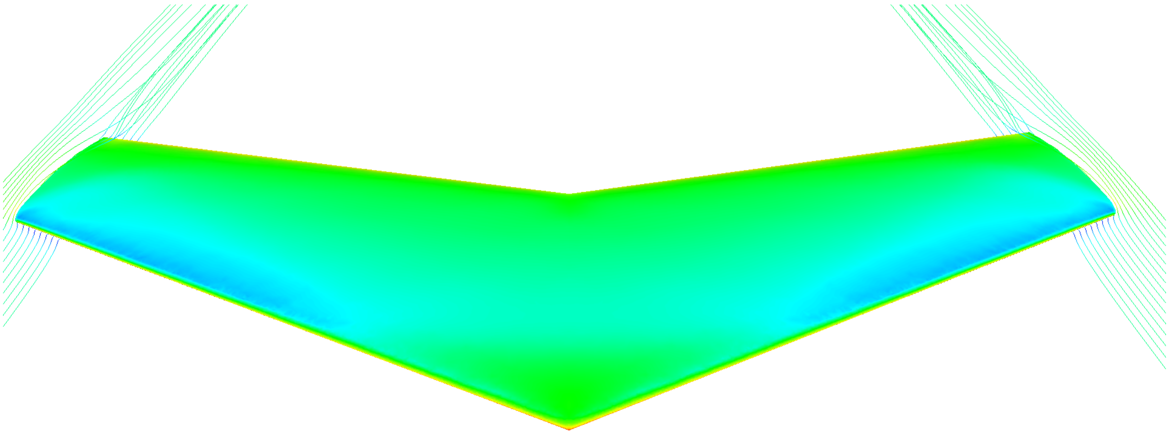


Efficient Large Scale Aerodynamic Design Based on Shape Calculus

Dissertation

zur Erlangung des Akademischen Grades eines Doktors der Naturwissenschaften
(Dr. rer. nat.)

Dem Fachbereich IV der Universität Trier vorgelegt von

Dipl.-Math. Stephan Schmidt

to morph, verb, (third-person singular simple present morphs, present participle morphing, simple past and past participle morphed)

1. *Shortening of metamorphose: to change in shape or form.*
2. *(colloquial) To undergo dramatic change in a seamless and barely noticeable fashion.*

Abstract

Large scale non-parametric applied shape optimization for computational fluid dynamics is considered. Treating a shape optimization problem as a standard optimal control problem by means of a parameterization, the Lagrangian usually requires knowledge of the partial derivative of the shape parameterization and deformation chain with respect to input parameters. For a variety of reasons, this mesh sensitivity Jacobian is usually quite problematic. For a sufficiently smooth boundary, the Hadamard theorem provides a gradient expression that exists on the surface alone, completely bypassing the mesh sensitivity Jacobian. Building upon this, the gradient computation becomes independent of the number of design parameters and all surface mesh nodes are used as design unknowns in this work, effectively allowing a free morphing of shapes during optimization.

Contrary to a parameterized shape optimization problem, where a smooth surface is usually created independently of the input parameters by construction, regularity is not preserved automatically in the non-parametric case. As part of this work, the shape Hessian is used in an approximative Newton method, also known as Sobolev method or gradient smoothing, to ensure a certain regularity of the updates, and thus a smooth shape is preserved while at the same time the one-shot optimization method is also accelerated considerably. For PDE constrained shape optimization, the Hessian usually is a pseudo-differential operator. Fourier analysis is used to identify the operator symbol both analytically and discretely. Preconditioning the one-shot optimization by an appropriate Hessian symbol is shown to greatly accelerate the optimization.

As the correct discretization of the Hadamard form usually requires evaluating certain surface quantities such as tangential divergence and curvature, special attention is also given to discrete differential geometry on triangulated surfaces for evaluating shape gradients and Hessians.

The Hadamard formula and Hessian approximations are applied to a variety of flow situations. In addition to shape optimization of internal and external flows, major focus lies on aerodynamic design such as optimizing two dimensional airfoils and three dimensional wings. Shock waves form when the local speed of sound is reached, and the gradient must be evaluated correctly at discontinuous states. To ensure proper shock resolution, an adaptive multi-level optimization of the Onera M6 wing is conducted using more than $36,000$ shape unknowns on a standard office workstation, demonstrating the applicability of the shape-one-shot method to industry size problems.

Zusammenfassung

Der Gegenstand dieser Arbeit ist die hochdimensionale nicht-parametrische angewandte Formoptimierung für die numerische Strömungssimulation. Wird ein Formoptimierungsproblem durch eine Parametrisierung wie ein gewöhnliches nichtlineares Optimierungsproblem behandelt, so benötigt die Lagrange–Funktion Kenntnis der partiellen Ableitungen der Parametrisierung und der Deformationskette bezüglich der Eingabeparameter. Aus verschiedensten Gründen sind diese Mesh- oder Metriksensitivitäten für gewöhnlich sehr problematisch. Für eine hinreichend glatte Oberfläche bietet das Hadamard–Theorem einen Ausdruck für den Gradienten, welcher ausschließlich auf der Oberfläche der Form existiert und die Metriksensitivitäten komplett umgeht. Darauf aufbauend wird die Berechnung des Gradienten unabhängig von der Anzahl der Variablen und im Rahmen dieser Arbeit werden alle Oberflächenknoten des Gitters als Unbekannte benutzt, wodurch effektiv ein freies Morphing der Form während der Optimierung ermöglicht wird.

Im Gegensatz zu einem parametrisierten Formoptimierungsproblem, bei dem die Glattheit der Oberfläche fast immer unabhängig von den Eingabeparametern entsprechend der Konstruktion der Parameterisierung gewährleistet ist, muss die Regularität bei dem nicht-parametrischen Ansatz nicht zwingend erhalten bleiben. In dieser Arbeit wird die Hesse–Abbildung des Formoptimierungsproblems in einem approximativen Newton–Verfahren, auch bekannt als Sobolev–Verfahren oder Gradientenglätten, genutzt, um die Regularität der Updates sicherzustellen und somit eine glatte Oberfläche zu erhalten, wodurch gleichzeitig die Optimierung in One-Shot deutlich beschleunigt wird. Für Optimierungsprobleme mit PDEs ist die Hesse–Abbildung gewöhnlich ein Pseudo-Differentialoperator. Fourieranalysis wird benutzt, um das Symbol des Operators sowohl analytisch als auch diskret zu bestimmen. Es wird gezeigt, wie eine Präkonditionierung des One-Shot Verfahrens durch ein entsprechendes Symbol der Hesse–Abbildung die Optimierung stark beschleunigt.

Da die korrekte Diskretisierung der Hadamard–Form für gewöhnlich die Auswertung von Oberflächengrößen wie Tangentialdivergenz oder Krümmung benötigt, liegt besonderes Augenmerk auf diskreter Differentialgeometrie zur Auswertung des Formgradienten und der Hesse–Abbildung auf unstrukturierten, triangulierten Oberflächen.

Die Hadamard–Form und die Hesse–Approximationen werden auf eine Vielfalt von Strömungssituationen angewendet. Neben der Formoptimierung von internen und externen Strömungen liegt der eigentliche Anwendungsschwerpunkt im aerodynamischen Entwurf, zum Beispiel die Optimierung zweidimensionaler Profilquerschnitte und dreidimensionaler Flügel. Schockwellen bilden sich aus, wenn die lokale Schallgeschwindigkeit erreicht wird, und der Gradient muss an einem unstetigen Zustand richtig ausgewertet werden. Um eine korrekte Auflösung der Schockwelle zu gewährleisten, wird eine adaptive multi-level Optimierung am Onera M6 Flügel mit mehr als 36.000 Unbekannten auf einer gewöhnlichen Workstation durchgeführt, was auch die Anwendbarkeit der Methodik auf Probleme industriellen Ausmaßes demonstriert.

Acknowledgments

First of all, I would like to thank my supervisor Prof. Dr. Volker Schulz for the many helpful discussions and the ongoing support, but also for the frequent opportunities to visit external conferences with inspiring discussions, including Oberwolfach and SIAM conferences in the United States. His group at the University of Trier is a very creative, challenging, and pleasant environment to work in. I also would like to thank Prof. Dr. Jan Sokolowski for the many discussions on shape optimization and for accepting the position as second referee.

Furthermore, I would like to thank Prof. Dr. Nicolas Gauger, with whom I have worked closely together. His detailed knowledge about computational fluid dynamics and adjoint solvers have been as valuable to me as his moral support.

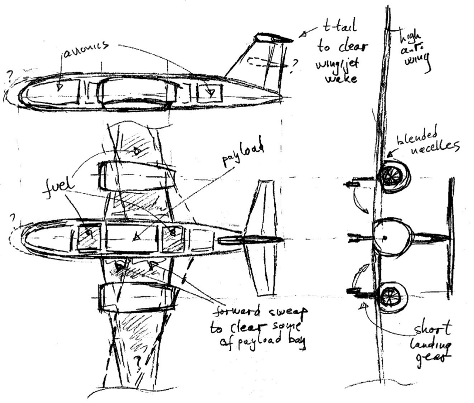
I would also like to thank Prof. Dr. Karsten Eppler. He first introduced me to shape calculus, and the many discussions we had till late at night during his stay at the University of Trier were extremely helpful. Additionally, I would like to thank Dipl.-Ing. Caslav Ilic at the German Aerospace Center (DLR), Braunschweig, who handled most of the DLR flow solvers and miscellaneous software. Spending countless hours on the phone, our many discussions have also greatly helped coming up with discrete solutions for analytical problems. While I was raving about optimizing an Airbus into a Concorde, he made sure none of the fine details of aerodynamic design were overlooked. He also greatly helped by proof-reading this thesis.

I would also like to thank my fellow graduate students at the University of Trier for creating such a nice work atmosphere: Claudia Schillings, Roland Stoffel, Benjamin Rosenbaum, Christian Wagner, Christoph Käbe, Andre Lörx, Timo Hylla, Matthias Schu, Bastian Groß, Nils Langenberg, and Christina Jager. Additionally, I would like to thank Dipl.-Phys. Christian Haake for writing output routines for my flow solver in postscript, thereby creating restart files which visualize themselves, and for proof-reading this thesis.

Finally and naturally, I would like to thank my parents for their support and encouraging comments during the last three years.

This work has been supported by the German science foundation (DFG) priority program SPP-1253: “Optimization with Partial Differential Equations” as the project “Multilevel Parameterizations and Fast Multigrid Methods for Aerodynamic Shape Optimization”, a joint project between Prof. Dr. Volker Schulz, University of Trier, and Prof. Dr. Nicolas Gauger, DLR Braunschweig.

Contents



Chapter 1

Introduction

1.1 Paradigms in Shape Optimization

As a special field of optimization subject to partial differential equations (PDEs), shape optimization and control of fluids has seen steady research interest over the past decades. Especially in aerodynamic design, the transition from simulation alone to a coupled simulation and optimization is progressing continuously. Although heuristic and derivative free optimization methods are still used in practice, only structure exploiting gradient based methods are efficient enough for optimizing industry size large scale systems.

Two major advancements in the field of derivative based general PDE constraint optimization and its application to aerodynamic design have been the introduction of gradient computation via adjoint calculus [23, 29, 40, 50] and the optimization in one-shot [27, 31, 66, 72, 73]. A good overview on applying general control theory to fluid control problems can also be found in [32]. When considering the special sub-class of shape optimization problems and fluid flow, such problems are almost always interpreted as a general non-linear optimization problem via a parameterization. By choosing a finite set of design parameters, such as the popular Hicks–Henne functions [35], B-splines, free-form deformation, or general computer aided design (CAD) software, the shape optimization problem is reduced to a standard optimization problem in finite dimensions. Thus, a parameterization is also a discretization. This means that the Lagrangian of the parameterized shape optimization problem is studied without considering the original nature of the problem, and possibilities for shape optimization structure exploitation are therefore neglected. Even worse, the Lagrangian system of a parameterized shape optimization problem requires knowledge of so called mesh sensitivities, i.e. the partial derivatives of the PDE constraint and objective functions with respect to the parameterization.

Discretizing the flow equations by e.g. finite volumes or finite elements results in a dependence of the state residual on the positions of the volume nodes. During shape optimization, the volume mesh must somehow be made to fit the new surface, which results in a perturbed volume mesh. Consequently, the mesh deformation mapping becomes part of the total parameterization chain and must also be differentiated. This chain is also shown in figure 1.1. The propagation of the deformed surface into the volume mesh is often subject to solving additional PDEs or, mostly in case of a CAD approach, a proprietary black box. Also, high fidelity computational fluid dynamics (CFD) software is very complex, and computing the derivative of a flow solver residual with respect to volume mesh node positions is also very challenging. Thus, one is frequently forced into finite differencing. This, however, can become extremely time consuming because a perturbed volume mesh must be made for each design parameter. For complex three dimensional aircraft configurations, these deformed meshes are very costly to create. Being dense matrices, the resulting Jacobians are also very memory consuming. Although computing these sensitivities can be acceptable for a very limited number of design parameters such as span, sweep angle, and twist, they very quickly become prohibitive for large scale deformations, effectively preventing any morphing of shapes. Furthermore, they tend to make the one-shot approach inefficient in terms of total CPU wall-clock-time to solution. One of the ideas that make one-shot such a fast optimization procedure is to trade an exact state and adjoint solution for more optimization steps and hence more gradient evaluations. This produces a speed-up only under the assumption that a gradient evaluation is numerically inexpensive compared to a forward and adjoint flow solution, which is no longer the case when one has to compute such mesh sensitivities.

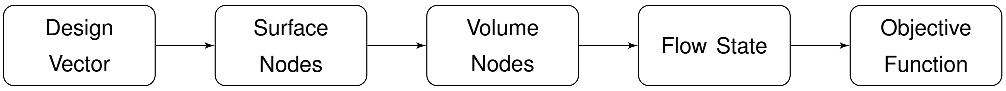


Figure 1.1: Mesh deformation chain for parameterized shape optimization.

Possible remedies mentioned in the literature are the treatment of the whole shape parameterization chain via algorithmic or automatic differentiation (AD) [24, 25] or introducing limited and local area of influences for volume mesh nodes and their perturbations. Sometimes, it is also possible to exploit special properties of structured meshes [41, 55]. However, there can be complexity problems and memory limitations when trying to apply a reverse mode AD on sophisticated flow solvers, especially if the code heavily depends on pointers. Also, for AD one needs complete access to the software sources, which can be problematic. Limiting the area of influence of perturbed volume nodes enforces sparsity in the Jacobian, but often the resulting derivative is not very accurate. Either way, one is quite limited in the possible deformations and useable structured meshes. Thus, the reachable shapes are very limited, resulting in a narrow search space.

General shape optimization, as pioneered in [70] and later studied as an aspect of geometry in [11], allows ignoring the parameterization step altogether and conduct a gradient derivation without considering discrete aspects at all. The Hadamard theorem states that for sufficiently smooth boundaries, the gradient of a shape optimization problem exists on the surface alone, thus analytically bypassing the volume mesh altogether. This is a profound advantage for large scale shape optimization in general and one-shot shape optimization in specific. The analytic expression of the Hadamard form can be evaluated very quickly wherever desired, making the gradient computation indeed independent of the number of design parameters. It is thus possible to use the position of each mesh surface node as a design parameter, utilizing extraordinary possibilities of shape deformations and morphing. It also supports the full optimization speed-up by one-shot methods nicely, possibly helping advanced aerodynamic optimizations such as Pareto curve computations [59] and optimization under uncertainties [67]. Additionally, the non-parametric approach is inherently suited for multi-level optimization and local adaptivity, as the Hadamard form can be evaluated at any surface point desired, independent of mesh topology changes between optimization steps. Interestingly, the non-parametric paradigm is seldom used in aerodynamic design, except to show optimality of certain rotationally symmetric ogive shaped bodies in supersonic, irrotational, inviscid potential flows [33]. Furthermore, the non-parametric approach is also used to derive optimal shapes in a viscous Stokes flow [49]. Although certain non-parametric shape optimization ideas are present in the literature [10, 15, 46, 47], it is almost never applied in any actual optimization.

Also, there is the effect of loss of regularity. While the parametric approach ensures a smooth shape for any choice of design parameters, this is no longer the case when considering the non-parameterized infinite dimensional shape optimization problem. The parameterization determines the regularity class of which shapes are constructed, and in a general non-parametric shape optimization approach, the desired regularity class must be enforced otherwise. Imagining a simple steepest descent algorithm, it is easy to see that updates must be in the same regularity space as the original shape. Consequently, the gradient based update must be manipulated to maintain regularity. When applicable, the gradient can be thought of as the Riesz representative of the derivative, and the regularity of the gradient depends on the appropriate scalar product used in the underlying space. This procedure is sometimes also called gradient smoothing or Sobolev gradient method [41, 69], and thus questions about the appropriate space, i.e. scalar product, arise. As it turns out, the scalar product induced by the shape Hessian is an excellent choice because it not only cures the loss of regularity, but it also greatly accelerates the optimization as the Sobolev steepest descent method essentially becomes Newton's method, i.e. an SQP method.

Thus, the loss of regularity in specific and gradient based optimization in general rises questions about efficient Hessian computations and approximations. Literature on shape Hessians for parameterized problems is rare, possibly because the parameterization camouflages the structure of shape Hessians. The application of non-parameterized shape Hessians in a preconditioned conjugate gradient iteration for image segmentation is studied in [36, 37]. Also, shape Hessians and optimality conditions for shape optimization problems are considered in [16, 17, 18] with various applications in liquid metal shaping, electrical impedance tomography, and general elliptic problems. In general, shape Hessians are quite complex objects even for problems that appear manageable at first glance. They also no longer satisfy the Hadamard form of a scalar product of the normal component with a perturbation direction on the boundary. Thus, more accessible approximations are usually advantageous. For a PDE constrained optimization problem the shape Hessian usually is a pseudo-differential operator, and the effect of such a pseudo-differential operator on the regularity of the shape update must be studied. Another advantage of infinite dimensional shape optimization is the applicability of Fourier analysis to problems of moderate complexity, which allows the identification of the pseudo-differential operator order governing the Hessian. This, in return, defines the amount of re-smoothing that must be applied in the Sobolev smoothing step, i.e. by the Hessian approximation. Fourier tracking of perturbations has for example also been used in [2, 3, 4, 28, 61] for similar purposes.

Summarizing the above, it can be said that shape optimization is a field with a surprisingly strong gap between first optimizing analytically followed by a discretization of the expressions versus first discretizing and then optimizing the discrete problem. Very recently, this gap is studied in [6], where both approaches are unified in the finite element context. The optimize-then-discretize approach by using shape differentiation techniques and the Hadamard formula of the shape gradient as part of the present work has considerable advantages over a discretization by parameterization as it bypasses all volume mesh deformations of the problem, enables very fast gradient evaluations for an arbitrary number of design parameters, and makes a Hessian analysis much more accessible. Thus, large scale morphing of shapes by a one-shot optimization is possible. Correct gradient evaluations and Hessian approximations require discretizing surface quantities, such as tangential divergence and curvature, on unstructured meshes, thereby creating an interesting bridge between optimization with PDE constraints and other research fields such as computer graphics and discrete differential geometry.

1.2 Aim and Scope of this Work

An exhaustive analysis of a PDE constrained shape optimization problem requires a well-posed model, i.e. weak solutions for the geometries under consideration exist. Additionally, the set of solutions over the family of admissible domains needs to be compact such that there is a solution of the shape optimization problem. Once the existence of an optimal shape is established, methods to compute it can be discussed. The desire to use efficient gradient based methods naturally leads to the question of shape differentiability. Therefore, the family of solutions under consideration should be Lipschitz continuous with respect to boundary variations such that a shape sensitivity analysis can be conducted.

For the compressible Navier–Stokes equations with constant temperature, the well-posedness and existence of optimal shapes is established in [53]. There, it is first stated that assuming the existence of a domain and corresponding flow solution of finite internal energy, the drag minimization problem has a solution. Afterwards, the existence is ensured by constructing one such domain. The question of shape differentiability is answered in [51] using small perturbations of the so-called approximate solutions, which are determined from Stokes problems.

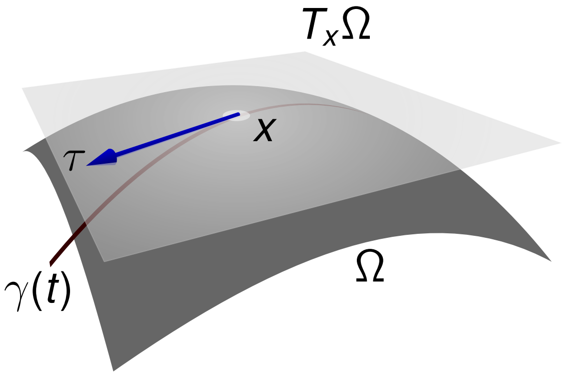
The aim and scope of this work, however, is to study how the information can be used to improve a given design numerically. Less focus lies on analytical existence and uniqueness of critical shapes. Therefore, a formal shape sensitivity analysis is conducted, followed by a study of the shape Hessian and an actual numerical optimization for a variety of problems in computational fluid dynamics and aerodynamic design. Special attention lies on the industrial applicability to very large scale shape optimization problems. This work is also used to study accelerating the one-shot approach for shape optimization problems by preconditioning, i.e. Hessian approximation, based on pseudo-differential operator approximation by Fourier analysis. Where possible, this is conducted analytically, otherwise discrete substitutes are considered. Thus, not only the applicability of infinite dimensional shape calculus to discrete problems is shown, but also the possible acceleration of the optimization procedure by analytically exploiting the structure of shape optimization problems.

1.3 Structure of this Work

The structure of this work is as follows: Chapter 2 is used to give a very brief overview about differential geometry. The purpose of this chapter is to prepare for chapter 3, i.e. special attention is given to introducing terms and definitions which are seldom encountered in the context of general PDE constrained optimization, such as tangential gradient, tangential divergence, and a variety of other lemmas and definitions. Combining many results scattered in the literature, chapter 3 then tries to give a complete overview about shape sensitivity analysis, i.e. shape differentiation techniques, especially when a PDE constraint is present. Special attention is given to providing the expressions in general formulations, as often in the literature, simplifications for standard Laplacian based problems, i.e. elliptic PDEs, are made, which prohibit application to the mixed parabolic/hyperbolic PDEs governing fluid flow. Chapter 4 is then used to give a brief overview on fluid dynamics. It not only introduces the PDEs governing the problems considered afterwards, but also shows what kind of objective functions are physically relevant for the different flow regimes of viscous, inviscid, compressible, and incompressible flows. One important fact is that in an inviscid compressible flow, a shape producing a shock-free flow solution can always be assumed to be drag optimal.

The following chapters 5 to 9 are then used to conduct the actual shape sensitivity analysis, Hessian approximation, and numerical optimization for a variety of CFD problems. The problems are studied in order of increasing difficulty of shape differentiation and not according to the sophistication of the fluid model, which is why the incompressible Navier–Stokes equations are considered before potential flow. First, chapter 5 considers shape optimization and Stokes fluids. This makes a very good introduction, as the optimal shape of the energy dissipation problem is known to be a rugby ball of 60° front and back angle, creating a perfect validation test-case for numerics. Due to its self-adjoint nature, the energy dissipation problem in a Stokes flow also allows for a very elegant Hessian derivation, and consequently this Hessian derivation is measured against the Fourier operator identification, familiarizing these concepts with a well structured example application. The chapter concludes by showing the optimization speed-up based on the Fourier symbol identification. By considering the incompressible Navier–Stokes equations in a general setting, the next chapter 6 both increases the complexity of the fluid model and the objective functions. Since the incompressible Navier–Stokes equations describe a very wide range of flow phenomena with numerous opportunities for application, very general objective shape functionals are considered. Since they are no longer self-adjoint as in the Stokes case, it is interesting to observe what kind of objective shape functionals allow consistent adjoint calculus. Due to their complex nature, the Fourier Hessian analysis is conducted in the discrete, again greatly accelerating a variety of optimizations. The chapter concludes with optimizing a variety of flow situations, such as internal flows through pipes or external flows around obstacles in the fluid. In chapter 7, the classical inverse design or pressure matching is considered. Assuming one has an intuition about what the pressure distribution on a good airfoil should be, a shape must be found which produces the desired pressure distribution in a potential flow. After a non-parametric shape sensitivity analysis is conducted, the shape Hessian is derived for star-shaped domains, and the optimization can again be greatly accelerated by a proper Hessian identification.

Starting with chapter 8, compressible flow models are considered. Since the density is assumed variable now, shock waves and discontinuities in the flow states form when the local speed of sound is reached. Evaluating the shape derivative at discontinuous states does not appear to be problematic, and after a detailed derivation of the shape gradient in Hadamard form, a variety of industry size aerodynamic shape optimization problems is considered. Here, the compressible Euler equations are used to model the fluid. Starting with supersonic two dimensional non-lifting airfoil optimizations, the chapter concludes with a three dimensional adaptively refined multi-level transonic Onera M6 wing optimization consisting of more than $36,000$ design parameters and multiple shock waves on the surface. Special attention is also given to discrete differential geometry and the correct evaluation of the shape gradient and Hessian approximations on triangulated unstructured surface meshes. The work concludes with chapter 9, where a formal shape differentiation for the compressible Navier–Stokes equations is conducted. Including viscosity makes a shape differentiation considerably more complex. Therefore, a frozen viscosity approach is used, meaning the shape differentiation is conducted for the mean flow of an averaged turbulent flow only. The thesis concludes with a summary in chapter 10.



Chapter 2

Differential Geometry

2.1 Basic Concepts

This chapter is used to give a very brief overview about differential geometry, preparing for the shape sensitivity analysis in chapter 3. Special attention is given to introducing terms and definitions which are seldom encountered in the context of general PDE constrained optimization, such as tangential gradient, tangential divergence, and a variety of other lemmas and definitions. More details can for example be found in [13, 71] or in numerous other works.

Definition 2.1.1 (Immersion)**.** *Let U be an open subset of $\mathbb{R}^n$. A function $h : U \to \mathbb{R}^{n+k}$ is called immersion, if $h \in C^\infty$ and* $\operatorname{rank}(Dh(x)) = n$ *for all $x \in U$.*

Definition 2.1.2 (Submanifold of $\mathbb{R}^m$, Parameterization, Chart, Co-Dimension)**.** *A set $\Omega \subset \mathbb{R}^m$ is called d-dimensional submanifold of $\mathbb{R}^m$ if for each $x \in \Omega$ there exists an open neighborhood $U_1(x) \subset \mathbb{R}^m$ and an injective immersion $h : U_2 \to \mathbb{R}^m$ with $U_2 \subset \mathbb{R}^d$ open and with continuous inverse mapping $h^{-1} : h(U_2) \to U_2$ such that*

$$h(U_2) \subset U_1 \cap \Omega.$$

Furthermore, h is called (local) parameterization, h^{-1} is called map, and the pair $(h^{-1}, h(U_2))$ is called chart. Thus, $x \in \Omega \subset \mathbb{R}^m$ is given by $x = h(\xi_1, ..., \xi_d)$ for $(\xi_1, ..., \xi_d) \in U_2 \subset \mathbb{R}^d$. The value $m - d$ is called co-dimension.

Definition 2.1.3 (Atlas)**.** *For a submanifold Ω of $\mathbb{R}^m$ the set of all charts covering Ω is called atlas*

$$A := \bigcup_{\alpha \in I} \{(h_\alpha^{-1}, U_\alpha)\},$$

where I is some index set.

Remark 2.1.4 (Surface of a Submanifold)**.** For a d-dimensional submanifold Ω of $\mathbb{R}^m$ the boundary is defined by

$$\Gamma := \partial\Omega := \bar{\Omega} \setminus \operatorname{int}\Omega,$$

where $\bar{\Omega}$ is the closure and $\operatorname{int}\Omega$ is the interior of Ω. In the following, for $\xi = (\xi_1, ..., \xi_{d-1}, \xi_d)$, the interior is thought to be given by $\xi_d > 0$, while the boundary is thought to be given by $\xi_d = 0$. Thus, the short notation $h(\xi, 0)$ is used in the following instead of $h(\xi_1, ..., \xi_{d-1}, 0)$ when referring to the boundary of Ω.

Definition 2.1.5 (Tangent Space)**.** *Let Ω be a d-dimensional submanifold of $\mathbb{R}^m$. Let (g, U) be a chart with $x \in U$. The space tangent to Ω at x is then defined as*

$$T_x\Omega := span(\{Dh(\xi, 0)e_i : i = 1, ..., d-1\}),$$

where $x = h(\xi, 0)$. Here, e_i denotes the unit vectors in $\mathbb{R}^d$.

Lemma 2.1.6 (Unit Normal Field on $\partial\Omega$)**.** *For a regular surface $\partial\Omega$, the unit normal field at $x = h(\xi, 0)$ on $\partial\Omega$ is given by*

$$n(x) = \frac{Dh(\xi, 0)^{-T} e_d}{\|Dh(\xi, 0)^{-T} e_d\|}.$$

Proof. The tangent space is given by

$$T_x\Omega = \text{span}(Dh(\xi, 0)e_i, i = 1, ..., d-1),$$

i.e. one (non-unit) tangent direction is given by $\tau_i := Dh(\xi, 0)e_i$. Hence,

$$\begin{aligned} \langle \tau_i, Dh(\xi, 0)^{-T} e_d \rangle &= \langle Dh(\xi, 0)e_i, Dh(\xi, 0)^{-T} e_d \rangle \\ &= \langle Dh(\xi, 0)^{-1} Dh(\xi, 0)e_i, e_d \rangle \\ &= \langle e_i, e_d \rangle \\ &= 0 \ \forall i = 1, ..., d-1 \end{aligned}$$

is normal to the tangent space. □

Definition 2.1.7 (Vector Field)**.** *Let $\Omega \subset \mathbb{R}^m$ be open. A (differentiable) mapping $V : \Omega \to \mathbb{R}^m$ is called a (differentiable) vector field.*

Definition 2.1.8 (Directional Derivative, Gateaux-Derivative)**.** *Let Ω be a submanifold of $\mathbb{R}^m$ and $f : \Omega \to \mathbb{R}^k$ differentiable. Furthermore, let $c : (-\epsilon, \epsilon) \to \Omega$ be a differentiable curve with $c(0) = x$ and $\dot{c}(0) = v$. We then call*

$$\frac{\partial f(x)}{\partial v} := Df(x)v := \left.\frac{d}{dt}\right|_{t=0} f(c(t))$$

the directional derivative of f in direction v, or alternatively the Gateaux-derivative. It is possible to show that the above definition does not depend on the particular choice of c.

Definition 2.1.9 (Tangential Gradient, Tangential Divergence, Curvature). *For a d-dimensional submanifold $\Omega \subset \mathbb{R}^m$ and a function $f \in C^2(\Omega, \mathbb{R})$, the tangential gradient of f is defined as the orthogonal projection of the classical gradient onto the tangent space:*

$$\nabla_\Omega f := P_T(\nabla f) = \sum_{i=1}^{d-1} \frac{\partial f}{\partial \tau_i}\tau_i \ \in \mathbb{R}^{d-1},$$

where τ_i forms an orthonormal basis of the tangent space. For a differentiable vector field V, the tangential divergence is defined by

$$\operatorname{div}_\Omega V := \sum_{i=1}^{d-1} \left\langle \frac{\partial V}{\partial \tau_i}, \tau_i \right\rangle \ \in \mathbb{R}.$$

This definition is independent of the choice of the orthonormal basis of the tangent space. Furthermore, the curvature is defined as the tangential divergence of the unit normal field:

$$\kappa := \operatorname{div}_\Omega n.$$

Remark 2.1.10. In the following, we assume that all submanifolds Ω are of co-dimension 1, such that the normal is unique and $\{n, \tau_1, ..., \tau_{d-1}\}$ forms an orthonormal basis of $\mathbb{R}^d$. The gradient ∇f can then be expressed in this basis:

$$\nabla f = \langle \nabla f, n\rangle n + \sum_{i=1}^{d-1} \langle \nabla f, \tau_i\rangle \tau_i.$$

Assuming f also exists in a neighborhood of Ω, such that $\frac{\partial f}{\partial n}$ exists, then the tangential gradient is equivalently given by

$$\nabla_\Omega f = \nabla f - \frac{\partial f}{\partial n} n$$

and likewise

$$\operatorname{div}_\Omega V = \operatorname{div} V - \langle DVn, n\rangle.$$

Lemma 2.1.11. *Tangential gradient and tangential divergence are related to each other just like their ordinary counterparts, i.e. for a differentiable scalar function f and a differentiable vector field V one has*

$$\operatorname{div}_\Omega fV = \langle \nabla_\Omega f, V\rangle + f \operatorname{div}_\Omega V.$$

Proof. A simple computation shows

$$\langle V \frac{\partial f}{\partial \tau_i}, \tau_i\rangle = \sum_{j=1}^{d-1} V^j \frac{\partial f}{\partial \tau_i} \tau_i^j = \langle \frac{\partial f}{\partial \tau_i}\tau_i, V\rangle = \langle \nabla_\Omega f, V\rangle,$$

where upper indices denote vector components. Furthermore,

$$\begin{aligned}\operatorname{div}_\Omega fV = \sum_{i=1}^{d}\left\langle \frac{\partial fV}{\partial \tau_i}, \tau_i \right\rangle &= \sum_{i=1}^{d-1}\langle V\frac{\partial f}{\partial \tau_i}, \tau_i\rangle + f\left\langle \frac{\partial V}{\partial \tau_i}, \tau_i \right\rangle \\ &= \langle \nabla_\Omega f, V\rangle + f\operatorname{div}_\Omega V.\end{aligned}$$

□

Lemma 2.1.12. *Let Ω be a d-dimensional submanifold with boundary Γ. For a differentiable scalar function f and a differentiable vector field V, the following relation holds on the boundary*

$$\operatorname{div} fV = f \operatorname{div} V + \frac{\partial f}{\partial n}\langle V, n\rangle + \langle \nabla f, V_\Gamma\rangle$$

where

$$V_\Gamma := \sum_{i=1}^{d-1}\langle V, \tau_i\rangle$$

is the tangential component of V.

