible incompressible fluids, i.e., air and water. The outer boundary is subdivided into an inlet part Γ_{in} and an outlet part Γ_{out}. The boundary of the obstacle Γ_{obs} is split into a non-deformable part Γ_{obsD} and a deformable part Γ_{obsN}. The optimization aims at minimal resistance of the obstacle Ω_{obs} by deforming the boundary Γ_{obsN}, and, as in the previous chapters, the transformation vector field is computed on Ω.

On the one hand, the state is given by the velocity $v_\Omega : \mathbb{R}_+ \times \Omega \to \mathbb{R}^d$, the total pressure $p_\Omega : \mathbb{R}_+ \times \Omega \to \mathbb{R}$ and the indicator function/volume concentration $c_\Omega : \mathbb{R}_+ \times \Omega \to [0, 1]$ to distinguish between the two immiscible fluid phases air and water. On the other hand, the additional state variables kinetic turbulent energy $k_\Omega : \mathbb{R}_+ \times \Omega \to \mathbb{R}_+$ and dissipation rate of kinetic turbulent energy $\omega_\Omega : \mathbb{R}_+ \times \Omega \to \mathbb{R}$ are introduced for turbulence modeling, using the Wilcox k-ω turbulence model is considered [Wil98]. The total pressure p_Ω consists of the pressure $\mathfrak{p}_\Omega$ and the hydrostatic pressure $-\rho_\Omega \mathfrak{g} \cdot r$ with the acceleration due to gravity $\mathfrak{g}$ and the position r. For the case sketched in Figure 7.1, the gravitation is pointing in negative x_2-direction, and thus, with the basis vector of unit length $r = e_2$, the total pressure is $p_\Omega = \mathfrak{p}_\Omega - \rho_\Omega \mathfrak{g} x_2$. The variables of the turbulent two-phase

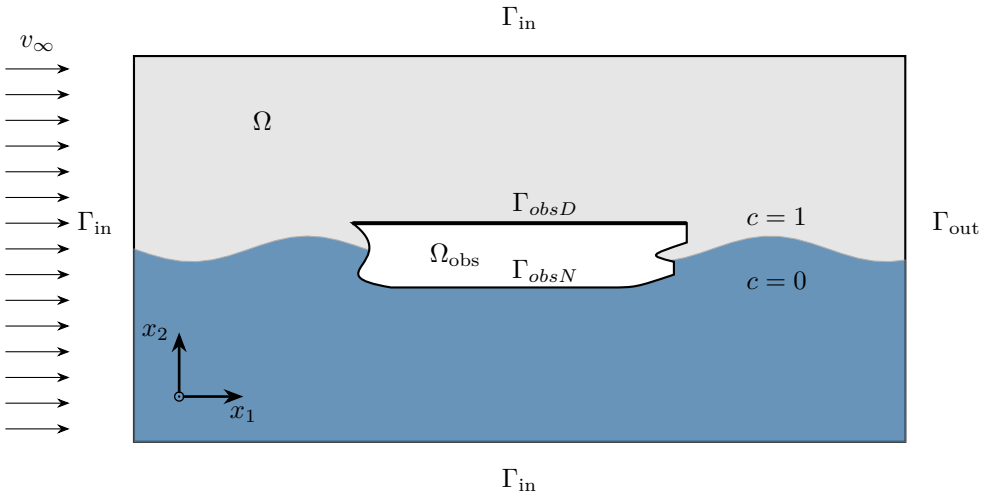


Figure 7.1: Illustration of the flow and obstacle domains and the boundaries. The fluid phase distribution is indicated by the air volume concentration c, i.e., $c = 1[0]$ for air[water] filled regions.

flow satisfy the following RANS equations in a weak sense on the space-time domain $(0, T] \times \Omega$

$$
\begin{aligned}
\operatorname{div}(v_\Omega) &= 0, \\
\frac{\partial(\rho_\Omega v_\Omega)}{\partial t} + \operatorname{div}\left(\rho_\Omega(v_\Omega \otimes v_\Omega) - (\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega})\left(\nabla v_\Omega + \nabla v_\Omega{}^{\intercal}\right)\right) + \nabla p_\Omega - f &= 0, \\
\frac{\partial c_\Omega}{\partial t} + (v_\Omega \cdot \nabla) c_\Omega &= 0, \\
\frac{\partial(\rho_\Omega k_\Omega)}{\partial t} + \operatorname{div}\left(\rho_\Omega v_\Omega k_\Omega - (\mu_\Omega + \sigma_k \frac{\rho k_\Omega}{\omega_\Omega})\nabla k_\Omega\right) - P_\Omega + \beta^* \rho_\Omega k_\Omega \omega_\Omega &= 0, \\
\frac{\partial(\rho_\Omega \omega)}{\partial t} + \operatorname{div}\left(\rho_\Omega v_\Omega \omega_\Omega - (\mu_\Omega + \sigma_\omega \frac{\rho k_\Omega}{\omega_\Omega})\nabla \omega_\Omega\right) - \gamma \frac{\omega_\Omega}{k_\Omega} P_\Omega & \\
+ \beta \rho_\Omega \omega_\Omega^2 - \sigma_d \frac{\rho_\Omega}{\omega_\Omega} \nabla k_\Omega \cdot \nabla \omega_\Omega &= 0
\end{aligned}
\tag{7.5}
$$

where the turbulent production refers to $P_\Omega = \frac{\rho k_\Omega}{\omega_\Omega}\left(\nabla v_\Omega + \nabla v_\Omega{}^{\intercal}\right) : \nabla v_\Omega$ and $f : \mathbb{R}_+ \times \Omega \to \mathbb{R}^d$ is a general source term that does not depend on the state. The parameters $\mu_\Omega, \rho_\Omega > 0$ are the molecular viscosity and density of the respective fluids. They are composed of the bulk properties for the air and water phase, i.e., $\rho_{\text{air}}, \rho_{\text{water}}$ and $\mu_{\text{air}}, \mu_{\text{water}}$, which are considered constant and the linear algebraic equation of state, viz. $\rho_\Omega = \rho_{\text{air}} c_\Omega + \rho_{\text{water}}(1 - c_\Omega)$ and $\mu_\Omega = \mu_{\text{air}} c_\Omega + \mu_{\text{water}}(1 - c_\Omega)$, holds. This finally yields a solenoidal velocity field with $\operatorname{div}(v) = 0$. The system is closed by the

following set of initial and boundary conditions

$$
\begin{aligned}
v_\Omega = 0, \quad \frac{\partial c_\Omega}{\partial n} = 0, \quad \frac{\partial k_\Omega}{\partial n} = 0, \quad \frac{\partial \omega_\Omega}{\partial n} = 0 \qquad & \text{on } \Gamma_{\text{obs}} \\
v_\Omega = v_\infty,\ c_\Omega = c_\infty, \quad k_\Omega = \frac{3}{2}\|v_\infty\|_2^2 \alpha^2,\ \omega_\Omega = \frac{\rho_\Omega k}{\mu_\Omega \nu_t^+} \qquad & \text{on } \Gamma_{\text{in}}, \\
(\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega})\,(\nabla v_\Omega + \nabla v_\Omega{}^\intercal)\cdot n = p_\Omega n,\ \frac{\partial c_\Omega}{\partial n} = 0,\ \frac{\partial k_\Omega}{\partial n} = 0,\ \frac{\partial \omega_O}{\partial n} = 0 \qquad & \text{on } \Gamma_{\text{out}} \\
\text{and} \quad v_\Omega(0) = v_0,\ c_\Omega(0) = c_0,\ k_\Omega(0) = \frac{3}{2}\|v_0\|^2 \alpha^2,\ \omega_\Omega(0) = \frac{\rho_\Omega k}{\mu_\Omega \nu_t^+} \qquad & \text{in } \Omega.
\end{aligned}
\tag{7.6}
$$

The parameters $\sigma_k, \sigma_\omega, \sigma_d, \gamma, \beta, \beta^*, \nu_t^+$ are real valued positive constants of the turbulence model and $\alpha \in (0, 1]$.

Recall the force functional (3.5), but here, considering the turbulent modeling. Following Appendix A, and introducing a sufficiently smooth extension $\Phi : \Omega \to \mathbb{R}^d$ with $\Phi|_{\Gamma_{\text{obs}}} = -e_1$ and $\Phi|_{\Gamma \setminus \Gamma_{\text{obs}}} = 0$, one obtains the equivalent volume formulation of (3.5) with integration by parts [Bra+09, Section 5.1]

$$
\mathcal{J}((v, p, c), \Omega) = \int_\Omega \left[\Big(\operatorname{div}\big(\rho(v \otimes v)\big) - f \Big) \cdot \Phi + (\mu + \frac{\rho k}{\omega})\,(\nabla v + \nabla v^\intercal) : \nabla \Phi - p \operatorname{div}(\Phi) \right] dx\,. \tag{7.7}
$$

Because the focus is on the steady-state resistance, and assuming that the becomes flow stationary if the time interval $[0, T]$, all the terms with time derivatives in (7.5) vanish in the converged state. In practice, using the average of the state variables over a sufficient pseudo-time/iteration period suppresses minor remaining variations of the flow and the objective functional. Different from the geometric constraints in Chapters 3, 5 and 6, preserving the water displacement of Ω_{obs} and the center of buoyancy of the obstacle incorporates the concentration c_Ω and c_{Ω_V}:

$$
\begin{aligned}
g_i(c_\Omega, \Omega_V) := \int_{\Omega_V} (1 - c_{\Omega_V}) x_i \, dx - \int_\Omega (1 - c_\Omega) x_i \, dx = 0, \qquad i = 1, \dots, d \quad \text{and} \\
g_{d+1}(c_\Omega, \Omega_V) := \int_{\Omega_V} (1 - c_{\Omega_V}) \, dx - \int_\Omega (1 - c_\Omega) \, dx = 0.
\end{aligned}
\tag{7.8}
$$

The appearance of the concentration c in (7.8) secures that the displacement of the underwater hull of the vessel (water-wetted part) is preserved rather than the volume of the whole hull. Here, the formulation of the geometric constraints is substantially different from previous works, e.g., [OS21; Mül+23].

This chapter follows the standard practice of neglecting the variables of the turbulence model in the adjoint system because the derivation and the implementation of the corresponding adjoint problem are complicated and error-prone. This simplification is also known as *frozen turbulence assumption* [DB06b; Oth08; SR13], which means that the state variables k and ω are treated as constants when deriving the adjoint equations, and when computing the shape and the deformation field, respectively. Hence, the state variable is considered $y = (v, p, c)$ in the following.

The augmented Lagrange function is defined as

$$\begin{aligned} L((v,p,c),V,(w,q,h),\lambda,\lambda_{\mathrm{BC}},\lambda_{\mathrm{Vol}}) :=\ & \mathcal{J}((v,p,c),\Omega_V) \\ + \int_{\Omega_V} \Big(\operatorname{div}\big(\rho(v\otimes v) - (\mu+\rho\frac{k}{\omega})\,(\nabla v + \nabla v^{\intercal})\,\big) + \nabla p - f\Big)\cdot w\,dx & \\ - \int_{\Omega_V} \operatorname{div}(v)q + \operatorname{div}(vc)h\,dx & \\ + \int_{F(\Gamma_{\mathrm{obs}})} \lambda\cdot v\,ds & \\ + \sum_{i=1}^{d} \lambda_{\mathrm{BC},i}\int_{\Omega_V}(1-c)\,x_i\,dx + \lambda_{\mathrm{Vol}}\int_{\Omega_V}(1-c)\,dx & \end{aligned} \tag{7.9}$$

where $z=(w,q,h)$ is the adjoint state, the multiplier $\lambda\in\mathbb{R}^d$ corresponds to the Dirichlet boundary conditions of the velocity $v=0$ on Γ_{obs}, and $\lambda_{\mathrm{BC}}\in\mathbb{R}^d$ and $\lambda_{\mathrm{Vol}}\in\mathbb{R}$ are associated with the center of buoyancy and displacement constraint. The adjoint state is characterized by the derivative of (7.9) w.r.t. the state $y=(v,p,c)$ which leads to the variational form

$$\begin{aligned} 0 = \int_{\Omega} (\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega})\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal}) : \nabla\delta_v - \rho v_\Omega\cdot(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\cdot\delta_v\,dx & \\ - \int_{\Omega} \operatorname{div}(\delta_v)\,q_\Omega\,dx & \\ - \int_{\Omega} \delta_p\operatorname{div}(w_\Omega)\,dx & \\ + \int_{\Omega} \operatorname{div}(\delta_v c_\Omega)h_\Omega + \operatorname{div}(v_\Omega\delta_c)\,h_\Omega & \\ + \int_{\Omega} \delta_c d_\rho (v_\Omega\otimes v_\Omega) : \nabla w_\Omega\,dx & \\ + \int_{\Omega} \delta_c d_\mu (\nabla w + \nabla w^{\intercal}) : \nabla v\,dx & \\ + \int_{\Gamma_{\mathrm{obs}}} \lambda\cdot\delta_v\,dx & \\ + \sum_{i=1}^{d}\lambda_i\int_{\Omega} -\delta_c\,x_i\,dx + \lambda_{d+1}\int_{\Omega} -\delta_c\,dx & \\ \forall \delta_y = (\delta_v,\delta_p,\delta_c) & \end{aligned} \tag{7.10}$$

where $d_\rho = \rho_{\mathrm{air}} - \rho_{\mathrm{water}}$ and $d_\mu = \mu_{\mathrm{air}} - \mu_{\mathrm{water}}$. The boundary integrals vanish if the boundary conditions

$$\begin{aligned} w &= -\Phi \quad \text{on } \Gamma\setminus\Gamma_{\mathrm{out}}, \quad \text{and} \\ (\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega})\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\cdot n &= (q_\Omega - c_\Omega h_\Omega)n \quad \text{on } \Gamma_{\mathrm{out}}. \end{aligned} \tag{7.11}$$

hold and by choosing

$$\lambda = -(\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega})\,(\nabla w_\Omega + \nabla w_\Omega^{\intercal})\cdot n + (q_\Omega - c_\Omega h_\Omega)n. \tag{7.12}$$

Céa's method is formally applied to derive the directional derivative of the reduced objective $J'(\Omega)V$, see [ADJ21, Section 4.6]. Generally, the shape derivative of an objective function $J(\Omega)$ has a volume and an equivalent surface formulation. The discretization used here requires considering

the surface formulation even though it needs higher regularity of the solutions (v, p, c) and (w, q, h). Utilizing [ADJ21, Theorem 4.2 and 4.3] and assuming that (v, p, c) and (w, q, h) have sufficient regularity one obtains

$$\begin{aligned} J'(\Omega)V = &\int_{\Gamma_{\text{obs}}} \left(-(\mu_\Omega + \frac{\rho k_\Omega}{\omega_\Omega}) \frac{\partial w_\Omega}{\partial n} \cdot \frac{\partial v_\Omega}{\partial n} \right) (V \cdot n)\, dx \\ &+ \sum_{i=1}^{d} \lambda_i \int_{\Gamma_{\text{obs}}} (1 - c_\Omega) x_i\, (V \cdot n)\, dx + \lambda_{d+1} \int_{\Omega_V} (1 - c_\Omega)(V \cdot n)\, dx \end{aligned} \tag{7.13}$$

As mentioned in Chapter 2, the deformation field is obtained from the directional shape derivative by solving the minimization problem (7.4)

$$\min_{V \in W^p} \frac{1}{p} \int_\Omega (DV : DV)^{\frac{p}{2}}\ dx + J'(\Omega)V. \tag{7.14}$$

The Dirichlet condition $V = 0$ is imposed on $\Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ to keep the outer boundary unchanged. In addition, parts of the obstacle may be fixed, and thus $V = 0$ also holds a.e. on Γ_{obsD} and natural boundary conditions hold on Γ_{obsN} where the boundary is deformed.
To approximate the shape deformation field V defined in Algorithm 10, Line 4, the minimization problem (7.14) is solved using the Picard iteration outlined in Algorithm 11.

Algorithm 11 Picard Iteration for Augmented p-Laplacian Problem

1: $\lambda \leftarrow 0$, $p \leftarrow 2$, $V \leftarrow 0$
2: **while** $p < p_{max}$ **do**
3: $\quad k \leftarrow 0$
4: $\quad$ **repeat**
5: $\quad\quad$ Obtain a preliminary $\tilde{V}^k$ by solving the linearized problem

$$\int_\Omega (\nabla V^{k-1} : \nabla V^{k-1})^{\frac{p-2}{2}}\, \nabla \tilde{V}^k : \nabla U\ dx + J'(\Omega)U = 0 \qquad \text{for all } U \tag{7.15}$$

6: $\quad\quad$ Relax update $V^k \leftarrow V^k + \omega(\tilde{V}^k - V^{k-1})$ with $\omega \in (0, 2)$.
7: $\quad\quad$ Update multiplier $\lambda^k \leftarrow \lambda^{k-1} + \tau \langle g_V(\Omega, y), V^k \rangle$ with a suitable step size $\tau > 0$.
8: $\quad\quad k \leftarrow k + 1$
9: $\quad$ **until** $R^k = \left(\|V^k - V^{k-1}\|_{L^2}^2 + \|\lambda^k - \lambda^{k-1}\|_2^2 \right)^{1/2} \leq tol$
10: $\quad p \leftarrow p + p_{inc}$
11: **end while**

7.3 Numerical Results

As mentioned in the introduction to this chapter, results are obtained from the finite volume procedure *FreSCo*$^+$[Run+09; SR11; Küh+22] for the KCS ship in model scale [MEI] at Reynolds and Froude numbers of $Re = 1.43 \cdot 10^7$ and $Fn = 0.26$. Besides the fluid dynamics and problem and the adjoint equations, the variational formulation in Algorithm 11, Line 5 is discretized using the finite volume method (cf. Appendix B). The resulting linear system is solved with the help of the parallel conjugated gradient method outlined in Appendix D.
Figure 7.2 shows the initial configuration with the hull of the KCS and free surface elevation. This chapter investigates two cases to obtain the shape deformation field from (7.14), which differ in the boundary conditions along the hull. The first case considers deforming the whole hull. In the

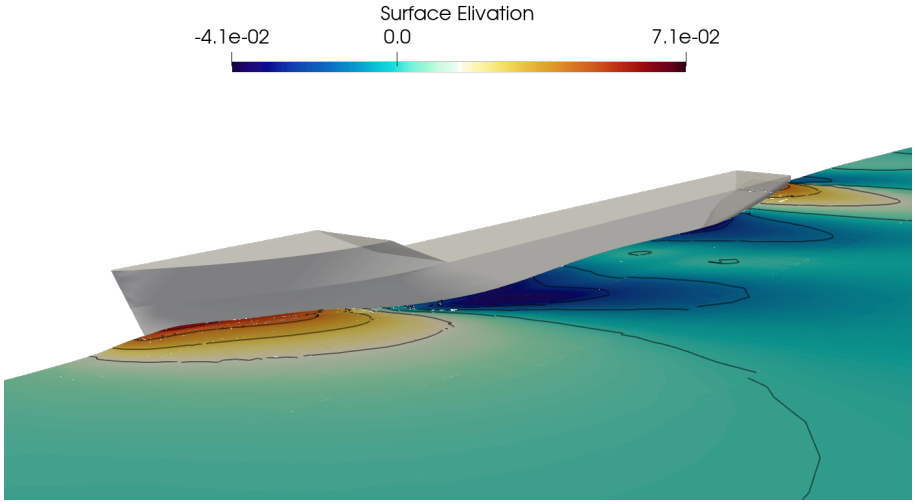
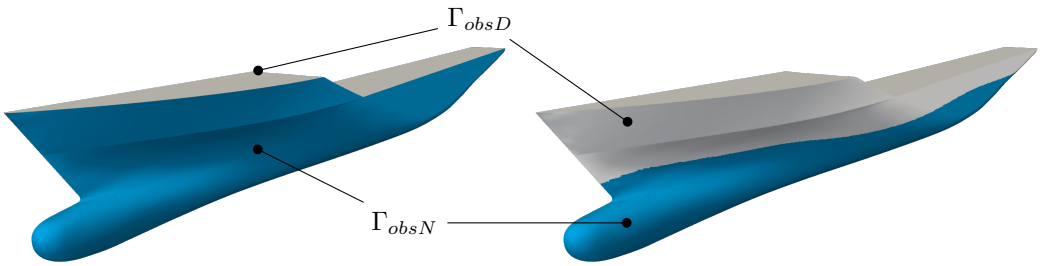


Figure 7.2: Initial hull shape and elevation of the free surface.

second case, the air-wetted part of the hull remains fixed, and only the underwater part Γ_{obsN} of the hull is deformed. In both studies, the deck, the transom, and a part of the propeller shaft remain fixed. For compatibility, both cases use a fixed step size $\alpha = 1/100$, corresponding to a maximum displacement of approximately $1/1000$ of the ship's length.



1^{st} test case. 2^{nd} test case.

Figure 7.3: Boundary type layout for the first test case (left) and second test case (right). The fixed part Γ_{obsD} is colored in light gray and the deformed part Γ_{obsN} in blue. The first test case assigns the complete hull to Γ_{obsN}. The second test case only assigns the underwater hull to Γ_{obsN}, whereas air-wetted hull parts above the water line belong to Γ_{obsD}.

As stated in [DHH23; IL05; Mül+21] the values for p should be significant to obtain a sufficient approximation for a descent direction in $W^{1,\infty}$. However, due to the nonlinearity of (7.14) and the size of the problem, the numerical computation for large values of p is demanding [Loi20]; thus, both cases use a maximum p-value of $p_{max} = 2.6$. For Algorithm 11 to converge for $p > 2$ requires a good initial guess V^0. Therefore, Algorithm 11 iterates over the sequence $p = \{2, 2.3, 2.6\}$ [Mül+21; Mül+23].

To review the convergence of Algorithm 11, Figure 7.4 exemplarily shows the residuals for the first iteration of the shape optimization procedure. The graphs display the individual contributions to the residual R^k from Algorithm 11, Line 9 for the tolerance $tol = 3 \cdot 10^{-5}$. The procedure is stable with the penalty factor $\tau = 10$, and the multipliers converge faster than V^k. Note that because the geometric constraints depend on the primal state, the multipliers $\lambda_{\text{BC}}, \lambda_{\text{Vol}}$, and thus the choice of

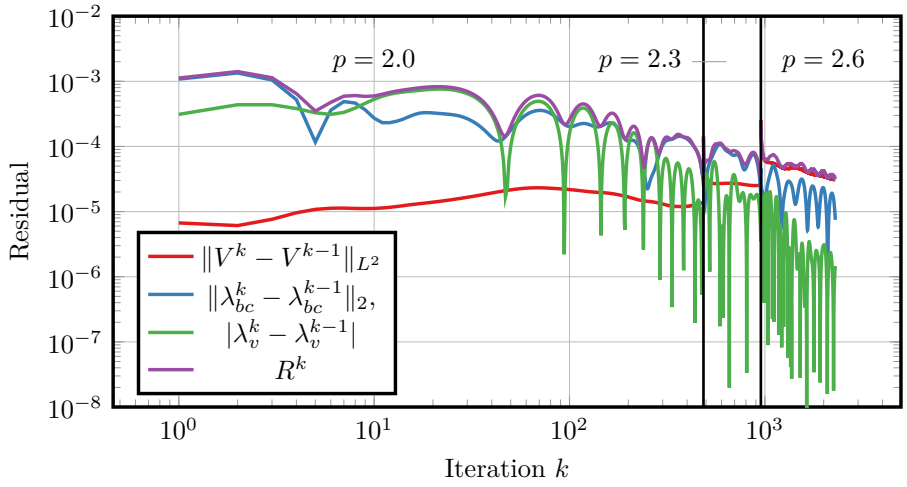


Figure 7.4: Residual plots of the p-Laplacian problem for the sequence of $p = \{2, 2.3, 2.6\}$. Displayed graphs refer to the residuals of the deformation field V, the Lagrangian multipliers $\lambda_{bc} = (\lambda_1, \ldots, \lambda_d)$ and $\lambda_v = \lambda_{d+1}$ as well as the total residual R^k.

the step size τ is particular to this problem.

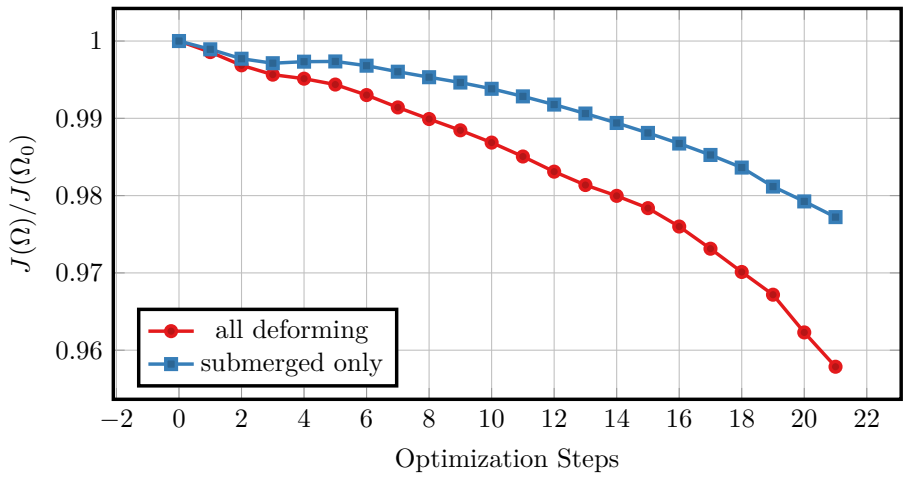


Figure 7.5: Evolution of the normalized drag force objective function obtained with $p = 2.6$ when deforming the whole hull (red line, test case 1) and only the underwater hull (blue line, test case 2).