Proof.

$$\begin{aligned}\operatorname{div} fV &= f \operatorname{div} V + \langle \nabla f, V\rangle \\ &= f \operatorname{div} V + \left\langle \frac{\partial f}{\partial n} n + \sum_{i=1}^{d-1}\frac{\partial f}{\partial \tau_i}\tau_i, V\right\rangle \\ &= f \operatorname{div} V + \frac{\partial f}{\partial n}\langle V, n\rangle + \langle \nabla f, V_\Gamma\rangle.\end{aligned}$$

□

Definition 2.1.13 (Tangential Jacobian Matrix)**.** *Similar to definition 2.1.9, the tangential Jacobian matrix for a differentiable vector valued function V is defined as*

$$D_\Omega V = [\nabla_\Omega V_i]_i^T ,$$

i.e. the rows of the tangential Jacobian are the tangential gradients of the respective component functions.

Remark 2.1.14. Similar to remark 2.1.10, there also exists the equality

$$D_\Omega V = \left[\sum_{k=1}^{d-1}\frac{\partial V_i}{\partial \tau_k}\tau_k\right]_i^T = \left[\nabla V_i - \frac{\partial V_i}{\partial n} n\right]_i^T = DV - DVnn^T$$

should the required derivative in normal direction exist. This property is needed later in lemma 3.3.7.

2.2 Integration over Submanifolds

Definition 2.2.1 (Integral Over Submanifolds)**.** *Let Ω be a d-dimensional compact submanifold in $\mathbb{R}^m$ with finite open atlas*

$$\Omega \subset \bigcup_{j=1}^{l} h_j(M_j)$$

such that $\Omega_j := h_j(M_j)$ and a corresponding partition of unity

$$\sum_{j=1}^{l} r_j(x) = 1$$

with r_j infinitely continuously differentiable with compact support $\subset \Omega_j$ for all j. Then, the integral over Ω is defined by

$$\begin{aligned}\int\limits_{\Omega} g \; d\Omega := \sum_{j=1}^{l} \int\limits_{\Omega_j} g r_j \; d\Omega &:= \sum_{j=1}^{l} \int\limits_{M_j} g(h_j(s)) r_j(h_j(s)) \sqrt{\det({Dh_j}^T Dh_j)(s)} \; ds \\ &=: \int\limits_{M} g(h(s)) \sqrt{\det(Dh^T Dh)(s)} \; ds,\end{aligned} \tag{2.1}$$

where Dh_j is the Jacobian of h_j.

Definition 2.2.2 (Minor, Cofactor Matrix)**.** *For a matrix $A \in R^{m \times m}$ the ij-minor*

$$[A]_{ij} \in \mathbb{R}^{m-1 \times m-1}$$

is defined as the matrix, which results from removing the i-th row and j-th column. The cofactor matrix $M(A)$ is defined by

$$M(A) := \left[(-1)^{i+j} \det([A]_{ij})\right]_{ij} \in \mathbb{R}^{m \times m}.$$

The entries of the cofactor matrix are the subdeterminants of A. For an invertible matrix A, Cramer's rule results in

$$M(A) = \det(A) A^{-T}.$$

Lemma 2.2.3 (Integral Over the Surface of Submanifolds)**.** *Let Ω be as in the definition 2.2.1. The integral over the surface of Ω is then given by*

$$\int\limits_{\partial\Omega} g \; dS = \int\limits_{B_0} g(h(s)) |\det Dh| \| (Dh)^{-T} e_d \| \; ds, \tag{2.2}$$

where $B_0 = \{\xi \in \mathbb{R}^d : \|\xi\| \leq 1, \; \xi_d = 0\}$ is the intersection of the open d-dimensional unit ball with the $\xi_d = 0$ hyperplane and e_d is the d-th unit vector.

Proof. Let $B := \{\xi \in \mathbb{R}^d : \|\xi\| \leq 1\} \subset \mathbb{R}^d$ be the open unit Ball in $\mathbb{R}^d$. The unit ball is segmented by a cut with the $\xi_d = 0$ hyperplane in

$$\begin{aligned} B_+ &:= \{\xi \in B : \xi_d > 0\} \\ B_- &:= \{\xi \in B : \xi_d < 0\} \\ B_0 &:= \{\xi \in B : \xi_d = 0\}. \end{aligned}$$

Without loss of generality, one can assume that the interior of Ω_j is given by

$$\operatorname{int} \Omega_j = h_j(B_+)$$

and consequently the boundary is given by

$$\partial\Omega_j = h_j(B_0),$$

i.e. $\partial\Omega_j = \{h_j(\xi, 0) : (\xi, 0) := (\xi_1, ..., \xi_{d-1}, 0) \in B_0\}$. Hence, for a proper computation of the surface integral it is necessary to project the integration density

$$\det({Dh_j}^T Dh_j)$$

of the volume case above to the $(\xi, 0)$-hyperplane, i.e. dropping the last column and last row from the matrix, which is the dd-minor $[{Dh_j}^T Dh_j]_{dd}$ of ${Dh_j}^T Dh_j$. By the definition of the cofactor-matrix, the determinant of the dd-minor is exactly the m_{dd}-entry of the cofactor-matrix $M({Dh_j}^T Dh_j)$. Thus, the proper integration density for the surface integral is given by

$$\begin{aligned} \sqrt{m_{dd}} &= \sqrt{e_d^T M({Dh_j}^T Dh_j) e_d} \\ &= \sqrt{e_d^T M({Dh_j}^T) M(Dh_j) e_d} \\ &= \sqrt{\|M(Dh_j)e_d\|_2^2} \\ &= \|M(Dh_j)e_d\|_2 \\ &= |\det(Dh_j)| \|Dh_j^{-T} e_d\|_2, \end{aligned}$$

where in the last line the property $M(A) = \det(A)A^{-T}$ was used. Hence, the corresponding boundary integral is given by

$$\begin{aligned} \int\limits_{\partial\Omega} g \, dS &:= \sum_{j=1}^{l} \int\limits_{\partial\Omega_j} gr_j \, dS \\ &= \sum_{j=1}^{l} \int\limits_{B_0} gr_j(h_j(s)) |\det Dh_j| \| \left(Dh_j\right)^{-T} e_d\| \, ds \\ &=: \int\limits_{B_0} g(h(s)) |\det Dh| \| (Dh)^{-T} e_d\| \, ds, \end{aligned}$$

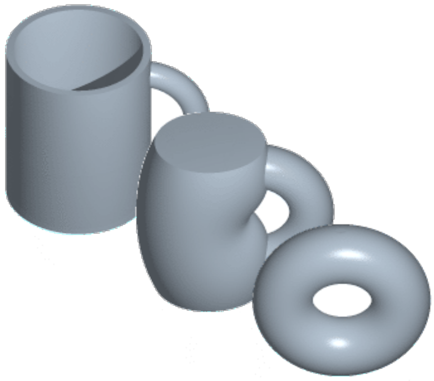
where $s = (\xi, 0) = (\xi_1, ..., \xi_{d-1}, 0)$. □

Remark 2.2.4 (Alternative Representations)**.** Since $M(A) = \det(A)A^{-T}$, the boundary integral can also be expressed as

$$\int\limits_{\partial\Omega} g \, dS = \int\limits_{B_0} g(h(s)) \| M(Dh(s)) e_d \| \, ds.$$

Analogously, the outer normal is given by

$$n(x) = \frac{M(Dh(\xi, 0)) e_d}{\| M(Dh(\xi, 0)) e_d \|_2}.$$



Chapter 3

Shape Sensitivity Analysis

3.1 Shape Optimization and Hadamard Theorem

The main purpose of this chapter is to derive the general expression of shape derivatives. Most of them are known from the literature [1, 8] and especially [11, 70]. However, listing them here will create a much more consistent work. The first part of this section formally defines a shape optimization problem. Approaches for deforming shapes are discussed next. Finally, the Hadamard formula for the shape derivative is elaborated. This formula provides a very efficient way of solving shape optimization problems numerically, as an analytic expression about how to deform the shape in order to improve the objective function is given. The focus lies on first order derivatives, but an exemplified shape Hessian derivation can be found in section 5.2 later on.

Definition 3.1.1 (Shape Functional, Shape Optimization Problem)**.** *A real-valued shape functional J is defined by*

$$\begin{aligned} J : \mathcal{P}(\mathbb{R}^d) &\to \mathbb{R} \\ \Omega &\mapsto J(\Omega), \end{aligned}$$

and a shape optimization problem is given by

$$\min_{\Omega} J(\Omega).$$

For a real valued shape functional J and a vector valued shape functional c, a constrained shape optimization problem is likewise given by

$$\begin{aligned} &\min_{(u,\Omega)} J(u,\Omega) \\ &s.t.\ c(u,\Omega) = 0. \end{aligned}$$

Here, u is called the state variable. Compared to a classical optimization problem, Ω takes the role of the control variable.

Definition 3.1.2 (Deformed Submanifold)**.** *Let $T_t : (t,x) \mapsto T_t(x)$ with $t \in \mathbb{R}$ be a family of bijective mappings. Let Ω be a closed submanifold with boundary Γ. A deformed submanifold Ω_t is given by*

$$\Omega_t := T_t(\Omega) = \{T_t(x_0) : x_0 \in \Omega\}.$$

For $x \in \Gamma$ parameterized by $x = h(\xi, 0)$ the point x_t on the deformed boundary Γ_t of Ω_t is parameterized by

$$x_t = T_t(h(\xi,0)) =: h^t(\xi,0).$$

It remains to define the actual deformation by choosing T_t. In the literature, two approaches are most common: the perturbation of identity and the speed method.

Definition 3.1.3 (Perturbation of Identity)**.** *Choosing $T_t[V]$ as*

$$T_t[V](x) = x + tV(x)$$

results in a deformation according to the perturbation of identity.

Definition 3.1.4 (Speed Method)**.** *For a sufficiently smooth vector field V, where*

$$\begin{aligned} V : \mathbb{R} \times \Omega &\to \mathbb{R}^d \\ (t,x) &\mapsto V(t,x), \end{aligned}$$

the speed method considers the flow equation

$$\frac{dx}{dt} = V(t,x),\ \ x(0) = x_0$$

and defines the family of bijective mappings as

$$T_t[V](X) := x(t,X).$$

Thus, the speed method allows non-constant perturbation fields V.

Remark 3.1.5. Both approaches are special cases of one another. They are equivalent for first order shape derivatives but not for higher derivatives.

Definition 3.1.6 (Shape Differentiability, Shape Derivative)**.** *Let $D \subset \mathbb{R}^d$ open and $\Omega \subset D$ measurable. Let V be a continuous vector field. A shape functional J is called shape differentiable at Ω, if the Eulerian derivative*

$$dJ(\Omega)[V] := \lim_{t \to 0^+} \frac{J(\Omega_t) - J(\Omega)}{t}, \quad \Omega_t := T_t(\Omega)$$

exists for all directions V and the mapping $V \mapsto dJ(\Omega)[V]$ is linear and continuous. The expression $dJ(\Omega)[V]$ is called the shape derivative of J at Ω in direction V.

The key ingredient for computing shape derivatives very efficiently is the so-called Hadamard formula.

Theorem 3.1.7 (Hadamard Theorem)**.** *Let J be shape differentiable as in definition 3.1.6. Then the relation*

$$dJ(\Omega)[V] = dJ(\Gamma)[\langle V, n \rangle n]$$

holds for all vector fields $V \in C^k(\bar{D}; \mathbb{R}^d)$.

Proof. See proposition 2.26, pages 59–60, in [70]. □

Remark 3.1.8. In reference [70], the Hadamard theorem actually states the existence of a scalar distribution

$$g(\Gamma) \in \mathcal{D}^{-k}(\Gamma),$$

such that the shape gradient $G(\Omega) \in \mathcal{D}^{-k}(\Omega, \mathbb{R}^d)$ is given by

$$G(\Omega) = \gamma_\Gamma^*(g \cdot n),$$

where γ_Γ^* is the adjoint of the trace operator on Γ. Here, however, it is always assumed that $G(\Omega)$ is an integrable function, i.e. Ω has piecewise smooth boundaries. Then the shape gradient g is much more conveniently expressed by

$$dJ(\Omega)[V] = \int_\Gamma \langle V, n \rangle \, g \, dS.$$

The requirement of piecewise smooth boundaries can for example be seen in equation (3.5).

The general strategy for solving the aerodynamic shape optimization problem considered in this work is to derive g and then conduct a gradient based optimization by discretizing g and tracking the shape by conducting updates of the type

$$\Gamma_{i+1} = \{x + \langle V(x), n(x) \rangle n(x) g(x) : \; x \in \Gamma_i\}.$$

Since g is known analytically and does not involve dependencies on the discretization of the domain, i.e. the mesh, the above update is numerically extremely cheap while also allowing maximal freedom in the deformability of the shape. Because the unit normal n changes with the shape in each iteration, updates of the above type also have the interesting side-effect that the shape Γ_i in iteration i is only expressed in terms of a deformation of the shape Γ_{i-1} from iteration $i-1$ and not from the initial shape Γ_0.

Before the shape gradient g is derived for aerodynamic shape optimization problems in chapter 5 to 9, shape calculus from the literature [1, 8] and especially [11, 70] is listed.

3.2 Hadamard Formula for Volume Objectives

When considering shape functionals of the type

$$J(\Omega) = \int\limits_{\Omega} f \, dA,$$

the integration formula in definition 2.2.1 is much more convenient. Using this definition, the integral over the deformed domain Ω_t can be brought back to the original domain.

$$\begin{aligned} \int\limits_{\Omega_t} f \, dA &= \int\limits_{\Omega} f(T_t(x)) \sqrt{\det DT_t^T DT_t(x)} \, dA(x) \\ &= \int\limits_{\Omega} f(T_t(x)) | \det DT_t(x) | \, dA(x). \end{aligned}$$

The bijective mapping T_t is assumed to preserve the orientation of Γ, i.e. $\det DT_t(x) \geq 0$ for all $x \in \Omega$, and the absolute is discarded in the following. For differentiation with respect to t, the derivative of the determinant is required.

Lemma 3.2.1 (Derivative of the Determinant)**.** *Let*

$$\begin{array}{rccc} A: & \mathbb{R} & \rightarrow & \mathbb{R}^{n \times n} \\ & t & \mapsto & A(t) \end{array}$$

be a matrix valued function on $\mathbb{R}$ with differentiable component functions. The derivative of the determinant is then given by

$$\frac{d(\det(A(t)))}{dt} = tr(A'(t) A^{-1}(t)) \det A(t).$$

Proof. Let a_i denote the columns of the matrix, i.e. $A(t) = [a_1, \dots, a_n]$. Leibniz formula for determinants results in

$$\begin{aligned} \frac{d(\det(A(t)))}{dt} &= \frac{d}{dt} \sum_{\sigma} s(\sigma) \, a_{1\,\sigma(1)} \cdot \dots \cdot a_{n\,\sigma(n)} \\ &= \sum_{\sigma} s(\sigma)(a'_{1\,\sigma(1)} \cdot a_{2\,\sigma(2)} \cdot \dots \cdot a_{n\,\sigma(n)} + \dots \\ &\qquad \dots + a_{1\,\sigma(1)} \cdot \dots \cdot a_{n-1,\,\sigma(n-1)} \cdot a'_{n,\,\sigma(n)}) \\ &= \sum_{i=1}^{n} \det(a_1, \dots, a_{i-1}, \frac{da_i}{dt}, a_{i+1}, \dots, a_n). \end{aligned}$$

Hence, for a matrix $A(t)$ with $A(t_0) = I$ one has

$$\left. \frac{d \det(A(t))}{dt} \right|_{t=t_0} = \sum_{i=1}^{n} a'_{ii}(t_0) = \mathrm{tr}(A'(t_0)).$$

Using $B(t) := A(t)A^{-1}(t_0) \Rightarrow B(t_0) = I$ results in

$$\begin{aligned} \left.\frac{d\left(\det(A(t))\det(A^{-1}(t_0))\right)}{dt}\right|_{t=t_0} &= (\det(B))'(t_0) \\ &= \operatorname{tr}(B'(t_0)) = \operatorname{tr}(A'(t_0)A^{-1}(t_0)). \end{aligned} \tag{3.1}$$

Futhermore, the product rule provides

$$\begin{aligned} \left.\frac{d}{dt}\right|_{t=t_0}\left(\det(A(t))\det(A^{-1}(t_0))\right) &= \left(\left.\frac{d}{dt}\right|_{t=0}\det(A(t))\right)\det(A^{-1}(t_0)) \\ &\quad + \det(A(t_0))\left(\left.\frac{d}{dt}\right|_{t=t_0}\det(A^{-1}(t_0))\right) \\ &= \left(\left.\frac{d}{dt}\right|_{t=0}\det(A(t))\right)\det(A^{-1}(t_0)). \end{aligned} \tag{3.2}$$

Taking (3.1) and (3.2) together, one has

$$\frac{d}{dt}\det(A(t)) = \operatorname{tr}(A'(t)A^{-1}(t))\det(A(t)).$$

□

Lemma 3.2.2 (Derivative of the Deformation Determinant)**.** *The derivative of the determinant of the perturbation of identity approach is given by:*

$$\left.\frac{d}{dt}\right|_{t=0}\det DT_t(x) = \operatorname{div} V(x). \tag{3.3}$$

Proof. Using lemma 3.2.1, one has

$$\frac{d}{dt}\det A(t) = \operatorname{tr}\left(\frac{dA(t)}{dt}A(t)^{-1}\right)\det A(t).$$

Since $DT_0(x) = I$, we have

$$\begin{aligned} \left.\frac{d}{dt}\right|_{t=0}\det DT_t(x) &= \operatorname{tr}\left(\left.\frac{dDT_t(x)}{dt}\right|_{t=0}\right) \\ &= \operatorname{tr}(DV(x)) \\ &= \operatorname{div} V(x). \end{aligned}$$

□

Theorem 3.2.3 (Divergence Theorem)**.** *Let Ω be compact with piecewise smooth boundary Γ. If F is a continuously differentiable vector field on a neighborhood of Ω, then the following relation holds:*

$$\int_\Omega \operatorname{div} F\, dA = \int_\Gamma \langle F, n\rangle\, dS.$$

Proof. The expression follows immediately from integration by parts. See also proposition 7.6.1 and theorem 13.1.2 in [5]. □

Lemma 3.2.4 (Hadamard Formula for Volume Objective Functions)**.** *For a general volume objective function $f : \Omega \to \mathbb{R}$, not depending on a PDE constraint or the shape of Ω, i.e.*

$$J(\Omega) = \int_{\Omega} f \, dA,$$

the shape derivative is given by

$$dJ(\Omega)[V] = \int_{\Gamma} \langle V, n \rangle f \, dS.$$

Proof. By definition one has

$$\begin{aligned} dJ(\Omega)[V] : &= \frac{d}{dt}\bigg|_{t=0} J(\Omega_t) := \frac{d}{dt}\bigg|_{t=0} \int_{\Omega_t} f(x) \, dA(x) \\ &= \frac{d}{dt}\bigg|_{t=0} \int_{\Omega} f(T_t(x)) \det(DT_t(x)) \, dA(x). \end{aligned}$$

Swapping differentiation and integration and applying lemma 3.2.2 leads to

$$\begin{aligned} dJ(\Omega)[V] &= \int_{\Omega} \frac{d}{dt}\bigg|_{t=0} f(T_t(x)) \det(DT_t(x)) \, dA(x) \\ &= \int_{\Omega} \langle \nabla f(x), V(x) \rangle + f(x) \mathrm{div}\, V(x) \, dA(x). \end{aligned} \tag{3.4}$$

Chain rule backwards results in

$$dJ(\Omega)[V] = \int_{\Omega} \mathrm{div}\, (f(x) V(x)) \, dA(x).$$

The final boundary formula can now be found using the divergence theorem 3.2.3. However, this requires piecewise smooth boundaries:

$$dJ(\Omega)[V] = \int_{\Gamma} \langle V, n \rangle \, f \, dS. \tag{3.5}$$

□

3.3 Hadamard Formula for Surface Objectives

The Hadamard formula for surface integrals is considerably more complex than the one for domain integrals. Comparison of equations (2.1) and (2.2) shows that the more complex integration density will create new terms in the gradient formula. Also, surface integrals often depend on additional geometric quantities such as the outer normal n, which must also be differentiated.

Lemma 3.3.1 (Perturbed Surface Integral)**.** *The surface integral over the perturbed surface Γ_t is given by*

$$\int\limits_{\Gamma_t} g \, d\Gamma_t = \int\limits_{\Gamma} g(T_t(x)) \| M(DT_t(x)) n(x) \|_2 \, d\Gamma(x),$$

where n is the unit normal of the unperturbed boundary Γ.

Proof. The perturbed submanifold Γ_t can be described by

$$h^t(\xi, 0) := T_t(h(\xi, 0)). \tag{3.6}$$

According to remark 2.2.4, the surface integral is given by

$$\int\limits_{\partial\Omega_t} g \, dS_t = \int\limits_{B_0} g(h^t(s)) \| M(Dh^t(s)) e_d \|_2 \, ds.$$

The chain rule results in

$$Dh^t(\xi, 0) = D[T_t(h(\xi, 0))] = DT_t(h(\xi, 0)) Dh(\xi, 0) \tag{3.7}$$

and

$$\begin{aligned} M(Dh^t(\xi, 0)) &= M(DT_t(h(\xi, 0) Dh(\xi, 0))) \\ &= M(DT_t(h(\xi, 0))) M(Dh(\xi, 0)). \end{aligned}$$

Using the alternative representation of the normal,

$$\begin{aligned} \| M(Dh^t(s)) e_d \|_2 &= \| M(DT_t(h(\xi, 0))) M(Dh(\xi, 0)) e_d \|_2 \\ &= \| M(DT_t(h(\xi, 0))) \| M(Dh(\xi, 0)) e_d \|_2 n(h(\xi, 0)) \|_2 \\ &= \| M(Dh(\xi, 0)) e_d \|_2 \| M(DT_t(h(\xi, 0))) n(h(\xi, 0)) \|_2. \end{aligned}$$

Thus,

$$\begin{aligned} \int\limits_{\partial\Omega_t} g \, dS_t &= \int\limits_{B_0} g(T_t(h(s)) \| M(DT_t(h(s))) n(h(s)) \|_2 \| M(Dh(s)) e_d \|_2 \, ds \\ &= \int\limits_{\partial\Omega} g(T_t(x)) \| M(DT_t(x)) n(x) \|_2 \, d\Gamma(x), \end{aligned}$$

where again $s = (\xi, 0)$ and $x = h(s)$. □

Remark 3.3.2 (Alternative Representation)**.** Due to the definition of the cofactor matrix, the perturbed surface integral can also be written as

$$\begin{aligned} \int\limits_{\partial\Omega_t} g \, dS_t &= \int\limits_{\partial\Omega} g(T_t(x)) \| M(DT_t(x)) n(x) \|_2 \, d\Gamma(x) \\ &= \int\limits_{\partial\Omega} g(T_t(x)) | \det DT_t(x) | \| (DT_t(x))^{-T} n(x) \|_2 \, d\Gamma(x). \end{aligned}$$

Since we assume the deformation mapping T_t does not change the orientation of Ω_t relative to Ω, we can assume $\det DT_t > 0$ in subsequent considerations.

Lemma 3.3.3 (Derivative Through Matrix Inverse)**.** *Let $A(t) \in \mathbb{R}^{m\times m}$ be a matrix where each entry is a differentiable function such that $A(t)^{-1}$ exists for some interval $I \subset \mathbb{R}$. The derivative of the matrix inverse with respect to t is then given by*

$$\frac{d}{dt}A(t)^{-1} = -A(t)^{-1}\frac{dA(t)}{dt}A(t)^{-1}.$$

Proof. Let $a_{ij}(t)$ be the component functions of $A(t)$ and let $a^{jk}(t)$ be the components of $A(t)^{-1}$, i.e.:

$$\begin{aligned} \delta_{ik} &= \sum_{j=1}^{m} a_{ij}(t)a^{jk}(t) \\ \Rightarrow 0 &= \sum_{j=1}^{m} \frac{da_{ij}(t)}{dt}a^{jk}(t) + a_{ij}(t)\frac{da^{jk}(t)}{dt} \\ \Rightarrow 0 &= \frac{dA(t)}{dt}A(t)^{-1} + A\frac{dA(t)^{-1}}{dt}. \end{aligned}$$

□

Lemma 3.3.4 (Preliminary Shape Derivative for Surface Objectives)**.** *For $g \in C(T(\Gamma))$, where $T(\Gamma)$ is a tubular neighborhood of Γ such that ∇g is defined on Γ, the preliminary shape derivative, not yet in Hadamard form, for the surface integral is given by*

$$\begin{aligned} \frac{d}{dt}\bigg|_{t=0}\int_{\Gamma_t} g\, dS_t &= \int_{\Gamma} \langle \nabla g, V\rangle + g\cdot \big(\operatorname{div} V - \langle DVn, n\rangle\big)\ dS \\ &= \int_{\Gamma} \langle \nabla g, V\rangle + g\operatorname{div}_{\Gamma} V\, dS. \end{aligned}$$

Proof. For simplicity reasons, perturbation of identity is assumed. The alternative representation from remark 3.3.2 provides:

$$\begin{aligned} &\frac{d}{dt}\bigg|_{t=0}\int_{\partial\Omega_t} g\, dS_t \\ &= \int_{\partial\Omega} \frac{d}{dt}\bigg|_{t=0} (g(T_t(x)) \det DT_t(x) \|(DT_t(x))^{-T} n(x)\|_2)\, d\Gamma(x). \end{aligned}$$

Furthermore,

$$\gamma(t) := DT_t^{-T} n = ((I + tDV)^T)^{-1} n$$

gives

$$\frac{d}{dt}\bigg|_{t=0} \|\gamma(t)\|_2 = \frac{d}{dt}\bigg|_{t=0}\left(\sum_{i=1}^{d} \gamma_i(t)^2\right)^{\frac{1}{2}} = \frac{1}{\|\gamma(0)\|_2}\left(\gamma^T(0)\frac{d}{dt}\bigg|_{t=0}\gamma(t)\right).$$

Due to lemma 3.3.3 one has

$$\begin{aligned}\gamma(0) &= n\\ \left.\frac{d}{dt}\right|_{t=0}\gamma(t) &= -I^{-1}\left.\frac{d}{dt}\right|_{t=0}(I+tDV)^T I^{-1}n\\ &= -DV^T n.\end{aligned}$$

Thus,

$$\left.\frac{d}{dt}\right|_{t=0}\|\gamma(t)\|_2 = -n^T DV^T n = -\langle DVn, n\rangle.$$

Using $\det DT_0 = \det I = 1$ and the product rule, the above results in

$$\begin{aligned}\left.\frac{d}{dt}\right|_{t=0}\int\limits_{\partial\Omega_t} g\ dS_t &= \int\limits_{\partial\Omega}\left[\left.\frac{d}{dt}\right|_{t=0}(g(T_t)\det DT_t)\right] n - g\cdot\langle DVn, n\rangle\ dS\\ &= \int\limits_{\partial\Omega}\langle\nabla g, V\rangle + g\cdot\big(\operatorname{div} V - \langle DVn, n\rangle\big)\ dS,\end{aligned}$$

where formula (3.3) for the determinant was used again. The final expression follows with remark 2.1.10. □

3.3.1 Shape Derivatives of Geometric Quantities

Before the construction of the Hadamard formula for surface objectives is finished, a shape sensitivity analysis of some geometric quantities, especially the outer normal n, is conducted. While the derivative of the normal is needed for many objective functions and PDE constraints in itself, the resulting tangential Stokes formula makes the Hadamard expression for a surface functional quite convenient to derive.

Lemma 3.3.5 (Unit Normal on Perturbed Domain)**.** *The unit normal on the perturbed domain Ω_t is given by*

$$n_t(T_t(x)) = \frac{(DT_t(x))^{-T}n(x)}{\|(DT_t(x))^{-T}n(x)\|_2}.$$

Proof. According to lemma 2.1.6, the unit normal on the perturbed domain is given by

$$n_t(x) = \frac{Dh^t(\xi, 0)^{-T}e_d}{\|Dh^t(\xi, 0)^{-T}e_d\|}.$$

Using equations (3.6) and (3.7) results in

$$\begin{aligned}n_t(T_t(x)) &= \frac{(DT_t(h(\xi, 0)))^{-T}(Dh(\xi, 0))^{-T}e_d}{\|(DT_t(h(\xi, 0)))^{-T}(Dh(\xi, 0))^{-T}e_d\|}\\ &= \frac{(DT_t(x))^{-T}n(x)}{\|(DT_t(x))^{-T}n(x)\|},\end{aligned}$$

where lemma 2.1.6 was used again for the unperturbed domain. □

Lemma 3.3.6 (Preliminary Shape Derivative of the Unit Normal)**.** *The preliminary shape derivative of the unit normal is given by*

$$dn[V](x) := \left.\frac{d}{dt}\right|_{t=0} n_t(T_t(x)) = \langle n, (DV(x))^T n(x)\rangle n(x) - (DV(x))^T n(x).$$

Proof. Since $DT_0(x) = I$, the quotient rule simplifies to

$$\begin{aligned} dn[V](x) := &\left(\left.\frac{d}{dt}\right|_{t=0} \left[(DT_t(x))^{-T} n(x)\right]\right) \\ &- n(x)\left(\left.\frac{d}{dt}\right|_{t=0} \|(DT_t(x))^{-T} n(x)\|_2\right). \end{aligned}$$

Using lemma 3.3.3, the above transforms to

$$dn[V](x) = n(x)\left(\left.\frac{d}{dt}\right|_{t=0} \|(DT_t(x))^{-T} n(x)\|_2\right) - (DV(x))^T n(x).$$

For any vector $v(t)$, where the components are differentiable functions, the chain rule gives

$$\left.\frac{d}{dt}\right|_{t=0} \|v(t)\|_2 = \left.\frac{d}{dt}\right|_{t=0} \left(\sum_i v_i(t)^2\right)^{\frac{1}{2}} = \frac{\langle v(0), v'(0)\rangle}{\|v(0)\|_2}.$$

Hence, for $v(t) = (DT_t(x))^{-T} n(x)$ one has $v(0) = n(x)$ and again due to lemma 3.3.3 we have $v'(0) = (DV(x))^T n(x)$, resulting in

$$\left.\frac{d}{dt}\right|_{t=0} \|DT_t(x) n(x)\|_2 = \langle n(x), (DV(x))^T n(x)\rangle,$$

which gives the desired expression. □

Unfortunately, lemma 3.3.6 does not yet fulfill the Hadamard form, and additional transformations using tangential Jacobians from definition 2.1.13 are required.

Lemma 3.3.7. *The shape derivative of the normal is equivalently given by*

$$dn[V] = -\left(D_\Gamma V\right)^T n.$$

Proof. Assuming that the perturbation field V extends into a tubular neighborhood, we have

$$D_\Gamma V = DV - DVnn^T$$

due to remark 2.1.14. Likewise,

$$\left(D_\Gamma V\right)^T n = (DV)^T n - n\,(DVn)^T n = -dn[V]$$

due to lemma 3.3.6. □

Lemma 3.3.8. *The local shape derivative of the normal $dn[V]$ at a point x lies in the tangent space $T_x\Omega$.*

Proof.

$$\begin{aligned}\langle dn[V], n\rangle =&\langle -\left(D_\Gamma V\right)^T n, n\rangle \\ =&\langle -\left(DV\right)^T n + n\left(DVn\right)^T n, n\rangle \\ =&-\left(\left(DV\right)^T n\right)^T n + \left(n\left(DVn\right)^T n\right)^T n \\ =&-n^T DVn + n^T\left(DVn\right) \\ =&0.\end{aligned}$$

□

Remark 3.3.9. The tangential Jacobian of the unit normal field $n(x)$ at a point x lies in the tangent space $T_x\Omega$, i. e.

$$0 = D_M 1 = D_M\left(n(x)^T n(x)\right) = 2\left(D_M n(x)\right) n(x) = 2\langle\nabla_M n, n\rangle,$$

meaning $D_M n \perp n$. This result is needed in the following lemma 3.3.10.