Figure 7.5 shows the normalized objective function for both cases. It can be observed that the functional values decline faster for the first test case, where the deformation is not limited to the submerged part (solid line). The greater maximum displacement value in the first case might be a possible explanation. The geometric constraints in the second case are more restrictive and might result in smaller values of the displacement fields. One could choose a greater step size for the second case, but the strategy for determining a valid step size might be computationally demanding. However, this is simultaneous with large deformations at the intersection of the hull and the deck, particularly in the bow regime, cf. Figure 7.6. The magnification in Figure 7.6 points out that locally vanishing cell volumes after 22 iterations occur in this regime, and the

simulation terminate. The issue only occurs for the first test case, where the whole vessel can

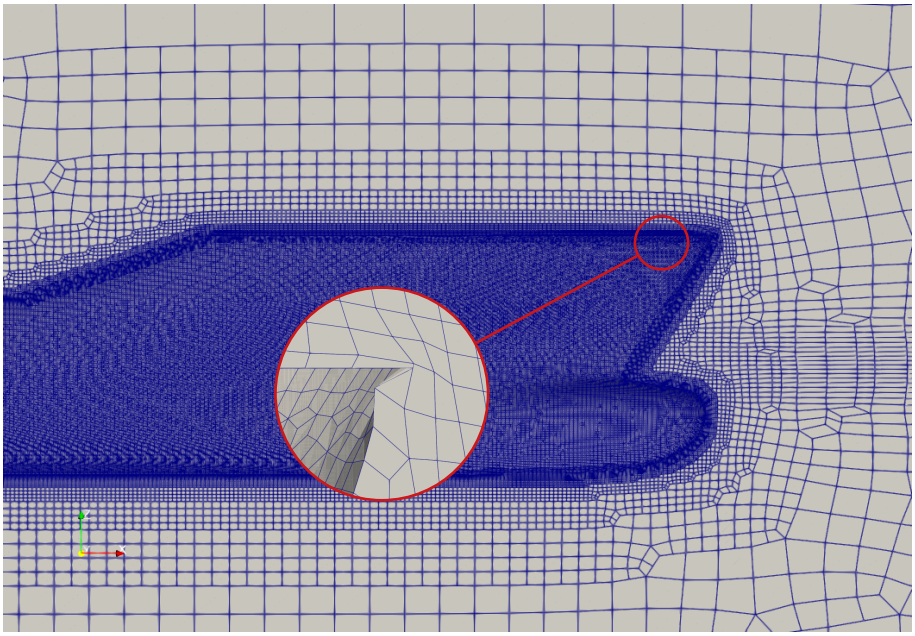


Figure 7.6: Illustration of the experienced grid deterioration in the bow region of the KCS container vessel after 22 design iterations in combination with the first test case, where both the water and air wetted hull sections are deformed (cf. Figure 7.3).

deform. Using the second approach, the deformation is confined to the water-wetted surface, and one could perform further iterations. Figure 7.7 compares the body plans of the initial (black) and the two modified designs of the 22^{nd} iteration. The two strategies predict virtually the same underwater hull deformations. However, differences occur when the free surface is approached, and more pronounced deformations are experienced in the first case, where the whole vessel can deform. Moreover, differences also occur in the bow regime, where the submerged-only design (blue) predicts a stronger displacement in the upper part, as indicated by the magnification of the section lines close to the bow in Figure 7.7.

7.4 Summary of the Container Ship Application

The chapter presents an algorithmic approach for fluid dynamic shape optimization of floating ships exposed to turbulent two-phase flows under geometric constraints. The presented algorithm is based on the augmented Lagrange method of multipliers for the geometric constraints, and the PDE constraints are treated utilizing the corresponding adjoint operator. Simulations were performed for a KRISO Container Ship in model scale at $Re = 1.4 \cdot 10^7$, $Fn = 0.26$ [MEI].
Results show that the suggested approach leads to deformation fields that fulfill the geometric constraints up to a pre-defined tolerance. However, the attainable drag reduction is limited by the degeneration of the computational grid with the successive shape updates. Because the deformation field is computed only on the flow domain and not on the whole domain (i.e., without the obstacle), it is possible for the geometry to overlap. Future research may thus consider a discretization of the entire domain, including the interior of the obstacle. Moreover, the algorithm may be applied to free-floating vessels subjected to rigid-body dynamics, representing a more realistic optimization problem.

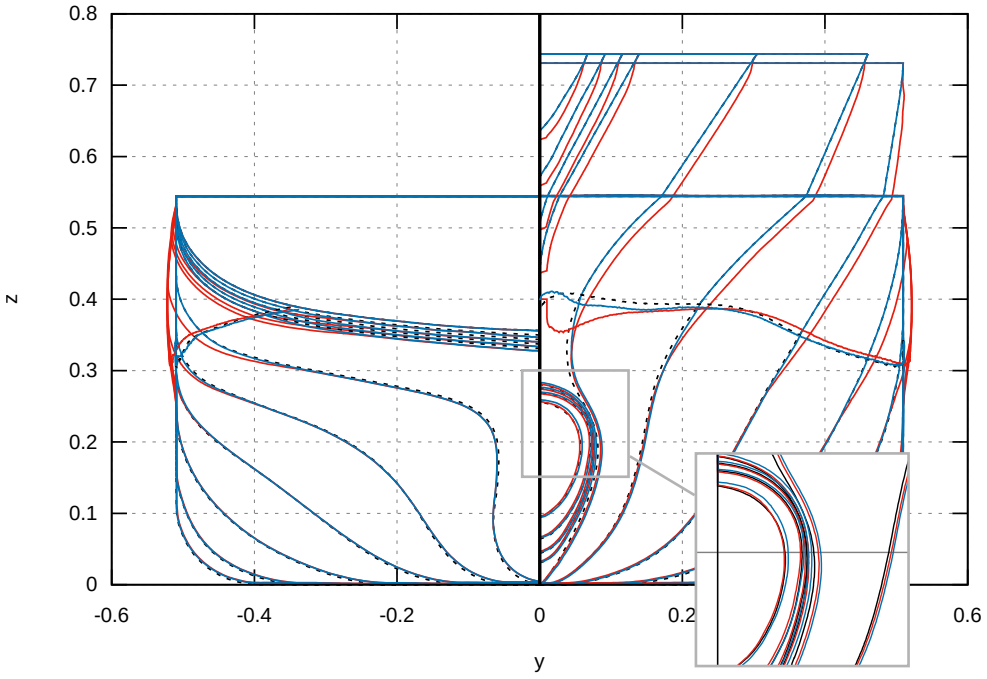


Figure 7.7: Body plan of the initial hull (dashed black line) and the optimized hulls of the 22nd design candidate predicted by the all deforming (red line) and only underwater hull deforming (blue line) approaches.

Chapter 8

Summary and Outlook

This chapter briefly summarizes the findings of the dissertation and addresses interesting possibilities for future work from the author's perspective. As the introduction suggests, smooth shapes may be desirable but not necessarily optimal. Previously published Hilbert-space strategies mainly generate smoothed, i.e., rounded, shape updates and displayed difficulties in capturing optima that feature kinks and corners. The same applies to opposite situations, i.e., converting an initially kinked shape into a curved optimum. Using previous methods, one may only obtain optimal shapes that (almost) resolve such features at the expense of an extremely fine resolution by the computational mesh. Moreover, the respective convergence tends to be slow in an industrial simulation context.

The thesis outlines a viable concept for a node-based (CAD-free) shape transformation approach for optimizing Lipschitz domains. The proposed method is related to the computation of the steepest descent directions of the shape functional in the $W^{1,\infty}$-topology. It is characterized by the fact that the domain topology is retained, while corners and edges might appear or disappear during a mesh-morphing procedure if this supports the optimization. On the one side, the investigated p-Laplace relaxation of the steepest descent direction enables approximating $W^{1,\infty}$ functions with mathematical rigor bases. On the other hand, using the p-Laplace operator in a volume-based formulation supports the mesh-morphing strategy and simplifies the optimization with unstructured, polyhedral meshes using a traditional second-order accurate finite-volume CFD solver.
Various 2D and 3D, laminar and turbulent, internal and external flow examples included in this thesis reveal that using the p-Laplace operator improves the convergence of the optimizer with increasing p. At the same time, the obtained shapes improve regarding the attainable value of the objective function and the ability to return shapes with edges or rapidly smooth sub-optimal initial edges. More importantly, the quality of the computational grid is virtually preserved even when large local deformations of the initial shape occur. Results suggest that $p \approx 4$ seems a sufficiently large p-value to gather most benefits of the p-harmonic approach. While the convergence rate improves with increasing p-values, the iterative effort increases substantially and is thus the subject of subsequent efforts. Moreover, the procedural consideration of technical constraints, which ensures the practical applicability of the optimization results, causes a major challenge. This motivated efforts to reduce the computational effort and, at the same time, increase the robustness of the optimization algorithm. Future work regarding the p-Laplace method may consider more advanced strategies, which avoid iterating over a sequence of p-values.

Hybridization of the p-Laplace and the Steklov-Poincaré (SP) method was suggested to reduce the computational cost while retaining the benefits, The hybrid "SP+p" method outperforms the SP approach in terms of the quality of the optimized mesh but lags behind the p-Laplace method in this regard. At the same time, however, it requires only about half the computational effort of the p-Laplace approach. Despite these reductions, the costs of the hybrid method are still

considerable. This is particularly true if geometrical constraints occur that significantly increase the time to solution.

Alternative ways to incorporate technical constraints on the shape into the optimization problem were investigated to reduce the computational effort and increase the robustness of the optimization algorithm. So far, technical and geometrical constraints, incorporated together with the PDE constraints into the shape optimization problem, are mostly handled using an augmented Lagrange approach or penalty methods in general. Instead, the present thesis suggests understanding the technical constraints as restrictions to the shape transformation. This allows a more efficient treatment by modifying the procedure for computing the descent direction. The resulting saddle point problem was solved iterative or most efficiently using a Schur complement approach.

In addition, a novel strategy was researched for approximating the direction of the steepest descent in $W^{1,\infty}$, based on the alternating direction method of multipliers (ADMM). Instead of solving a non-linear p-Laplace problem for p as large as possible, the aim was to directly approximate directions in $W^{1,\infty}$ rather than $W^{1,p}$. It was seen that p-Laplace solutions tend to the ∞-ADMM solution when p is increased. Moreover, the ∞-ADMM method displays a faster convergence of the optimizer, improves the grid quality, and reveals numerical benefits for great values of p by reducing the influence of rounding errors and limited machine precision. However, simulation efforts remain an aspect to be improved in future research. Nevertheless, the research results show the great potential for the ADMM-based $W^{1,\infty}$ method as an adequate descent method for fluid dynamic shape optimization problems.
Continued research might consider improving the ∞-ADMM approach to reduce computational efforts. A straightforward advancement would be using the Schur complement approach from Chapter 5 for the constraint Poisson problem, solved in each iteration of the ADMM algorithm. Moreover, considering other discretization and solution strategies dedicated to fast Poisson solvers may further reduce the computational time; for example, multigrid methods, like being used for the scalable constrained p-Laplace solver in this thesis, or boundary element methods. The ADMM-based approach for fluid dynamic shape optimization could possibly improve further considering a second-order method, as suggested in [DHH23]. Besides the investigated methods in this thesis, another method for approximating the $W^{1,\infty}$ solution could be based on a Moreau-Yosida type regularization.

Finally, the applicability of implemented concepts was demonstrated using unstructured mesh configurations for a two-phase flow drag-reduction example of a container vessel. Using the investigated descent method for shape optimization problems of maritime applications requires further improvement to advance the physical model. For example, when optimizing a free-floating ship, one must consider rigid body motions during the shape optimization problem, even in calm water conditions. In addition, it would be desirable to consider sea states and random weather conditions, as a ship usually operates only for a short period in calm water conditions.

Appendix A

Shape Derivative of the Drag Functional

Computing the shape derivative of the drag functional requires the derivative of the functional determinant $\det(I + DV)$, which is computed first. Let us first note that the perturbation of the identity matrix $I \in \mathbb{N}^{d\times d}$, with a matrix $\varepsilon \in \mathbb{R}^{d\times d}$ has a determinant of the form

$$\det(I + \varepsilon) = 1 + \operatorname{tr}(\varepsilon) + o(\varepsilon), \tag{A.1}$$

where $\operatorname{tr}(\cdot)$ is the trace of matrix and $o(\varepsilon)$ denotes higher order terms. Setting $j(U) := \det(I+DU)$ and with $j(U+V) = \det(I + D(U+V))$, the Frechét derivative is compuded using the definition

$$|j(U+V) - j(U) - j'(U)V| \leq |o(V)|. \tag{A.2}$$

Using the fact that the determinant is a multiplicative map and (A.1), it follows that

$$\begin{aligned} j(U+V) = \det(I + D(U+V)) &= \det\left(I + DU + DV\right) \\ &= \det\Big((I+DU)\Big((I + (I+DU)^{-1}DV\Big)\Big) \\ &= \det(I+DU)\det\Big(I + (I+DU)^{-1}DV\Big) \\ = \det(I+DU)\left(1 + \operatorname{tr}\Big((I+DU)^{-1}DV\Big) + o((I+DU)^{-1}DV)\right)&. \end{aligned} \tag{A.3}$$

Inserting the above expression in the definition (A.2) reveals that

$$j'(U)V = \det(I+DU)\operatorname{tr}\left((I+DU)^{-1}DV\right). \tag{A.4}$$

The shape derivative of the objective J' often is evaluated on the reference domain, that is, at $U = 0$; Then, the expression simplifies to

$$j'(0)V = \operatorname{tr}(DV) = \operatorname{div}(V). \tag{A.5}$$

Recall the geometric configuration from Figure 3.1 and the drag functional from (3.5)

$$\mathcal{J}((v_\Omega, p_\Omega), \Omega) = -v_\infty/\|v_\infty\| \cdot \int_{\Gamma_{\text{des}}} \nu\left(\nabla v_\Omega + \nabla v_\Omega{}^\intercal\right)\cdot n - p_\Omega n\, ds.$$

where $\Gamma_{\text{des}} \subseteq \Gamma_{\text{wall}}$ is the design surface, which is the optimized part of a no-slip wall, and (v_Ω, p_Ω) are the velocity and the volume-specific pressure that satisfy

$$\begin{cases} -\operatorname{div}(v_\Omega) = 0 & \text{in } \Omega, \\ \operatorname{div}\big(v_\Omega \otimes v_\Omega - \nu\left(\nabla v_\Omega + \nabla v_\Omega^\intercal\right) + I p_\Omega\big) = f & \text{in } \Omega, \\ v_\Omega = 0 & \text{on } \Gamma_{\text{wall}}, \\ v_\Omega = v_\infty & \text{on } \Gamma_{\text{in}}, \\ \nu\left(\nabla v_\Omega + \nabla v_\Omega^\intercal\right) = p_\Omega n & \text{on } \Gamma_{\text{out}}, \end{cases} \tag{A.6}$$

in a weak sense. The drag functional is reformulated to an equivalent volume formulation to avoid using the change of variables formula for surface integrals. It is assumed that the reformulation is feasible for problems in this thesis. Therefore, the vector, pointing in the direction of the approaching flow, is extended to the domain with the help of the smooth auxiliary function $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ for which holds that $\Phi = v_\infty / \|v_\infty\|$ on Γ_{obs} and $\Phi = 0$ on $\Gamma \setminus \Gamma_{\text{obs}}$. Inserting Φ into Using the divergence theorem gives

$$\mathcal{J}((v_\Omega, p_\Omega), \Omega) = -\int_\Omega \nu\left(\nabla v_\Omega + \nabla v_\Omega^\intercal\right) : \nabla\Phi + p_\Omega \operatorname{div}(\Phi) + \big(\operatorname{div}(v_\Omega \otimes v_\Omega) - f\big) \cdot \Phi \, dx. \tag{A.7}$$

Integration by parts of the term $\operatorname{div}(v \otimes v) \cdot \Phi$, that is,

$$\int_\Omega \operatorname{div}(v \otimes v) \cdot \Phi \, dx = \int_\Gamma (v \cdot n)(v \cdot \Phi) \, ds - \int_\Omega (v \otimes v) : \nabla\Phi \, dx. \tag{A.8}$$

and using that $\Phi \neq 0$ only on Γ_{obs} and that $v = 0$ on Γ_{obs}, the boundary integral disappears and one obtains

$$\mathcal{J}((v_\Omega, p_\Omega), \Omega) = \int_\Omega \big(v_\Omega \otimes v_\Omega - \nu\left(\nabla v_\Omega + \nabla v_\Omega^\intercal\right)\big) : \nabla\Phi - p_\Omega \operatorname{div}(\Phi) + f \cdot \Phi \, dx. \tag{A.9}$$

The associated Lagrange function is defined as

$$\begin{aligned} L((v,p), V, (w,q)) := &\int_{\Omega_V} \big(v \otimes v - \nu\left(\nabla v + \nabla v^\intercal\right)\big) : \nabla\Phi - p \operatorname{div}(\Phi) + f \cdot \Phi \, dx \\ &+ \int_{\Omega_V} \big(v \otimes v - \nu\left(\nabla v + \nabla v^\intercal\right) : \nabla w - p \operatorname{div}(w) + f \cdot w \, dx \\ &\int_{\Omega_V} -\operatorname{div}(v)\, q \, dx \end{aligned} \tag{A.10}$$

or in a more compact form

$$\begin{aligned} L((v,p), V, (w,q)) := \int_{\Omega_V} \big(v \otimes v - \nu\left(\nabla v + \nabla v^\intercal\right)\big) &: \nabla(\Phi + w) + p \operatorname{div}(\Phi + w) \\ &+ f \cdot (\Phi + w) - \operatorname{div}(v)\, q \, dx \end{aligned} \tag{A.11}$$

From (2.30) and (2.31) follows the variational formulation, which characterizes the adjoint state: Find an adjoint state $(w, q) = (w_\Omega, q_\Omega)$ such that

$$\begin{aligned} &\langle L_v((v_\Omega, p_\Omega), V, (w_\Omega, q_\Omega)), \delta_v \rangle + \langle L_p((v_\Omega, p_\Omega), V, (w_\Omega, q_\Omega)), \delta_p \rangle = \\ &\int_\Omega \big(\delta_v \otimes v_\Omega + v_\Omega \otimes \delta_v - \nu\left(\nabla \delta_v + \nabla \delta_v^\intercal\right)\big) : \nabla(\Phi + w_\Omega) - \delta_p \operatorname{div}(\Phi + w_\Omega) \\ &\qquad -\operatorname{div}(\delta_v)\, q_\Omega \, dx = 0 \end{aligned} \tag{A.12}$$

holds for all test functions δ_v and δ_p. Integration by parts and using the identity $(\delta_v \otimes v_\Omega + v_\Omega \otimes \delta_v) : \nabla w_\Omega = \delta_v \cdot (\nabla w_\Omega + \nabla w_\Omega{}^\intercal) \cdot v_\Omega$ one obtains the adjoint systems

$$\begin{cases} -\operatorname{div}(w_\Omega) = 0 & \text{in } \Omega, \\ -(\nabla w_\Omega + \nabla w_\Omega{}^\intercal) \cdot v_\Omega - \operatorname{div}\big(\nu\,(\nabla w_\Omega + \nabla w_\Omega{}^\intercal)\big) = -\nabla q_\Omega & \text{in } \Omega \\ w_\Omega = \Phi & \text{on } \Gamma \setminus \Gamma_{\text{out}}, \\ \nu\,(\nabla w_\Omega + \nabla w_\Omega{}^\intercal) \cdot n = q_\Omega n & \text{on } \Gamma_{\text{out}}, \end{cases} \tag{A.13}$$

where (w_Ω, q_Ω) are solutions in a weak sense. Setting $(v, p) = (v_{\Omega_V}, p_{\Omega_V})$, and $(w, q) = (w_{\Omega_V}, q_{\Omega_V})$ and with $\Phi + w_{\Omega_V} =: \tilde{w}_{\Omega_V}$ the Lagrange function over the transformed domain is

$$\begin{aligned} L((v_{\Omega_V}, p_{\Omega_V}), V, (\tilde{w}_{\Omega_V}, q_{\Omega_V})) = \\ \int_{\Omega_V} \big(v_{\Omega_V} \otimes v_{\Omega_V} - \nu\,(\nabla v_{\Omega_V} + \nabla v_{\Omega_V}{}^\intercal)\big) : \nabla \tilde{w}_{\Omega_V}\, dx \\ - \int_{\Omega_V} p_{\Omega_V} \operatorname{div}(\tilde{w}_{\Omega_V})\, dx \\ - \int_{\Omega_V} \operatorname{div}(v_{\Omega_V})\, q_{\Omega_V}\, dx \\ - \int_{\Omega_V} f \cdot \tilde{w}_{\Omega_V}\, dx, \end{aligned} \tag{A.14}$$

where $(v_{\Omega_V}, p_{\Omega_V})$ and $(w_{\Omega_V}, q_{\Omega_V})$ are weak solutions to (A.6) and (A.13) on Ω_V. In the following step, the above integral is pulled back to the reference domain Ω:

$$\begin{aligned} L(((v_{\Omega_V}, p_{\Omega_V}) \circ F), V, ((\tilde{w}_{\Omega_V}, q_{\Omega_V}) \circ F)) = \\ \int_\Omega \Big((v_{\Omega_V} \circ F) \otimes (v_{\Omega_V} \circ F) - \nu(\nabla v_{\Omega_V}) \circ F + (\nabla v_{\Omega_V})^\intercal \circ F\Big) : \big((\nabla \tilde{w}_{\Omega_V}) \circ F\big) j(V)\, dx \\ - \int_\Omega (p_{\Omega_V} \circ F)\, (\operatorname{div}(\tilde{w}_{\Omega_V}) \circ F) j(V)\, dx \\ - \int_\Omega (\operatorname{div}(v_{\Omega_V}) \circ F)\, (q_{\Omega_V} \circ F) j(V)\, dx \\ - \int_\Omega (f \circ F) \cdot ((\tilde{w}_{\Omega_V}) \circ F) j(V)\, dx \end{aligned} \tag{A.15}$$

where $j(V)$ is the functional determinant $\det(I + DV)$, and, e.g., $(v_{\Omega_V}, p_{\Omega_V}) \circ F = (v_{\Omega_V} \circ F, p_{\Omega_V} \circ F)$, cf. the change of variables formula (2.8). Assuming that $DF = I + DV$ is invertible it follows from the chain rule that $(\nabla v_{\Omega_V}) \circ F = DF^{-\intercal} \nabla (v_{\Omega_V} \circ F)$. With this, and using the identity $\operatorname{div}(V) = \operatorname{tr}(DV)$ the above expression can be reformulated as follows:

$$\begin{aligned} L(((v_{\Omega_V}, p_{\Omega_V}) \circ F), V, ((\tilde{w}_{\Omega_V}, q_{\Omega_V}) \circ F)) = \\ \int_\Omega \Big(B(V)^\intercal ((v_{\Omega_V} \circ F) \otimes (v_{\Omega_V} \circ F)) - \nu A(V)\, (\nabla (v_{\Omega_V} \circ F) + \nabla (v_{\Omega_V} \circ F)^\intercal)\Big) : \nabla (\tilde{w}_{\Omega_V} \circ F)\, dx \\ - \int_\Omega (p_{\Omega_V} \circ F) \operatorname{tr}(B(V) \nabla \tilde{w}_{\Omega_V})\, dx \\ - \int_\Omega \operatorname{tr}(B(V) \nabla v_{\Omega_V})\, (q_{\Omega_V} \circ F)\, dx \\ - \int_\Omega (f \circ F)\big) \cdot ((\tilde{w}_{\Omega_V}) \circ F) j(V)\, dx, \end{aligned} \tag{A.16}$$

with the abbreviations

$$A(V) = j(V)DF^{-1}DF^{-\intercal}, \quad \text{and} \quad B(V) = j(V)DF^{-\intercal}. \tag{A.17}$$

According to (2.32) the shape derivative $J'(\Omega)V$ is obtained by taking the derivative of $L((v,p),V,(\tilde{w},q))$ with respect to V evaluated at $V = 0$ while (v,p) and (w,q) remain fixed. Thus, taking the derivative of

$$\begin{aligned} &L((v,p),V,(\tilde{w},q)) = \\ &\int_\Omega \Big(B(V)^\intercal (v \otimes v) - \nu A(V) \left(\nabla v + \nabla v^\intercal\right) \Big) : \nabla \tilde{w} \, dx \\ &\qquad - \int_\Omega p \operatorname{tr}(B(V)\nabla \tilde{w}) \, dx \\ &\qquad - \int_\Omega \operatorname{tr}(B(V)\nabla v) \, q \, dx \\ &\qquad - \int_\Omega (f \circ F)) \cdot \tilde{w} j(V) \, dx, \end{aligned} \tag{A.18}$$

breaks down to computing $A'(V)$ and $B'(V)$ and evaluating at $V = 0$, and the derivative of the forcing term follows directly from (2.12). Thus, one finally obtains the volume formulation

$$\begin{aligned} &J'(\Omega)V = L_V((v,p),0,(w,q))V = \\ &\int_\Omega \Big(B'(0)^\intercal (v \otimes v) - \nu A'(0) \left(\nabla v + \nabla v^\intercal\right) \Big) : \nabla w \, dx \\ &\qquad - \int_\Omega p \operatorname{tr}(B'(0)\nabla \tilde{w}) \, dx \\ &\qquad - \int_\Omega \operatorname{tr}(B'(0)\nabla v) \, q \, dx \\ &\qquad - \int_\Omega (\nabla f \cdot V - f \operatorname{div}(V)) \cdot \tilde{w} \, dx \end{aligned} \tag{A.19}$$

with

$$A'(0) = \operatorname{div}(V)I - DV - DV^\intercal, \quad \text{and} \quad B'(0) = \operatorname{div}(V)I - DV^\intercal \tag{A.20}$$

using the identities from (5.12).