Lemma 3.3.10. *For a perturbation normal to the boundary* Γ*, i.e.* $\tilde{V} := \langle V, n\rangle n$ *or equivalently* $\langle\tilde{V}, \tau\rangle = 0$ *for a vector* $\tau \in T_x\Omega$*, we have*

$$dn[\tilde{V}] = -\nabla_\Gamma\langle\tilde{V}, n\rangle.$$

Proof. Let $\{\tau_i \in T_x\Omega : 1 \leq i \leq d-1\}$ be an orthonormal basis of the tangent space and let the unit normal be given by n with components n_k. By definition 2.1.9 one has

$$\begin{aligned}\nabla_\Gamma\langle\tilde{V}, n\rangle &= \sum_{i=1}^{d-1}\frac{\partial\langle\tilde{V}, n\rangle}{\partial\tau_i}\tau_i \\ &= \sum_{i=1}^{d-1}\frac{\partial}{\partial\tau_i}\left[\sum_{k=1}^{d}\tilde{V}_k n_k\right]\tau_i \\ &= \sum_{i=1}^{d-1}\left[\sum_{k=1}^{d}\frac{\partial\tilde{V}_k}{\partial\tau_i}n_k + \tilde{V}_k\frac{\partial n_k}{\partial\tau_i}\right]\tau_i.\end{aligned}$$

According to remark 3.3.9, the variation of the normal in tangent directions is perpendicular to the normal, and with the particular choice of $\tilde{V}$, the second part vanishes. This results in

$$\begin{aligned}\nabla_\Gamma\langle\tilde{V}, n\rangle &= \sum_{i=1}^{d-1}\sum_{k=1}^{d}\frac{\partial\tilde{V}_k}{\partial\tau_i}n_k\tau_i \\ &= (D_\Gamma\tilde{V})^T n = -dn[\tilde{V}].\end{aligned}$$

□

The idea now is to apply the preliminary shape derivative of lemma 3.3.4 to the divergence theorem of lemma 3.2.3. However, the preliminary gradient expression requires certain derivatives for which the functional under consideration must extend into a tubular neighborhood of Γ. Unfortunately, this is not true for the outer normal n, such that an extension of the normal into a tubular neighborhood is needed.

Remark 3.3.11. When considering the shape functional

$$J(g,\Gamma) = \int\limits_{\Gamma} g(\varphi, n)\ dS,$$

where

$$\begin{aligned} g : \mathbb{R}^d \times \mathbb{R}^d &\to \mathbb{R} \\ (\varphi, \psi) &\mapsto g(\varphi, \psi) \end{aligned}$$

is a sufficiently smooth function, the preliminary gradient for surface objectives, lemma 3.3.4, requires the existence of the total derivative $\langle \nabla g(\varphi, \psi), V \rangle$. For the expression $g(\varphi, n)$ this existence is not given and a smooth unitary extension $\mathcal{N}$ of the unit normal n into a tubular neighborhood of Γ is needed. Just as in remark 3.3.9, this extension satisfies

$$0 = D\,1 = D\ \left(\mathcal{N}(x)^T \mathcal{N}(x)\right) = 2\,(D\mathcal{N}(x))\,\mathcal{N}(x) = 2\langle \nabla \mathcal{N}, \mathcal{N} \rangle$$

in the domain Ω. For more details see [70]. A popular choice for this extension $\mathcal{N}$ is the normalized gradient of the signed distance function $\nabla b / \|\nabla b\|$ due to the applicability in level-set methods [36]. The tangential Stokes formula can now be used to perform an integration by parts on surfaces to arrive at more convenient expressions for surface shape functionals.

Lemma 3.3.12 (Tangential Stokes Formula)**.** *Let g be a real valued differentiable function on Γ and v be a differentiable vector valued function on Γ. Then the following relation holds:*

$$\int\limits_{\Gamma} g \mathrm{div}_{\Gamma}\, v + \langle \nabla_{\Gamma} g, v \rangle\ dS = \int\limits_{\Gamma} \kappa\ g\ \langle v, n \rangle\ dS.$$

Proof. Applying the Hadamard formula for volume objectives, lemma 3.2.4, to the left side of the divergence theorem, lemma 3.2.3 and the preliminary gradient expression of lemma 3.3.4 to the right side, the expression

$$\int\limits_{\Gamma} \langle V, n \rangle \mathrm{div}\, F\ dS = \int\limits_{\Gamma} \langle \nabla \langle F, \mathcal{N} \rangle, V \rangle + \langle F, n \rangle\, (\mathrm{div}_{\Gamma}\, V) + \langle F, dn[V] \rangle\ dS$$

is created. The shape derivative of the normal $dn[V]$ enters due to the chain rule. Choosing $V = \mathcal{N}$ and applying lemma 3.3.7 result in

$$\int\limits_{\Gamma} \mathrm{div}\, F\ dS = \int\limits_{\Gamma} \langle \nabla \langle F, \mathcal{N} \rangle, \mathcal{N} \rangle + \langle F, \mathcal{N} \rangle\, (\mathrm{div}_{\Gamma}\, \mathcal{N})\ \ dS,$$

because $D\mathcal{N}\mathcal{N} = 0$. The above now transforms into

$$\int\limits_{\Gamma} \mathrm{div}\, F\ dS = \int\limits_{\Gamma} \langle DFn, n \rangle + \langle F, n \rangle \kappa\ dS.$$

Because $\mathrm{div}_{\Gamma}\, F = \mathrm{div}\, F - \langle DFn, n \rangle$, the desired expression is created by choosing $F := g \cdot v$ for a scalar g and a vector v. □

3.3.2 Shape Derivatives of General Surface Objectives

Using the tangential Stokes formula, the preliminary gradient expression from lemma 3.3.4 can now be brought into Hadamard form.

Lemma 3.3.13 (Hadamard Formula for Surface Objectives)**.** *For a general surface objective function $g : T(\Gamma) \to \mathbb{R}$, which is independent of the shape and for which $\frac{\partial g}{\partial n}$ exists, the shape derivative for the surface objective*

$$J(\Omega) := \int\limits_{\Gamma} g \, dS$$

is given by

$$dJ(\Omega)[V] = \int\limits_{\Gamma} \langle V, n\rangle \left[\frac{\partial g}{\partial n} + \kappa g\right] \, dS,$$

where $\kappa = \operatorname{div}_\Gamma n$ is the tangential divergence of the normal, i.e. the additive mean curvature of Γ.

Proof. Starting form the preliminary gradient of lemma 3.3.4, the derivative is given by

$$\begin{aligned}\left.\frac{d}{dt}\right|_{t=0} \int\limits_{\partial\Omega_t} g \, dS_t &= \int\limits_{\partial\Omega} \langle \nabla g, V\rangle + g\left(\operatorname{div} V - \langle DVn, n\rangle\right) \, dS \\ &= \int\limits_{\partial\Omega} \langle \nabla g, V\rangle + g\operatorname{div}_\Gamma V \, dS.\end{aligned}$$

The desired expression is immediately created due to the tangential Stokes formula, lemma 3.3.12 and the tangential quantities from definition 2.1.9 and remark 2.1.10. □

Lemma 3.3.14 (Hadamard Formula of the Shape Derivative of the Normal)**.** *Let the objective function be given by*

$$J(g, \Gamma) := \int\limits_{\Gamma} g(\varphi, D\varphi, n) \, dS,$$

where $g : \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \mathbb{R}^d \to \mathbb{R}, (\varphi, \zeta, \psi) \mapsto g(\varphi, \zeta, \psi)$ is a sufficiently smooth functional. The shape derivative of the above expression is then given by

$$dJ(g, \Gamma)[V] = \int\limits_{\Gamma} \langle V, n\rangle \left[D_\varphi g D\varphi \; n + D_\zeta g D^2\varphi \; n + \kappa\left(g - D_\psi g \; n\right) + \operatorname{div}_\Gamma \; \left(D_\psi g\right)^T\right] \, dS.$$

Proof. To ensure applicability of the Hadamard formula for boundary integrals, lemma 3.3.13, the objective

$$J(g, \Gamma) := \int\limits_{\Gamma} g(\varphi, D\varphi, \mathcal{N}) \, dS$$

is considered. Here, $\mathcal{N}$ is a unitary extension of the normal into Ω just as in remark 3.3.11. By construction, the extension fulfills $\mathcal{N} = n$ and $d\mathcal{N}[V] = dn[V]$ on Γ. The chain rule and lemma 3.3.13 then provide

$$dJ(g,\Gamma)[V] = \int_\Gamma \langle V, n\rangle \left[\langle \nabla g(\varphi, D\varphi, \mathcal{N}), n\rangle + \kappa g(\varphi, D\varphi, n)\right] + D_\psi g(\varphi, D\varphi, n)\, dn[V]\; dS.$$

The chain rule also leads to

$$\begin{aligned}\langle \nabla g(\varphi, D\varphi, \mathcal{N}), n\rangle &= Dg(\varphi, D\varphi, \mathcal{N})n\\ &= \left(D_\varphi g(\varphi, D\varphi, \mathcal{N})D\varphi + (D_\zeta g(\varphi, D\varphi, \mathcal{N})D^2\varphi + D_\psi g(\varphi, D\varphi, \mathcal{N})D\mathcal{N}\right) n\\ &= D_\varphi g(\varphi, D\varphi, \mathcal{N})D\varphi\; n + D_\zeta g(\varphi, D\varphi, \mathcal{N})D^2\varphi\; n + D_\psi g(\varphi, \mathcal{N})D\mathcal{N}\mathcal{N}\\ &= D_\varphi g(\varphi, D\varphi, \mathcal{N})D\varphi\; n + D_\zeta g(\varphi, D\varphi, \mathcal{N})D^2\varphi\; n,\end{aligned}$$

where the third part vanishes due to remark 3.3.11. Let $\tilde{V} := \langle V, n\rangle n$ be the perpendicular component of V. Applying lemma 3.3.10 and inserting the above results in

$$dJ(g,\Gamma)[\tilde{V}] = \int_\Gamma \langle \tilde{V}, n\rangle \left[D_\varphi g D\varphi\; n + D_\zeta g D^2\varphi\; n + \kappa g\right] - D_\psi g\, \nabla_\Gamma \langle \tilde{V}, n\rangle\; dS.$$

The tangential Stokes formula, lemma 3.3.12, gives

$$\int_\Gamma -D_\psi g \nabla_\Gamma \langle \tilde{V}, n\rangle\; dS = \int_\Gamma -\kappa \langle \tilde{V}, n\rangle D_\psi g\, n + \langle \tilde{V}, n\rangle \operatorname{div}_\Gamma \left(D_\psi g\right)^T\; dS,$$

which results in

$$dJ(g,\Gamma)[\tilde{V}] = \int_\Gamma \langle \tilde{V}, n\rangle \left[D_\varphi g D\varphi\; n + D_\zeta g D^2\varphi\; n + \kappa\left(g - D_\psi g\; n\right) + \operatorname{div}_\Gamma \left(D_\psi g\right)^T\right]\; dS.$$

According to the Hadamard theorem 3.1.7, the shape derivative depends only on the normal component of V. Hence, one has

$$dJ(g,\Gamma)[\tilde{V}] = dJ(g,\Gamma)[V],$$

and the above becomes the desired expression. □

Remark 3.3.15. Two objective functions often encountered are

$$\begin{aligned}J_1(\varphi_1, \Gamma) &:= \int_\Gamma \langle \varphi_1, n\rangle\; dS\\ J_2(\varphi_2, \Gamma) &:= \int_\Gamma \langle \nabla\varphi_2, n\rangle\; dS,\end{aligned}$$

where φ_1 is a vector and φ_2 is a scalar. As seen in remark 4.1.12 in chapter 4, J_1 and J_2 are closely related to the viscous and inviscid parts of the aerodynamic drag. Using the notation from lemma 3.3.14, one has for J_1

$$\begin{aligned} g(\varphi, \zeta, \psi) &= \langle \varphi, \psi \rangle \\ D_\varphi g &= \psi^T \\ D_\zeta g &= 0 \\ D_\psi g &= \varphi^T. \end{aligned}$$

Thus, the shape derivative is given by

$$\begin{aligned} dJ_1(\varphi_1, \Gamma)[V] &= \int\limits_\Gamma \langle V, n \rangle \left[n^T D\varphi_1 n + \kappa \left(\varphi_1^T n - \varphi_1^T n \right) + \operatorname{div}_\Gamma \varphi_1 \right] \, dS \\ &= \int\limits_\Gamma \langle V, n \rangle \left[\langle D\varphi_1 n, n \rangle + \operatorname{div}_\Gamma \varphi_1 \right] \, dS. \end{aligned}$$

For J_2 one has

$$\begin{aligned} g(\varphi, \zeta, \psi) &= \langle \zeta, \psi \rangle \\ D_\varphi g &= 0 \\ D_\zeta g &= \psi^T \\ D_\psi g &= \zeta^T, \end{aligned}$$

and the shape derivative is analogously given by

$$\begin{aligned} dJ_2(\varphi_2, \Gamma)[V] &= \int\limits_\Gamma \langle V, n \rangle \left[n^T D^2 \varphi_2 n + \kappa \left(\langle \nabla \varphi_2, n \rangle - D\varphi_2 n \right) + \operatorname{div}_\Gamma \nabla \varphi_2 \right] \, dS \\ &= \int\limits_\Gamma \langle V, n \rangle \left[\langle D^2 \varphi_2 n, n \rangle + \operatorname{div}_\Gamma \left(\nabla_\Gamma \varphi_2 + \frac{\partial \varphi_2}{\partial n} n \right) \right] \, dS \\ &= \int\limits_\Gamma \langle V, n \rangle \left[\langle D^2 \varphi_2 n, n \rangle + \kappa \frac{\partial \varphi_2}{\partial n} + \operatorname{div}_\Gamma \nabla_\Gamma \varphi_2 \right] \, dS. \end{aligned}$$

Finally, for the objective

$$J_3(\varphi_3, \Gamma) := \int\limits_\Gamma \langle \nabla \varphi_3, n \rangle^p \, dS$$

one has

$$\begin{aligned} g(\varphi, \zeta, \psi) &= \langle \zeta, \psi \rangle^T \\ D_\varphi g &= 0 \\ D_\zeta g &= p \langle \zeta, \psi \rangle^{p-1} \psi^T \\ D_\psi g &= p \langle \zeta, \psi \rangle^{p-1} \zeta^T, \end{aligned}$$

and lemma 3.3.14 provides

$$dJ(\varphi_3,\Gamma)[V] = \int\limits_\Gamma \langle V,n\rangle \left[p\langle\nabla\varphi_3,n\rangle^{p-1} n^T D^2\varphi_3 n + \kappa\left(\langle\nabla\varphi_3,n\rangle^p - p\langle\nabla\varphi_3,n\rangle^{p-1} D\varphi_3 n\right)\right]\, dS$$

$$+ \int\limits_\Gamma \langle V,n\rangle \mathrm{div}_\Gamma\,\left(p\langle\nabla\varphi_3,n\rangle^{p-1}\nabla\varphi_3\right)\, dS.$$

Furthermore, there is the equality

$$\begin{aligned}&\mathrm{div}_\Gamma\,\left(p\langle\nabla\varphi_3,n\rangle^{p-1}\nabla\varphi_3\right)\\ =&\mathrm{div}_\Gamma\,\left(p\langle\nabla\varphi_3,n\rangle^{p-1}\left(\nabla_\Gamma\varphi_3 + \frac{\partial\varphi_3}{\partial n}n\right)\right)\\ =&\mathrm{div}_\Gamma\,\left(p\langle\nabla\varphi_3,n\rangle^{p-1}\nabla_\Gamma\varphi_3\right) + p\langle\nabla\varphi_3,n\rangle^{p-1}\frac{\partial\varphi_3}{\partial n}\kappa,\end{aligned}$$

which results in

$$dJ(\varphi_3,\Gamma)[V] = \int\limits_\Gamma \langle V,n\rangle \left[p\langle\nabla\varphi_3,n\rangle^{p-1}\langle D^2\varphi_3 n,n\rangle + \kappa\langle\nabla\varphi_3,n\rangle^p + \mathrm{div}_\Gamma\,\left(p\langle\nabla\varphi_3,n\rangle^{p-1}\nabla_\Gamma\varphi_3\right)\right]\, dS.$$

3.4 Shape Derivatives and State Constraints

Definition 3.4.1 (Material Derivative, Local Derivative)**.** *Let u_t solve a PDE constraint on the perturbed domain $\Omega_t = T_t[V](\Omega)$ and let $x_t := T_t(x)$ be a shifted boundary point. The material derivative is then defined as the total derivative*

$$du[V](x) := \left.\frac{d}{dt}\right|_{t=0} u_t(x_t),$$

and the local shape derivative is defined as the partial derivative

$$u'[V](x) := \left.\frac{d}{dt}\right|_{t=0} u_t(x).$$

Remark 3.4.2. A straight forward linearization of the PDE boundary conditions usually results in an expression for the material derivative. The general strategy when deriving shape derivatives is to first transfer the problem back to the original boundary before computing the limit, resulting in the need to compute the local shape derivative. The chain rule combines both by the relation:

$$du[V] = u'[V] + \langle\nabla u, V\rangle.$$

Thus, if the right hand side of the boundary condition does not depend on the geometry, one has

$$du_b[V] = \langle\nabla u_b, V\rangle.$$

In the presence of a state constraint, i.e.

$$
\begin{aligned}
\min_{(u,\Omega)} J(u,\Omega) &:= \int_\Omega f_1(u)\, dA + \int_\Gamma f_2(u)\, dS \\
&\text{subject to} \\
L(u) &= u_f \quad \text{in} \quad \Omega \\
L_b(u) &= u_b \quad \text{on} \quad \Gamma,
\end{aligned}
$$

where f_1, f_2, u_f, and u_b do not depend on the shape, the chain rule immediately results in

$$
\begin{aligned}
dJ(u,\Omega) := & \int_\Gamma \langle V, n\rangle \left[f_1(u) + \frac{\partial f_2(u)}{\partial n} + \kappa f_2(u) \right] dS \\
& + \int_\Omega \frac{\partial f_1(u)}{\partial u} u'[V]\, dA + \int_\Gamma \frac{\partial f_2(u)}{\partial u} u'[V]\, dS \\
\text{subject to} & \\
& L(u) = u_f \quad \text{in} \quad \Omega \\
& L_b(u) = u_b \quad \text{on} \quad \Gamma \\
& \frac{\partial L(u)}{\partial u} u'[V] = 0 \quad \text{in} \quad \Omega \\
& \text{Boundary condition for } u'[V] \quad \text{on} \quad \Gamma.
\end{aligned}
$$

The above does not yet fulfill the Hadamard form that can now be found by the adjoint approach. Crucial for the adjoint approach is knowing the boundary conditions of the linearized problem which determines the local shape derivative $u'[V]$ of the state.

Lemma 3.4.3 (Shape Derivative of the Dirichlet Boundary Condition)**.** *Suppose the state u is given as the solution of a PDE of the form*

$$
\begin{aligned}
L(u) &= u_f \quad \textit{in} \quad \Omega \\
u &= u_b \quad \textit{on} \quad \partial\Omega,
\end{aligned}
$$

such that u_f and u_b do not depend on the geometry of Ω, e.g. the unit normal n. The local shape derivative under the perturbation V is then given as the solution of the problem

$$
\begin{aligned}
\frac{\partial L(u)}{\partial u} u'[V] &= 0 && \textit{in} \quad \Omega \\
u'[V] &= \langle V, n\rangle \frac{\partial (u_b - u)}{\partial n} && \textit{on} \quad \Gamma,
\end{aligned}
$$

where Γ is the variable part of the boundary of $\partial\Omega$.

Proof. The linearization in Ω is straight forward. Taking the total derivative of the boundary condition results in

$$
du[V] = du_b[V] \quad \text{on} \quad \Gamma.
$$

Using definition 3.4.1, the above can be transformed to

$$\begin{aligned} u'[V] + \langle \nabla u, V \rangle &= du[V] = du_b[V] = \langle \nabla u_b, V \rangle \\ \Rightarrow u'[V] &= \langle \nabla (u_b - u), V \rangle . \end{aligned}$$

The usual orthogonality argument gives the desired expression

$$u'[V] = \langle V, n \rangle \left(\frac{\partial (u_b - u)}{\partial n} \right).$$

□

Lemma 3.4.4 (Shape Derivative of the Slip Boundary Condition)**.** *The slip-boundary condition is often encountered in fluid dynamics, especially when an inviscid fluid is modeled:*

$$\langle u, n \rangle (x) = 0 \quad on \quad \Gamma.$$

The local shape derivative then satisfies the boundary condition

$$\begin{aligned} \langle u'[V], n \rangle &= -\langle DuV, n \rangle - \langle u, dn[V] \rangle \\ &= -\langle V, n \rangle \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \langle u, \nabla_\Gamma \langle V, n \rangle \rangle , \end{aligned}$$

where the second part of the identity holds for a perturbation in normal direction only and can be brought into Hadamard form using lemma 3.3.12.

Proof. The derivation is analog to lemma 3.4.3 using the product rule and the extension of definition 3.4.1 for a vector valued state u:

$$du[V] = u'[V] + \langle Du, V \rangle.$$

□

Lemma 3.4.5 (Shape Derivative of the Neumann Boundary Condition)**.** *Suppose the state u is given as the solution of a PDE of the form*

$$\begin{aligned} L(u) &= u_f \quad in \quad \Omega \\ \frac{\partial u}{\partial n} &= u_b \quad on \quad \partial\Omega, \end{aligned}$$

such that u_f and u_b do not depend on the geometry of Ω, e.g. the unit normal n, etc. The local shape derivative under the perturbation V is then given as the solution of the problem

$$\begin{aligned} \frac{\partial L(u)}{\partial u} u'[V] &= 0 \; in \, \Omega \\ \frac{\partial u'[V]}{\partial n} &= \langle \nabla u_b, V \rangle - \langle D^2 u V, n \rangle - \langle \nabla_\Gamma u, dn[V] \rangle \\ &= \langle V, n \rangle \left[\frac{\partial u_b}{\partial n} - \frac{\partial^2 u}{\partial n^2} \right] + \langle \nabla_\Gamma u, \nabla_\Gamma \langle V, n \rangle \rangle , \end{aligned}$$

where the second identity holds for the orthogonal component of the perturbation field only.

Proof. The Neumann boundary condition at $x_t = T_t(x)$ on the deformed domain Ω_t reads

$$\begin{aligned} u_b \circ x_t &= \langle \nabla u_t, n_t \rangle \circ x_t \\ &= \langle \nabla u_t, n_t \rangle \circ T_t(x) \\ &= \langle (\nabla u_t) \circ T_t(x), n_t(x_t) \rangle. \end{aligned}$$

The chain rule results in

$$\begin{aligned} \nabla(u_t \circ T_t(x)) &= ((\nabla u_t) \circ T_t(x))^T \cdot DT_t(x) \\ &= (DT_t(x))^T \cdot [(\nabla u_t) \circ T_t(x)]\,, \end{aligned}$$

and the boundary condition becomes

$$\begin{aligned} u_b(x_t) &= \langle (DT_t(x))^{-T}\, \nabla(u_t \circ T_t(x)), n_t(x_t) \rangle \\ &= (\nabla(u_t(x_t)))^T\, DT_t(x)^{-1} \cdot n_t(x_t). \end{aligned}$$

The total derivative with respect to t now yields the material derivative of $u_t(x_t)$. Using lemma 3.3.3 results in:

$$du_b[V] = (\nabla du[V])^T\, n + (\nabla u)^T\, (-DV)\, n + \langle \nabla u, dn[V] \rangle,$$

which results in

$$\frac{\partial du[V]}{\partial n} = du_b[V] - \langle \nabla u, (-DV)\, n \rangle - \langle \nabla u, dn[V] \rangle.$$

Using the relationship

$$\begin{aligned} du[V] &= u'[V] + \langle \nabla u, V \rangle \\ du_b[V] &= \langle \nabla u_b, V \rangle \\ dn[V] &= -\nabla_\Gamma \langle V, n \rangle, \end{aligned}$$

we have

$$\frac{\partial du[V]}{\partial n} = \frac{\partial u'[V]}{\partial n} + \langle D^2 u V, n \rangle + \langle \nabla u, DVn \rangle,$$

and the above can now be expressed in terms of the local shape derivatives:

$$\frac{\partial u'[V]}{\partial n} = \langle \nabla u_b, V \rangle - \langle D^2 u V, n \rangle - \langle \nabla u, dn[V] \rangle.$$

Since $\langle \nabla u, n \rangle = 0$, we have $\nabla u = \nabla_\Gamma u$, and with the usual orthogonality argument the boundary condition can be expressed as

$$\frac{\partial u'[V]}{\partial n} = \langle V, n \rangle \left[\frac{\partial u_b}{\partial n} - \frac{\partial^2 u}{\partial n^2} \right] + \langle \nabla_\Gamma u, \nabla_\Gamma \langle V, n \rangle \rangle,$$

where the last part can be brought into Hadamard form using lemma 3.3.12, i.e.

$$\begin{aligned}\int\limits_{\Gamma} \langle \nabla_\Gamma \langle V, n\rangle, \nabla_\Gamma u\rangle \; dS &= \int\limits_{\Gamma} -\langle V, n\rangle \mathrm{div}_\Gamma\, \nabla_\Gamma u + \kappa \langle V, n\rangle \langle \nabla_\Gamma u, n\rangle \; dS \\ &= \int\limits_{\Gamma} \langle V, n\rangle \left[\kappa \langle \nabla_\Gamma u, n\rangle - \Delta_\Gamma u\right] \; dS.\end{aligned}$$

□

Remark 3.4.6. Note that in the setting considered in the above lemma 3.4.5, it can be possible that the problem does not possess a unique solution u. However, this has no consequence for the shape derivative of the Neumann boundary condition.

Remark 3.4.7. A simpler formula than lemma 3.4.5 can be given in the special case of the standard Laplace problem

$$\begin{aligned} -\Delta u &= u_f \quad \text{in} \quad \Omega \\ \frac{\partial u}{\partial n} &= u_b \quad \text{on} \quad \partial\Omega. \end{aligned}$$

The Laplace-Beltrami operator

$$\Delta_\Gamma u := \mathrm{div}_\Gamma\, \nabla_\Gamma u = \Delta u - \kappa \frac{\partial u}{\partial n} - \frac{\partial^2 u}{\partial n^2}$$

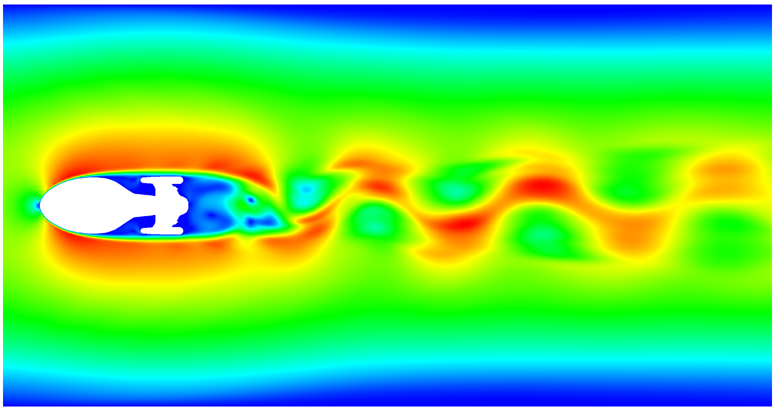
provides

$$\frac{\partial^2 u}{\partial n^2} = -\Delta_\Gamma u - u_f - \kappa u_b$$

which results in

$$\frac{\partial u'[V]}{\partial n} = \mathrm{div}_\Gamma\, \left(\langle V, n\rangle \nabla_\Gamma u\right) + \langle V, n\rangle \left(\frac{\partial u_b}{\partial n} + \kappa u_b + u_f\right).$$

For more details see [70].



Chapter 4

Fluid Mechanics

4.1 Derivation of the State Equations

Before considering shape optimization in fluids, this chapter is used to give a brief overview about partial differential and integral equations governing fluid flow. First, the governing equations are derived in a general setting. Afterwards, possible simplifications of inviscid or incompressible flows are introduced. More detailed overviews about the derivation of the state equations can for example also be found in [7, 20]. The derivation of the partial differential and integral equations describing fluids are a direct consequence of the continuum hypothesis, conservation of mass, conservation of momentum, and conservation of energy.

For consistency reasons with the literature the nomenclature is redefined. For example, t is now used to denote the physical time as opposed to being responsible for the amount of shape deformation, for which the symbol was used in chapter 3.

Definition 4.1.1 (Intensive and Extensive Quantity)**.** *A physical property is called intensive if it is scale invariant, meaning it does not depend on the system size or the amount of material in the system. Examples of intensive properties are temperature, density, or specific energy. By contrast, a property is called extensive if it does depend on scale, such as mass, length, volume, enthalpy, or energy. Let ϕ be an intensive quantity. The corresponding extensive quantity φ is then given by*

$$\varphi = \int\limits_M \rho\phi \, dA, \tag{4.1}$$

where M is a control volume under consideration and ρ is the fluid density.

Remark 4.1.2 (Reynolds Transport Theorem)**.** The Reynolds transport theorem is a three-dimensional generalization of the Leibniz integral rule for differentiation under the integral sign. It relates the change of extensive quantities to the change of intensive quantities by

$$\frac{d\varphi}{dt} = \int\limits_M \frac{\partial}{\partial t}(\rho\phi)\; dA + \int\limits_{\partial M} \langle \rho\phi \cdot (u - u_b), n\rangle\; dS,$$

where φ is the extensive quantity under consideration and ϕ is the corresponding intensive quantity. The fluid density is denoted by ρ, and the fluid velocity is given by u. The velocity of the control surface ∂M is given by u_b. More details can be found in [7].

Lemma 4.1.3 (Conservation of Mass)**.** *Let $M \subset \Omega$ be an arbitrary control volume. The conservation of mass results in the first state equation*

$$\int\limits_M \frac{\partial\rho}{\partial t} + \operatorname{div}\,(\rho u)\; dA = 0. \tag{4.2}$$

Proof. The mass m of a fluid contained in the volume M is given by

$$m = \int\limits_M \rho\; dA,$$

thus, when comparing the above with equation (4.1), one can see that mass is the extensive quantity corresponding to the intensive quantity $\phi = 1$. The mass of the fluid in a fixed control volume is considered to be conserved, resulting in

$$0 = \frac{dm}{dt} = \frac{d}{dt}\int\limits_M \rho\; dA.$$

Considering a fixed control volume, i.e. $u_b = 0$, a straight application of the Reynolds transport theorem results in

$$0 = \frac{dm}{dt} = \int\limits_M \frac{\partial\rho}{\partial t}\; dA + \int\limits_{\partial M} \langle \rho u, n\rangle\; dS.$$

The desired expression follows with remark 3.2.3. □

Lemma 4.1.4 (Conservation of Momentum)**.** *The conservation of momentum results in the second state equation governing fluid flow:*

$$\int\limits_M \frac{\partial}{\partial t}(\rho u_i) + \operatorname{div}\,(-\mathbb{T}_i + \rho u_i u) + \frac{\partial p}{\partial x_i}\; dA = \int\limits_M \rho g_i\; dA, \tag{4.3}$$

where $i = 1, 2, 3$ are the three spacial dimensions and $\mathbb{T}_i \in \mathbb{R}^3$ is the corresponding stress tensor row describing the distortion of the control volume M under forces. The fluid pressure is denoted by p. Also, g_i is the volume force in the i-th coordinate direction.

Proof. Newton's second law states that the change of momentum equals the sum of all active forces:

$$\sum \tilde{g} = \frac{d(mu)}{dt} = \frac{d}{dt} \int\limits_M \rho u \, dA,$$

where $\tilde{g}$ are the total forces acting on the control volume M. Hence, remark 4.1.2 is applicable with $\phi := u_i$ and results in

$$\left(\sum \tilde{g}\right)_i = \int\limits_M \frac{\partial}{\partial t}(\rho u_i) \, dA + \int\limits_{\partial M} \langle \rho u_i u, n \rangle \, dS.$$

However, the total forces $\tilde{g}$ acting on the control volume also depend on the pressure and stresses contained within the fluid. In order to arrive at a closed system of equations, the internal quantities pressure and stress on the boundary ∂M of the control volume must be related to the external forces $\tilde{g}$:

$$\sum \tilde{g} = \int\limits_M \rho g \, dA + \int\limits_{\partial M} \langle \sigma, n \rangle \, dS \in \mathbb{R}^3. \tag{4.4}$$

Here, g is the volume force and $\sigma \in \mathbb{R}^{3\times 3}$ is the corresponding stress tensor describing the internal friction and pressure. The stress tensor σ can now be split into two terms:

$$\begin{aligned}\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} &= \begin{bmatrix} \sigma_{11} + p & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} + p & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} + p \end{bmatrix} - \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} \\ &=: \mathbb{T} - p \cdot I.\end{aligned}$$

The second part containing the pressure p is called mean hydrostatic stress tensor, volumetric stress tensor, or mean normal stress tensor and is responsible for a change of volume of the control volume M. The first part part $\mathbb{T}$ is called stress deviator tensor and results in a distortion of the control volume M. Thus, the preliminary conservation of momentum for the i-th coordinate direction becomes

$$\int\limits_M \frac{\partial}{\partial t}(\rho u_i) \, dA + \int\limits_{\partial M} \langle -\mathbb{T}_i + \rho u_i u + p \cdot e_i, n \rangle \, dS = \int\limits_M \rho g_i \, dA,$$

where e_i is the i-th unit vector. The desired expression follows with remark 3.2.3. □

Definition 4.1.5 (Newtonian Fluid)**.** *In order to close the momentum equations, the stress tensor $\mathbb{T}$ must be related to the other variables. Let the strain tensor of the fluid be given by*

$$S = \frac{1}{2}\left[\left(\nabla u + (\nabla u)^T\right) + \lambda \, (\operatorname{div} u) \cdot I\right]$$

where $I \in \mathbb{R}^{3\times 3}$ is the identity matrix and $\lambda = -\frac{2}{3}$ is the bulk viscosity. For Newtonian fluids the following assumptions are generally made:

- *The stress tensor $\mathbb{T}$ is a linear function of the strain S.*
- *The fluid is isotropic.*

- *A fluid at rest does not distort control volumes.*

The last assumption links the equations of fluid dynamics to hydrostatics. As a consequence, the stress tensor must be divergence free, i.e. $\operatorname{div} \mathbb{T} = 0$ *for a fluid at rest. For a Newtonian fluid it is therefore often assumed that*

$$\mathbb{T} := \tau := 2\mu S,$$

where μ is the shear or dynamic viscosity of the fluid, which is related to the kinematic viscosity ν by

$$\nu = \frac{\mu}{\rho}.$$

Viscosity describes the ratio of viscous forces to inertia forces. Dynamic viscosity μ and bulk viscosity λ are closely related to the Lamé coefficients in linear elasticity. A wide array of real world fluids are considered to be Newtonian such as water, air, and oil. Examples for non-Newtonian fluids are polymer solutions, blood, paint, quicksand, toothpaste, and ketchup. Thus, for Newtonian fluids the conservation of momentum in the i-th coordinate direction is given by

$$\int_M \frac{\partial}{\partial t}(\rho u_i) + \sum_{j=1}^{3} \left[\frac{\partial}{\partial x_j} \left(-\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{2}{3} (\operatorname{div} u)\, \delta_{ij} \right) \right) + \frac{\partial(\rho u_i u_j)}{\partial x_j} \right] + \frac{\partial p}{\partial x_i}\, dA = \int_M \rho g_i \, dA, \quad (4.5)$$

where δ_{ij} is the Kronecker symbol.