Surface Formulation of the Drag Functional

The above volume formulation of the shape derivative can be reformulated to obtain an equivalent surface formulation, like in (2.17), if the boundary Γ is sufficiently regular, cf. [DZ11; HP18; ADJ21]. In the following, it is assumed that the boundary Γ, the state (v,p), and the adjoint state (w,q) fulfill the regularity requirements such that all performed operations are permitted, and all symbols exist. Essentially, the individual terms in (A.19) are rearranged such that the non-conservative terms of the volume integral disappear. Integration by parts leads to a formulation with only surface integrals left. By inserting the expressions in (A.20) in (A.19) and reordering

the terms the volume formulation reads

$$\begin{aligned} J'(\Omega)V = &\int_\Omega \Big[\Big(\big(v\otimes v - \nu\left(\nabla v + \nabla v^\intercal\right)\big) : \nabla\tilde{w} - p\,\mathrm{div}(\tilde{w}) - f\cdot\tilde{w} - \mathrm{div}(v)\,q\Big)\mathrm{div}(V)\Big]\,dx \\ &- \int_\Omega \Big[\Big(v\otimes v - \nu\left(\nabla v + \nabla v^\intercal\right)\Big) : DV^\intercal\nabla\tilde{w} - \nu DV^\intercal\left(\nabla v + \nabla v^\intercal\right) : \nabla\tilde{w}\Big]\,dx \\ &+ \int_\Omega \Big[p\mathrm{tr}(DV^\intercal\nabla\tilde{w})\Big]\,dx \\ &+ \int_\Omega \Big[\mathrm{tr}(DV^\intercal\nabla v)q\Big]\,dx \\ &+ \int_\Omega \Big[\tilde{w}\cdot\nabla f\cdot V\Big]\,dx. \end{aligned} \tag{A.21}$$

For some vector fields s, a, b and V, the identities

$$\int_\Omega \Big[s\mathrm{div}(V)\Big]\,dx = \int_\Gamma \Big[s(V\cdot n)\Big]\,ds - \int_\Omega \Big[\nabla s\cdot V\Big]\,dx \tag{A.22}$$

and

$$\int_\Omega \Big[\nabla V a\cdot b\Big]\,dx = \int_\Gamma \Big[(a\cdot V)(b\cdot n)\Big]\,ds - \int_\Omega \Big[\mathrm{div}(b)(a\cdot V) + b\cdot\nabla a\cdot V\Big]\,dx \tag{A.23}$$

from [ADJ21, Section 4.5] are useful for reformulating the above volume-formulation. The first identity can directly be applied to the first line in (A.21). The remaining terms, however, require a vector-valued version of the second identity. That is, with the matrices $A = (a_1,\dots,a_j,\dots,a_n)$ and $B = (b_1,\dots,b_j,\dots,b_n)$ and where a_j and b_j are the j-th column of A and B, the double inner product for matrices can be written as a sum of vector inner products. Thus, the second identity reads

$$\begin{aligned} \int_\Omega \Big[\nabla V A : B\Big]\,dx &= \sum_{j=1}^n \int_\Omega \Big[\nabla V a_j\cdot b_j\Big]\,dx \\ = \sum_{j=1}^n \int_\Gamma \Big[(a_j\cdot V)(b_j\cdot n)\Big]\,ds &- \int_\Omega \Big[\mathrm{div}(b_j)(a_j\cdot V) + b_j\cdot\nabla a_j\cdot V\Big]\,dx. \end{aligned} \tag{A.24}$$

for the vector-valued case. The first line in (A.21) is of the form $s\mathrm{div}(V)$, and the second line has the structure $DV^\intercal A : B + DV^\intercal B : A$. Reformulating the first line is straightforward. For the second line, one obtains a term of the structure

$$\begin{aligned} \int_\Omega \Big[DV^\intercal A : B + DV^\intercal B : A\Big]\,dx = \\ \sum_{j=1}^d \int_\Gamma \Big[(a_j\cdot V)(b_j\cdot n) + (b_j\cdot V)(a_j\cdot n)\Big]\,ds \\ - \int_\Omega \Big[\mathrm{div}(b_j)(a_j\cdot V) + \mathrm{div}(a_j)(b_j\cdot V) + \nabla(a_j\cdot b_j)\cdot V\Big]\,dx \end{aligned} \tag{A.25}$$

using the identity $\nabla(a_j\cdot b_j) = \nabla a_j\cdot b_j + \nabla b_j\cdot a_j$. Using that $\mathrm{tr}(AB) = B^\intercal : A = A : B^\intercal$, and $\mathrm{div}(Tv) = \mathrm{div}(T)\cdot v + T^\intercal : \nabla v$ for some tensor T and vector field v, the integral in the third line in (A.21), for example, can be rewritten as follows:

$$\int_\Omega \Big[p\mathrm{tr}\big(DV^\intercal\nabla\tilde{w}\big)\Big]\,dx = \int_\Gamma \Big[pV\cdot\nabla\tilde{w}\cdot n\Big]\,ds - \int_\Omega \Big[V\cdot\nabla\tilde{w}\cdot\nabla p + p\,\nabla\big(\mathrm{div}(\tilde{w})\big)\cdot V\Big]\,dx. \tag{A.26}$$

The term in the fourth row is rewritten analog, and the integrand within the last row is replaced by $\nabla(f \cdot \tilde{w}) \cdot V - f \cdot \nabla\tilde{w} \cdot V$. Gathering all the terms and inserting them leads to

$$
\begin{aligned}
J'(\Omega)V = &\int_\Gamma \Big[\Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla\tilde{w} + f \cdot \tilde{w} - p\,\mathrm{div}(\tilde{w}) - \mathrm{div}(v)\,q\Big)(V \cdot n)\Big]\,ds \\
&- \int_\Omega \Big[\nabla\Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla\tilde{w} + f \cdot \tilde{w} - p\,\mathrm{div}(\tilde{w}) - \mathrm{div}(v)\,q\Big) \cdot V\Big]\,dx \\
&- \sum_{j=1}^{d} \int_\Gamma \Big[\Big(\nabla\tilde{w}_j \cdot V\Big)\Big(\Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big)\Big) \cdot n\Big)\Big]\,ds \\
- \int_\Omega \Big[\mathrm{div}\Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big)\Big)\Big(\nabla\tilde{w}_j \cdot V\Big) + \Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big)\Big) \cdot \nabla(\nabla\tilde{w}_j) \cdot V\Big]\,dx \\
&+ \sum_{j=1}^{d} \int_\Gamma \Big[\Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big) \cdot V\Big)\Big(\nabla\tilde{w}_j \cdot n\Big)\Big]\,ds \\
- \int_\Omega \Big[\mathrm{div}(\nabla\tilde{w}_j)\Big(\Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big)\Big) \cdot V\Big) + \nabla\tilde{w}_j \cdot \nabla\Big(v\,v_j - \nu\Big(\nabla v_j + \frac{\partial v}{\partial x_j}\Big)\Big) \cdot V\Big]\,dx \\
&+ \int_\Gamma \Big[pV \cdot \nabla\tilde{w} \cdot n\Big]\,ds \\
&- \int_\Omega \Big[V \cdot \nabla\tilde{w} \cdot \nabla p + p\,\nabla\Big(\mathrm{div}(\tilde{w})\Big) \cdot V\Big]\,dx \\
&+ \int_\Gamma \Big[qV \cdot \nabla v \cdot n\Big]\,ds \\
&- \int_\Omega \Big[V \cdot \nabla v \cdot \nabla q + q\,\nabla\Big(\mathrm{div}(v)\Big) \cdot V\Big]\,dx \\
&+ \int_\Omega \Big[\nabla(f \cdot \tilde{w}) \cdot V - f \cdot \nabla\tilde{w} \cdot V\Big]\,dx.
\end{aligned}
\tag{A.27}
$$

After reordering the terms, writing the terms in a more compact form, and using that $\nabla(p\,\mathrm{div}(\nabla\tilde{w})) =$

$\nabla p \operatorname{div}(\tilde{w}) + p\,\nabla(\operatorname{div}(\tilde{w}))$ and $\nabla(f \cdot \tilde{w}) = \nabla f \cdot \tilde{w} + \nabla \tilde{w} \cdot f$, one obtains an expression

$$
\begin{aligned}
J'(\Omega)V = \int_\Gamma \Big[\Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla \tilde{w} + f \cdot \tilde{w} - p \operatorname{div}(\tilde{w}) - \operatorname{div}(v)\, q \Big)(V \cdot n) \Big]\, ds \\
+ \int_\Gamma \Big[pV \cdot \nabla \tilde{w} \cdot n + qV \cdot \nabla v \cdot n \Big]\, ds \\
- \sum_{j=1}^{d} \int_\Gamma \Big[(\nabla \tilde{w}_j \cdot V) \Big(\Big(v\, v_j - \nu \Big(\nabla v_j + \frac{\partial v}{\partial x_j} \Big) \Big) \cdot n \Big) \Big]\, ds \\
+ \sum_{j=1}^{d} \int_\Gamma \Big[\Big(v\, v_j - \nu \Big(\nabla v_j + \frac{\partial v}{\partial x_j} \Big) \cdot V \Big) (\nabla \tilde{w}_j \cdot n) \Big]\, ds \\
- \int_\Omega \Big[\nabla \Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla \tilde{w} + f \cdot \tilde{w} - p \operatorname{div}(\tilde{w}) - \operatorname{div}(v)\, q \Big) \cdot V \Big]\, dx \\
+ \int_\Omega \Big[\nabla \Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla \tilde{w} + f \cdot \tilde{w} - p \operatorname{div}(\tilde{w}) - \operatorname{div}(v)\, q \Big) \Big]\, dx \\
+ \int_\Omega \Big[\Big((\nabla \tilde{w} + \nabla \tilde{w}^\intercal) \cdot v + \operatorname{div}(\nabla \tilde{w} + \nabla \tilde{w}^\intercal) - \nabla q \Big) \cdot \Big(\Big(v \otimes v - \nu\,(\nabla v + \nabla v^\intercal) \Big) \cdot V \Big) \Big]\, dx \\
+ \int_\Omega \Big[\operatorname{div}(\tilde{w}) (\nabla p \cdot V) \Big]\, dx \\
- \int_\Omega \Big[\Big(\operatorname{div} \Big(v \otimes v - \nu\,(\nabla v + \nabla v^\intercal) \Big) + \nabla p - f \Big) \cdot (\nabla \tilde{w} \cdot V) \Big]\, dx \\
+ \int_\Omega \Big[\operatorname{div}(v)\, (\nabla q \cdot V) \Big]\, dx,
\end{aligned}
\tag{A.28}
$$

where the volume integrals in the last four lines of the equation disappear. Thus the remaining terms are

$$
\begin{aligned}
J'(\Omega)V = \int_\Gamma \Big[\Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) : \nabla \tilde{w} + f \cdot \tilde{w} - p \operatorname{div}(\tilde{w}) - \operatorname{div}(v)\, q \Big)(V \cdot n) \Big]\, ds \\
+ \int_\Gamma \Big[pV \cdot \nabla \tilde{w}^\intercal \cdot n + qV \cdot \nabla v^\intercal \cdot n \Big]\, ds \\
- \int_\Gamma \Big[(\nabla \tilde{w} \cdot V) \cdot \Big((v \otimes v - \nu\,(\nabla v + \nabla v^\intercal)) \cdot n \Big) \Big]\, ds \\
+ \int_\Gamma \Big[\Big(v \otimes v - \nu\,(\nabla v + \nabla v^\intercal) \cdot V \Big) \cdot (\nabla \tilde{w} \cdot n) \Big]\, ds
\end{aligned}
\tag{A.29}
$$

To further simplify the above expression, one must consider the boundary conditions of the primal and adjoint variables and for the descent direction V. The descent direction is assumed to be non-zero on the design surface Γ_{des} and disappears on all other parts of the boundary else, i.e. $V = 0$ a.e. on $\Gamma \setminus \Gamma_{\text{des}}$. From $v = 0$ and $\tilde{w} = \Phi + w = 0$ on Γ_{des} follows that $v \otimes v = 0$ and the tangent gradients of v and $\tilde{w}$ disappear, that is,

$$
\nabla_\Gamma v = \nabla v - n \otimes \frac{\partial v}{\partial n} = 0 \quad \text{and} \quad \nabla_\Gamma \tilde{w} = \nabla \tilde{w} - n \otimes \frac{\partial \tilde{w}}{\partial n} = 0, \tag{A.30}
$$

which is also true for the tangent divergence:

$$
\operatorname{div}_\Gamma(v) = \operatorname{div}(v) - n \cdot \frac{\partial v}{\partial n} = 0, \quad \text{and} \quad \operatorname{div}_\Gamma(\tilde{w}) = \operatorname{div}(\tilde{w}) - n \cdot \frac{\partial \tilde{w}}{\partial n} = 0. \tag{A.31}
$$

Therewith, $\nabla v \cdot V = \frac{\partial v}{\partial n}(V \cdot n)$ and $\nabla \tilde{w} \cdot V = \frac{\partial \tilde{w}}{\partial n}(V \cdot n)$ and with

$$\begin{aligned} \frac{\partial \tilde{w}}{\partial n} \cdot (\nabla v + \nabla v^{\intercal}) \cdot n &= (\nabla v + \nabla v^{\intercal}) : \left(n \otimes \frac{\partial \tilde{w}}{\partial n} \right) \\ &= (\nabla v + \nabla v^{\intercal}) : \nabla \tilde{w} = (\nabla \tilde{w} + \nabla \tilde{w}^{\intercal}) : \nabla v \\ = (\nabla \tilde{w} + \nabla \tilde{w}^{\intercal}) : \left(n \otimes \frac{\partial v}{\partial n} \right) &= \frac{\partial v}{\partial n} \cdot (\nabla \tilde{w} + \nabla \tilde{w}^{\intercal}) \cdot n \end{aligned} \tag{A.32}$$

one obtains the final surface formulation

$$\boxed{J'(\Omega)V = -\int_{\Gamma_{\text{des}}} \left[\left(\nu \left(\nabla v + \nabla v^{\intercal} \right) : \nabla w \right) (V \cdot n) \right] ds.} \tag{A.33}$$

Appendix B

Finite Volume Discretization

Typically, the finite volume method is the approximation method of choice for hyperbolic and parabolic linear, quasi-linear, and non-linear equations. However, the method is also appropriate for approximating elliptic and degenerate elliptic equations, like the Poisson problem in Sections 4.1 and 6.1 and the p-Laplace problem in Section 3.1. This chapter discusses the discretization and implementation details for non-linear diffusive and source terms. For details on the finite volume discretization for the Navier-Stokes equations, which also require the discretization of a convective term, see [FPS20]. In this section, the symbols ϕ, $\nabla\phi$, etc. are continuous values while ϕ_i, $\nabla\phi_{ij}$, etc. denote constant mean values, which are defined below. Consider the tow- or 3D domain $\Omega \subset \mathbb{R}^d$, with Lipschitz boundary Γ, which is split in two parts, Γ_D and Γ_N such that $\Gamma = \Gamma_D \cup \Gamma_N$, and the generic problem finding a function $\phi : \Omega \to \mathbb{R}$ that satisfies

$$\begin{cases} -\operatorname{div}(\eta\nabla\phi) = f & \text{in } \Omega, \\ \phi = g_D & \text{on } \Gamma_D, \\ \eta\nabla\phi \cdot n = g_N & \text{on } \Gamma_N, \end{cases} \tag{B.1}$$

where η is a diffusivity, and f is a given source term. In this thesis, the diffusivity also depends on the gradient, that is, $\eta = \eta(\nabla\phi)$, see (3.16). The source term, however, is assumed to be independent of the solution ϕ. The domain is split into a finite number of disjoint control volumes Ω_i, $i = 1, \dots, m_{\text{cv}}$, (polygons if $d = 2$ or polyhedra if $d = 3$). The boundary of the control volume Ω_i is denoted as Γ_i. In the context of finite volumes, the term "cells" is regularly used for the tesselation rather than "elements", which is common for finite element methods. Usually, the cell's boundaries are composed of straight lines (for $d = 2$) or planar surfaces (for $d = 3$), which are called "faces" in the following. By integrating the PDE in (B.1) over the domain Ω, utilizing the linearity of the integral, and applying the divergence theorem, one obtains the expression

$$-\int_\Omega \Big[\operatorname{div}(\eta\nabla\phi) - f\Big] dx = -\sum_{i=1}^{m_{\text{cv}}} \int_{\Gamma_i} \eta\nabla\phi \cdot n \, ds - \int_{\Omega_i} f \, dx = 0. \tag{B.2}$$

The above equation is implied by

$$\int_{\Gamma_i} \eta\nabla\phi \cdot n \, ds - \int_{\Omega_i} f \, dx = 0 \tag{B.3}$$

for each cell Ω_i. The integral over the cell's boundary Γ_i is split into two parts, one that coincides with the Neumann boundary $\Gamma_i \cap \Gamma_N$ and an internal part $\Gamma_i \setminus \Gamma_N$, to take the natural boundary condition $\nabla\phi \cdot n = g_N$ on Γ_N into account. The above expression then reads

$$-\int_{\Gamma_i \setminus \Gamma_N} \eta\nabla u \cdot n \, ds = \int_{\Omega_i} f \, dx + \int_{\Gamma_i \cap \Gamma_N} g_N \, ds. \tag{B.4}$$

The second term on the right-hand side occurs only for cells adjacent to the Neumann boundary and disappears everywhere else. Figure B.1 illustrates a cell adjacent to the boundary Γ_N and shows the naming scheme for the points and faces. In the following step, the integral over Γ_i is

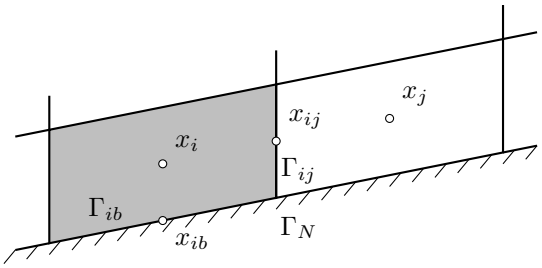


Figure B.1: Considering the i-th cell next to the boundary Γ_N, where x_i is the cell's center point, x_{ij} is the midpoint of the face between the i-the and j-th cell, and x_{ib} is the midpoint of the face that coincides with the boundary Γ_N.

split into a sum of integrals over the faces Γ_{ij}, $j = 1, \dots, m_i - 1$, and $\Gamma_i \cap \Gamma_N$.

$$-\sum_{j=1}^{m_i-1} \int_{\Gamma_{ij}} \eta \nabla u \cdot n \, ds = \int_{\Omega_i} f \, dx + \int_{\Gamma_i \cap \Gamma_N} g_N \, ds. \tag{B.5}$$

The finite volume method approximates mean cell values. Therefore, the following notation is used for the cellwise means of ϕ and f and the mean face value of g_N:

$$\phi_i := \frac{1}{|\Omega_i|} \int_{\Omega_i} \phi \, dx, \quad f_i := \frac{1}{|\Omega_i|} \int_{\Omega_i} f \, dx, \quad \text{and} \quad g_{ib} := \frac{1}{|\Gamma_i \cap \Gamma_N|} \int_{\Gamma_i \cap \Gamma_N} g_N \, ds, \tag{B.6}$$

where $|\Omega_i| := \int_{\Omega_i} 1 \, dx$ is the volume of the control volume Ω_i, $|\Gamma_i \cap \Gamma_N| := \int_{\Gamma_i \cap \Gamma_N} 1 \, ds$ the surface of the boundary face where Neumann data applies. Inserting the definition from (B.6) on the right-hand side of (B.5) does not change the actual equation. The left-hand side, however, requires an expression for the derivative in the normal direction $\nabla u \cdot n$. To explain the approximation strategy, consider the polygon sketched in Figure B.2. The sketch shows the i-th control volume

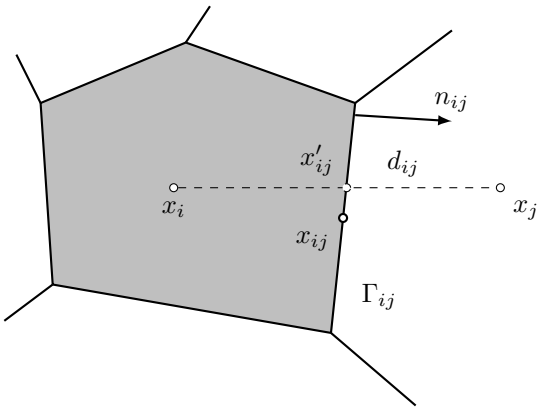


Figure B.2: The i-th polygon cell and it's j-th neighboring cell. S_{ij} is the interface between the two adjacent cells and n_{ij} with $|n_{ij}| = 1$ is the face unit normal vector. d_{ij} denotes the connecting straight line between the cell midpoints x_i and x_j.

and the j-th neighbor cell. The two cells share the interface Γ_{ij}, with surface area $|\Gamma_{ij}|$, the unit

normal vector n_{ij}, and the midpoint of the face x_{ij}. The straight line $d_{ij} = x_j - x_i$ connects the midpoints x_i and x_j of the two neighboring cells, and $x'_{ij} = x_j - \lambda d_{ij}$ is the point where the face and the line d_{ij} intersect, where the interpolation factor is determined as follows:

$$m_{ij}^{\perp} := (x_{ij} - x'_{ij}), \quad 0 = m_{ij}^{\perp} \cdot n_{ij} = (x_{ij} - x'_{ij})\, n_{ij} \overset{(x'_{ij} = x_j - \lambda d_{ij})}{\Longleftrightarrow} \quad \lambda = -\frac{(x_{ij} - x_j) \cdot n_{ij}}{d_{ij} \cdot n_{ij}}. \tag{B.7}$$

Diffusive flux discretization. (B.5) requires the derivative in the direction of the face normal n_{ij}. To approximate the normal derivative $\nabla\phi \cdot n$, in the first step, the exact solution at the cell midpoints, $\phi(x_i)$ and $\phi(x_j)$, is approximated by the cellwise means that is, $\phi(x_i) \approx \phi_i$ and $\phi(x_j) \approx \phi_j$. In the second step, the directional derivative $\nabla\phi \cdot d_{ij}$ is approximated using a simple difference quotient, i.e., $\nabla\phi \cdot d_{ij} \approx \frac{\phi_j - \phi_i}{|d_{ij}|}$, which gives a second-order accurate approximateion of $\nabla\phi \cdot d_{ij}$ at the point x'_{ij}. Because on general grids d_{ij} and n_{ij} are not necessarily parallel, the following differed correction approach accounts for the deviation:

$$\nabla\phi \cdot n = \nabla\phi \cdot \frac{d_{ij}}{|d_{ij}|} - \nabla\phi \cdot \left(\frac{d_{ij}}{|d_{ij}|} - n\right) \approx \underbrace{\frac{\phi_i - \phi_j}{|d_{ij}|}}_{\text{implicit}} - \underbrace{\nabla\phi \cdot \left(\frac{d_{ij}}{|d_{ij}|} - n\right)}_{\text{explicit}}, \tag{B.8}$$

where the first term is implicitly on the left-hand side, and the second is treated explicitly and goes to the right-hand side. Thus, the resulting scheme must be solved iteratively, and from here, the index k denotes the iteration number. The gradient $\nabla\phi = \nabla\phi^{k-1}$ on the right-hand side is computed using the solution ϕ^{k-1}, from the previous iteration. Inserting (B.8) in (B.5) leads to

$$-\sum_{j=1}^{m_i - 1} \int_{\Gamma_{ij}} \eta \frac{\phi_j^k - \phi_i^k}{|d_{ij}|}\, ds = |\Omega_i| f_i + |\Gamma_i \cap \Gamma_N| g_{ib} + \sum_{j=1, j\neq i}^{m_i} |\Gamma_{ij}| F_{ij}^D, \tag{B.9}$$

with the deferred correction term

$$F_{ij}^D = \frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} \nabla\phi^{k-1} \cdot \left(\frac{d_{ij}}{|d_{ij}|} - n\right) ds. \tag{B.10}$$

Because ϕ_i and ϕ_j are constant, the left-hand side can be reformulated as follows:

$$\int_{\Gamma_{ij}} \eta \frac{\phi_j^k - \phi_i^k}{|d_{ij}|}\, ds = \frac{\phi_j^k - \phi_i^k}{|d_{ij}|} \int_{\Gamma_{ij}} \eta\, ds = \frac{\eta_{ij} |\Gamma_{ij}|}{|d_{ij}|} (\phi_j^k - \phi_i^k), \quad j = 1, \ldots, m_i, \tag{B.11}$$

with the mean face value of the diffusivity $\eta_{ij} = \frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} \eta\, ds$. In the non-linear case (e.g. (3.16)), the mean diffusivity is approximated by the diffusivity at the point x'_{ij}:

$$\eta_{ij} \approx \eta'_{ij} = \eta(\nabla\phi'_{ij}) \tag{B.12}$$

using the interpolated value $\nabla\phi_{ij}^{k-1} = \lambda \nabla\phi_i^k + (1 - \lambda)\nabla\phi_j^k$ from the previous iteration, and the interpolation factor λ from (B.7).

Gradient calculation. When assuming piecewise constant solutions, the gradients must be reconstructed from the mean cell values. There are two usual approaches to compute the mean gradient at the cell center x_i. The first approach uses the divergence theorem and the approxima-

tion $\phi'_{ij} \approx \phi_{ij}$:

$$\begin{aligned}\nabla\phi_i &= \frac{1}{|\Omega_i|}\int_{\Omega_i}\nabla\phi\,dx = \frac{1}{|\Omega_i|}\sum_{j=1}^{m_i-1}\int_{\Gamma_{ij}}\phi n\,ds \\ &\approx \frac{1}{|\Omega_i|}\sum_{j=1}^{m_i-1}\phi_{ij}n_{ij}|\Gamma_{ij}| \\ &\approx \frac{1}{|\Omega_i|}\sum_{j=1}^{m_i-1}\phi'_{ij}n_{ij}|\Gamma_{ij}|\end{aligned} \tag{B.13}$$

The second approach solves the minimization problem

$$\min_{\nabla\phi_i}\| \begin{pmatrix}\vdots\\(x_j - x_i)\\\vdots\end{pmatrix}\cdot\nabla\phi_i - \begin{pmatrix}\vdots\\\phi_j-\phi_i\\\vdots\end{pmatrix}\|^2, \tag{B.14}$$

which results in solving a linear equation system for each cell with a 3×3 matrix. The second approach is known for giving accurate results and can be computed efficiently by storing the inverse of the 3×3 matrix for each cell.