Lemma 4.1.6 (Conservation of Energy)**.** *The conservation of energy results in the following equation of state:*

$$\int_M \frac{\partial}{\partial t}(\rho E) + \operatorname{div}(\rho H u - \kappa \nabla T - \mathbb{T} u)\, dA = \int_M \rho g u \, dA,$$

where H is the enthalpy, T the absolute static temperature, and κ is the thermal conductivity of the fluid.

Proof. The total energy per unit mass E of a fluid is given by kinetic energy and internal energy e

$$E = e + \frac{1}{2}\|u\|^2 = e + \frac{1}{2}\left(u_1^2 + u_2^2 + u_3^2\right).$$

The change of energy in a control volume equals the work performed by exterior forces plus heat supply:

$$\frac{d}{dt}\int_M \rho E \, dA = \underbrace{\int_M \rho g u \, dA}_{\text{work performed by volume forces}} - \underbrace{\int_{\partial M} q n \, dA}_{\text{heat supply over the boundary}} + \underbrace{\int_{\partial M} (\mathbb{T} - p \cdot I)\, u n \, dS}_{\text{work performed by surface forces}}.$$

Fourier's law of heat conduction states that

$$q = -\kappa \nabla T,$$

where κ is the thermal conductivity of the fluid and T is the absolute static temperature. Applying the Reynolds transport theorem 4.1.2 with $\phi = E$ results in

$$\int_M \frac{\partial}{\partial t}(\rho E)\, dA + \int_{\partial M} \langle \rho E u, n\rangle \, dS = \int_M \rho g u \, dA + \int_{\partial M} \kappa \nabla T n \, dS + \int_{\partial M} (\mathbb{T} - p \cdot I)\, u n \, dS.$$

The energy equation is often rewritten in terms of total enthalpy H

$$H := E + \frac{p}{\rho} \Rightarrow p = \rho H - \rho E, \tag{4.6}$$

which eliminates the pressure p:

$$\int_M \frac{\partial}{\partial t}(\rho E)\, dA + \int_{\partial M} \langle \rho H u, n \rangle\, dS = \int_M \rho g u\, dA + \int_{\partial M} \kappa \nabla T n\, dS + \int_{\partial M} \mathbb{T} u n\, dS.$$

The desired expression follows with remark 3.2.3. □

The conservation of mass, momentum, and energy results in five equations for the seven unknowns ρ, u, p, E, and T. Hence, additional closure assumptions must be made, linking pressure p and temperature T to the unknowns ρ, u, and E.

Definition 4.1.7 (Perfect Gas)**.** *A gas is said to be perfect or ideal if the pressure is given by the relation*

$$p = \rho R T,$$

where R is the specific gas constant. Furthermore, the specific heat capacity is the measure of the heat energy required to increase the temperature of a unit quantity of a substance by a certain temperature interval. For a compressible body, one may distinguish between heat capacity at constant volume c_v and heat capacity at constant pressure c_p. For an ideal gas, the heat capacity is constant with temperature resulting in

$$R = c_p - c_v,$$

and the internal energy is related to the temperature by

$$e = c_v T.$$

Since $E = e + \frac{1}{2}\|u\|^2$, the above results in

$$p = \rho R \frac{E - \frac{1}{2}\|u\|^2}{c_v} = \frac{R}{c_v}\rho \left(E - \frac{1}{2}\|u\|^2 \right).$$

Introducing the adiabatic exponent γ as

$$\gamma = \frac{c_p}{c_v} \Rightarrow \frac{R}{c_v} = \gamma - 1,$$

one arrives at the final equation linking the pressure to the unknowns:

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2}\|u\|^2 \right).$$

Closing the circle, the temperature can thus also be expressed as $T = \frac{p}{R\rho}$.

Remark 4.1.8 (Speed of Sound, Mach Number)**.** A very important property is the speed of sound. As the fluid velocity approaches the local speed of sound, shock waves begin to form, and the equations of state are then dominated by their hyperbolic nature. For a perfect gas, the speed of sound c is given by

$$c = \sqrt{\gamma RT} = \sqrt{\gamma R \frac{1}{c_v} e} = \sqrt{\gamma(\gamma - 1)(E - \|u\|^2)} = \sqrt{\gamma \frac{p}{\rho}}.$$

The local Mach number M is then given by

$$M = \frac{\|u\|}{c}.$$

Remark 4.1.9 (Scaling, Non-Dimensionalization)**.** Different scales can lead to similar fluid flow: Flow around a sphere of radius $r_1 = 10\ m$ with a velocity of $10\ km/h$ will be the same as flow around a sphere of radius $r_2 = 1\ m$ with a velocity of $100\ km/h$. This is often exploited in experiments when smaller models are used in wind tunnels. The Navier–Stokes equations are thus often non-dimensionalized. The subscript ∞ denotes the respective value in the farfield:

$p_\infty = 1$	reference pressure
$\rho_\infty = 1$	reference density
$T_\infty = 1$	reference temperature
$c_\infty = \sqrt{\gamma \frac{p_\infty}{\rho_\infty}} = \sqrt{\gamma}$	reference speed of sound
$u_\infty = M_\infty a_\infty = M_\infty \sqrt{\gamma}$	reference velocity
$\mu_\infty = \frac{\rho_\infty u_\infty L}{Re_\infty} = \frac{M_\infty \sqrt{\gamma}}{Re_\infty}$	reference dynamic viscosity.

Additionally, M_∞ is the freestream Mach number and

$$Re = \frac{\rho u L}{\mu}$$

is the Reynolds number where L is the characteristic length scale. Furthermore, the thermal conductivity is non-dimensionalized by

$$\kappa_\infty = \frac{\gamma \mu_\infty}{(\gamma - 1)Pr},$$

where Pr is the Prandtl number. As a consequence, the non-dimensional gas constant reduces to

$$R = \frac{p}{\rho T} = 1.$$

Usually, the Reynolds number Re, Prandtl number Pr, Mach number M, adiabatic exponent γ, and the reference length L are externally given, defining the flow conditions to simulate.

Remark 4.1.10 (Sutherland Law)**.** The viscosity of a fluid is often decreasing with temperature, which is modeled by Sutherland's formula

$$\mu = \mu_\infty \left(\frac{T}{T_\infty} \right)^{\frac{3}{2}} \frac{T_\infty + S}{T + S},$$

where S is the Sutherland temperature, usually 110.4 K for air.

Taking everything together results in the compressible Navier–Stokes equations.

Definition 4.1.11 (Compressible Navier–Stokes Equations)**.** *The compressible Navier–Stokes equations for a Newtonian fluid are given by*

$$\int_M \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u)\; dA = 0 \tag{4.7}$$

$$\int_M \frac{\partial}{\partial t}(\rho u_i) + \sum_{j=1}^{3} \left[\frac{\partial}{\partial x_j} \left(-\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{2}{3} (\operatorname{div} u)\, \delta_{ij} \right) \right) + \frac{\partial(\rho u_i u_j)}{\partial x_j} \right] + \frac{\partial p}{\partial x_i}\; dA = \int_M \rho g_i\; dA \tag{4.8}$$

$$\int_M \frac{\partial}{\partial t}(\rho E) + \operatorname{div} \left(\rho H u - \kappa \nabla T + \left[-\mu \left(\nabla u + (\nabla u)^T + \frac{2}{3} (\operatorname{div} u) \cdot I \right) \right] u \right)\; dA = \int_M \rho g u\; dA. \tag{4.9}$$

Note that equation (4.8) *is also of divergence type when written in vectorial form using the Kronecker symbol for the pressure. The dynamic viscosity μ is given by the Sutherland law. The boundary condition for the velocity on fluid obstacles usually is*

$$u = 0 \textit{ on } \Gamma_0.$$

Appropriate boundary conditions for the heat fluxes and on inlets/outlets will be provided later on based on the situation to be simulated.

Remark 4.1.12 (Forces in Fluids)**.** Equation (4.4) can also be used to compute the force a fluid exerts on an immersed body with boundary ∂M. According to (4.4), the force a solid body exerts on the fluid is given by

$$F = \int_M \rho g\; dA + \int_{\partial M} \langle \sigma, n \rangle\; dS \in \mathbb{R}^3.$$

According to Newton's third law of motion, the sign switches when considering the force the fluid exerts on the solid body, and in a Newtonian fluid without body forces, the fluid force in the i-th coordinate axis is thus given by

$$\begin{aligned} F_i &= \left(\int_{\partial M} \langle -\mathbb{T} + p, n \rangle\; dS \right)_i \\ &= \int_{\partial M} \left[\sum_{j=1}^{3} \left(-\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{2}{3} (\operatorname{div} u)\, \delta_{ij} \right) \right) n_j \right] + p n_i\; dS. \end{aligned} \tag{4.10}$$

4.2 Simplifications

There are many simplifications making the complex governing equations more accessible. Two assumptions most often used are incompressibility and diminishing viscosity, i.e. inviscid flow.

4.2.1 Incompressible Flow

Definition 4.2.1 (Incompressible Fluid)**.** *A fluid is considered incompressible if the density ρ does not change in time, i.e.*

$$\frac{d\rho(x(t), t)}{dt} = 0.$$

Here, it is also assumed that the density is constant in space.

When using the above assumption and omitting Sutherland's law, the conservation of mass (4.7) and momentum (4.8) for a Newtonian fluid become

$$\int\limits_M \rho \operatorname{div} u \, dA = 0$$

$$\int\limits_M \rho \frac{\partial}{\partial t} u_i - \mu \Delta u_i + \rho u \nabla u_i + \frac{\partial p}{\partial x_i} \, dA = \int\limits_M \rho g_i \, dA.$$

For an incompressible flow the conservation of mass and momentum already create a closed system of equations. The energy of the fluid can be computed using equation (4.9) after the above equations have been solved for u and p. Since the incompressible equation cannot develop shock waves, the above system is often written as a partial differential equation:

Definition 4.2.2 (Incompressible Navier–Stokes Equations)**.** *The incompressible Navier–Stokes equations are given by*

$$\begin{aligned} \rho \frac{\partial}{\partial t} u - \mu \Delta u + \rho u \nabla u + \nabla p &= \rho g \quad && in \quad \Omega \\ \operatorname{div} u &= 0 \\ u &= u_+ && on \quad \Gamma_+ \\ u &= 0 && on \quad \Gamma_0 \\ pn - \mu \frac{\partial u}{\partial n} &= 0 && on \quad \Gamma_- . \end{aligned} \tag{4.11}$$

Here, $\partial\Omega = \Gamma = \Gamma_+ \overset{d}{\cup} \Gamma_0 \overset{d}{\cup} \Gamma_-$. The inflow is given by Γ_+ and the surface of any obstacles in the fluid is given by Γ_0. The boundary condition on Γ_0 is the no-slip boundary condition of a viscous fluid. The outflow condition on Γ_- is chosen such that mass conservation holds. This is the natural boundary condition for finite element discretizations. When enclosed flows are considered, i.e. $\Gamma_- = \emptyset$, the missing boundary condition results in the pressure being defined only up to an additive constant. Omitting the time derivative results in the steady state Navier–Stokes equations.

Since the energy equation decouples from the conservation of mass and momentum, there exists a convenient representation of the kinetic energy loss of an incompressible fluid. In order to derive the dissipation rate of kinetic energy into heat, the following result is needed:

Lemma 4.2.3. *Let u solve the incompressible Navier–Stokes equations* (4.11)*. It then follows that*

$$\int_\Omega u(u\nabla u)\;dA = 0.$$

Proof.

$$\begin{aligned}\int_\Omega u(u\nabla u)\;dA &= \frac{1}{2}\int_\Omega u\,\nabla\left(\|u\|^2\right)\;dA - \int_\Omega u\,(u\times \operatorname{rot} u)\;dA\\ &= -\frac{1}{2}\int_\Omega \operatorname{div} u\|u\|^2\;dA + \frac{1}{2}\int_\Gamma \|u\|^2 u\cdot n\;dS\\ &= 0.\end{aligned}$$

□

Lemma 4.2.4 (Energy Dissipation in an Incompressible Fluid)**.** *In the absence of body forces g, the dissipation of kinetic energy into heat in a viscous incompressible flow is given by*

$$\dot{E}_u := \frac{\partial}{\partial t}E_u = -\mu\sum_{i,j=1}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_j}\right)^2\;dA. \tag{4.12}$$

Since the above value is always negative, one can see that in an incompressible flow, heat is never converted back into kinetic energy.

Proof. The kinetic energy E_u is given by

$$\begin{aligned}E_u &= \frac{1}{2}m\|u\|^2 = \frac{1}{2}\int_\Omega\sum_{i=1}^{3}\rho u_i^2\;dA\\ \Rightarrow \dot{E}_u &= \int_\Omega\sum_{i=1}^{3}u_i(\rho\dot{u}_i)\;dA.\end{aligned}$$

Inserting the incompressible Navier–Stokes equations without body forces for $\rho\dot{u}_i$, one arrives at

$$\dot{E}_u = \int_\Omega\sum_{i=1}^{3}u_i(\mu\Delta u_i - \rho u\nabla u_i - \frac{\partial p}{\partial x_i})dA.$$

Using integration by parts on the pressure term, the equation becomes

$$\begin{aligned}\dot{E}_u &= \int_\Omega\sum_{i=1}^{3}(\mu u_i\Delta u_i - u_i\rho u\nabla u_i - u_i\frac{\partial p}{\partial x_i})\;dA\\ &= \int_\Omega\sum_{i=1}^{3}\left[\mu u_i\Delta u_i - u_i\rho u\nabla u_i + \frac{\partial u_i}{\partial x_i}p\right]\;dA - \int_\Gamma\sum_{i=1}^{3}pu_in_i\;dS.\end{aligned}$$

Furthermore, $\sum_{i=1}^{3} \frac{\partial u_i}{\partial x_i} = 0$ because of the divergence freedom and $\sum_{i=1}^{3} pu_i n_i = 0$ because of the no-slip boundary condition, which leads to

$$\dot{E}_u = \int_{\Omega} \sum_{i=1}^{3} [\mu u_i \Delta u_i - u_i \rho u \nabla u_i] \, dA.$$

With lemma 4.2.3 this becomes:

$$\dot{E}_u = \int_{\Omega} \sum_{i=1}^{3} \mu u_i \Delta u_i \, dA.$$

Swapping over Δ and eliminating the boundary integrals because of the no-slip boundary condition results in:

$$\dot{E}_u = -\mu \sum_{i,j=1}^{3} \int_{\Omega} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \, dA.$$

□

The above formula has both advantages and disadvantages when compared to using forces according to equation (4.10). On the one hand, the value of (4.12) depends on the size of the simulation domain Ω, making results from two different domains difficult to compare: Considering an obstacle in a fluid channel, the value of (4.12) will increase when a longer channel is simulated, although the force the fluid exerts on any obstacle should stay the same. Also, in case of multiple obstacles, (4.12) does not allow identifying the contribution each obstacle has to the total energy loss. On the other hand, the shape differentiation of a volume objective function is much more straight forward than the shape differentiation of a surface functional such as (4.10). This is especially true for higher order derivatives. Hence, for considerations on shape optimization in fluids later on, (4.12) will be considered first.

Remark 4.2.5. The incompressible Navier–Stokes equations still possess several numerical difficulties. The missing time derivative in the conservation of mass requires at least semi-implicit solution strategies. The divergence freedom of *u* results in the fact that the discretization of *u* and *p* cannot be chosen independently. Instead, a stable discretization without checkerboarding must satisfy the inf-sup condition [30, 76], and the resulting discrete saddle-point problem is hard to solve iteratively. Last, a low viscosity μ, i.e. a high Reynolds number *Re*, results in an emphasis on the non-linearity. Unless the resulting turbulence is not averaged in some way, the equation does no longer have a stable steady state, and all turbulent length scales must be resolved in the computational grid. For most flow phenomena of interest, the computational power of present and future computers does not suffice to resolve turbulence. The problems due to turbulence can be avoided by dropping the non-linearity from the momentum conservation, resulting in the Stokes equation.

Definition 4.2.6 (Incompressible Stokes Equations)**.** *Using the same notation as in definition 4.2.2, the incompressible Stokes equations are given by*

$$\begin{aligned} -\mu\Delta u + \nabla p &= \rho g && \text{in} \quad \Omega \\ \operatorname{div} u &= 0 \\ u &= u_+ && \text{on} \quad \Gamma_+ \\ u &= 0 && \text{on} \quad \Gamma_0 \\ pn - \mu\frac{\partial u}{\partial n} &= 0 && \text{on} \quad \Gamma_- . \end{aligned} \tag{4.13}$$

Due to the simplified momentum conservation, the Stokes equations are only applicable to very viscous “creeping” flows or inside boundary layers. However, the simplicity and linearity allows for a detailed mathematical analysis.

4.2.2 Inviscid Flow

The other alternative to remove turbulence and boundary layers as described in remark 4.2.5 is by neglecting viscosity altogether.

Definition 4.2.7 (Compressible Euler Equations)**.** *The compressible Euler equations result from dropping the viscous terms from the compressible Navier–Stokes equations. For a Newtonian fluid they are given by*

$$\int_M \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) \, dA = 0 \tag{4.14}$$

$$\int_M \frac{\partial}{\partial t}(\rho u_i) + \sum_{j=1}^{3} \left[\frac{\partial(\rho u_i u_j)}{\partial x_j}\right] + \frac{\partial p}{\partial x_i} \, dA = \int_M \rho g_i \, dA \tag{4.15}$$

$$\int_M \frac{\partial}{\partial t}(\rho E) + \operatorname{div}(\rho H u) \, dA = \int_M \rho g u \, dA. \tag{4.16}$$

Note that in addition to setting the dynamic viscosity to zero, the temperature diffusion is also removed from the energy equation. The solid wall boundary condition for an inviscid fluid is given by the non-permeability or “slip” condition

$$\langle u, n \rangle = 0.$$

Since turbulence is a result of low viscosity, it at first appears counter-productive to remove the viscous terms altogether. However, the slip condition on solid walls removes boundary layers and hence eliminates turbulence. Additionally, the compressible Euler equations no longer possess a unique solution. The physically relevant solution has to be chosen such that entropy is increasing when crossing shock waves. Technically, however, most numerical schemes feature a vanishing viscosity approach such that the physically correct solution is automatically found.

Remark 4.2.8 (Euler Equations, Flux Functions)**.** The compressible Euler equations, definition 4.2.7, can also be written in the following way:

$$\frac{\partial U}{\partial t} + \operatorname{div} F = \frac{\partial U}{\partial t} + \sum_{k=1}^{3} \frac{\partial F_k}{\partial x_k} = 0 \in \mathbb{R}^5, \tag{4.17}$$

where the “inviscid fluxes” F_i are given by

$$\begin{aligned} F_1 &:= \left(\rho u_1, p + \rho u_1^2, \rho u_1 u_2, \rho u_1 u_3, u_1(E+p)\right)^T \\ F_2 &:= \left(\rho u_2, \rho u_1 u_2, p + \rho u_2^2, \rho u_2 u_3, u_2(E+p)\right)^T \\ F_3 &:= \left(\rho u_3, \rho u_1 u_3, \rho u_2 u_3, p + \rho u_3^2, u_3(E+p)\right)^T . \end{aligned}$$

The vector of conserved variables is given by

$$U := (\rho, \rho u, \rho E)^T,$$

and

$$U_{\mathrm{p}} := (\rho, u, E)^T$$

is the vector of primitive variables.

Remark 4.2.9 (Euler Flux Jacobians)**.** Applying the chain rule on (4.17), the conserved variables U can be made visible

$$\begin{aligned} 0 = \sum_{k=1}^{3} \frac{\partial F_k}{\partial x_k} &= \sum_{k=1}^{3} \frac{\partial F_k}{\partial U} \frac{\partial U}{\partial x_k} = \sum_{k=1}^{3} \frac{\partial F_k}{\partial U_{\mathrm{p}}} \frac{\partial U_{\mathrm{p}}}{\partial U} \frac{\partial U}{\partial x_k} \\ &= \sum_{k=1}^{3} \frac{\partial F_k}{\partial U_{\mathrm{p}}} \left[\frac{\partial U}{\partial U_{\mathrm{p}}}\right]^{-1} \frac{\partial U}{\partial x_k} =: \sum_{k=1}^{3} A_k \frac{\partial U}{\partial x_k}. \end{aligned}$$

The Euler flux Jacobians A_k are given by

$$A_1 := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ (\gamma-1)H - u_1^2 - c^2 & (3-\gamma)u_1 & -(\gamma-1)u_2 & -(\gamma-1)u_3 & \gamma-1 \\ -u_1 u_2 & u_2 & u_1 & 0 & 0 \\ -u_1 u_3 & u_3 & 0 & u_1 & 0 \\ u_1[(\gamma-2)H - c^2] & H - (\gamma-1)u_1^2 & -(\gamma-1)u_1 u_2 & -(\gamma-1)u_1 u_3 & \gamma u_1 \end{bmatrix}$$

$$A_2 := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -u_1 u_2 & u_2 & u_1 & 0 & 0 \\ (\gamma-1)H - u_2^2 - c^2 & -(\gamma-1)u_1 & (3-\gamma)u_2 & -(\gamma-1)u_3 & \gamma-1 \\ -u_2 u_3 & 0 & u_3 & u_2 & 0 \\ u_2[(\gamma-2)H - c^2] & -(\gamma-1)u_1 u_2 & H - (\gamma-1)u_2^2 & -(\gamma-1)u_2 u_3 & \gamma u_2 \end{bmatrix}$$

$$A_3 := \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -u_1 u_3 & u_3 & 0 & u_1 & 0 \\ -u_2 u_3 & 0 & u_3 & u_2 & 0 \\ (\gamma-1)H - u_3^2 - c^2 & -(\gamma-1)u_1 & -(\gamma-1)u_2 & (3-\gamma)u_3 & \gamma-1 \\ u_3[(\gamma-2)H - c^2] & -(\gamma-1)u_1 u_3 & -(\gamma-1)u_2 u_3 & H - (\gamma-1)u_3^2 & \gamma u_3 \end{bmatrix},$$

where $H := E + \frac{p}{\rho}$ is the enthalpy, see also equation (4.6), and $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound according to remark 4.1.8.

Remark 4.2.10 (Incompressible Euler Equations)**.** It is also possible to assume an inviscid and incompressible flow simultaneously, which leads to the incompressible Euler equations

$$\begin{aligned} \rho\frac{\partial}{\partial t}u + \rho u\nabla u + \nabla p = \rho g &\quad \text{in} \quad \Omega \\ \operatorname{div} u = 0 & \\ u = u_+ &\quad \text{on} \quad \Gamma_+ \\ \langle u, n\rangle = 0 &\quad \text{on} \quad \Gamma_0 \\ pn = 0 &\quad \text{on} \quad \Gamma_-. \end{aligned}$$

However, when considering the dissipation of kinetic energy into heat for an incompressible flow according to equation (4.12), one can see that

$$\dot{E}_u \equiv 0.$$

Hence, a body immersed in an inviscid, incompressible fluid does not experience any force at all. Unfortunately, this is also true when the compressible equations are used to simulate an essentially incompressible flow. As such, the Euler equations only produce physically meaningful forces when compressibility effects are relevant, meaning the speed of the fluid is close to or exceeds Mach 1.0 and compression shock waves form.

4.2.3 Potential Flow

The final simplification is that of potential flow. When an inviscid fluid is also considered to be irrotational, meaning

$$\operatorname{rot} u = 0,$$

one can show that there exists a velocity potential ϕ such that $u = -\nabla\phi$. More information can for example be found in [43]. Using the irrotational assumption in the inviscid mass conservation div $u = 0$, one arrives at the equation for potential flow:

Definition 4.2.11 (Potential Flow)**.** *Potential flow is given by*

$$\begin{aligned} -\Delta\phi = 0 &\quad \textit{in} \quad \Omega \\ \frac{\partial\phi}{\partial n} = 0 &\quad \textit{on} \quad \Gamma_0 \\ \phi = \phi_0 &\quad \textit{on} \quad \Omega \setminus \Gamma_0. \end{aligned}$$

This is now a scalar equation based on the inviscid conservation of mass. The boundary value ϕ_0 creates the necessary potential difference between inflow and outflow. It is possible to recover the pressure once the above equation is solved for ϕ. However, due to the limitations as discussed in remark 4.2.10, potential flow can mostly be used in inverse design only, meaning finding shapes that produce a desired pressure profile.

In the following chapters, both fluid dynamics and shape optimization will be combined with special attention on fast numerical procedures. However, first a very brief overview on finite volume methods for conservation laws will be given.

4.3 Numerical Schemes for Conservation Laws

In this section, some aspects of solving the flow equations discretely will be presented from the literature, especially [44, 82]. Special attention is given to finite volume schemes for compressible fluid dynamics. The DLR flow solver TAU, which is used in chapter 8, is also based on a finite volume discretization. However, in the incompressible case, chapter 5 and 6, a finite element based flow solver is used. Compared to the compressible case, finite element methods are more straight forward applicable to incompressible fluid dynamics, but for convection dominant high Reynolds-number flows they usually require some kind of stabilization.

4.3.1 The Finite Volume Method

For simplicity reasons, a general scalar conservation law of the type

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{4.18}$$

will be considered. The discretization of the space-time domain is given by a cartesian grid

$$(x_j, t_n) := (j\Delta x, n\Delta t),\ j \in \mathbb{Z}, n \in \mathbb{N}_0.$$

A discrete cell or finite control volume is then defined by the interval

$$[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] := [x_j - \frac{1}{2}\Delta x, x_j + \frac{1}{2}\Delta x],$$

and the discretized state $\hat{u}_j$ at node j is thought of as an approximation of the cell-averaged value of the function u

$$\hat{u}(x_j, t) \approx \frac{1}{\Delta x} \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t)\ dx.$$

The integral representation of the conservation law (4.18) provides

$$\int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_{n+1})\ dx = \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n)\ dx - \left[\int\limits_{t_n}^{t_{n+1}} f(u(x_{j+\frac{1}{2}}, t))\ dt - \int\limits_{t_n}^{t_{n+1}} f(u(x_{j-\frac{1}{2}}, t))\ dt \right]$$

for each finite control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. Hence, a numerical scheme should reproduce this property, which gives rise to the following definition.

Definition 4.3.1 (Conservative Scheme)**.** *A numerical scheme is called conservative if the following relation holds*

$$\hat{u}_j^{n+1} = \hat{u}_j^n - \frac{\Delta t}{\Delta x} \left[F(\hat{u}_{j-p}^n, \hat{u}_{j-p+1}^n, ..., \hat{u}_{j+q}^n) - F(\hat{u}_{j-p-1}^n, \hat{u}_{j-p}^n, ..., \hat{u}_{j+q-1}^n) \right],$$

where lower indices denote the spacial dimension and upper indices denote time. In short notation the above reads

$$\hat{u}_j^{n+1} = \hat{u}_j^n - \frac{\Delta t}{\Delta x}\left[F(\hat{u}^n;j) - F(\hat{u}^n;j-1)\right].$$

Finally, the numerical flux F is given as the approximation of the average flux at position $x_{j+\frac{1}{2}}$ during the time interval $[t_n, t_{n+1}]$

$$F(\hat{u}^n;j) \approx \frac{1}{\Delta t}\int\limits_{t_n}^{t_{n+1}} f(u(x_{j+\frac{1}{2}}, t))\, dt.$$

Remark 4.3.2. Often, the numerical flux F depends only on one left and right neighbor. Thus, one frequently has the situation $p = 0$ and $q = 1$ in definition 4.3.1 above. This leads to the more common expression

$$\hat{u}_j^{n+1} = \hat{u}_j^n - \frac{\Delta t}{\Delta x}\left[F(\hat{u}_j^n, \hat{u}_{j+1}^n) - F(\hat{u}^n, \hat{u}_j)\right] =: \hat{u}_j^n - \frac{\Delta t}{\Delta x}\left[F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right].$$

Definition 4.3.3 (Consistent Scheme)**.** *A finite volume scheme is call consistent, if the numerical flux function F satisfies for all $u \in \mathbb{R}^m$*

$$F(u, u, ..., u) = f(u)$$

and is Lipschitz-continuous, i.e. for an arbitrary $u \in \mathbb{R}^m$ fulfills

$$\|F(\hat{u}_{j-p}, \hat{u}_{j-p+1}, ..., \hat{u}_{j+q}) - f(u)\| \leq C \max_{-p\leq i\leq q} \|\hat{u}_{j+i} - u\|$$

for all $\hat{u}_{j+i}$ in a neighborhood of u with C independent of u.

Remark 4.3.4. The first condition in definition 4.3.3 basically states that constant functions must be integrated correctly and is in fact necessary for the Lipschitz-continuity. One can show that a consistent finite volume scheme must also satisfy the discrete conservation property of definition 4.3.1. As a consequence, shock waves will be captured at correct positions during computation. For more details see [44].

Unless special provisions are taken, numerical schemes for conservation laws tend to produce spurious oscillations when the solution of the conservation law features discontinuities and shock waves. Thus, it is little surprising that the total variation of the computed solution must be diminishing in order to achieve convergence of the scheme, which is stated by Lax-Wendroff's theorem later on.

Definition 4.3.5 (Total Variation)**.** *The total variation of a function u is given by*

$$\mathrm{TV}(u) := \sup\left\{\sum_{j=1}^{N} |u(\xi_j) - u(\xi_{j-1})| : \xi_0 < \xi_1 < ... < \xi_N, N \in \mathbb{N}\right\}.$$

For the discrete case, the total variation of a vector, i.e. the piecewise constant approximation of the state $\hat{u}^n$ at time n, the total variation is given by

$$\mathrm{TV}(\hat{u}^n) := \sum_{j=-\infty}^{\infty} |\hat{u}_{j+1}^n - \hat{u}_j^n|.$$

Using the total variation, the theorem of Lax–Wendroff now ensures the approximation property of finite volume schemes.

Theorem 4.3.6 (Lax–Wendroff)**.** *Let a sequence of meshes*

$$(x_j^\ell, t_n^\ell) := (jh_\ell, nk_\ell),\ j \in \mathbb{Z}, n \in \mathbb{N}_0, \ell \in \mathbb{N}$$

be given, such that $h_\ell \to 0$ *and* $k_\ell \to 0$ *for* $\ell \to \infty$*. Let* $(\hat{u}_\ell)_{\ell\in\mathbb{N}}$ *be a sequence of solutions generated by a conservative and consistent scheme on the respective mesh. Furthermore, let the sequence* $(\hat{u}_\ell)_{\ell\in\mathbb{N}}$ *converge to a locally integrable function* u^* *such that for each finite set* $[a, b] \times [0, T]$ *the relation*

$$\lim_{\ell\to\infty} \int_0^T \int_a^b |\hat{u}_\ell(x, t) - u^*(x, t)|\ dxdt = 0$$

holds. Additionally, let the total variation be uniformly bounded, such that for each $T > 0$ *there exists an* $R > 0$ *with*

$$\mathrm{TV}(\hat{u}_\ell(., t)) < R$$

for all $0 \le t \le T$ *and all* $\ell \in \mathbb{N}$*. The limit function* u^* *is then a weak solution of the conservation law.*

Proof. See for example theorem 12.1 in [44]. □

Remark 4.3.7. The Lax–Wendroff theorem neither states that convergence occurs at all nor that the computed solution of the conservation law is the physically correct solution. To ensure convergence to a physically relevant solution, more considerations are necessary, some of which are based on monotonicity conditions.