Differed correction. The differed correction approach form (B.8) adds a source term to the right-hand side. The additional term can be treated as follows:

$$\begin{aligned}|\Gamma_{ij}|F_{ij}^D = \int_{\Gamma_{ij}}\nabla\phi^{k-1}\cdot\left(\frac{d_{ij}}{|d_{ij}|}-n\right)ds &= |\Gamma_{ij}|\underbrace{\int_{\Gamma_{ij}}\nabla\phi^{k-1}\,ds}_{=:\nabla\phi_{ij}^{k-1}}\cdot\left(\frac{d_{ij}}{|d_{ij}|}-n_{ij}\right) \\ &= |\Gamma_{ij}|\nabla\phi_{ij}^{k-1}\cdot\left(\frac{d_{ij}}{|d_{ij}|}-n_{ij}\right) \\ &\approx |\Gamma_{ij}|\nabla\phi'_{ij}\cdot\left(\frac{d_{ij}}{|d_{ij}|}-n_{ij}\right)\end{aligned} \tag{B.15}$$

using the iterpolated gradient $\nabla\phi_{ij}^{k-1} \approx \nabla\phi'^{k-1}_{ij} = \lambda\nabla\phi_i^k + (1-\lambda)\nabla\phi_j^k$.

Interpolaiton improvement The error for the discretization is only of second order if the points x'_{ij} and x_{ij} coincide. If this is the case, $\phi_{ij} = \phi'_{ij} = \lambda\phi_i + (1-\lambda)\phi_j$ holds for the linear interpolation of the face values. However, if the connecting line d_{ij} does not go through the face midpoint x_{ij}, the approximation is only first-order accurate. Because the above discretization method considers an iterative procedure, the interpolated value ϕ'_{ij} can be improved with the excentricity correction

$$\phi_{ij}^k \approx \phi'^k_{ij} + \nabla\phi'^{k-1}_{ij}\cdot(x_{ij} - x'_{ij}) \tag{B.16}$$

using the gradient $\nabla\phi'_{ij}$ computed in the previous iteration $k-1$.

Appendix C

Node Value Interpolation

To deform the shape, the nodes of the computational mesh are moved using the formula (2.3). However, the discrete solution consists of cell-wise average values when utilizing the finite volume method to compute the vector field V for the shape deformation. Thus, the nodal values of V must be reconstructed using the cell-wise average values. The implementation used for the optimization in this thesis uses an inverse distance weighted interpolation strategy, following [She68], of the form

$$V(x_i) \approx \frac{\sum_{j \in \mathcal{C}_i} w_{ij}^p V_j}{\sum_{j \in \mathcal{C}_i} w_{ij}^p}, \tag{C.1}$$

where $\mathcal{C}_i$ denotes the index set of adjacent cells of the i-th node, and w_{ij}^p are weights given by

$$w_{ij}^p = \frac{1}{|x_i - x_j|^p}. \tag{C.2}$$

In the above expression $V(x_i)$ denotes the value of V at the i-th node with the coordinate x_i, and V_j denotes the cell average of the j-th control volume (cf. (B.6)). In this work, the weights (C.2) exclusively use an exponent $p = 2$ for the interpolation.

Appendix D

Parallel Conjugated Gradient Method

Solving large linear systems, typically occurring in the numerical approximation of PDE solutions, requires some parallelization. The algorithmic approach for distributed-memory machines is based on a domain decomposition of the physical domain Ω. Hence, each process has only a segment of the solution vector, and the subsystem can be handled individually. This section considers the global linear system of the form $Ax = b$ where $A \in \mathbb{R}^{n\times n}$, $x, b \in \mathbb{R}^n$, and $n \in \mathbb{N}$ is the total number of unknowns. The computational mesh is decomposed in n_{part} parts, and each process i has one part of the domain of size $n_i \in \mathbb{N}$ such that $n = \sum_{i=1}^{n_{\text{part}}} n_i$. The parts overlap by one layer of cells, and each process stores the results of the neighboring cells for the so-called ghost cells; The number of ghost cells for each process is denoted as n_{ghost}. Due to the discretization method and domain decomposition, the matrix A is sparse and has a block diagonal structure with some entries in the off-diagonal blocks. The entries in the diagonal block correspond to the process's unknowns, and those outside the diagonal blocks correspond to the ghost cell values from neighbor processes.
The preconditions system, with matrix $M \in \mathbb{R}^{n\times n}$ that is simple to invert, the original system is transformed such that $M^{-1}Ax = M^{-1}b$. The linear equation systems with a symmetric matrix $A = A^\intercal \in \mathbb{R}^{n\times n}$, which arise from the discretization of a PDE, are solved using the Conjugate Gradient method (CG) method, with symmetric over-relaxation (SOR) preconditioner

$$M = \frac{1}{2-\omega}(\frac{1}{\omega}D + L)(\frac{1}{\omega}D)^{-1}(\frac{1}{\omega}D + L)^T. \tag{D.1}$$

In parallel, the preconditioning is realized with a block Jacobi and SOR on each block [Bar+93] The implementation utilizes the Message Passing Interface (MPI) [GLS99] for distributed memory machines due to a large number of unknowns. The matrix L is the lower triangular part, and $D = \text{diag}(a_{11}, \ldots, a_{nn})$ is a diagonal matrix with the diagonal elements of A. The parallel preconditioned CG method is sketched in Algorithm 12. For each process, the arrays that store the variables r, v, x and y hold n_{part} double precision values; only the array that stores p holds $n_{\text{part}} + n_{\text{ghost}}$ variables. Due to the structure of M, solving with M^{-1} can be done in the three steps

$$\begin{aligned} \frac{1}{2-\omega}(\frac{1}{\omega}D + L)x &= r, \\ (\frac{1}{\omega}D)^{-1} y &= x, \\ (\frac{1}{\omega}D + L)^T z &= y. \end{aligned} \tag{D.2}$$

or respectively

Algorithm 12 The parallel preconditioned CG

Require: $A = A^T \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $\mathrm{tol} > 0$, $\omega \in (0, 2)$, $m \in \mathbb{N}$

1: $x \leftarrow x_0$
2: $r \leftarrow b - Ax$, $r_0 \leftarrow r$
3: $\mathrm{res}_0 \leftarrow r^T r$
4: ParallelSum(res_0)
5: $z \leftarrow M^{-1} r$
6: $p \leftarrow z$
7: ParallelExchange(p)
8: $\rho\_ \leftarrow r^T z$
9: ParallelSum($\rho\_$)
10: **for** $k = 1, \ldots, m$ **do**
11: $v \leftarrow Ap$
12: $\alpha \leftarrow \rho / p^T z$
13: $r \leftarrow r - v\alpha$
14: $x \leftarrow x + p\alpha$
15: $\mathrm{res} \leftarrow r^T r$
16: ParallelSum(res)
17: **if** $\mathrm{res} \leq \mathrm{res}_0 \, \mathrm{tol}$ **then**
18: exit
19: **end if**
20: $z \leftarrow M^{-1} r$
21: $\rho \leftarrow r^T z$
22: ParallelSum(ρ)
23: $p \leftarrow z + p\rho / \rho\_$
24: ParallelExchange(p)
25: $\rho\_ \leftarrow \rho$
26: **end for**

for $i = 1, \ldots, n$ **do**
$x_i = \frac{\omega}{d_{ii}} \left((2 - \omega) r_i - \sum_{j=1}^{i-1} l_{ij} x_j \right)$
end for
$y_i = \omega d_{ii} x_i$
for $i = n, \ldots, 1$ **do**
$z_i = \frac{\omega}{d_{ii}} \left(y_i - \sum_{j=n}^{i+1} l_{ij} y_j \right)$
end for

Because the inner products in Algorithm 12 require communication, this formulation contains blocking communication points where processes are synchronized. It is possible to reformulate the algorithm such that communication overlaps with practical computations [Bar+93] or reduce the number of communication points

Bibliography

[ADJ21] 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.

[AFA13] H. Azegami, S. Fukumoto, and T. Aoyama. "Shape optimization of continua using NURBS and basis functions". In: *Structural and Multidisciplinary Optimization* 47.2 (2013). DOI: 10.1007/s00158-012-0822-4.

[Age20] International Energy Agency. *CO2 emissions by sector, World 1990-2018*. 2020. URL: https://www.iea.org/data-and-statistics/data-browser.

[AJT04] 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.

[All+05] G. Allaire et al. "Structural optimization using topological and shape sensitivity via a level set method". In: *Control and cybernetics* 34.1 (2005), p. 59.

[And+08] R. Andreani et al. "On augmented Lagrangian methods with general lower-level constraints". In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1286–1309. DOI: 10.1137/060654797.

[And+99] E. Anderson et al. *LAPACKA Users' Guide*. 1999. ISBN: 0-89871-447-8.

[AW96] H. Azegami and Z. c. Wu. "Domain Optimization Analysis in Linear Elastic Problems : Approach Using Traction Method". In: *JSME international journal. Ser. A, Mechanics and material engineering* 39.2 (1996). DOI: 10.1299/jsmea1993.39.2_272.

[Bäc96] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996. DOI: 10.1093/oso/9780195099713.001.0001.

[Bar+93] R. Barrett et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. A electronic copy is publicly available on https://netlib.org/templates. SIAM, 1993. ISBN: 978-0-898713-28-2.

[Bel+97] 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.

[Bla21] S. Blauth. "cashocs: A Computational, Adjoint-Based Shape Optimization and Optimal Control Software". In: *SoftwareX* 13 (2021). DOI: 10.1016/j.softx.2020.100646.

[BM20] S. Bartels and M. Milicevic. "Efficient iterative solution of finite element discretized nonsmooth minimizationproblems". In: *Computers and Mathematics with Applications* 80.5 (2020). DOI: j.camwa.2020.04.026.

[Bra+09] 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*. International Series of Numerical Mathematics. Birkhäuser, Basel, 2009. DOI: 10.1007/978-3-7643-8923-9_2.

[DB06a] R. Dwight and J. Brézillon. “Effects of Various Approximations of the Discrete Adjoint on Gradient-Based Optimization”. In: *AIAA paper 2006* 690 (2006).

[DB06b] 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.

[De 15] J. C. De los Reyes. *Numerical PDE-Constraint Optimization*. Ed. by P. M. Pardalos et al. Springer, Cham Heidelberg New York Dordrecht London, 2015. DOI: 10.1007/978-3-319-13395-9.

[DHH22] K. Deckelnick, P. J. Herbert, and M. Hinze. “A novel $W^{1,\infty}$ approach to shape optimisation with Lipschitz domains”. In: *ESIAM: COCV* 28.2 (2022).

[DHH23] K. Deckelnick, P. J. Herbert, and M. Hinze. “Shape optimisation in the $W^{1,\infty}$ topology with the ADMM algorithm”. E-print available in arXiv: https://arxiv.org/abs/2301.08690. 2023.

[DZ11] M. C. Delfour and J.-P. Zolesio. *Shapes and Geometries, Metrics, Analysis, Differential Calculus, and Optimization*. Second. Advances in Design and Control. SIAM, 2011. ISBN: 978-0-89871-936-9. DOI: 10.1137/1.9780898719826.

[ESW14] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Itertative Solvers With Applications in Incompressible Fluid Dynamics*. Vol. 1. Oxford Science Publications, 2014.

[FPS20] J. H. Ferziger, M. Peri, and R. L. Steet. *Computational Methods for Fluid Dynamics*. 4th ed. Springer Cham, 2020. DOI: 10.1007/978-3-319-99693-6.

[GLS99] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 1999.

[Gou06] Frédéric de Gournay. “Velocity Extension for the Level-set Method and Multiple Eigenvalues in Shape Optimization”. In: *SIAM Journal on Control and Optimization* 45.1 (2006), pp. 343–367. DOI: 10.1137/050624108. URL: https://doi.org/10.1137/050624108.

[GP00] M. Giles and N. Pierce. “An Introduction to the Adjoint Approach to Design”. In: *Flow, Turbulence and Combustion* 65.3 (2000). DOI: 10.1023/A:1011430410075.

[GP97] M. Giles and N. Pierce. “Adjoint equations in CFD-Duality, boundary conditions and solution behaviour”. In: *13th Computational Fluid Dynamics Conference*. AIAA, 1997. DOI: 10.2514/6.1997-1850.

[GPK11] K. C. Giannakoglou, D. I. Papadimitriou, and I. C. Kampolis. “Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels”. In: *Computer Methods in Applied Mechanics and Engineering* 195.44 (2011), pp. 6312–6329. DOI: 10.1016/j.cma.2005.12.008.

[GR09] 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.

[GRS07] C. Grossmann, H.-G. Roos, and M. Stynes. *Numerische Behandlung Partieller Differentialgleichungen*. Springer Berlin Heidelberg New-York, 2007. ISBN: 978-3-540-71582-5.

[Hac85] W. Hackbusch. *Multi-grid methods and applications*. Vol. 4. Springer, 1985. DOI: 0.1007/978-3-662-02427-0.

[Her23] P. J. Herbert. "Shape Optimisation with $W^{1,\infty}$: A connection between the steepest descent and Optimal Transport". Eprint available in arXiv https://arxiv.org/pdf/2301.07994.pdf. 2023.

[HH] R. M. Hicks and P. A. Henne. In: ().

[Hin+09] M. Hinze et al. *Optimization with PDE Constraints*. Vol. 23. Mathematical Modelling: Theory and Application. Springer, 2009. DOI: 10.1007/978-1-4020-8839-1.