Remark 4.3.8 (Monotonicity Preservation)**.** For some flow velocity $c > 0$, the linear convection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, t > 0, x \in \mathbb{R} \tag{4.19}$$

with initial condition $u(x, 0) = u_0(x)$ is fulfilled by the function

$$u(x, t) = u_0(x - ct).$$

Thus, if $u(x, 0)$ has some monotonicity in x, then $u(x, t)$ is also monotonic for all $t > 0$. Numerical schemes for such a conservation law should therefore also preserve monotonicity.

Theorem 4.3.9 (Godunov's Order Barrier Theorem)**.** *Linear one-step second-order accurate numerical schemes for the convection equation* (4.19) *cannot be monotonicity preserving unless*

$$c\frac{\Delta t}{\Delta x} \in \mathbb{N}.$$

Proof. See chapter 9.2 in [82]. □

A consequence of Godunov's theorem is that any numerical discretization scheme should only be of first order in the vicinity of discontinuities, e.g. shock waves. Since higher order accuracy is generally desired in smooth regions of the flow, Godunov's theorem has given rise to higher order methods that employ flux and slope limiters at shock positions. These limiters are thus required to reduce the approximation order of any numerical scheme to 1 in the vicinity of shocks.

4.3.2 The Jameson–Schmidt–Turkel Scheme

Due to Godunov's theorem, any flow discretization must be of first order in the vicinity of discontinuities, which can be achieved by flux and slope limiters. Another approach is to start from an unstable central second order scheme and add a non-linear stabilization term, which usually leads to a version of the Jameson–Schmidt–Turkel (JST) scheme [42].

The domain Ω is discretized into a finite number of non-overlapping control volumes $\Omega_{i,j}$, and the approximation $\hat{u}_{i,j}$ of the flow state is assumed to reside in the center of each control volume. The situation is also illustrated in figure 4.1. The conservation law in integral formulation provides

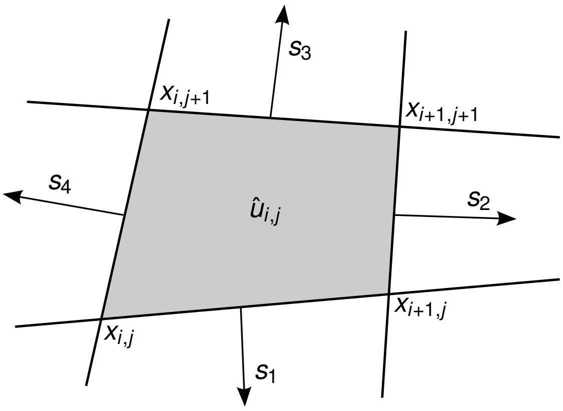


Figure 4.1: Cell-centered finite volume discretization.

$$\begin{aligned} 0 = \int_{\Omega} \frac{\partial u}{\partial t} + \operatorname{div} f(u)\, dA &= \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \int_{\Omega_{i,j}} \frac{\partial u}{\partial t} + \operatorname{div} f(u)\, dA \\ &= \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \int_{\Omega_{i,j}} \frac{\partial u}{\partial t}\, dA + \int_{\partial\Omega_{i,j}} f(u) n\, dS \end{aligned}$$

due to the divergence theorem. Thus, an approximation of u on the faces of each finite control volume is needed. Assuming the approximation $\hat{u}$ to be constant inside each control volume, the value of $\hat{u}$ on the face is given by

$$\begin{aligned} \hat{u}_{i,j+\frac{1}{2}} &= \frac{1}{2}\left(\hat{u}_{i,j} + \hat{u}_{i,j+1}\right) \\ \hat{u}_{i,j-\frac{1}{2}} &= \frac{1}{2}\left(\hat{u}_{i,j} + \hat{u}_{i,j-1}\right) \\ \hat{u}_{i+\frac{1}{2},j} &= \frac{1}{2}\left(\hat{u}_{i,j} + \hat{u}_{i+1,j}\right) \\ \hat{u}_{i-\frac{1}{2},j} &= \frac{1}{2}\left(\hat{u}_{i,j} + \hat{u}_{i-1,j}\right), \end{aligned}$$

which leads to

$$0 = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \left[\frac{\partial \hat{u}_{i,j}}{\partial t} |\Omega_{i,j}| + f(\hat{u}_{i,j-\frac{1}{2}}) s_1 + f(\hat{u}_{i+\frac{1}{2},j}) s_2 + f(\hat{u}_{i,j+\frac{1}{2}}) s_3 + f(\hat{u}_{i-\frac{1}{2},j}) s_4 \right].$$

Here, the face vectors s_i are orthogonal to the edges of $\Omega_{i,j}$. By not normalizing them, the corresponding integral over each of the edges can be computed straight forward. Thus, the equation can be integrated in time until a steady state is reached by solving for each control volume

$$\frac{\partial \hat{u}_{i,j}}{\partial t} = -\frac{1}{|\Omega_{i,j}|} Q_{i,j},$$

where

$$Q_{i,j} = \left[f(\hat{u}_{i,j-\frac{1}{2}}) s_1 + f(\hat{u}_{i+\frac{1}{2},j}) s_2 + f(\hat{u}_{i,j+\frac{1}{2}}) s_3 + f(\hat{u}_{i-\frac{1}{2},j}) s_4 \right].$$

Remark 4.3.10. One can show that the above central discretization scheme is of 2nd order and thus violates Godunov's theorem. In order to ensure stability, a certain degree of numerical viscosity has to be added by conducting flux updates of the type

$$\frac{\partial \hat{u}_{i,j}}{\partial t} = -\frac{1}{|\Omega_{i,j}|} Q_{i,j} + \frac{1}{|\Omega_{i,j}|} D_{i,j}.$$

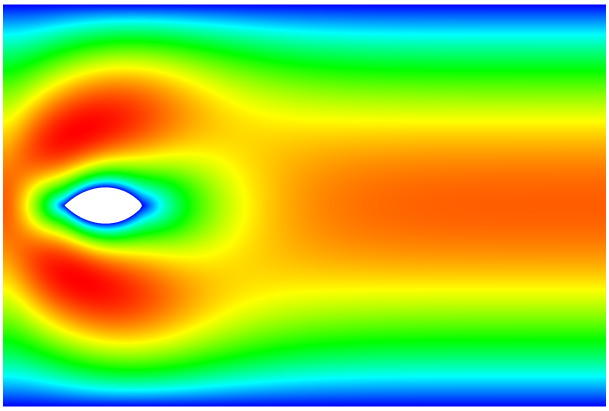
The dissipative flux $D_{i,j}$ is usually given by

$$D_{i,j} = d_{i+\frac{1}{2},j} + d_{i-\frac{1}{2},j} + d_{i,j+\frac{1}{2}} + d_{i,j-\frac{1}{2}}$$

with $d_{i,j}$ being

$$d_{i+\frac{1}{2},j} = \alpha_{i+\frac{1}{2},j} \left(\epsilon^{(2)}_{i+\frac{1}{2},j} \left(\hat{u}_{i+1,j} - \hat{u}_{i,j} \right) - \epsilon^{(4)}_{i+\frac{1}{2},j} \left(\hat{u}_{i+2,j} - 3\hat{u}_{i+1,j} + 3\hat{u}_{i,j} - \hat{u}_{i-1,j} \right) \right).$$

The other directions are given analogeously. The coefficients $\epsilon^{(2)}_{i,j}$ and $\epsilon^{(4)}_{i,j}$ are adaptive coefficients determining the amount of 1st and 3rd order numerical dissipation. They usually depend on the pressure discontinuity of the respective finite volume. The coefficient $\alpha_{i,j}$ is usually chosen with respect to the discrete time-step size. For more details, especially concerning accelerating the convergence to a steady state by local and adaptive time-stepping and the treatment of boundary conditions, see [7].



Chapter 5

Shape Optimization and Stokes Fluids

5.1 Problem Introduction and First Order Calculus

The self-adjoint nature of the Stokes equations creates an ideal introductory problem for studying shape Hessians. This shape optimization problem is also mentioned in [46] and makes for a perfect test-case for validating numerics, since the optimal shape is analytically known to be similar to a pointed prolate spheroid, e.g. a rugby ball, with 60° front and back angle [49, 50]. After defining the objective function and conducting first order calculus, the Stokes case will be used as a model problem for studying shape Hessians with respect to accelerating numerical shape optimization. Two dimensional considerations have already been published in parts in [61].

The novelty of this chapter lies in the detailed analysis of the shape Hessian for a Stokes fluid, lemma 5.2.2. Furthermore, knowledge of the correct shape Hessian is used to validate a Fourier mode operator symbol approximation both analytically and discretely, which is used to greatly accelerate applied shape optimization. Therefore, the following Stokes shape optimization problem is considered:

Definition 5.1.1 (Stokes Problem)**.** *The Stokes problem consists of minimizing the energy dissipation of kinetic energy* (4.12) *into heat in a Stokes flow* (4.13) *in absence of body forces.*

$$\min_{(u,p,\Omega)} J(u,p,\Omega) := \int_{\Omega} \mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \, dA$$

subject to

$$
\begin{aligned}
-\mu\Delta u + \nabla p &= 0 \quad && \textit{in} \quad \Omega \\
\operatorname{div} u &= 0 \\
u &= u_+ && \textit{on} \quad \Gamma_+ \\
u &= 0 && \textit{on} \quad \Gamma_0 \\
pn - \mu\frac{\partial u}{\partial n} &= 0 && \textit{on} \quad \Gamma_- \\
\mathrm{Vol} &= V_0.
\end{aligned}
$$

The last constraint, Vol = V$_0$*, means preserving the volume and prevents a degeneration of the shape. In order to formulate a minimization problem, the sign in the objective function has been switched compared to* (4.12)*. The domain is shown in figure 5.1.*

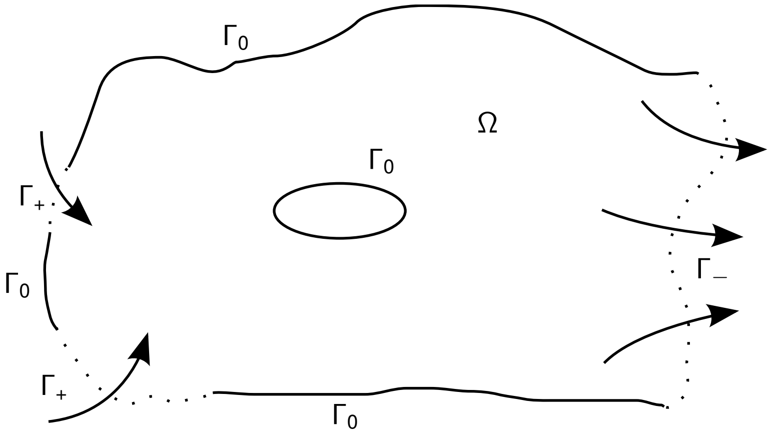


Figure 5.1: A possible layout of the domain.

Lemma 5.1.2 (Shape Derivative of the Stokes Problem)**.** *The shape derivative for a variation of the shape of a fluid obstacle* Γ_0 *for the Stokes problem is given by*

$$
dJ(u,p,\Omega)[V] = -\mu \int_{\Gamma_0} \langle V, n\rangle \sum_{k=1}^{3} \left(\frac{\partial u_k}{\partial n}\right)^2 \, dS.
$$

The expression does not involve an adjoint state, which is a consequence of the self-adjoint nature of the problem.

Proof. Linearizing the state equation according to section 3.4 results in the following partial differential equation for the local shape derivatives $u'[V]$ and $p'[V]$:

$$
\begin{aligned}
-\mu\Delta u'[V] + \nabla p'[V] &= 0 \quad && \text{in} \quad \Omega \\
\operatorname{div} u'[V] &= 0 \\
u'[V] &= 0 && \text{on} \quad \Gamma_+ \\
u'[V] &= -\langle V, n\rangle \frac{\partial u}{\partial n} && \text{on} \quad \Gamma_0 \\
p'[V]n - \mu\frac{\partial u'[V]}{\partial n} &= 0 && \text{on} \quad \Gamma_-.
\end{aligned}
\tag{5.1}
$$

The outflow boundary Γ_- does not feature the local shape derivative of the normal $dn[V]$ since the fluid obstacle Γ_0 is assumed to be deformed only. Furthermore, the preliminary gradient according to section 3.4 is given by

$$
\begin{aligned}
& dJ(u,p,\Omega)[V] \\
= & \int_{\Gamma_0} \langle V,n\rangle \mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2 \, dS \; + \; \int_{\Omega} \mu \sum_{i,j=1}^{3} 2 \left(\frac{\partial u_i}{\partial x_j}\right) \left(\frac{\partial u_i'[V]}{\partial x_j}\right) \, dA \\
= & \; \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2\mu \sum_{i=1}^{3} \left[- \int_{\Omega} (\Delta u_i) u_i'[V] \, dA + \int_{\Gamma} \left(\frac{\partial u_i}{\partial n}\right) u_i'[V] \, dS \right] ,
\end{aligned}
$$

where $\Gamma = \Gamma_+ \cup \Gamma_0 \cup \Gamma_-$ disjoint. Replacing Δu_i by the state equation results in:

$$
\begin{aligned}
& dJ(u,p,\Omega)[V] \\
= & \; \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2 \sum_{i=1}^{3} \left[\int_{\Omega} -\frac{\partial p}{\partial x_i} \cdot u_i'[V] \, dA + \mu \int_{\Gamma} \left(\frac{\partial u_i}{\partial n}\right) u_i'[V] \, dS \right] .
\end{aligned}
$$

Another integration by parts and using div $u'[V] = 0$ gives

$$
\begin{aligned}
dJ(u,p,\Omega)[V] = & \; \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + \\
& +2 \left[\int_{\Omega} p \operatorname{div} u'[V] \, dA + \sum_{i=1}^{3} \int_{\Gamma} -p u_i'[V] n_i + \mu \left(\frac{\partial u_i}{\partial n}\right) u_i'[V] \, dS \right] \\
= & \; \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2 \sum_{i=1}^{3} \left[\int_{\Gamma} -p u_i'[V] n_i + \mu \left(\frac{\partial u_i}{\partial n}\right) u_i'[V] \, dS \right] \\
= & \; \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2 \sum_{i=1}^{3} \left[\int_{\Gamma} \left(\mu \left(\frac{\partial u_i}{\partial n}\right) - p n_i \right) u_i'[V] \, dS \right] .
\end{aligned}
$$

Using the boundary conditions $u'[V] = 0$ on Γ_+ and $pn_i - \mu \frac{\partial u_i}{\partial n} = 0$ on Γ_-, the above transforms to

$$
dJ(u,p,\Omega)[V] = \mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2 \sum_{i=1}^{3} \left[\int_{\Gamma_0} \left(\mu \left(\frac{\partial u_i}{\partial n}\right) - p n_i \right) u_i'[V] \, dS \right] ,
$$

and due to the boundary condition $u'[V] = -\langle V,n\rangle \frac{\partial u}{\partial n}$ on Γ_0, one arrives at

$$
dJ(u,p,\Omega)[V] = -\mu \int_{\Gamma_0} \langle V,n\rangle \|\nabla u\|^2 \, dS + 2 \sum_{i=1}^{3} \left[\int_{\Gamma_0} \langle V,n\rangle p \frac{\partial u_i}{\partial n} n_i \, dS \right] .
$$

Since the divergence freedom is also valid at the boundary, the last term can be dropped, and one arrives at

$$dJ(u,p,\Omega)[V] = -\mu \int_{\Gamma_0} \langle V, n\rangle \|\nabla u\|^2 \, dS = -\mu \int_{\Gamma_0} \langle V, n\rangle \sum_{k=1}^{3} \left(\frac{\partial u_k}{\partial n}\right)^2 \, dS$$

due to the boundary conditions for u. □

5.2 Shape Hessian Analysis

Since very fast gradient based optimization procedures require Hessian information, the shape Hessian for the Stokes problem is now derived using the concept of repeated differentiation, i.e. the shape gradient is differentiated again using the same techniques. A good overview about the practical derivation of shape Hessians can for example also be found in [74]. The shape Hessian for the Stokes problem will require the following pseudo-differential operator.

Definition 5.2.1 (Divergence Free Dirichlet-to-Neumann Map)**.** *Using the same setting as in definition 5.1.1, let the operator S be defined by*

$$S\psi = \left.\frac{\partial \varphi}{\partial n}\right|_{\Gamma_0},$$

where $\psi : \Gamma_0 \to \mathbb{R}^d$*. Also,* $\varphi : \Omega \to \mathbb{R}^d$ *and* $\varphi_p : \Omega \to \mathbb{R}$ *are the solution of*

$$\begin{aligned} -\mu\Delta\varphi + \nabla\varphi_p &= 0 && \text{in } \Omega \\ \operatorname{div}\varphi &= 0 && \\ \varphi &= 0 && \text{on } \Gamma_+ \\ \varphi &= -\psi && \text{on } \Gamma_0 \\ \varphi_p n - \mu\frac{\partial\varphi}{\partial n} &= 0 && \text{on } \Gamma_-. \end{aligned}$$

Lemma 5.2.2 (Shape Hessian for the Stokes Problem)**.** *Using repeated differentiation, the shape Hessian for the Stokes problem is given by*

$$\begin{aligned} d^2J(u,p,\Omega)[V,W] = &\int_{\Gamma} \langle W, n\rangle \left[-\mu \sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right] \operatorname{div} V \, dS \\ &+ \int_{\Gamma} \langle W, n\rangle \frac{\partial}{\partial n}\left[-\mu \sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right] \langle V, n\rangle \, dS \\ &+ \int_{\Gamma} \langle W, n\rangle \left\langle \nabla \left[-\mu \sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right], V_\Gamma \right\rangle dS \\ &+ \int_{\Gamma} \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \frac{\partial u_i}{\partial n} S \frac{\partial u_i}{\partial n} \langle W, n\rangle\right] dS, \end{aligned}$$

where S is the pseudo-differential operator from definition 5.2.1 above.

Proof. Undoing the final step (3.5) in lemma 3.2.4, the gradient for the Stokes problem can be transformed back to a volume integral:

$$\begin{aligned} dJ(u,p,\Omega)[V] &= -\mu \int\limits_{\Gamma_0} \langle V, n\rangle \|\nabla u\|^2 \, dS \\ &= -\mu \int\limits_{\Gamma} \langle V, n\rangle \|\nabla u\|^2 \, dS \\ &= \int\limits_{\Omega} \operatorname{div}\left(\left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \right] V \right) dA. \end{aligned}$$

To derive the shape Hessian, let W be another sufficiently smooth perturbation vector field and let $\Omega_t[W] := \{x + tW(x) : x \in \Omega\}$ be given by another perturbation of identity. Hence, the second deformation $T_t[W]$ fulfills:

$$\begin{aligned} T_t[W](x) &= x + tW(x) \\ T_0[W](x) &= x \Rightarrow DT_0[W](x) \equiv I \\ \left.\frac{d}{dt}\right|_{t=0} T_t[W](x) &= W(x) \end{aligned} \tag{5.2}$$

Furthermore, let

$$\varphi(t,x) := \operatorname{div}\left(\left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i^t(x)}{\partial x_j} \right)^2 \right] V(x) \right),$$

where u^t and p^t is the solution of the Stokes equations on the perturbed domain $\Omega_t[W]$. Hence, for the shape Hessian, the limit

$$\left.\frac{d}{dt}\right|_{t=0} \int\limits_{\Omega_t[W]} \varphi(t,x)\, dA(x) = \left.\frac{d}{dt}\right|_{t=0} \int\limits_{\Omega} \varphi(t, T_t[W](x)) \det(DT_t[W](x))\, dA(x)$$

must be computed. The multiplication and total derivative rule immediately result in

$$\begin{aligned} & d^2J(u,p,\Omega)[V,W] := \left.\frac{d}{dt}\right|_{t=0} \int\limits_{\Omega_t[W]} \varphi(t,x)\, dA(x) \\ &= \int\limits_{\Omega} \left[\left.\frac{d}{dt}\right|_{t=0} \varphi(t, T_t[W](x)) \right] \det(DT_0[W](x)) + \varphi(0, T_0[W](x)) \left[\left.\frac{d}{dt}\right|_{t=0} \det(DT_t[W](x)) \right] dA(x) \\ &= \int\limits_{\Omega} \left[\frac{\partial}{\partial x} \varphi(0, T_0[W](x)) \left.\frac{d}{dt}\right|_{t=0} T_t[W](x) + \left.\frac{\partial}{\partial t}\right|_{t=0} \varphi(t, T_0[W](x)) \right] \det(DT_0[W](x)) \\ &\qquad + \varphi(0, T_0[W](x)) \left[\left.\frac{d}{dt}\right|_{t=0} \det(DT_t[W](x)) \right] dA(x). \end{aligned}$$

Using (5.2), the above transforms to

$$\begin{aligned}&\left.\frac{d}{dt}\right|_{t=0}\int\limits_{\Omega_t[W]}\varphi(t,x)\ dA(x)\\ =&\int\limits_{\Omega}\frac{\partial}{\partial x}\varphi(0,x)W+\left.\frac{\partial}{\partial t}\right|_{t=0}\varphi(t,x)+\varphi(0,x)\left[\left.\frac{d}{dt}\right|_{t=0}\det(DT_t[W](x))\right]\ dA(x).\end{aligned}$$

Using the derivative of the deformation determinant, lemma 3.2.2, and regrouping results in

$$\begin{aligned}&\left.\frac{d}{dt}\right|_{t=0}\int\limits_{\Omega_t[W]}\varphi(t,x)\ dA(x)\\ =&\int\limits_{\Omega}\langle\nabla_x\varphi(0,x),\,W(x)\rangle+\varphi(0,x)\mathrm{div}\ W(x)+\left.\frac{\partial}{\partial t}\right|_{t=0}\varphi(t,x)\ dA(x),\end{aligned}$$

where the first part is now an analogon to equation (3.4), leading to

$$d^2J(u,p,\Omega)[V,W]=\int\limits_{\Gamma}\langle W(x),n(x)\rangle\ \varphi(0,x)\ dS(x)+\int\limits_{\Omega}\left.\frac{\partial}{\partial t}\right|_{t=0}\varphi(t,x)\ dA(x).$$

Computing the derivative of the second part provides

$$\begin{aligned}\int\limits_{\Omega}\left.\frac{\partial}{\partial t}\right|_{t=0}\varphi(t,x)\ dA(x)&=\int\limits_{\Omega}\left.\frac{\partial}{\partial t}\right|_{t=0}\mathrm{div}\ \left(\left[-\mu\sum_{i,j=1}^{3}\left(\frac{\partial u_i^t(x)}{\partial x_j}\right)^2\right]V(x)\right)\ dA(x)\\ &=\int\limits_{\Omega}\mathrm{div}\ \left(\left[-2\mu\sum_{i,j=1}^{3}\left(\frac{\partial u_i(x)}{\partial x_j}\right)\frac{\partial u_i'[W](x)}{\partial x_j}\right]V(x)\right)\ dA(x)\\ &=\int\limits_{\Gamma}\langle V(x),n(x)\rangle\left[-2\mu\sum_{i,j=1}^{3}\left(\frac{\partial u_i(x)}{\partial x_j}\right)\frac{\partial u_i'[W](x)}{\partial x_j}\right]\ dS(x),\end{aligned}$$

where $u'[W]$ again solves the linearized Stokes equations (5.1), only this time for perturbation W. Combining everything together leads to the preliminary expression

$$\begin{aligned}&d^2J(u,p,\Omega)[V,W]\\ =&\int\limits_{\Gamma}\langle W,n\rangle\ \mathrm{div}\ \left(\left[-\mu\sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right]V\right)\ dS\\ &\qquad+\int\limits_{\Gamma}\langle V,n\rangle\left[-2\mu\sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)\frac{\partial u_i'[W]}{\partial x_j}\right]\ dS\\ =&I_1+I_2.\end{aligned}$$

Here, I_1 stems from a variation of the geometry and I_2 stems from a variation of the state u. Although the above expression is already a boundary integral, the desired structure of a scalar product of the Hessian and the two perturbation directions V and W is not yet visible and more manipulations are required.

Using lemma 2.1.12 on I_1 results in

$$\begin{aligned} I_1 &= \int_\Gamma \langle W, n\rangle \operatorname{div}\left(\left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right] V\right) dS \\ &= \int_\Gamma \langle W, n\rangle \left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right] \operatorname{div} V \, dS \\ &+ \int_\Gamma \langle W, n\rangle \frac{\partial}{\partial n}\left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right] \langle V, n\rangle \, dS \\ &+ \int_\Gamma \langle W, n\rangle \left\langle \nabla\left[-\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right], V_\Gamma \right\rangle dS, \end{aligned}$$

and for I_2 one has due to the no-slip condition for u

$$\begin{aligned} I_2 &= \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right) \frac{\partial u_i'[W]}{\partial x_j}\right] dS = \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \left\langle \nabla u_i, \nabla u_i'[W]\right\rangle\right] dS \\ &= \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \left\langle \frac{\partial u_i}{\partial n} n + \sum_{k=1}^{d-1} \frac{\partial u_i}{\partial \tau_k}\tau_k, \nabla u_i'[W]\right\rangle\right] dS \\ &= \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \left\langle \frac{\partial u_i}{\partial n} n, \nabla u_i'[W]\right\rangle\right] dS = \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \frac{\partial u_i}{\partial n}\frac{\partial u_i'[W]}{\partial n}\right] dS. \end{aligned}$$

Unfortunately, the above expression already is a boundary integral. Hence, integration by parts cannot be used easily to remove the normal derivative from the variation. Since the shape derivative of the Dirichlet boundary condition, lemma 3.4.3, does not make any statement concerning the (normal) derivative of the perturbation $\frac{\partial u_i'[W]}{\partial n}$, the above expression can only be further manipulated by introducing a pseudo-differential operator. Using the divergence free Dirichlet-to-Neumann map from definition 5.2.1, the local shape derivative can also be expressed as

$$\frac{\partial u'[W]}{\partial n} = S\langle \nabla u, W\rangle = S\frac{\partial u}{\partial n}\langle W, n\rangle.$$

Thus, I_2 is given by

$$I_2 = \int_\Gamma \langle V, n\rangle \left[-2\mu \sum_{i=1}^{3} \frac{\partial u_i}{\partial n} S \frac{\partial u_i}{\partial n}\langle W, n\rangle\right] dS.$$

Combining I_1 and I_2 creates the desired expression. □

Remark 5.2.3. The structure of the shape Hessian depends on the deformation approach. Using speed method, the perturbation fields V and W will be time-dependent which introduces additional acceleration terms, see equation (5.2). Also, it is in general not possible to write shape Hessians as a scalar product of the normal components of the two perturbation directions. Since discrete Hessians are almost always symmetric, the non-symmetry casts some doubts on the applicability of the analytic shape Hessian without further considerations. Furthermore, the two perturbation directions V and W are not interchangeable, meaning the shape Hessian is not symmetric and the presence of the pseudo-differential operator S results in a loss of regularity during the optimization.

5.3 Loss of Regularity, Sobolev Gradient, and Newton Direction

Definition 5.3.1 (Hilbert Space). *A vector space H combined with a scalar product $\langle .,. \rangle$ is denoted Hilbert space, if the pairing $(H, d(.,.))$ is a complete metric space. The metric $d(.,.)$ is induced by the scalar product $\langle .,. \rangle$.*

Definition 5.3.2 (Sobolev Space). *For $d \geq 1$, $\Omega \subset \mathbb{R}^d$ open, $p \in [1, +\infty]$, and $s \in \mathbb{N}$, the Sobolev space $W^{s,p}(\Omega)$ is defined by*

$$W^{s,p}(\Omega) := \{f \in L^p(\Omega) : \forall |a| \leq s, \partial_x^a f \in L^p(\Omega)\},$$

where $a = (a_1, \cdots, a_d)$, $|a| := a_1 + \cdots + a_d$, and $\partial_x^a f := \partial_{x_1}^{a_1} \cdots \partial_{x_d}^{a_d} f$ weak.

Remark 5.3.3. The above definition of Sobolev spaces can be extended to real and negative exponents. For more details on Sobolev spaces see [5].

Let $q \in W^{1,2}$ be the control of a standard optimization problem. The steepest descent optimization method is defined by updating according to the sequence

$$q_{k+1} = q_k - \alpha_k \nabla f(q_k),$$

where α_k is the steplength. Since a differentiable solution in $W^{1,2}$ is desired, the update $\nabla f(q_k)$ must be an element of $W^{1,2}$ for all k. Thus, a Sobolev gradient $\nabla_S f$ is defined as the Riesz-representative of the directional derivative in $W^{1,2}$

$$Df(q)h = \langle \nabla_S f(q), h \rangle_{W^{1,2}} \; \forall h \in W^{1,2}.$$

Consequently, the Sobolev steepest descent is given by

$$q_{k+1} = q_k - \alpha_k \nabla_S f(q_k), \tag{5.3}$$

which ensures that all updates of the control remain in the same regularity class. If the scalar product of the Sobolev space relates to the scalar product of the canonical space L^2 by

$$\langle \phi, \psi \rangle_{W^{1,2}} = \langle M\phi, \psi \rangle_{L^2},$$

then the Sobolev gradient can be computed from the ordinary L^2 gradient by

$$\nabla_S f = M^{-1} \nabla f, \tag{5.4}$$

thus solving an additional linear system is required. Consequently, the Sobolev steepest descent (5.3) results in updates according to Newton's direction if the scalar product of the Hilbert space is induced by the Hessian of the problem.

For a shape optimization problem, the control q can be associated with the boundary Γ of the domain. However, as seen in lemma 5.2.2, the Hessian of a shape optimization problem need not be symmetric. Thus, it is problematic to define a scalar product based on this Hessian. The strategy now is to identify the amount of regularity lost when using ∇f instead of $\nabla_S f$ for a shape optimization problem and choosing the smoothing operator M in equation (5.4) to re-introduce this regularity. This difference is responsible for the phenomenon of loss of regularity in shape optimization procedures. Essentially, the order of the pseudo-differential operator inside the Hessian must be identified, for which Fourier analysis is employed.

5.4 Operator Symbols and Fourier Analysis

In this section, Fourier analysis similar to [2, 3, 4] will be used to identify the pseudo-differential operator nature of the Hessian. The method is for example also employed in [34]. The Stokes problem will serve as an example application of the procedure.

Definition 5.4.1 (Symbol of an Operator). *Considering a sinusoidal perturbation $\tilde{q}(x) = \hat{q}e^{i\omega x}$ of the control q, the pseudo-differential operator nature of the Hessian H can be seen by comparing the input $\tilde{q}$ with the output $H\tilde{q}$. For example, if*

$$H\tilde{q} = i\omega\hat{q}e^{i\omega x} = i\omega\tilde{q}$$

then $\Sigma(\omega) := i\omega$ is the symbol of the Hessian, and this corresponds to a classical differential operator of order +1. If, for example, one has

$$H\tilde{q} = -\omega^2\hat{q}e^{i\omega x} = -\omega^2\tilde{q}$$

then $\Sigma(\omega) := -\omega^2$ is the symbol of the Hessian, and this corresponds to a classical differential operator of order +2. However, if

$$H\tilde{q} = |\omega|\hat{q}e^{i\omega x} = |\omega|\tilde{q},$$

then H is a pseudo-differential operator of order +1.

Lemma 5.4.2. *The Hessian of the Stokes shape optimization problem is a pseudo-differential operator with the symbol*

$$\Sigma(\omega) := \pm\sqrt{\omega_1^2 + \omega_2^2}$$

in three dimensions or

$$\Sigma(\omega) := \pm|\omega_1|$$

in two dimensions.

Proof. Assuming flow over a flat plate, the domain Ω is considered to be given by

$$\Omega = \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_3 \geq 0\}.$$

Thus, at the boundary $\Gamma = \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_3 = 0\}$, the outer normal is given by

$$n = (0, 0, 1)^T,$$

and a complex valued oscillation of the two dimensional flat plate is described by

$$x_3 = \alpha(x_1, x_2) := e^{i(\omega_1 x_1 + \omega_2 x_2)},$$

where i is the imaginary unit. Using this setting, a disturbed gradient from lemma 5.1.2 is given by

$$\begin{aligned} \tilde{G} :=& -2\mu \sum_{k=1}^{3} \frac{\partial u_k}{\partial n} \frac{\partial u_k'[\alpha]}{\partial n} \\ =& -2\mu \sum_{k=1}^{3} \frac{\partial u_k}{\partial x_3} \frac{\partial u_k'[\alpha]}{\partial x_3} \end{aligned}$$

due to the flat initial domain under consideration. To identify the symbol of the Hessian, the mapping

$$S\alpha := -2\mu \sum_{k=1}^{3} \frac{\partial u_k}{\partial x_3} \frac{\partial u_k'[\alpha]}{\partial x_3}$$

must now be characterized. Furthermore, the perturbed states $u'[\alpha]$ and $p'[\alpha]$ are also considered to be oscillatory:

$$\begin{aligned} u_k'[\alpha] &= \hat{u}_k e^{i(\omega_1 x_1 + \omega_2 x_2)} e^{\omega_3 x_3} \\ p'[\alpha] &= \hat{p} e^{i(\omega_1 x_1 + \omega_2 x_2)} e^{\omega_3 x_3}. \end{aligned}$$

The second part not containing the imaginary unit i will lead to some more convenient expressions later on. The no-slip boundary condition on $x_3 = 0$ leads to

$$u_k'[\alpha] = \hat{u}_k \alpha e^0,$$

which results in

$$\hat{u}_k = -\frac{\partial u_k}{\partial x_3} \neq 0. \tag{5.5}$$

However, the linearized Stokes PDE must also be solved inside the domain. Applying the Laplace and gradient operator on the disturbances $u'[\alpha]$ and $p'[\alpha]$ is equivalent to

$$A \begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \\ \hat{p} \end{pmatrix} \alpha(x_1, x_2) e^{\omega_3 x_3} = 0,$$

where A is given by

$$A := \begin{bmatrix} \mu(\omega_1^2+\omega_2^2-\omega_3^2) & 0 & 0 & i\omega_1 \\ 0 & \mu(\omega_1^2+\omega_2^2-\omega_3^2) & 0 & i\omega_2 \\ 0 & 0 & \mu(\omega_1^2+\omega_2^2-\omega_3^2) & \omega_3 \\ i\omega_1 & i\omega_2 & \omega_3 & 0 \end{bmatrix}.$$

This is only non-contradictory to the consequences of the boundary condition (5.5), if the linearized system matrix A does not have full rank, which means the determinant of A must vanish:

$$\begin{aligned} \det(A) &= \left[\mu\left(\omega_1^2+\omega_2^2-\omega_3^2\right)\right]^2\left(\omega_1^2+\omega_2^2-\omega_3^2\right) \overset{!}{=} 0 \\ \Rightarrow \omega_3 &= \pm\sqrt{\omega_1^2+\omega_2^2}. \end{aligned}$$

Hence, it is possible to remove ω_3 from the equations and the local shape derivative of the velocity is given by

$$\begin{aligned} u_k'[\alpha] &= -\frac{\partial u_k}{\partial x_3}\alpha e^{\pm\sqrt{\omega_1^2+\omega_2^2}x_3} \\ \Rightarrow \left.\frac{\partial u_k'[\alpha]}{\partial x_3}\right|_{x_3=0} &= \left[-\frac{\partial^2 u_k}{\partial x_3^2} \mp \frac{\partial u_k}{\partial x_3}\sqrt{\omega_1^2+\omega_2^2}\right]\alpha, \end{aligned}$$

and the operator S is given by

$$S = -2\mu\sum_{k=1}^{3}\frac{\partial u_k}{\partial x_3}\left[-\frac{\partial^2 u_k}{\partial x_3^2} \pm \frac{\partial u_k}{\partial x_3}\sqrt{\omega_1^2+\omega_2^2}\right]. \tag{5.6}$$

In two dimensions, one can assume $\omega_2 \equiv 0$ and the symbol becomes

$$\pm|\omega_1|,$$

which is the symbol of a pseudo-differential operator of order +1 closely related to the Dirichlet-to-Neumann map. □

Crucial for this method is finding the roots of the characteristic polynomial of the linearized state equation. For the Stokes case, this is manageable. However, as discussed later on, this polynomial is considerably more complex in case of the Navier–Stokes equations.

5.5 Application

Knowledge of the Hessian operator symbol will now be used to accelerate the actual optimization of the shape of an obstacle in a Stokes fluid. First, the in-house flow solver will be discussed briefly. Afterwards, the analytic derivation of the shape Hessian will be repeated discretely, resulting in a re-smoothing procedure for a Sobolev or approximative Newton-method which greatly accelerates the numerical optimization scheme.

5.5.1 Flow Solver

The two dimensional Stokes equations are discretized by mixed Taylor–Hood finite elements. The velocities are discretized using a six node triangular element with quadratic ansatz functions while the pressure exists on standard three node triangular elements using linear ansatz functions. This leads to a linear system of the type

$$\begin{bmatrix} A & B^* \\ B & 0 \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = 0,$$

where A is the symmetric discrete diffusion operator and B is the divergence operator. Both A and B^* also contain the inflow and no-slip boundary conditions but not B, as the divergence freedom must be valid everywhere, including the boundary nodes. This is denoted by the symbol B^* instead of B^T. Since the outflow boundary condition is the natural finite element boundary condition, no special treatment is necessary. Due to the saddle-point structure, this system is surprisingly difficult to solve iteratively without carefully constructed preconditioners. Here, the system is solved using a sparse direct linear solver such as [12].

5.5.2 Discrete Hessian Symbol

The real valued analogon to the complex valued Fourier mode α is a perturbation of the type

$$\tilde{q}_\omega(x) = \sin(2\pi\omega x).$$

When the perimeter of Γ is denoted with ℓ, one can see that a standing wave on Γ can be expressed by

$$\tilde{q}_\omega(\varphi) = \sin(2\pi\omega\frac{\varphi}{\ell}),$$

where φ is the parameterization of the curve. This perturbation is called the “input signal”. Thus, a deformed domain is of the type

$$\Gamma_\epsilon[\tilde{q}_\omega] = \left\{x(\varphi) + \epsilon\tilde{q}_\omega(\varphi)n(\varphi) : \varphi \in [0, \ell]\right\}.$$

The shape Hessian in direction $[\tilde{q}_\omega]$ is the limit

$$d^2J(u, p, \Gamma)[V][\tilde{q}_\omega] = \lim_{\epsilon\to 0} \frac{dJ(u_\epsilon, p_\epsilon, \Gamma_\epsilon)[V] - dJ(u, p, \Gamma)[V]}{\epsilon},$$

which is replaced by finite differences and called the “output signal”. Assuming a standard differential operator, the output signal can be interpreted as

$$\begin{aligned} d^2J(u, p, \Gamma)[V][\tilde{q}_\omega](\varphi) &= \sum_{k=1}^{2n} c_k(\varphi)\frac{d^k\tilde{q}_\omega}{d\varphi^k}(\varphi) \\ &= \left(\sum_{k=0}^{n}(-1)^k c_{2k}(\varphi)\left(\frac{2\pi\omega}{\ell}\right)^{2k}\right)\sin(2\pi\omega\frac{\varphi}{\ell}) \\ &\quad + \left(\sum_{k=1}^{n}(-1)^{k-1} c_{2k-1}(\varphi)\left(\frac{2\pi\omega}{\ell}\right)^{2k-1}\right)\cos(2\pi\omega\frac{\varphi}{\ell}). \end{aligned}$$

That means, the output signal is first split in a wave that oscillates in phase, which corresponds to the sin-part, and a wave that oscillates out of phase, which corresponds to the cos-part. Next, the value of each sum is determined: For this, the above discrete shape Hessian is evaluated multiple times with different values for ω, and the change in amplitude of the output signal is observed.

Assuming a Hessian symbol of $\Sigma(\omega) = \omega^2$, which is a standard differential operator, the procedure works as follows: For two input oscillations of frequencies ω and 2ω, the respective output signals are computed. Because the symbol is real valued, both output signals should have the same phase as the respective input oscillations. The exponent of the symbol can then be approximated by the amplitude of the output signals. Under the assumption of $\Sigma(\omega) = \omega^2$, the output signal corresponding to the input oscillation of 2ω should have an amplitude four times stronger than the output signal corresponding to the input oscillation of ω.

However, the behavior of pseudo-differential operators can also be observed nicely. If one observes an output signal that is oscillating in the same phase as the input signal but with an amplitude that scales linearly with the input frequency, one can conclude that the symbol of the discrete operator must be $|\omega|$ as the discrete approach translates the imaginary part of the fourier mode to phase shifts. This comparison is now conducted for the Stokes equations:

Flow around a circular obstacle in a channel is considered. Parts of the initial geometry and fluid velocity are depicted in figure 5.2. A parabolic inflow profile is used, such that the Reynolds number would be *Re* = 80 in a Navier–Stokes fluid. The initial circle consists of 500 surface nodes. The position of each of these nodes is the control of the optimization problem. Due to the Taylor–Hood finite elements, 500 additional velocity nodes are also present on each mid-section of the surface triangles. However, these nodes are not part of the design unknowns, as each side of the Taylor–Hood triangles must remain a straight line. For a proper Hessian analysis, oscillations of high frequency are required, and the state must be computed correctly, resulting in the need of 500

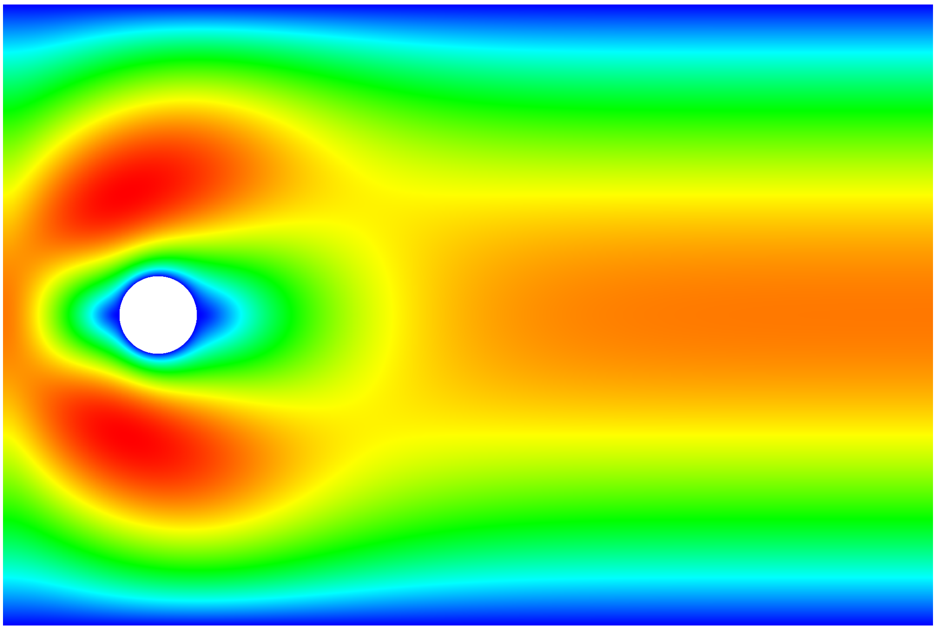


Figure 5.2: Initial domain for the Stokes problem. Color denotes speed.

design parameters. However, due to the Hadamard form of the shape derivative, evaluating the shape gradient at 500 or more nodes is no problem at all.

First, a sin-wave of frequency $\omega = 50$ and with amplitude $\epsilon = 0.002$ is modulated onto the circle. The phase portrait of incoming and outgoing wave is presented in figure 5.3. One can clearly see how the output signal consists of a single wave of opposite phase compared to the input wave. Therefore, the cos-part of the output signal must be zero, which precisely matches the analytical prediction in lemma 5.4.2. Next, the input frequency is halved to $\frac{1}{2}\omega = 25$. As a consequence, the maximum amplitude decreases by almost exactly a factor of 2. Thus, the symbol of the discrete Hessian is $|\omega|$ as output amplitude scales linearly with input frequency. The actual waves are presented in figure 5.4. Altogether, one can see that the discrete method matches the analytical prediction from lemma 5.4.2 extremely well.

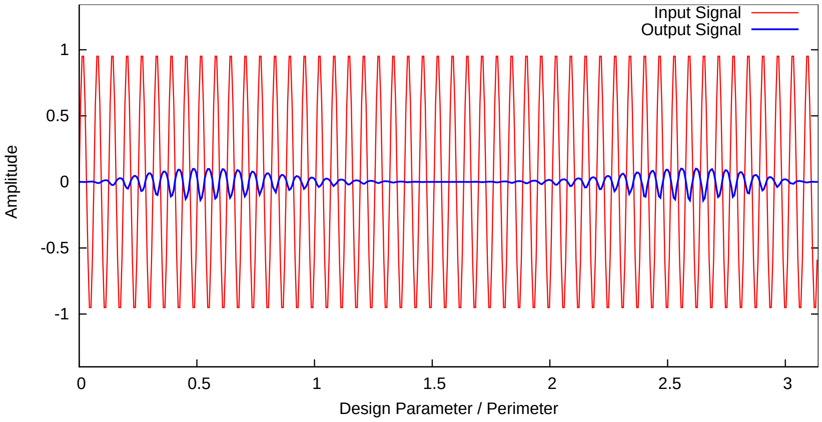


Figure 5.3: Incoming and outgoing wave for the Stokes problem.

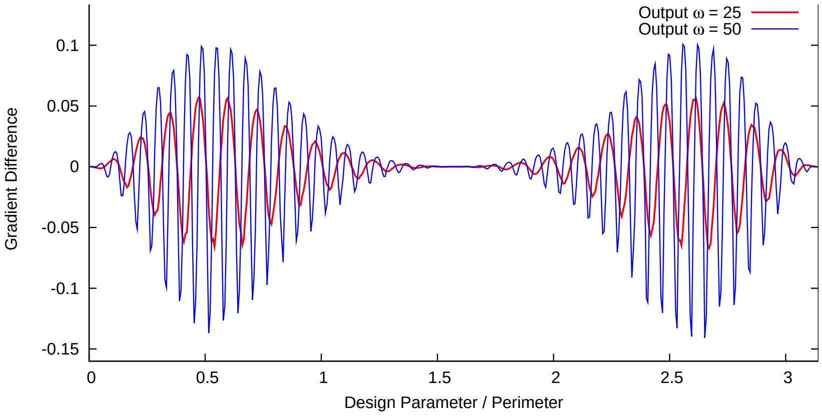


Figure 5.4: Amplitude of the Stokes problem response scales linearly with the input frequency.

5.5.3 Optimization

A good Hessian approximation for the Stokes problem must mimic a pseudo-differential operator with symbol $|\omega|$. In two dimensions, the shape boundary Γ is a line, and finite differences can very efficiently be used to approximate standard differential operators, e.g. operators with symbols 1, $i\omega$, $-\omega^2$. Three dimensional flow around a two dimensional surface with the need for a two dimensional operator approximation on an unstructured surface mesh will be discussed in chapter 8. For a true pseudo-differential operator with symbol $|\omega|$, a discretization using finite differences is not straight forward. A symbol which can easily be approximated using finite differences is ω^2, and this corresponds to the tangential Laplace operator, also known as Laplace–Beltrami operator Δ_Γ. Thus, the discrete Hessian is approximated by

$$H_h \approx k\Delta_\Gamma^h + I, \tag{5.7}$$

where $I \in \mathbb{R}^{m\times m}$ is the identity matrix for m surface mesh nodes and k is a smoothing parameter. Including the identity ensures coercivity of the operator and has led to superior results during application. The matrix Δ_Γ^h is given by the central difference stencil

$$\frac{g_{i-1} - 2g_i + g_{i+1}}{h^2},$$

where g_i denotes the gradient or any other surface quantity to smooth at node i on the surface. If τ_ℓ and τ_r denote the tangent left and right of node i, i.e.

$$\begin{aligned}\tau_\ell &:= \frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|_2}\\ \tau_r &:= \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|_2},\end{aligned}$$

then $h = \|\frac{1}{2}(\tau_\ell + \tau_r)\|$. The smoothing parameter k allows some fine tuning. Using the symbol ω^2 instead of $|\omega|$ effectively results in an under-smoothing of low frequencies and in an over-smoothing of high frequencies. Since in the discrete setting the maximum frequency that can be represented depends on the number of nodes defining the surface, one can assume a finite range of frequencies $[0, \omega_{\max}]$. Consequently, the smoothing parameter k can be chosen such that the average smoothing of the operators $|\omega|$ and $k\omega^2 + 1$ are the same on average:

$$\int\limits_0^{\omega_{\max}} k\omega^2 + 1 \, d\omega \overset{!}{=} \int\limits_0^{\omega_{\max}} \beta|\omega| \, d\omega.$$

Solving the above for k results in

$$k = \frac{\frac{3}{2}\beta\omega_{\max} - 3}{\omega_{\max}^2}, \tag{5.8}$$

where

$$\beta = -2\mu \sum_{i=1}^{2} \left(\frac{\partial u_i}{\partial x_2}\right)^2 \tag{5.9}$$

as seen in equation (5.6) after dropping one spatial dimension. Assuming that resolving a single wave properly requires two to four mesh nodes, a surface mesh of 200 design parameters would lead to a value of $\omega_{\max}$ between 50 and 100. Hence, $\omega_{\max} = 75$ is used.

A Sobolev method based on this preconditioning is now compared with an unpreconditioned one, i.e. standard steepest descent. Since the unpreconditioned steepest descent algorithm is not level independent, the number of variable boundary nodes is reduced to 200 as with 500 variable nodes discussed earlier a direct comparison of the preconditioned and the non-preconditioned iterations would not have been possible. The steplength d for the unpreconditioned iteration was constant with $d = 0.02$, which was found to be the maximum steplength useable with 200 variable

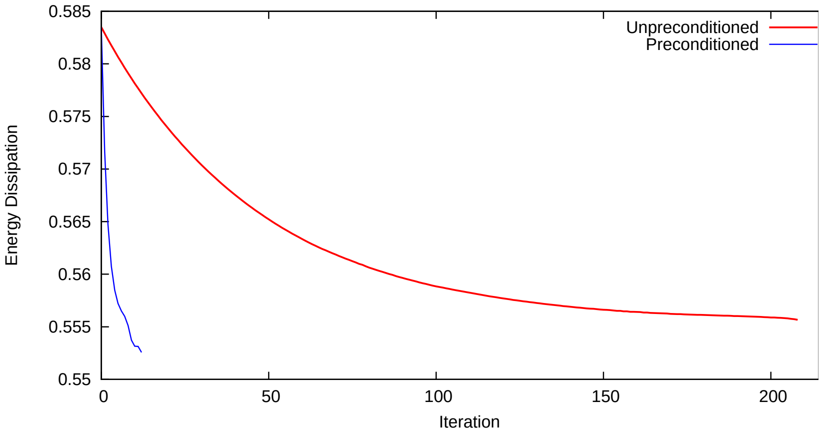


Figure 5.5: Convergence rates of the preconditioned and unpreconditioned Stokes problem.

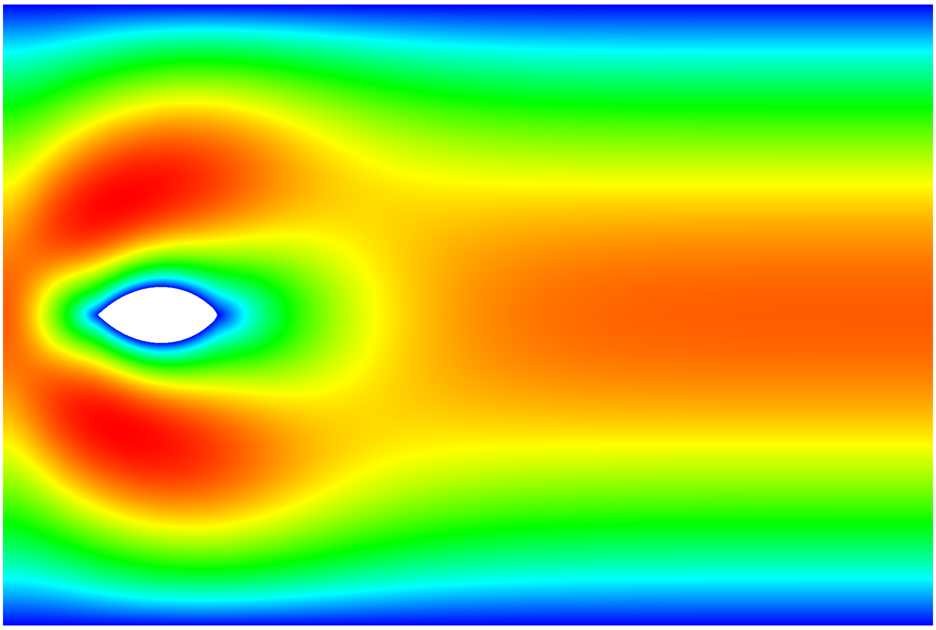


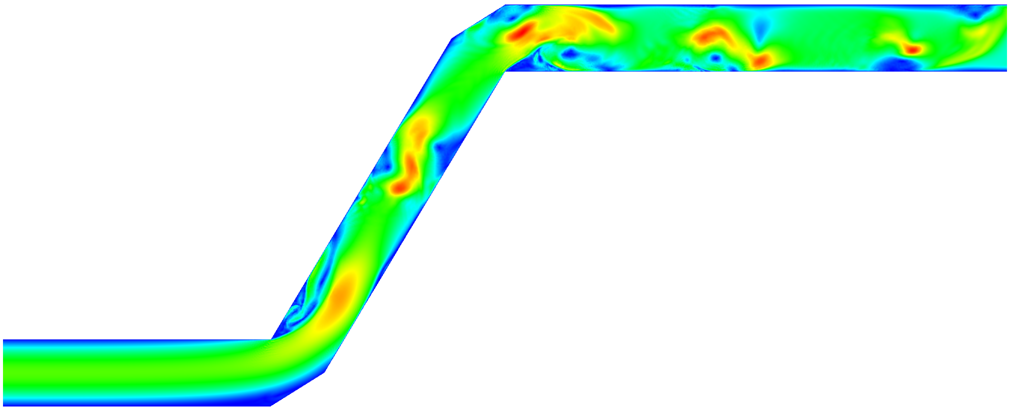
Figure 5.6: Final shape for the preconditioned Stokes problem.

boundary nodes. The steplength for the preconditioned iteration was chosen to be $d = 0.25$, and a backtracking linesearch was used to determine convergence. The backtracking linesearch has reported optimality in iteration 12. This means a saving of 94% if iteration 200 is accepted as optimal in the unpreconditioned case. The history of the optimization is presented in figure 5.5.

According to [49], the optimal shape is similar to a pointed prolate spheroid with 60° front and back angle. The computation with preconditioning matches this very well, whereas without preconditioning or an improper choice of ω_{max} the rear end is computed slightly too round. This can also be seen in the value of the unpreconditioned objective function shown in figure 5.5, which is slightly higher than the preconditioned one. The computed optimal wedge shape with preconditioning is shown in figure 5.6. Since the volume constraint is also given by

$$\int_{\text{int}\,\Gamma_0} 1\; dA,$$

it is very easy to see that the shape derivative of the volume constraint is constant 1. Thus, a degeneration of the shape is prevented by keeping the volume constant, which is done by a projection step in normal direction after each shape optimality update. After one update of the boundary, the flow domain is simply remeshed with unstructured triangular Taylor–Hood elements.



Chapter 6

Shape Optimization and Navier–Stokes Fluids

6.1 Problem Introduction and First Order Calculus

This chapter extends shape optimization for Stokes fluids to the nonlinear incompressible Navier–Stokes equations. The Navier–Stokes problem is no longer self-adjoint, such that the shape derivative will now require solving an adjoint PDE. Parts of the results of this chapter have already been published in [61, 62]. When it comes to incompressible Newtonian fluids such as water or air moving considerably slower than the local speed of sound, the incompressible Navier–Stokes equations are considered to describe a wide variety of real world problems properly. As such, many more objective functions than the energy dissipation, equation (4.12), can be relevant for practical application. For this reason, the Navier–Stokes shape optimization problem will be considered in a more general problem setting. In the literature, the problem is seldom considered from a pure shape optimization point of view, except for [38], where the existence of the shape gradient for specific objective functions is shown using surprisingly weak regularity assumptions, and [51, 52, 54], where many details on the differentiability and existence of optimal domains can be found.

Definition 6.1.1 (Navier–Stokes Problem)**.** *Using the same notation as before, the Navier–Stokes* *problem is given by*

$$\min_{(u,p,\Omega)} J(u,p,\Omega) := \int\limits_{\Omega} f(u,Du,p)\ dA + \int\limits_{\Gamma_0} g(u,D_n u,p,n)\ dS \tag{6.1}$$

$$\begin{aligned} &\textit{subject to} \\ -\mu\Delta u + \rho u\nabla u + \nabla p &= \rho G \quad \textit{in} \quad \Omega \\ \operatorname{div} u &= 0 \\ u &= u_+ \quad \textit{on} \quad \Gamma_+ \\ u &= 0 \quad\ \ \textit{on} \quad \Gamma_0 \\ pn - \mu\frac{\partial u}{\partial n} &= 0 \quad\ \ \textit{on} \quad \Gamma_- . \end{aligned} \tag{6.2}$$

Here, $f : \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ are assumed to be continuously differentiable in each argument. They form the volume and surface part of the objective function. The outflow boundary condition on Γ_- is the finite element "do nothing" outflow condition that naturally arises due to integration by parts during the finite element matrix assembly. It is also needed to satisfy the conservation of mass of the fluid. Due to the no-slip boundary condition on Γ_0, the tangent derivative of the velocities is zero, and it is sufficient to consider the derivative in normal direction $D_n u := Du \cdot n$ on Γ_0 only. In order to keep the notation readable, components of the Jacobian are denoted as follows:

$$\begin{aligned} Du &=: \left[a_{ij}\right]_{ij} \in \mathbb{R}^{d\times d} \\ D_n u = Du \cdot n &= \frac{\partial u}{\partial n} =: [b_i]_i \in \mathbb{R}^d . \end{aligned}$$

Since the pressure has no explicit boundary condition on Γ_0 but is implicitly linked with the velocity, the following restriction needs to be imposed on g, the boundary part of the objective, such that one can later arrive at a consistent adjoint boundary condition: The surface part g is chosen such that there exists a functional $\lambda : \Omega \to \mathbb{R}^d$ satisfying the following conditions on Γ_0:

$$\begin{aligned} \lambda_i &= \frac{1}{\mu}\frac{\partial g}{\partial b_i}\ \forall i = 1,\dots,d \\ \langle \lambda, n\rangle &= -\frac{\partial g}{\partial p}. \end{aligned}$$

Remark 6.1.2. The restriction on the surface part g is less limiting than it might appear. A consequence is that for a force minimization, the forces should be chosen in line with the state equation, i.e. since the state equation describes a viscous fluid, the objective function should also include the viscous forces. For a drag minimization at zero angle of attack, one has according to remark 4.1.12

$$g(u,D_n u,p,n) = -\mu\frac{\partial u_1}{\partial n} + pn_1,$$

which leads to

$$\begin{aligned} \frac{\partial g}{\partial p} &= n_1 \\ \frac{\partial g}{\partial b_i} &= -\mu\delta_{1,i}, \end{aligned}$$

and the above is satisfied with $\lambda_i = -\delta_{1,i}$. The inclusion of higher derivatives on the velocities within the objective (6.2) is possible, but further limits the allowed surface functionals g and will not be considered.

Theorem 6.1.3 (Shape Derivative in Sensitivity Formulation)**.** *The shape derivative of* (6.1) *and* (6.2) *in sensitivity formulation is given by:*

$$
\begin{aligned}
&dJ(u,p,\Omega)[V] = \\
&\int_{\Gamma_0} \langle V,n\rangle f(u,Du,p)\, dS && (6.3)\\
&+\int_{\Omega} \left(\sum_{i=1}^{d} \frac{\partial f}{\partial u_i} u_i'[V]\right) + \left(\sum_{i,j=1}^{d} \frac{\partial f}{\partial a_{ij}} \frac{\partial u_i'[V]}{\partial x_j}\right) + \frac{\partial f}{\partial p} p'[V]\, dA && (6.4)\\
&+\int_{\Gamma_0} \langle V,n\rangle \left[D_{(u,b,p)} g(u,D_n u,p,n)\cdot n + \kappa g(u,D_n u,p,n)\right]\, dS && (6.5)\\
&+\int_{\Gamma_0} \left(\sum_{i=1}^{d} \frac{\partial g}{\partial u_i} u_i'[V]\right) + \left(\sum_{i,j=1}^{d} \frac{\partial g}{\partial b_i} \frac{\partial u_i'[V]}{\partial x_j} n_j\right) + \frac{\partial g}{\partial p} p'[V]\, dS && (6.6)\\
&+\int_{\Gamma_0} \sum_{i=1}^{d} \frac{\partial g}{\partial n_i} dn_i[V]\, dS, && (6.7)
\end{aligned}
$$

where $u'[V]$ and $p'[V]$ are given as the solution of the linearized Navier–Stokes equations

$$
\begin{aligned}
-\mu\Delta u'[V] + \rho\left(u'[V]\nabla u + u\nabla u'[V]\right) + \nabla p'[V] &= 0 \quad \text{in} \quad \Omega\\
\operatorname{div} u'[V] &= 0
\end{aligned}
$$

with boundary conditions

$$
\begin{aligned}
u_i'[V] &= -\langle V,n\rangle \frac{\partial u_i}{\partial n} && \text{on} \quad \Gamma_0 && (6.8)\\
u_i'[V] &= 0 && \text{on} \quad \Gamma_+ && (6.9)\\
p'[V]n_i - \mu\langle \nabla u_i'[V], n\rangle &= 0 && \text{on} \quad \Gamma_-. && (6.10)
\end{aligned}
$$

Proof. Formal shape differentiation of (6.1) and (6.2) according to chapter 3 and lemma 3.3.14. The boundary condition on Γ_0 is given by lemma 3.4.3. Since the other boundaries are considered fixed, one does not have to consider differences between the material and the local shape derivative, and a linearization is straight forward. □

For the adjoint formulation of the shape derivative further discussions including adjoint functionals $\lambda : \Omega \to \mathbb{R}^d$ and $\lambda_p : \Omega \to \mathbb{R}$ are necessary.

Lemma 6.1.4. *For a sufficiently smooth, arbitrary $\lambda : \Omega \to \mathbb{R}^d$ and $\lambda_p : \Omega \to \mathbb{R}$ the relation*

$$
0 = \int_{\Omega} \sum_{i=1}^{d} \left[-\mu\Delta\lambda_i - \rho\left(\sum_{j=1}^{d} \frac{\partial \lambda_j}{\partial x_i} u_j + \frac{\partial \lambda_i}{\partial x_j} u_j\right) - \frac{\partial \lambda_p}{\partial x_i}\right] u_i'[V]\, dA \tag{6.11}
$$

$$- \int_\Omega \sum_{i=1}^{d} \frac{\partial \lambda_i}{\partial x_i} p'[V] \, dA \tag{6.12}$$

$$+ \int_\Gamma \sum_{i=1}^{d} \left[\mu \frac{\partial \lambda_i}{\partial n} + \rho \sum_{j=1}^{d} \left(\lambda_j u_j n_i + \lambda_i u_j n_j \right) \right] u_i'[V] \, dS \tag{6.13}$$

$$+ \int_\Gamma \lambda_p \sum_{i=1}^{d} u_i'[V] n_i \, dS + \int_\Gamma \sum_{i=1}^{d} \lambda_i n_i p'[V] \, dS + \int_\Gamma \sum_{i=1}^{d} -\mu \lambda_i \frac{\partial u_i'[V]}{\partial n} \, dS \tag{6.14}$$

holds.