[HMT07] S. Hofmann, M. Mitrea, and M. Taylor. "Geometric and Transformational Properties of Lipschitz Domains, Semmes-Kenig-Toro Domains, and Other Classes of Finite Perimeter Domains". In: *Journal of Geomtric Analysis* 17.4 (2007). DOI: 10.1007/BF02937431.

[HP18] A. Henrot and M. Pierre. *Shape Variation and Optimization, A Geometrical Analysis*. Ed. by M. Faber et al. EMS, 2018. DOI: 10.4171/178.

[HSB14] M. Hojjat, E. Stavropoulou, and K.-U. Bletzinger. "The Vertex Morphing method for node-based shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014). DOI: 10.1016/j.cma.2013.10.015.

[HSU21] J. Haubner, M. Siebenborn, and M. Ulbrich. "A Continuous Perspective on Shape Optimization via Domain Transformations". In: *SIAM Journal of Scientific Computing* 43.3 (2021). DOI: 10.1137/20M1332050.

[IL05] 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.

[IMO20] IMO. *Fourth IMO Greenhouse Gas Study 2020*. 2020. URL: https://www.imo.org/en/OurWork/Environment/Pages/Fourth-IMO-Greenhouse-Gas-Study-2020.aspx.

[Jam+06] A. Jameson et al. "Aerodynamic Simulation and Shape Optimization in High Speed Flow". In: *44th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA, 2006. DOI: 10.2514/6.2006-708.

[Jam04] A. Jameson. "Advances in Aerodynamic Shape Optimization". In: *Computational Fluid Dynamics 2004*. Ed. by C. Groth and D.W. Zingg. Springer, Berlin, Heidelberg, 2004. DOI: 10.1007/3-540-31801-1_100.

[Jam88] A. Jameson. "Aerodynamic Design via Control Theory". In: *Journal of Scientific Computing* 3.3 (1988).

[JS11] A. Jahangirian and A. Shahrokhi. "Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver". In: *Computers & Fluids* (2011). DOI: 10.1016/j.compfluid.2011.02.010.

[Kaw+00] B. Kawohl et al. *Optimal Shape Design. Lectures given at the Joint C.I.M./C.I.M.E. Summer School held in Troia (Portugal), June 1-6, 1998*. Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg, 2000. DOI: 10.1007/BFb0106739.

[Kel20] S. Keller. *Handelsschiffe weltweit nach Handelsflagge 2019*. 2020. URL: https://de.statista.com/statistik/daten/studie/29152/umfrage/anzahl-der-handelsschiffe-weltweit-nach-handelsflagge/.

[Kit] Kitware. *Visialization Toolkit*. URL: https://vtk.org.

[KKR18] J. Kröger, N. Kühl, and T. Rung. "Adjoint Volume-of-Fluid Approaches for the Hydrodynamic Optimization of Ships". In: *Ship Technology Research* 65.1 (2018). DOI: 10.1080/09377255.2017.1411001.

[KMR21] N. Kühl, P. M. Müller, and T. Rung. "Adjoint Complement to the Universal Momentum Law of the Wall". In: *Flow, Turbulence and Combustion* 108.2 (2021), pp. 329–315. DOI: 10.1007/s10494-021-00286-7.

[KR15] J. Kröger and T. Rung. "CAD-free hydrodynamic optimisation using consistent kernel-based sensitivity filtering". In: *Ship Technology Research* 62.3 (2015), pp. 111–130. DOI: 10.1080/09377255.2015.1109872.

[Krö16] J. C. Kröger. "A numerical process for the hydrodynamic optimisation of ships". PhD thesis. Hamburg University of Technology, 2016.

[KSK13] 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).

[Küh+19] N. Kühl et al. "Decoupling of Control and Force Objective in Adjoint-Based Fluid Dynamic Shape Optimization". In: *AIAA journal* 57.9 (2019), pp. 4110–4114. DOI: 10.2514/1.J058376.

[Küh+22] N. Kühl et al. "Adjoint node-based shape optimization of free floating vessels". In: *Structural and Multidisciplinary Optimization* 65 (2022), p. 247. DOI: 10.1007/s00158-022-03338-2.

[Küh21] N. Kühl. "Adjoint-based shape optimization constraint by turbulent two-phase Navier-Stokes systems". PhD thesis. Hamburg University of Technology, 2021. DOI: 10.15480/882.3794.

[Lan] NASA Langley. *Turbulence Model Benchmarking Working Group Turbulence Modeling Resource*. URL: https://turbmodels.larc.nasa.gov/.

[Loi20] S. Loisel. "Efficient algorithms for solving the p-Laplacian in polynomial time". In: *Numerische Mathematik* 146.2 (2020), pp. 369–400. DOI: 10.1007/s00211-020-01141-z.

[MBF23] P. Marty, C. Boehm, and A. Fichtner. "Shape Optimization for Transcranial Ultrasound Computed Tomography". In: *Proceedings of SPIE*. 2023. DOI: 10.1117/12.2654328.

[MBR23] P. M. Müller, G. Bletsos, and T. Rung. "Transformation Approaches for Fluid Dynamic Optimization". In: *Aerospace* 10.6 (2023). DOI: 10.3390/aerospace10060519.

[MEI] Maritime, Ocean Engineering, and Research Institute. *KRISO container ship geometry*. http://www.simman2008.dk/KCS/kcs_geometry.htm.

[MP04] B. Mohammadi and O. Pironneau. "Shape Optimization in Fluid Mechanics". In: *Annual Review of Fluid Mechanics* 39 (2004), pp. 255–279. DOI: 10.1146/annurev.fluid.36.050802.121926.

[MP09] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2009. DOI: 10.1093/acprof:oso/9780199546909.001.0001.

[MSR23] P. M. Müller, M. Siebenborn, and T. Rung. "Application of p-Laplace relaxed steepest descent to shape optimization in two-phase flows". In: *Modeling, Simulation and Optimization of Fluid Dynamic Applications*. Ed. by A. Iske and T. Rung. Vol. 148. Lecture Notes in Computational Science and Engineering. Springer, Heidelberg, 2023. DOI: 978-3-031-45158-4_8.

[Mül] Peter Marvin Müller. *w-1-infinity-model-problems*. DOI: 10.5281/zenodo.10683873. URL: https://zenodo.org/records/10683873.

[Mül+21] P. M. Müller et al. "A novel p-harmonic descent approach applied to fluid dynamic shape optimization". In: *Struct. Multidic. Optim.* 64 (2021), pp. 3489–3503. DOI: 10.1007/s00158-021-03030-x.

[Mül+23] 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.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Ed. by T. V. Mikosch, S. I. Resnick, and S. M. Robinson. 2nd ed. Springer, New York, 2006.

[OS21] S. Onyshkevych and M. Siebenborn. “Mesh Quality Preserving Shape Optimization Using Nonlinear Extension Operators”. In: *Journal of Optimization Theory and Applications* (2021). DOI: 10.1007/s10957-021-01837-8.

[Oth08] C. Othmer. “A Continuous Adjoint Formulation for the Computation of Topological and Surface Sensitivities of Ducted Flows”. In: *International Journal for Numerical Methods in Fluids* 58.8 (2008), pp. 861–877. DOI: 10.1002/fld.1770.

[PD10] J.E.V. Peter and R. P. Dwight. “Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches”. In: *Computers & Fluids* 39.3 (2010). DOI: 10.1016/j.compfluid.2009.09.013.

[PG16] E. M. Papoutsis-Kiachagias and K. C. Giannakoglou. “Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications”. In: *Archives of Computational Methods in Engineering* 23.2 (2016). DOI: 10.1007/s11831-014-9141-9.

[Pir73] O. Pironneau. “On optimum profiles in Stokes flow”. In: *Journal of Fluid Mechanics* 59.1 (1973), pp. 117–128. DOI: 10.1017/S002211207300145X.

[PS10] P. I. Plotnikov and J. Sokolowski. “Shape Derivative of Drag Functional”. In: *SIAM Journal on Control and Optimization* 48.7 (2010). DOI: 10.1137/090758179.

[PS21] J. Pinzon and M. Siebenborn. *PLaplaceOptim*. http://www.github.com/multigridshapeopt. Version 1.0. 2021. (Visited on 08/17/2021).

[PS23] J. Pinzon and M. Siebenborn. “Fluid dynamic shape optimization using self-adapting nonlinear extension operators with multigrid preconditioners”. In: *Optimization and Engineering* 24.2 (2023).

[Rei+13] S. Reiter et al. “A massively parallel geometric multigrid solver on hierarchically distributed grids”. In: *Comp. Vis. Sci.* 16.4 (2013), pp. 151–164. DOI: 10.1007/s00791-014-0231-x.

[Rob+12] T. T. Robinson et al. “Optimizing Parameterized CAD Geometries Using Sensitivities Based on Adjoint Functions”. In: *Computer-Aided Design and Applications* 9.3 (2012). DOI: 10.3722/cadaps.2012.253-268.

[Run+09] T. Rung et al. “Challenges and Perspectives for Maritime CFD Applications”. In: *Jahrbuch der Schiffbautechnischen Gesellschaft* 103 (2009), pp. 127–39.

[SHB14] E. Stavropoulou, M. Hojjat, and K.-U. Bletzinger. “In-plane mesh regularization for node-based shape optimization problems”. In: *Computer Methods in Applied Mechanics and Engineering* 275 (2014). DOI: 10.1016/j.cma.2014.02.013.

[She68] D. Shepard. “A two-dimensional interpolation function for irregularly-spaced data”. In: *ACM'68ACM '68: Proceedings of the 1968 23rd ACM national conference*. 1968, pp. 517–524. DOI: 10.1145/800186.810616.

[SL02] O. Soto and R. Löhner. “A mixed adjoint formulation for incompressible turbulent problems”. In: *40th AIAA Aerospace Sciences Meeting & Exhibit*. 2002. DOI: 10.2514/6.2002-451.

[SL04] O. Soto and R. Löhner. “On the Boundary Computation of Flow Sensitivities”. In: *42nd AIAA Aerospace Science Meeting and Exhibit*. 2004. DOI: abs/10.2514/6.2004-112.

[SLY04] O. Soto, R. Löhner, and C. Yang. "An adjoint-based design methodology for CFD problems". In: *International Journal of Numerical Methods for Heat & Fluid Flow* 14.6 (2004). DOI: 10.1108/09615530410544292.

[SR11] A. Stück and T. Rung. "Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation". In: *Computers & Fluids* 47.1 (2011), pp. 22–32. DOI: 10.1016/j.compfluid.2011.01.041.

[SR13] A. Stück and T. Rung. "Adjoint Complement to Viscous Finite-Volume Pressure-Correction Methods". In: *Journal of Computational Physics* 248 (2013), pp. 402–419. DOI: 10.1016/j.jcp.2013.01.002.

[SS16] 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.

[Stü12] A. Stück. "Adjoint Navier-Stokes methods for hydrodynamic shape optimisation". PhD thesis. Hamburg University of Technology, 2012. DOI: 10.15480/882.1061.

[SZ92] J. Sokolowski and J.-P. Zolesio. *Introduction to shape optimization*. Vol. 16. Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg, 1992. ISBN: 978-3-642-63471-0. DOI: 10.1007/978-3-642-58106-9_1.

[Tem10] R. Temam. "Navier-Stokes equaitons : theory and numerical analysis". In: Studies in Mathematics and Its Applications. North-Holland Pub. Co, 2010.

[Ten+] Univ. of Tennessee et al. *Liner Aalgebra PACKage*. URL: https://www.netlib.org/lapack/.

[Ulb03] M. Ulbrich. "Constrained Optimal Control of Navier-Stokes Flow by Semismooth Newton Methods". In: *Systems & Control Letters* 48 (2003). DOI: 10.1016/S0167-6911(02)00274-8.

[USD21] B. D. Upadhyay, S. S. Sonigra, and S. D. Daxini. "Numerical analysis perspective in structural shape optimization: A review post 2000". In: *Advances in Engineering Software* 155 (2021). DOI: 10.1016/j.advengsoft.2021.102992.

[VGJ05] J. C. Vassberg, A. Gopinath, and A. Jameson. "Revisiting the Vertical-Axis Wind-Turbine Design Using Advanced Computational Fluid Dynamics". In: *43rd AIAA Aerospace Sciences Meeting and Exhibit*. 2005. DOI: 10.2514/6.2005-47.

[Vog+13] A. Vogel et al. "UG 4: A novel flexible software system for simulating PDE based models on high performance computers". In: *Comp. Vis. Sci.* 16.4 (2013), pp. 165–179. DOI: 10.1007/s00791-014-0232-9.

[Wil98] D. C. Wilcox. *Turbulence modeling for CFD*. Vol. 2. DCW industries La Canada, CAMKSDHR2021, 1998.

[Yak+13] S. Yakubov et al. "Hybrid MPI/OpenMP Parallelization of an Euler-Lagrange Approach to Cavitation Modelling". In: *Computers & Fluids* 80 (2013), pp. 365–317. DOI: 10.1016/j.compfluid.2012.01.020.

[YMR15] S. Yakubov, T. Maquil, and T. Rung. "Experience Using Pressure-Based CFD Methods for Euler-Euler Simulations of Cavitating Flows". In: *Computers & Fluids* 111 (2015). DOI: 10.1016/j.compfluid.2015.01.008.