Proof. Multiplying the volume part of the linearized Navier–Stokes equations with an arbitrary λ and λ_p results in

$$\begin{aligned} 0 = &\int_\Omega \sum_{i=1}^{d} \lambda_i \left[-\mu \Delta u_i'[V] + \rho \left(\sum_{j=1}^{d} u_j'[V] \frac{\partial u_i}{\partial x_j} + u_j \frac{\partial u_i'[V]}{\partial x_j} \right) + \frac{\partial p'[V]}{\partial x_i} \right] dA \\ &+ \int_\Omega \lambda_p \text{div}\, u'[V] \, dA. \end{aligned}$$

Integration by parts gives

$$\begin{aligned} \int_\Omega \sum_{i=1}^{d} -\mu \lambda_i \Delta u_i'[V] \, dA = &\int_\Gamma \sum_{i=1}^{d} -\mu \left(\lambda_i \frac{\partial u_i'[V]}{\partial n} - u_i'[V] \frac{\partial \lambda_i}{\partial n} \right) dS \\ &+ \int_\Omega \sum_{i=1}^{d} -\mu u_i'[V] \Delta \lambda_i \, dA. \end{aligned}$$

Likewise, due to div $u'[V] = 0$,

$$\int_\Omega \sum_{i,j=1}^{d} \lambda_i u_j'[V] \frac{\partial u_i}{\partial x_j} \, dA = \int_\Gamma \sum_{i,j=1}^{d} \lambda_i u_j'[V] u_i n_j \, dS - \int_\Omega \sum_{i,j=1}^{d} \frac{\partial \lambda_i}{\partial x_j} u_j'[V] u_i \, dA. \tag{6.15}$$

Note that in the above equation the index on the local shape derivative is *j* and not *i*. To derive the desired expression, the indices *i* and *j* must be switched. Integration by parts on the second part of the linearized convection results in

$$\int_\Omega \sum_{i,j=1}^{d} \lambda_i u_j \frac{\partial u_i'[V]}{\partial x_j} \, dA = \int_\Gamma \sum_{i,j=1}^{d} \lambda_i u_j u_i'[V] n_j \, dS - \int_\Omega \sum_{i,j=1}^{d} \frac{\partial \lambda_i}{\partial x_j} u_j u_i'[V] \, dA. \tag{6.16}$$

The pressure variation provides

$$\int_\Omega \sum_{i=1}^{d} \lambda_i \frac{\partial p'[V]}{\partial x_i} \, dA = \int_\Gamma \lambda_i n_i p'[V] \, dS - \int_\Omega \sum_{i=1}^{d} \frac{\partial \lambda_i}{\partial x_i} p'[V] \, dA,$$

and the divergence constraint provides

$$\int_\Omega \lambda_p \sum_{i=1}^d \frac{\partial u_i'[V]}{\partial x_i}\, dA = \int_\Gamma \lambda_p \sum_{i=1}^d u_i'[V] n_i \, dS - \int_\Omega \sum_{i=1}^d \frac{\partial \lambda_p}{\partial x_i} u_i'[V]\, dA.$$

Summarizing the above creates the desired expression. □

Using lemma 6.1.4, it is now possible to derive the adjoint right hand side in the volume:

Lemma 6.1.5 (Adjoint Right Hand Side, Volume)**.** *The adjoint equation must fulfill in the domain Ω:*

$$-\mu\Delta\lambda_i - \rho \sum_{j=1}^d \left(\frac{\partial \lambda_j}{\partial x_i} u_j + \frac{\partial \lambda_i}{\partial x_j} u_j \right) - \frac{\partial \lambda_p}{\partial x_i} = \frac{\partial f}{\partial u_i} - \sum_{j=1}^d \frac{\partial}{\partial x_j} \frac{\partial f}{\partial a_{ij}}$$
$$\operatorname{div} \lambda = \frac{\partial f}{\partial p}.$$

Proof. Due to equations (6.11) – (6.14) summing to zero, they can be added to the preliminary gradient (6.3) – (6.7). Integration by parts on equation (6.4) yields

$$\begin{aligned}
&\int_\Omega \left(\sum_{i=1}^d \frac{\partial f}{\partial u_i} u_i'[V] \right) + \left(\sum_{i,j=1}^d \frac{\partial f}{\partial a_{ij}} \frac{\partial u_i'[V]}{\partial x_j} \right) + \frac{\partial f}{\partial p} p'[V]\, dA \\
&= \int_\Gamma \sum_{i,j=1}^d \frac{\partial f}{\partial a_{ij}} u_i'[V] n_j \, dS && (6.17)\\
&+ \int_\Omega \sum_{i=1}^d \left(\frac{\partial f}{\partial u_i} - \sum_{j=1}^d \frac{\partial}{\partial x_j} \frac{\partial f}{\partial a_{ij}} \right) u_i'[V]\, dA + \int_\Omega \frac{\partial f}{\partial p} p'[V]\, dA, && (6.18)
\end{aligned}$$

and a direct comparison between the above and equations (6.11) and (6.12) reveals the required adjoint right hand side in Ω. Note that this has introduced a new boundary term. □

Lemma 6.1.6 (Adjoint Boundary Condition at Inflow)**.** *The adjoint boundary condition on the inflow boundary Γ_+ is given by*

$$\lambda = 0$$
$$\lambda_p \quad \textit{free}.$$

Proof. Since the inflow velocity is fixed and independent of the shape of the fluid obstacle, we have $u'[V] = 0$ on Γ_+. Hence, the only term appearing on Γ_+ is the normal variation of $u'[V]$ and the pressure variation $p'[V]$ from equation (6.14):

$$\int_{\Gamma_+} \sum_{i=1}^d \lambda_i n_i p'[V]\, dS + \int_{\Gamma_+} \sum_{i=1}^d -\mu \lambda_i \frac{\partial u_i'[V]}{\partial n}\, dS,$$

which is removed by $\lambda = 0$ on Γ_+. □

Lemma 6.1.7 (Adjoint Boundary Condition at No-Slip)**.** *The adjoint boundary condition on the no-slip boundary Γ_0 is given by*

$$\begin{aligned} \lambda_i &= \frac{1}{\mu}\frac{\partial g}{\partial b_i}\ \forall i = 1, \dots, d \\ \langle \lambda, n\rangle &= -\frac{\partial g}{\partial p} \\ \lambda_p &\quad \textit{free}. \end{aligned}$$

Proof. The sensitivities on Γ_0 are equations (6.17), (6.6), (6.13), and (6.14):

$$\begin{aligned} &\int\limits_{\Gamma_0} \sum_{i,j=1}^{d} \frac{\partial f}{\partial a_{ij}} u_i'[V] n_j \; dS \\ &+ \int\limits_{\Gamma_0} \left(\sum_{i=1}^{d} \frac{\partial g}{\partial u_i} u_i'[V] \right) + \left(\sum_{i,j=1}^{d} \frac{\partial g}{\partial a_{ij}} \frac{\partial u_i'[V]}{\partial x_j} \right) + \frac{\partial g}{\partial p} p'[V] \; dS \\ &+ \int\limits_{\Gamma_0} \sum_{i=1}^{d} \left[\mu \frac{\partial \lambda_i}{\partial n} + \rho \sum_{j=1}^{d} \left(\lambda_j u_j n_i + \lambda_i u_j n_j \right) \right] u_i'[V] \; dS \\ &+ \int\limits_{\Gamma_0} \lambda_p \sum_{i=1}^{d} u_i'[V] n_i \; dS + \int\limits_{\Gamma_0} \sum_{i=1}^{d} \lambda_i n_i p'[V] \; dS + \int\limits_{\Gamma_0} \sum_{i=1}^{d} -\mu \lambda_i \frac{\partial u_i'[V]}{\partial n} \; dS. \end{aligned}$$

Using the no-slip boundary condition and the boundary condition for the local shape derivative, the above transforms to

$$\begin{aligned} &\int\limits_{\Gamma_0} \langle V, n\rangle \left[-\sum_{i=1}^{d} \left(\frac{\partial g}{\partial u_i} + \mu \frac{\partial \lambda_i}{\partial n} + \lambda_p n_i + \sum_{j=1}^{d} \frac{\partial f}{\partial a_{ij}} n_j \right) \frac{\partial u_i}{\partial n} \right] \; dS \\ &+ \int\limits_{\Gamma_0} \left(\sum_{i=1}^{d} \frac{\partial g}{\partial b_i} \frac{\partial u_i'[V]}{\partial n} \right) + \left(\frac{\partial g}{\partial p} + \sum_{i=1}^{d} \lambda_i n_i \right) p'[V] \; dS \\ &+ \int\limits_{\Gamma_0} \sum_{i=1}^{d} -\mu \lambda_i \frac{\partial u_i'[V]}{\partial n} \; dS, \end{aligned}$$

where the first part now also enters the gradient (6.3) – (6.7). Expressing ∇u_i in local coordinates on the boundary results in

$$\nabla u_i = \langle \nabla u_i, n\rangle n + \sum_{j=1}^{d} \langle \nabla u_i, \tau_j\rangle \tau_j,$$

hence

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial n} n_j \Rightarrow 0 = \lambda_p \sum_{i=1}^{d} \frac{\partial u_i}{\partial n} n_i$$

due to the mass conservation on Γ_0. Consequently, λ_p does not receive a boundary condition. The remaining sensitivities can be eliminated by

$$\begin{aligned} \lambda_i &= \frac{1}{\mu}\frac{\partial g}{\partial b_i} \; \forall i = 1, \dots, d \\ \langle \lambda, n \rangle &= -\frac{\partial g}{\partial p}. \end{aligned}$$

□

In order to arrive at a complete adjoint system, the boundary conditions for the adjoint variables at the outflow boundary are also needed:

Lemma 6.1.8 (Adjoint Boundary Condition at Outflow)**.** *The adjoint boundary condition on the outflow boundary Γ_- is given by*

$$\mu\frac{\partial \lambda_i}{\partial n} + \rho\left(\sum_{j=1}^{d} \lambda_j u_j n_i + \lambda_i u_j n_j\right) + \lambda_p n_i = 0.$$

Proof. Inserting equation (6.10) into equations (6.11) – (6.14), the remaining sensitivity is

$$\begin{aligned} &\int\limits_{\Gamma_-} \sum_{i=1}^{d}\left[\mu\frac{\partial \lambda_i}{\partial n} + \rho\sum_{j=1}^{d}\left(\lambda_j u_j n_i + \lambda_i u_j n_j\right)\right] u_i'[V]\, dS \\ +&\int\limits_{\Gamma_-} \lambda_p \sum_{i=1}^{d} u_i'[V] n_i \, dS. \end{aligned}$$

Hence, the required boundary condition is

$$\mu\frac{\partial \lambda_i}{\partial n} + \rho\left(\sum_{j=1}^{d} \lambda_j u_j n_i + \lambda_i u_j n_j\right) + \lambda_p n_i = 0.$$

□

Theorem 6.1.9 (Shape Derivative for the General Navier–Stokes Problem)**.** *The shape derivative in Hadamard form for the problem under consideration is given by*

$$\begin{aligned} &dJ(u, p, \Omega)[V] = \\ &\int\limits_{\Gamma_0} \langle V, n\rangle f(u, Du, p)\, dS \\ +&\int\limits_{\Gamma_0} \langle V, n\rangle \left[D_{(u,b,p)} g(u, D_n u, p, n)\cdot n + \kappa g(u, D_n u, p, n)\right]\, dS \\ +&\int\limits_{\Gamma_0} \langle V, n\rangle \left[-\sum_{i=1}^{d}\left(\frac{\partial g}{\partial u_i} + \mu\frac{\partial \lambda_i}{\partial n} + \sum_{j=1}^{d}\frac{\partial f}{\partial a_{ij}} n_j\right)\frac{\partial u_i}{\partial n}\right]\, dS \end{aligned}$$

$$+\int_{\Gamma_0} \langle V, n\rangle \left[(\mathrm{div}_\Gamma \nabla_n g) - \kappa\langle \nabla_n g, n\rangle\right] \, dS,$$

where $\nabla_n g$ denotes the vector consisting of components $\frac{\partial g}{\partial n_i}$. Furthermore, u and p solve the incompressible Navier–Stokes equations

$$\begin{aligned}
-\mu\Delta u + \rho u \nabla u + \nabla p &= \rho G \quad \text{in} \quad \Omega\\
\mathrm{div}\, u &= 0\\
u &= u_+ \quad \text{on} \quad \Gamma_+\\
u &= 0 \quad \text{on} \quad \Gamma_0\\
pn - \mu\frac{\partial u}{\partial n} &= 0 \quad \text{on} \quad \Gamma_-,
\end{aligned}$$

and λ and λ_p solve the adjoint incompressible Navier–Stokes equations

$$\begin{aligned}
-\mu\Delta\lambda_i - \rho\sum_{j=1}^{d}\left(\frac{\partial\lambda_j}{\partial x_i}u_j + \frac{\partial\lambda_i}{\partial x_j}u_j\right) - \frac{\partial\lambda_p}{\partial x_i} &= \frac{\partial f}{\partial u_i} - \sum_{j=1}^{d}\frac{\partial}{\partial x_j}\frac{\partial f}{\partial a_{ij}} \quad \text{in} \quad \Omega\\
\mathrm{div}\,\lambda &= \frac{\partial f}{\partial p}
\end{aligned}$$

with boundary conditions

$$\begin{aligned}
\lambda &= 0 \quad \text{on} \quad \Gamma_+\\
\lambda_i &= \frac{1}{\mu}\frac{\partial g}{\partial b_i} \quad \text{on} \quad \Gamma_0\\
\langle\lambda, n\rangle &= -\frac{\partial g}{\partial p} \quad \text{on} \quad \Gamma_0\\
\mu\frac{\partial\lambda_i}{\partial n} + \rho\left(\sum_{j=1}^{d}\lambda_j u_j n_i + \lambda_i u_j n_j\right) + \lambda_p n_i &= 0 \quad \text{on} \quad \Gamma_-.
\end{aligned}$$

Proof. The adjoint boundary conditions are derived in lemma 6.1.6, 6.1.7, and 6.1.8. The adjoint right hand side is derived in lemma 6.1.5, and removing the shape derivative of the normal is described in lemma 3.3.14 and remark 3.3.15. □

Lemma 6.1.10 (Incompressible Navier–Stokes Fluid Forces)**.** *When considering flow around an airfoil or any other obstacle, one does not want to make a new mesh in case the airfoil has a different angle of attack. Instead, most flow solver rotate the coordinate system internally. This leads to a rotated form of equation* (4.10)*. For drag at angle of attack α, the incident vector a is given by*

$$a := (\cos\alpha, 0, \sin\alpha)^T.$$

The first component denotes chord direction, the second span direction, and the third wing thickness. The drag force an incompressible Navier–Stokes fluid exerts on Γ_0 is given by

$$F_D := \int_{\Gamma_0} -\mu\,\langle D_n u, a\rangle + p\langle n, a\rangle \, dS.$$

The gradient of F_D is then given by

$$dF_D(u,p,\Omega)[V] = \int_{\Gamma_0} \langle V, n\rangle \left[-\mu (D_n)^2 ua + \frac{\partial p}{\partial n}\langle a, n\rangle - \sum_{i=1}^{3} \mu \frac{\partial \lambda_i}{\partial n}\frac{\partial u_i}{\partial n} \right] dS$$

$$+ \int_{\Gamma_0} \langle V, n\rangle \left[\operatorname{div}_\Gamma \left(-\mu Dua + pa\right)\right] dS$$

with adjoint boundary condition $\lambda = -a$ on Γ_0.

Proof. Here, the function g is given by

$$g := -\mu\langle D_n u, a\rangle + p\langle n, a\rangle.$$

Furthermore,

$$\begin{aligned}
\langle \nabla g, n\rangle &= -\mu (D_n)^2 ua + \frac{\partial p}{\partial n}\langle n, a\rangle \\
\frac{\partial g}{\partial u_i} &= 0 \\
\nabla_n g &= -\mu Dua + pa \\
\frac{\partial g}{\partial p} &= \langle a, n\rangle \\
\frac{\partial g}{\partial b} &= -\mu a,
\end{aligned}$$

where $(D_n)^2 ua$ refers to the second normal derivative tensor of u, e.g.

$$(D_n)^2 ua = \sum_{i,j,k=1}^{3} n_i \frac{\partial^2 u_k}{\partial x_i \partial x_j} n_j a_k.$$

The structure of the gradient and the adjoint boundary conditions are a direct consequence of theorem 6.1.9. Note that for this specific function, the terms $\kappa g(u, D_n u, p, n)$ and $\kappa\langle \nabla_n g, n\rangle$ cancel each other. □

6.2 Example Application

6.2.1 Energy Dissipation

As an example application, the optimization of a channel filled with water in two dimensions is considered. The objective is to minimize the dissipation of kinetic energy into heat, as given by equation (4.12).

Definition 6.2.1 (Energy Dissipation Problem in a Navier–Stokes Fluid)**.** *In the absence of body forces, minimizing the viscous energy dissipation in a Navier–Stokes fluid results in the following problem*

$$\min_{(u,p,\Omega)} J(u,p,\Omega) := \int_{\Omega} \mu \sum_{i,j=1}^{3} \left(\frac{\partial u_i}{\partial x_j}\right)^2 dA$$

subject to

$$\begin{aligned} -\mu\Delta u + \rho u \nabla u + \nabla p &= 0 && \textit{in} \quad \Omega \\ \operatorname{div} u &= 0 \\ u &= u_+ && \textit{on} \quad \Gamma_+ \\ u &= 0 && \textit{on} \quad \Gamma_0 \\ pn - \mu\frac{\partial u}{\partial n} &= 0 && \textit{on} \quad \Gamma_- \\ \mathrm{Vol} &= V_0. \end{aligned}$$

The volume constraint Vol = V_0 *will be considered discretely.*

Lemma 6.2.2 (Gradient of the Navier–Stokes Energy Dissipation Problem)**.** *The shape gradient for the viscous energy dissipation problem in a Navier–Stokes fluid is given by*

$$dJ(u,p,\Omega) = \int_{\Gamma_0} \langle V, n\rangle \left[-\mu \sum_{i=1}^{2} \frac{\partial \lambda_i}{\partial n}\frac{\partial u_i}{\partial n} + \left(\frac{\partial u_i}{\partial n}\right)^2 \right] dS \tag{6.19}$$

subject to

$$\begin{aligned} -\mu\Delta u + \rho u \nabla u + \nabla p &= 0 && \textit{in} \quad \Omega \\ \operatorname{div} u &= 0 \\ u &= u_+ && \textit{on} \quad \Gamma_+ \\ u &= 0 && \textit{on} \quad \Gamma_0 \\ pn - \mu\frac{\partial u}{\partial n} &= 0 && \textit{on} \quad \Gamma_- \\ -\mu\Delta\lambda_i - \rho\sum_{j=1}^{d}\left(\frac{\partial \lambda_j}{\partial x_i}u_j + \frac{\partial \lambda_i}{\partial x_j}u_j\right) - \frac{\partial \lambda_p}{\partial x_i} &= -2\mu\Delta u_i && \textit{in} \quad \Omega \\ \operatorname{div} \lambda_p &= 0 \\ \lambda &= 0 && \textit{on} \quad \Gamma_+ \\ \lambda &= 0 && \textit{on} \quad \Gamma_0 \\ \mu\frac{\partial \lambda_i}{\partial n} + \rho\left(\sum_{j=1}^{2} \lambda_j u_j n_i + \lambda_i u_j n_j\right) + \lambda_p n_i &= 0 && \textit{on} \quad \Gamma_-. \end{aligned}$$

Proof. A comparison with equation (6.1) shows that

$$\begin{aligned} f(u, Du, p) &= \mu \sum_{i,j=1}^{2}\left(\frac{\partial u_i}{\partial x_j}\right)^2 \\ g(u, D_n u, p, n) &= 0, \end{aligned}$$

which results in

$$\begin{aligned}\frac{\partial f}{\partial a_{ij}} &= 2\mu a_{ij} = 2\mu \frac{\partial u_i}{\partial x_j} \\ \frac{\partial f}{\partial u_i} &= 0.\end{aligned}$$

According to lemma 6.1.9, the adjoint equation is given by

$$\begin{aligned}-\mu\Delta\lambda_i - \rho\sum_{j=1}^{d}\left(\frac{\partial\lambda_j}{\partial x_i}u_j + \frac{\partial\lambda_i}{\partial x_j}u_j\right) - \frac{\partial\lambda_p}{\partial x_i} &= \frac{\partial f}{\partial u_i} - \sum_{j=1}^{d}\frac{\partial}{\partial x_j}\frac{\partial f}{\partial a_{ij}} \\ &= -2\mu\Delta u_i \qquad\qquad \text{in} \quad \Omega \\ \operatorname{div}\lambda_p &= \frac{\partial f}{\partial p} = 0\end{aligned}$$

with boundary conditions

$$\begin{aligned}\lambda &= 0 \quad \text{on} \quad \Gamma_+ \\ \lambda_i = \frac{1}{\mu}\frac{\partial g}{\partial b_i} &= 0 \quad \text{on} \quad \Gamma_0 \\ \langle\lambda, n\rangle = -\frac{\partial g}{\partial p} &= 0 \quad \text{on} \quad \Gamma_0 \\ \mu\frac{\partial\lambda_i}{\partial n} + \rho\left(\sum_{j=1}^{2}\lambda_j u_j n_i + \lambda_i u_j n_j\right) + \lambda_p n_i &= 0 \quad \text{on} \quad \Gamma_-.\end{aligned}$$

Both conditions on Γ_0 are satisfied by $\lambda = 0$, and consequently the gradient is given by

$$\begin{aligned}dJ(u,p,\Omega)[V] &= \int_{\Gamma_0}\langle V, n\rangle \left[\mu\sum_{i,j=1}^{2}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right] dS \\ &+ \int_{\Gamma_0}\langle V, n\rangle\left[-\sum_{i=1}^{2}\left(\mu\frac{\partial\lambda_i}{\partial n} + \sum_{j=1}^{2}\frac{\partial f}{\partial a_{ij}}n_j\right)\frac{\partial u_i}{\partial n}\right] dS \\ &= \int_{\Gamma_0}\langle V, n\rangle\left[\mu\sum_{i=1}^{2}\left(\frac{\partial u_i}{\partial n}\right)^2\right] dS \\ &+ \int_{\Gamma_0}\langle V, n\rangle\left[-\sum_{i=1}^{2}\left(\mu\frac{\partial\lambda_i}{\partial n} + \sum_{j=1}^{2}2\mu\frac{\partial u_i}{\partial x_j}n_j\right)\frac{\partial u_i}{\partial n}\right] dS \\ &= \int_{\Gamma_0}\langle V, n\rangle\left[-\mu\sum_{i=1}^{2}\frac{\partial\lambda_i}{\partial n}\frac{\partial u_i}{\partial n} + \left(\frac{\partial u_i}{\partial n}\right)^2\right] dS.\end{aligned}$$

□

6.2.2 Flow Solver

The flow solver from section 5.5.1 is extended to the Navier–Stokes equations, which introduces two new difficulties: the non-linearity and the need for an adjoint solver. Often, the non-linearity is treated in a Picard or Uzawa based approach, meaning that basically the non-linear term is lagged one iteration behind, resulting in the need to subsequently solve Oseen problems. This is also the case in the SIMPLE iteration. Usually, such methods converge only linearly, but are also only weakly dependent or possibly even independent of the initial guess. However, the adjoint system is based on the "proper" linearization of the system. Since the adjoint solver requires knowledge of the proper Jacobian of the Navier–Stokes system anyway, the non-linearity can as well be solved using Newton's method. An exact Newton method converges quadratically but is quite sensitive to the initial guess. For high Reynolds number flows finding a suitable starting solution can be quite problematic. Newton's method requires computing updates according to

$$\begin{bmatrix} A(u^k, p^k) & B^*(u^k, p^k) \\ B(u^k, p^k) & 0 \end{bmatrix} \begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = -c(u^k, p^k), \quad \begin{pmatrix} u^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} u^k \\ p^k \end{pmatrix} + \begin{pmatrix} \delta u \\ \delta p \end{pmatrix},$$

where $A(u^k, p^k)$ now contains both the discretized diffusion operator as well as the linearized convection operator and thus is no longer symmetric. Here, c denotes the residual of the discrete Navier–Stokes equations. The big advantage of using a matrix based exact Newton method lies in the ability to very easily create a discrete adjoint solver by simply solving the system

$$\begin{bmatrix} A & B^* \\ B & 0 \end{bmatrix}^T \begin{pmatrix} \lambda \\ \lambda_p \end{pmatrix} = \text{rhs} \Leftrightarrow \begin{bmatrix} A^T & B^T \\ B^{*T} & 0 \end{bmatrix} \begin{pmatrix} \lambda \\ \lambda_p \end{pmatrix} = \text{rhs}, \tag{6.20}$$

where rhs is the partial derivative of the discretized objective function. The Navier–Stokes shape gradient in equation (6.19) basically shows how to combine state and adjoint variables to form the gradient. However, the derivation of the shape gradient is based on an all analytic setting. Since the boundaries are part of the discrete adjoint system (6.20), the block B^* has zero rows for nodes with prescribed velocities, whereas B does not. Hence, the analytic no-slip boundary condition $\lambda = 0$ on Γ_0 will not be precisely kept, creating a discrepancy between the analytic shape derivative and the discrete adjoint solver. It was not possible to evaluate the shape gradient in the needed precision using such a discrete adjoint solution. As such, the off-diagonal blocks in (6.20) where modified to precisely enforce the adjoint viscous wall boundary condition $\lambda = 0$. More information on discrete adjoint Navier–Stokes solvers can be found in [63].

6.2.3 Flow Through a Pipe

As a first test-case, the shape of a tube connecting two points is optimized. The geometry is shown in figure 6.1. The channel has a cross section of 1.0 and the viscosity is $\mu = \frac{1}{400}$. The inflow velocity profile is parabolic up to a maximum value of $u_1 = 1.5$. The fluid enters the channel on the bottom left side. With a constant density of $\rho = 1.0$, the average mass influx results in a Reynolds number of 400 using the channel width as reference length. The Reynolds number is close to the maximum such that the Newton iteration converges, with the flow probably becoming instationary for higher Reynolds numbers. Each of the sharp bends creates a strong stationary vortex in the flow with the streamlines shown in figure 6.2. As a consequence, the initial tube has a comparatively

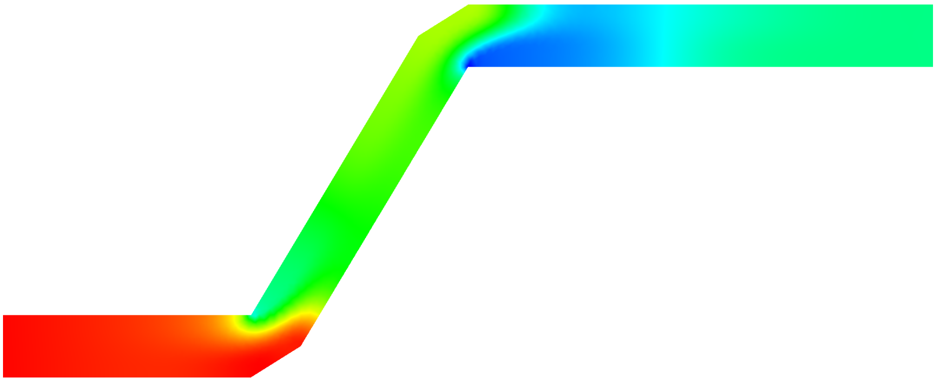


Figure 6.1: The initial tube. The middle area is the unknown to be found, the remainder outside is considered fixed. Color denotes pressure.

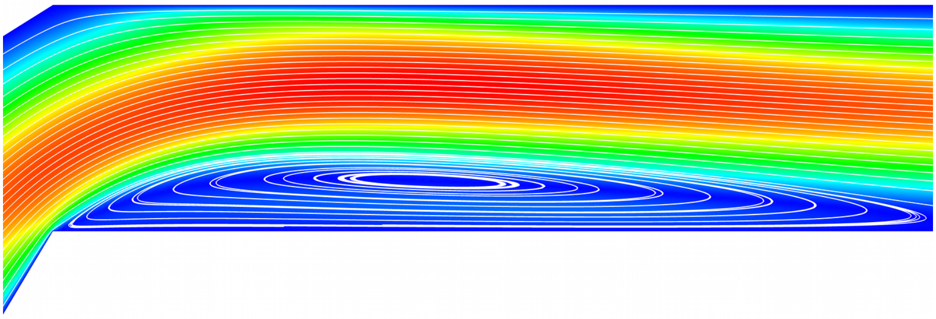


Figure 6.2: Magnification of the initial tube. Strong counter vortices develop after the bends. Color denotes speed.

strong pressure gradient, resulting in a distinct pressure loss of the flow, see figure 6.1. A straight forward steepest descent algorithm using a fixed step length and the shape derivative is used as an optimization procedure. The perturbation direction is chosen as $V = \delta \cdot n$, i.e. in each iteration each boundary node is shifted into the direction of the normal at that node. Thus, the boundary nodes follow a curved path during the optimization. In order to prevent a degeneration of the tube, the initial volume is preserved, which is enforced by a projection step after each shape update, just like in the Stokes case. The optimized tube is shown in figure 6.3. The initial channel has a net loss of kinetic energy into heat of $J = 0.9077$, which is reduced to $J = 0.4308$, a reduction by 52.54%.

6.2.4 Flow through a T-Connection

Before returning to smoothing and Hessian approximations, a T-junction is optimized. Inlet and outlet are positioned such that the flow enters on the bottom and must be redirected by 90°, leaving the pipe system on the left and right. The initial geometry is shown in figure 6.4 and the optimized in figure 6.5. A considerable amount of the channel after the actual fork is fixed, such that the

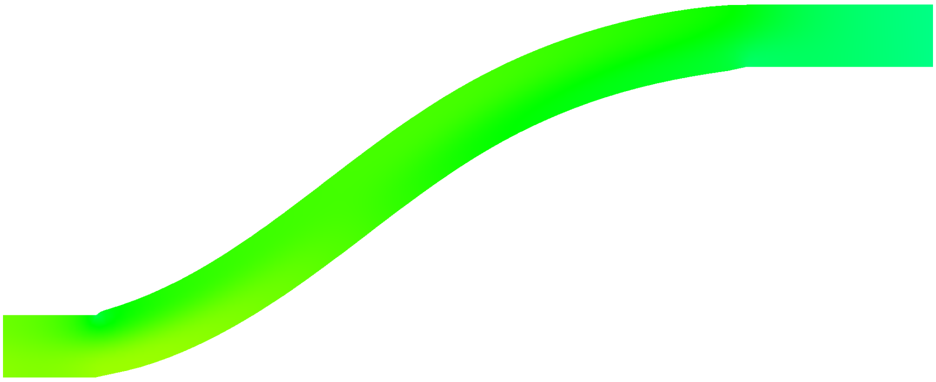


Figure 6.3: Pressure distribution optimized tube. The pressure loss is almost completely removed.

optimization cannot circumvent a net turning of 90°, and the flow has to exit parallel to the x-axis. The inflow profile is quadratic and the average mass influx results in a Reynolds number of $Re = 100$ with the channel cross-section as reference area. The initial junction has a net loss of kinetic energy into heat of $J = 0.9208$, which is reduced to $J = 0.6900$, a reduction by 25.05%. The total area occupied by the fluid was enforced to stay the same during the whole optimization.

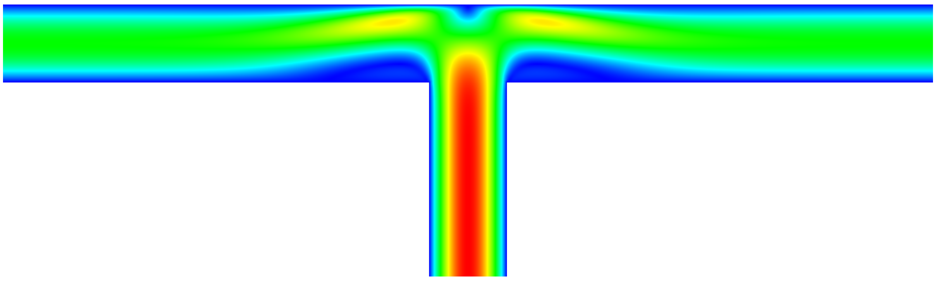


Figure 6.4: Initial T-connection. The fluid is entering on the bottom. Color denotes speed.

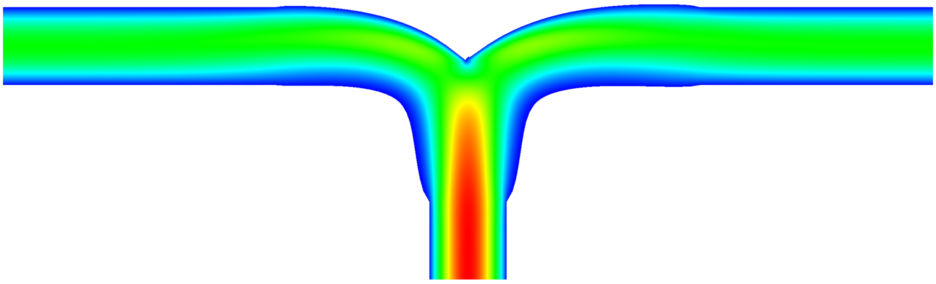


Figure 6.5: Optimized T-connection.

6.3 Hessian Approximation and Sobolev Optimization

Hessian approximation and smoothing procedures for Sobolev descent methods for aerodynamic shape optimization, i.e. an obstacle in a channel, are now considered. Unfortunately, the presence of adjoint equations and the non-linear convection operator make a similar analysis as in lemma 5.2.2 non-trivial. In two dimensions, the gradient from equation (6.19) is given by

$$\begin{aligned} G &= -\mu \sum_{k=1}^{2} \frac{\partial \lambda_k}{\partial x_2} \frac{\partial u_k}{\partial x_2} + \left(\frac{\partial u_k}{\partial x_2} \right)^2 \\ &= -\mu \sum_{k=1}^{2} \frac{\partial u_k}{\partial x_2} \left[\frac{\partial \lambda_k}{\partial x_2} + \frac{\partial u_k}{\partial x_2} \right]. \end{aligned}$$

The perturbed gradient for a flat domain $\Omega = \{(x_1, x_2) : x_1 \in \mathbb{R},\ x_2 \geq 0\}$ is given by

$$\tilde{G} = -\mu \sum_{k=1}^{2} \frac{\partial u_k'[\alpha]}{\partial x_2} \left(\frac{\partial \lambda_k}{\partial x_2} + \frac{\partial u_k}{\partial x_2} \right) + \frac{\partial u_k}{\partial x_2} \left(\frac{\partial \lambda_k'[\alpha]}{\partial x_2} + \frac{\partial u_k'[\alpha]}{\partial x_2} \right),$$

where $\alpha = e^{i\omega_1 x_1}$ is again an oscillation of the boundary with i being the imaginary unit. The local shape derivatives are assumed to have the following structure:

$$\begin{aligned} u_k'[\alpha] &= \hat{u}_k e^{i\omega_1 x_1} e^{\omega_2 x_2} \\ p'[\alpha] &= \hat{p} e^{i\omega_1 x_1} e^{\omega_2 x_2} \\ \lambda_k'[\alpha] &= \hat{\lambda}_k e^{i\omega_1 x_1} e^{\omega_2 x_2} \\ \lambda_p'[\alpha] &= \hat{\lambda}_p e^{i\omega_1 x_1} e^{\omega_2 x_2}. \end{aligned}$$

Applying the linearized Navier–Stokes equations to $u'[\alpha]$ and $p'[\alpha]$ alone results in a linear system

$$\begin{bmatrix} -\mu\left(-\omega_1^2 + \omega_2^2\right) + \rho\frac{\partial u_1}{\partial x_1} + \rho u_1 i\omega_1 + \rho u_2 \omega_2 & \rho\frac{\partial u_1}{\partial x_2} & i\omega_1 \\ \rho\frac{\partial u_2}{\partial x_1} & -\mu\left(-\omega_1^2 + \omega_2^2\right) + \rho\frac{\partial u_2}{\partial x_2} + \rho u_1 i\omega_1 + \rho u_2 \omega_2 & \omega_2 \\ i\omega_1 & \omega_2 & 0 \end{bmatrix},$$

of which the determinant must again vanish. The determinant of the above is given by

$$\begin{aligned} &\mu\omega_2^4 - \rho u_2 \omega_2^3 + \left(-\mu\omega_1^2 - \rho\frac{\partial u_1}{\partial x_1} - \rho u_1 i\omega_1 - \mu\omega_1^2 \right) \omega_2^2 \\ &+ \left(i\rho\frac{\partial u_2}{\partial x_1}\omega_1 + i\omega_1\rho\frac{\partial u_1}{\partial x_2} + \rho u_2 \omega_1^2 \right) \omega_2 + \mu\omega_1^4 + \rho\frac{\partial u_2}{\partial x_2}\omega_1^2 + \rho u_1 i\omega_1^3. \end{aligned}$$

For finding the operator symbol, the root in terms of ω_2 must now be found. However, for a polynomial of fourth order in complex coefficients, the roots are no longer elegantly given. Thus, the discrete approach is again used for the Navier–Stokes problem.

Flow around a circular obstacle in a channel is considered. Parts of the domain and the fluid are shown in figure 6.8. The Reynolds number is kept at 80, resulting in a steady state laminar flow. The circle is discretized using 1000 surface mesh nodes with variable positions, and a sin-wave

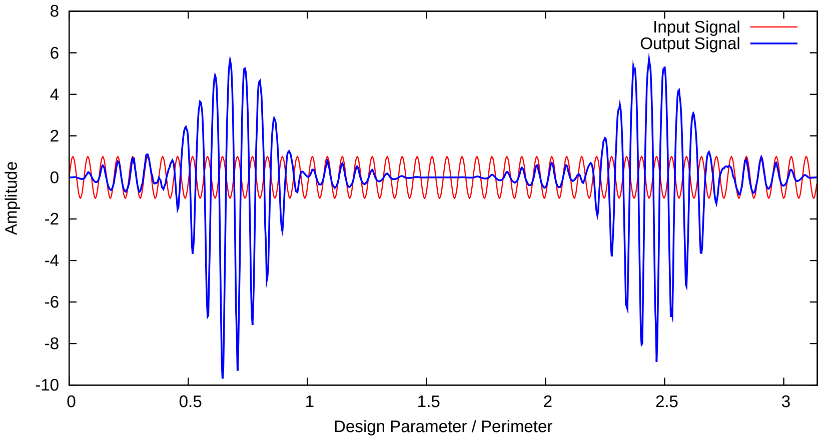


Figure 6.6: Incoming and outgoing wave for the Navier–Stokes problem.

of amplitude 0.002 and an angular frequency of $\omega = 50$ is modulated onto this circle. Due to the Taylor–Hood discretization, there are also 500 fixed nodes on the mid-sides. A comparison of the input and output signal is given in figure 6.6. One can again see that both input and output signal stay in phase, which again points to either a differential operator of even order, or a pseudo-differential operator very similar to the Stokes problem. Observed next is the scaling of the amplitude when the input frequency is halved to $\frac{1}{2}\omega = 25$. Similar to the Stokes problem, one can see the amplitude of the output signal scaling linearly with the frequency of the input signal. The corresponding waves are shown in figure 6.7. Thus, the discrete Hessian for the Navier–Stokes problem is also a pseudo-differential operator of symbol $|\omega|$ just as in the Stokes problem. Unfortunately, the discrete approach cannot reveal the dependence of the symbol of the Hessian and the Reynolds number. It is thus entirely possible that the behavior of the Hessian for the Navier–Stokes equations changes significantly with the occurrence of turbulence.

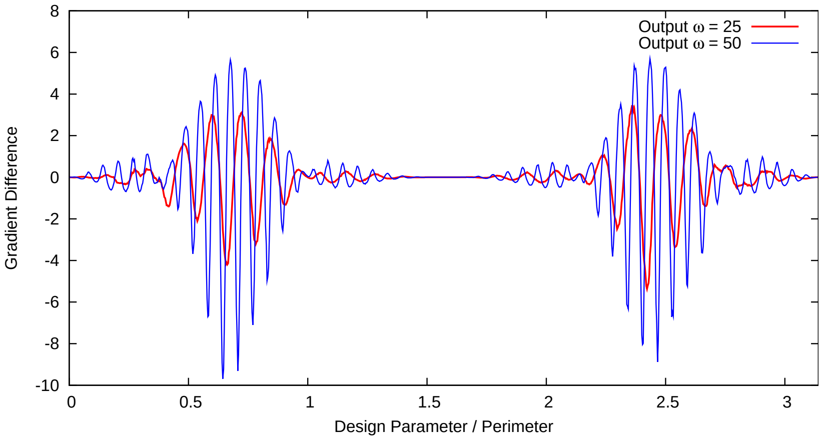


Figure 6.7: Amplitude of the Stokes problem response scales linearly with the input frequency.

The preconditioner is again based on the available information concering the Hessian. In the Stokes problem, the outer derivative of the gradient, equation (5.6) after dropping one dimension,

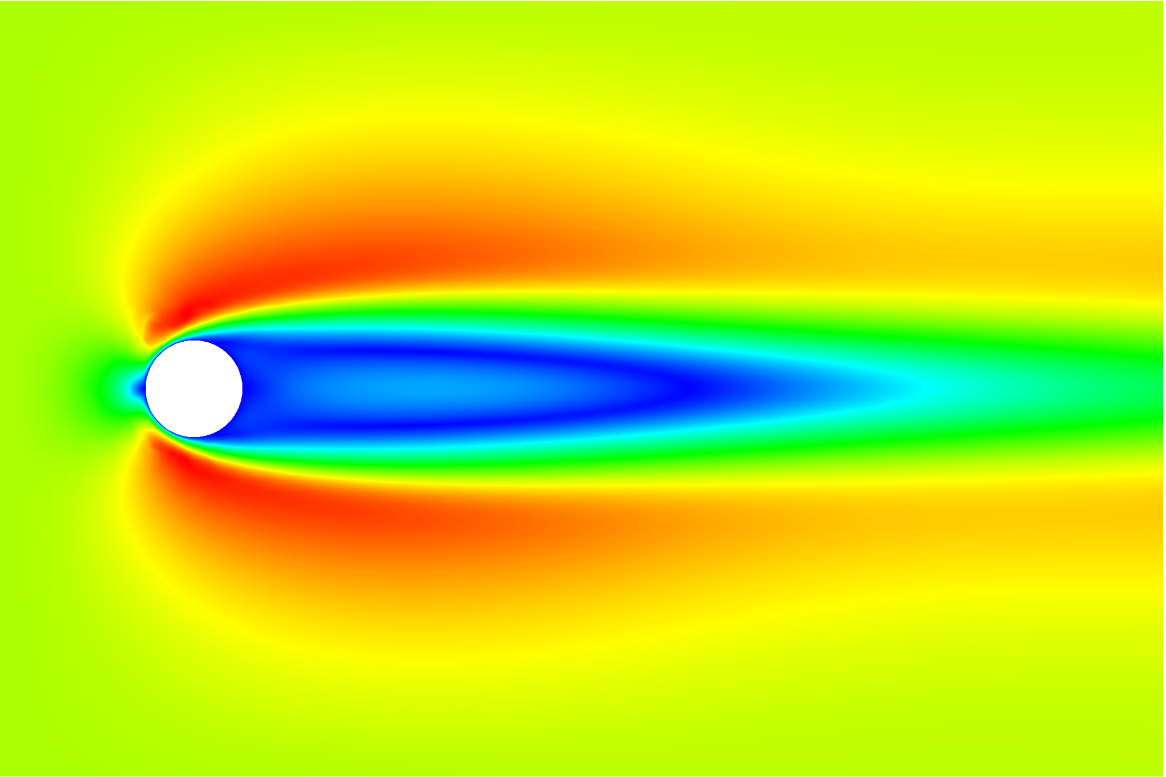


Figure 6.8: Initial shape of the Navier–Stokes problem. Color denotes speed.

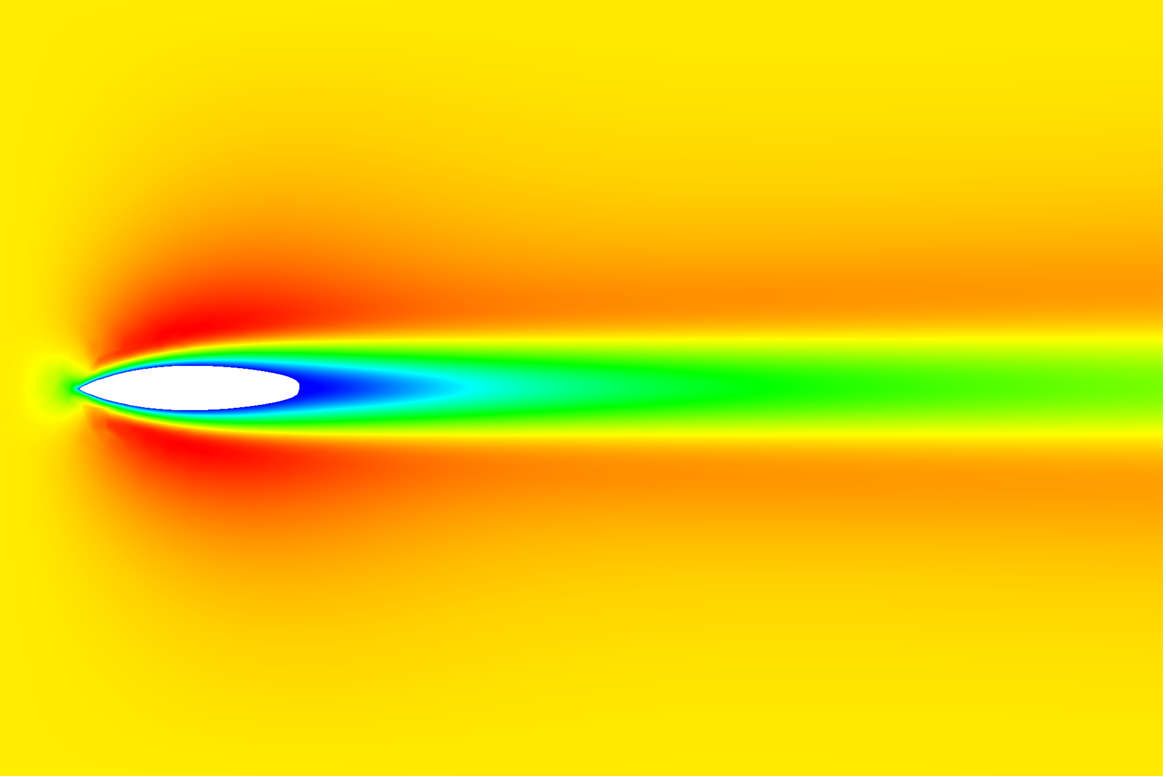


Figure 6.9: Optimal shape of the Navier–Stokes problem. Color denotes speed.

was used as the variable coefficient β in (5.9). In the Navier–Stokes case, this outer derivative is more complex, as it now involves the costate also. Therefore, the constant factor

$$k = \frac{\frac{3}{2}\omega_{\max} - 3}{\omega_{\max}^2} \tag{6.21}$$

is used in the Laplace–Beltrami preconditioner (5.7). In order to again be able to compare the speed-up with the unpreconditioned iteration, the number of variable surface mesh nodes is reduced from 1000 to 100. This results in a value of $\omega_{\max}$ between 25 and 50 which was kept fixed at 30. The initial and optimized shapes are shown in figure 6.8 and 6.9. The preconditioned optimization requires 71 steps till convergence using a steplength of $d = 0.06$, and the unpreconditioned optimization requires around 350 iterations using a steplength of $d = 0.005$, which is the longest steplength possible. The resulting optimal ogive shape is astonishing ship-like. Also, the preconditioned iteration requires only 20% of the unpreconditioned gradient steps. That is, for the Navier–Stokes problem, preconditioning reduces the computational effort by 80%. The precise comparison of the convergence history is plotted in figure 6.10. To prevent a degeneration of the shape, a volume constraint was again enforced by a discrete projection step in the direction of the shape derivative of the volume after each shape update.

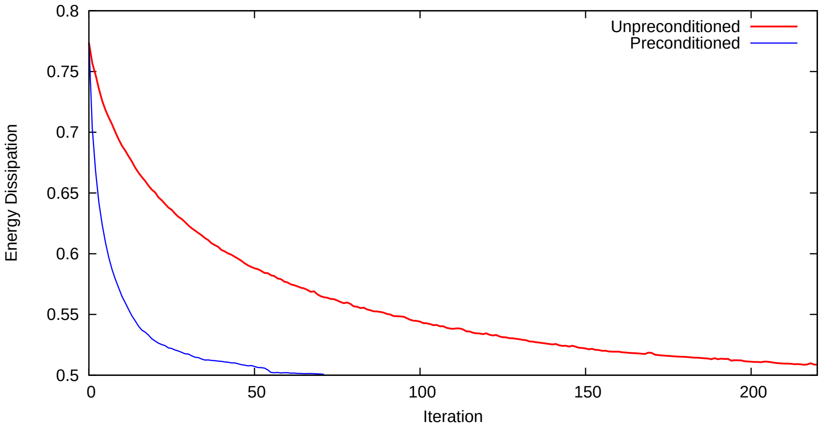
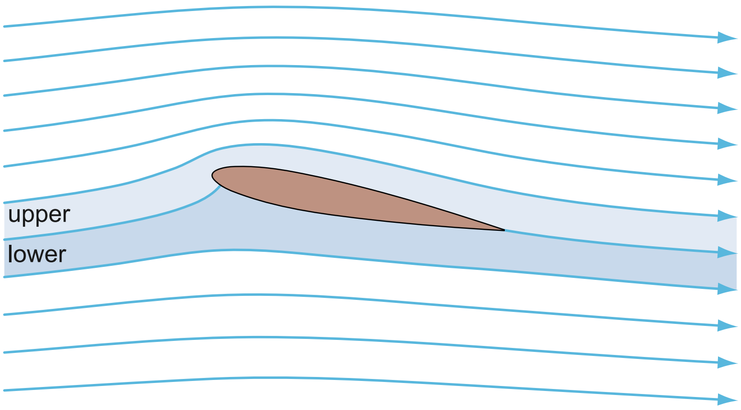


Figure 6.10: Convergence rates of the preconditioned Navier–Stokes problem.



Chapter 7

Potential Flow Pressure Tracking

7.1 Introduction

Potential flow pressure matching is a classical inverse design aerodynamic problem. Due to the simplifications of potential flow, remark 4.2.10, a meaningful drag value cannot be computed in potential flow. Thus, this flow model is almost always used in an inverse design setting, i.e. given a surface pressure distribution, a corresponding airfoil is to be found which produces this pressure field. Although potential flow is essentially the Laplace equations, the strong geometric dependencies of the pressure matching objective function make a shape differentiation surprisingly difficult, especially when compared to the self-adjoint Stokes energy dissipation problem. Part of this research has already been published in [19]. As mentioned in definition 4.2.11, the flow is modeled as the gradient of some potential ϕ. In order to reconstruct the pressure from the velocity $\nabla\phi$, Bernoulli's principle is used:

Remark 7.1.1 (Bernoulli's Principle)**.** Bernoulli's principle for incompressible, inviscid flows states that along a streamline

$$\frac{1}{2}\|u\|^2 + g + \frac{p}{\rho} = \text{const},$$

where g is the gravitational potential. A similar version exists for compressible inviscid flows.

Thus, velocity and pressure can be linked together, which creates the following shape optimization problem.

Definition 7.1.2 (Potential Flow Pressure Tracking)**.** *The potential flow pressure tracking problem is given by*

$$\min_{(\phi,\Omega)} J(\phi,\Omega) := \int_{\Gamma_0} \frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2 \, dS \tag{7.1}$$

subject to

$$\begin{aligned} -\Delta\phi &= 0 && \text{in } \Omega \\ \frac{\partial\phi}{\partial n} &= 0 && \text{on } \Gamma_0 \\ \phi &= \phi_0 && \text{on } \Gamma_+ \cup \Gamma_-, \end{aligned} \tag{7.2}$$

where ϕ_0 is used to create the necessary potential difference between inflow, outflow, and airfoil surface. The target surface pressure distribution is denoted by p_0. A two dimensional domain Ω is assumed, such that there is only one tangent vector τ.

Lemma 7.1.3 (Shape Derivative for Potential Flow Pressure Tracking)**.** *The shape derivative for potential flow pressure tracking in two dimensions is given by*

$$dJ(\phi,\Omega)[V] = \int_{\Gamma_0} \langle V,n\rangle \left[\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\left(\frac{\partial^2\phi}{\partial\tau\partial n} - \frac{\partial p_0}{\partial n}\right) + \kappa\frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2 + \frac{\partial\lambda}{\partial\tau}\frac{\partial\phi}{\partial\tau}\right] dS,$$

where λ solves the adjoint equation

$$\begin{aligned} -\Delta\lambda &= 0 && \text{in } \Omega \\ \frac{\partial\lambda}{\partial n} &= \frac{\partial}{\partial\tau}\left(\frac{\partial\phi}{\partial\tau} - p_0\right) && \text{on } \Gamma_0 \\ \lambda &= 0 && \text{on } \Gamma_+ \cup \Gamma_-. \end{aligned}$$

Proof. A formal differentiation according to chapter 3 results in

$$\begin{aligned} dJ(\phi,\Omega)[V] = &\int_{\Gamma_0} \langle V,n\rangle \left[\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\left(\frac{\partial^2\phi}{\partial\tau\partial n} - \frac{\partial p_0}{\partial n}\right) + \kappa\frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2\right] dS \\ &+ \int_{\Gamma_0} \left(\frac{\partial\phi}{\partial\tau} - p_0\right)\langle\nabla\phi'[V],\tau\rangle \, dS \\ &+ \int_{\Gamma_0} \left(\frac{\partial\phi}{\partial\tau} - p_0\right)\langle\nabla\phi,\tau'[V]\rangle \, dS, \end{aligned}$$

where according to remark 3.4.7, the local shape derivative $\phi'[V]$ is given by

$$\begin{aligned} -\Delta\phi'[V] &= 0 && \text{in } \Omega \\ \frac{\partial\phi'[V]}{\partial n} &= \operatorname{div}_\Gamma\left(\langle V,n\rangle\nabla_\Gamma\phi\right) && \text{on } \Gamma_0 \\ \phi'[V] &= 0 && \text{on } \Gamma_+ \cup \Gamma_-. \end{aligned}$$

According to lemma 3.3.8, the local shape derivative of the normal $dn[V]$ lies in the tangent plane. Thus, the local shape derivative of the tangent lies in the normal plane. Consequently,

$$\langle \nabla\phi, \tau'[V]\rangle = \langle \nabla\phi, \epsilon n\rangle = \epsilon\langle \nabla\phi, n\rangle = 0$$

due to the boundary conditions for ϕ on Γ_0. Therefore, the part of the gradient due to a variation of the tangent vanishes. Unfortunately, the objective function needs the derivative of $\phi'[V]$ in tangent direction, but the boundary conditions only provide information about the derivative in normal direction. Thus, an adjoint equation is needed. Multiplication with an arbitrary λ results in

$$\begin{aligned}
dJ(\phi,\Omega)[V] = &\int_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\left(\frac{\partial^2\phi}{\partial\tau\partial n} - \frac{\partial p_0}{\partial n}\right) + \kappa\frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2\right] dS \\
&+ \int_{\Gamma_0} \left(\frac{\partial\phi}{\partial\tau} - p_0\right)\frac{\partial\phi'[V]}{\partial\tau}\, dS \\
&+ \int_{\Omega} -\lambda\Delta\phi'[V]\, dA.
\end{aligned}$$

Integration by parts and Green's formula provide

$$\begin{aligned}
dJ(\phi,\Omega)[V] = &\int_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\left(\frac{\partial^2\phi}{\partial\tau\partial n} - \frac{\partial p_0}{\partial n}\right) + \kappa\frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2\right] dS \\
&- \int_{\Gamma_0} \frac{\partial}{\partial\tau}\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\phi'[V]\, dS \\
&+ \int_{\Gamma} -\lambda\frac{\partial\phi'[V]}{\partial n} + \phi'[V]\frac{\partial\lambda}{\partial n}\, dS + \int_{\Omega} -\phi'[V]\Delta\lambda\, dA \\
= &\int_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial\phi}{\partial\tau} - p_0\right)\left(\frac{\partial^2\phi}{\partial\tau\partial n} - \frac{\partial p_0}{\partial n}\right) + \kappa\frac{1}{2}\left(\langle\nabla\phi,\tau\rangle - p_0\right)^2\right] dS \\
&+ \int_{\Gamma_0} \phi'[V]\left[-\frac{\partial}{\partial\tau}\left(\frac{\partial\phi}{\partial\tau} - p_0\right) + \frac{\partial\lambda}{\partial n}\right] dS \\
&+ \int_{\Gamma} -\lambda\frac{\partial\phi'[V]}{\partial n}\, dS + \int_{\Omega} -\phi'[V]\Delta\lambda\, dA,
\end{aligned}$$

where $\Gamma = \Gamma_0 \cup \Gamma_+ \cup \Gamma_-$. Thus, if λ solves the adjoint equation

$$\begin{aligned}
-\Delta\lambda &= 0 && \text{in} \quad \Omega \\
\frac{\partial\lambda}{\partial n} &= \frac{\partial}{\partial\tau}\left(\frac{\partial\phi}{\partial\tau} - p_0\right) && \text{on} \quad \Gamma_0 \\
\lambda &= 0 && \text{on} \quad \Gamma_+ \cup \Gamma_-,
\end{aligned}$$

the shape derivative becomes

$$\begin{aligned} dJ(\phi, \Omega)[V] = &\int\limits_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial \phi}{\partial \tau} - p_0 \right) \left(\frac{\partial^2 \phi}{\partial \tau \partial n} - \frac{\partial p_0}{\partial n} \right) + \kappa \frac{1}{2} \left(\langle \nabla \phi, \tau \rangle - p_0 \right)^2 \right] \, dS \\ &+ \int\limits_{\Gamma_0} -\lambda \frac{\partial \phi'[V]}{\partial n} \, dS. \end{aligned}$$

The boundary condition for $\phi'[V]$ now provides

$$\begin{aligned} dJ(\phi, \Omega)[V] = &\int\limits_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial \phi}{\partial \tau} - p_0 \right) \left(\frac{\partial^2 \phi}{\partial \tau \partial n} - \frac{\partial p_0}{\partial n} \right) + \kappa \frac{1}{2} \left(\langle \nabla \phi, \tau \rangle - p_0 \right)^2 \right] \, dS \\ &+ \int\limits_{\Gamma_0} -\lambda \mathrm{div}_\Gamma \left(\langle V, n\rangle \nabla_\Gamma \phi \right) \, dS \\ = &\int\limits_{\Gamma_0} \langle V, n\rangle \left[\left(\frac{\partial \phi}{\partial \tau} - p_0 \right) \left(\frac{\partial^2 \phi}{\partial \tau \partial n} - \frac{\partial p_0}{\partial n} \right) + \kappa \frac{1}{2} \left(\langle \nabla \phi, \tau \rangle - p_0 \right)^2 \right] \, dS \\ &+ \int\limits_{\Gamma_0} \langle V, n\rangle \langle \nabla_\Gamma \lambda, \nabla_\Gamma \phi \rangle \, dS, \end{aligned}$$

due to integration by parts on Γ_0. The desired expression follows immediately due to the two dimensional domain Ω. □

7.2 Local Coordinates and Shape Hessian

In contrast to the Stokes case in section 5.2, a shape Hessian analysis will be conducted in local coordinates, i.e. a two dimensional star-shaped domain Ω is considered. This has the advantage that the derivation is more straight forward than using the shape differentiation techniques of chapter 3. For first order calculus, expressions for star-shaped domains can most of the time be interpreted on more general and three dimensional domains, too. Unfortunately, hopes that this would also be true for the Hessian did not come true.

Definition 7.2.1 (Star-shaped Domain)**.** *A domain Ω is said to be star-shaped if the boundary Γ is given by*

$$\Gamma := \{r(\varphi)e_r(\varphi) : \varphi \in [0, 2\pi]\},$$

where $e_r(\varphi) = (\sin\varphi, \cos\varphi)^T$. Thus, the boundary is a curve C with a parameterization c given by

$$c(\varphi) = r(\varphi)e_r(\varphi), \varphi \in [0, 2\pi].$$

A perturbed star-shaped domain is then given by

$$\Gamma_\epsilon := \{(r(\varphi) + \epsilon\tilde{r}(\varphi))\, e_r\}$$

with $r(0) = r(2\pi)$, which corresponds to a perturbation of identity using $V = \tilde{r}e_r$ and a curve parameterized by c_ϵ.

Remark 7.2.2 (Perturbed Line Integral)**.** For a curve C, the line integral of f over C is given by

$$\int\limits_C f(s)\ ds = \int\limits_a^b f(c(\varphi))\|c'(\varphi)\|\ d\varphi,$$

where $c : [a, b] \longrightarrow C$ is an arbitrary bijective parameterization of C. Thus, for shape optimization on star-shaped domains, the expression

$$\frac{d}{d\epsilon}\bigg|_{\epsilon=0} \int\limits_0^{2\pi} f_\epsilon(c_\epsilon(\varphi))\|c'_\epsilon(\varphi)\|\ d\varphi$$

must be computed, which is possible using standard differentiation techniques. For a star-shaped domain, a direct computation reveals

$$\|c'_\epsilon\| = \left(r'^2 + r^2 + 2\epsilon\left(r\tilde{r} + r'\tilde{r}'\right) + \epsilon^2\tilde{r}'^2 + \epsilon^2\tilde{r}^2\right)^{\frac{1}{2}},$$

where r' denotes a differentiation with respect to arc length, e.g.

$$r' = \frac{\partial r}{\partial\varphi}.$$

Switching differentiation and integration, the expression

$$\frac{\partial}{\partial\epsilon}\bigg|_{\epsilon=0}\|c'_\epsilon\| = \frac{r\tilde{r} + r'\tilde{r}'}{\sqrt{r'^2 + r^2}}$$

will be needed.

Remark 7.2.3 (Geometric Quantities on Perturbed Star-Shaped Domains)**.** A straight forward computation reveals for the normal n_ϵ and the tangent τ_ϵ on a perturbed star-shaped domain c_ϵ:

$$\begin{aligned} n_\epsilon &= \frac{re_r - r'e'_r + \epsilon\left(\tilde{r}e_r - \tilde{r}'e'_r\right)}{\sqrt{r'^2 + r^2 + 2\epsilon\left(r\tilde{r} + r'\tilde{r}'\right) + \epsilon^2\tilde{r}'^2 + \epsilon^2\tilde{r}^2}} \\ \tau_\epsilon &= \frac{r'e_r + re'_r + \epsilon\left(\tilde{r}'e_r + \tilde{r}e'_r\right)}{\sqrt{r'^2 + r^2 + 2\epsilon\left(r\tilde{r} + r'\tilde{r}'\right) + \epsilon^2\tilde{r}'^2 + \epsilon^2\tilde{r}^2}}. \end{aligned}$$

Note that depending on the definition of what is outside and inside, which direction the tangent vector is pointing, and due to the periodicity of e_r, the signs may differ. The appropriate quantities on an unperturbed star-shaped domain are easily attainable by setting $\epsilon = 0$. Consequently, on a star-shaped domain, one also has

$$e_r = \frac{r}{\sqrt{r^2 + r'^2}}\,n + \frac{r'}{\sqrt{r^2 + r'^2}}\,\tau.$$

The curvature of a curve parameterized by c is given by

$$\kappa = \frac{\langle c'', c'^\perp\rangle}{\|c'\|^3},$$

which results in

$$\kappa = \frac{r''r - 2r'^2 - r^2}{\left(\sqrt{r^2 + r'^2}\right)^3}$$

for a star-shaped domain. Finally, a straight forward computation also reveals

$$\langle V, n\rangle = \left\langle \tilde{r} e_r, \frac{r e_r - r' e_r'}{\sqrt{r^2 + r'^2}} \right\rangle = \frac{\tilde{r} r}{\sqrt{r^2 + r'^2}} = \frac{1}{\|c'\|} \tilde{r} r,$$

since $e_r^T e_r' = 0$.

Remark 7.2.4 (Potential Flow Shape Gradient on Star-Shaped Domains)**.** Using the expressions from remark 7.2.3, it is possible to express the shape derivative from lemma 7.1.3 for a star-shaped domain by

$$\begin{aligned} dJ(\phi, \Omega)[\tilde{r}] = &- \int_0^{2\pi} \tilde{r} r \langle \nabla\lambda, \nabla\phi\rangle \, d\varphi \\ &+ \int_0^{2\pi} \left(\frac{\partial\phi}{\partial\tau} - p_0 \right) \left[\tilde{r} e_r^T D^2 u \, \tau - \langle \nabla p_0, \tilde{r} e_r\rangle \right] \left(r^2 + r'^2 \right)^{\frac{1}{2}} \, d\varphi \\ &+ \int_0^{2\pi} \frac{1}{2} \left(\frac{\partial\phi}{\partial\tau} - p_0 \right)^2 \frac{r\tilde{r} + r'\tilde{r}'}{\sqrt{r^2 + r'^2}} \, d\varphi. \end{aligned}$$

Lemma 7.2.5 (Shape Hessian for Potential Flow Pressure Tracking on Star-Shaped Domains)**.** *A preliminary expression for the shape Hessian for potential flow pressure tracking on a star-shaped domain is given by*

$$\begin{aligned} & d^2 J(\phi, \Omega)[\tilde{r}_1][\tilde{r}_2] \\ &= - \int_0^{2\pi} \tilde{r}_1 \tilde{r}_2 \langle \nabla\lambda, \nabla\phi\rangle \, d\varphi \\ &\quad - \int_0^{2\pi} \tilde{r}_1 r \left[\langle \nabla\lambda'[\tilde{r}_2], \nabla\phi\rangle + \tilde{r}_2 e_r^T D^2 \lambda \nabla\phi + \langle \nabla\lambda, \nabla\phi'[\tilde{r}_2]\rangle + \tilde{r}_2 e_r^T D^2 \phi \nabla\lambda \right] \, d\varphi \\ &\quad + \int_0^{2\pi} \left(\langle \nabla\phi'[\tilde{r}_2], \tau\rangle + \tilde{r}_2 e_r^T D^2 \phi \, \tau - \langle \nabla p_0, \tilde{r}_2 e_r\rangle \right) \left(\tilde{r}_1 e_r^T D^2 \phi \, \tau - \langle \nabla p_0, \tilde{r}_1 e_r\rangle \right) \sqrt{r^2 + r'^2} \\ &\qquad\qquad + \left(\frac{\partial\phi}{\partial\tau} - p_0 \right) \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \left(\left[\tilde{r}_1 e_r^T D^2 \phi_\epsilon \, \tau_\epsilon - \langle \nabla p_{0,\epsilon}, \tilde{r}_1 e_r\rangle \right] \sqrt{r_\epsilon^2 + r_\epsilon'^2} \right) \, d\varphi \end{aligned}$$

$$+ \int_0^{2\pi} \left(\frac{\partial \phi}{\partial \tau} - p_0 \right) \left(\langle \nabla \phi'[\tilde{r}_2], \tau \rangle + \tilde{r}_2 e_r^T D^2 \phi \, \tau - \langle \nabla p_0, \tilde{r}_2 e_r \rangle \right) \frac{r \tilde{r}_1 + r' \tilde{r}_1'}{\sqrt{r^2 + r'^2}}$$

$$+ \frac{1}{2} \left(\frac{\partial \phi}{\partial \tau} - p_0 \right)^2 \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \left(\frac{r_\epsilon \tilde{r}_1 + r_\epsilon' \tilde{r}_1'}{\sqrt{r_\epsilon^2 + r_\epsilon'^2}} \right) \, d\varphi.$$

Note that the terms which still need to be differentiated will vanish close to the optimum, because

$$\frac{\partial \phi}{\partial \tau} - p_0 \approx 0$$

if p_0 is reachable.

Proof. Let a second perturbation of the star-shaped domain be given by

$$r_\epsilon := r + \epsilon \tilde{r}_2.$$

The shape Hessian is then the limit

$$\begin{aligned} d^2 J(\phi, \Omega)[\tilde{r}_1][\tilde{r}_2] = \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \Bigg[& - \int_0^{2\pi} \tilde{r}_1 r_\epsilon \langle \nabla \lambda_\epsilon, \nabla \phi_\epsilon \rangle \, d\varphi \\ & + \int_0^{2\pi} \left(\frac{\partial \phi_\epsilon}{\partial \tau_\epsilon} - p_{0,\epsilon} \right) \left[\tilde{r}_1 e_r^T D^2 u_\epsilon \, \tau_\epsilon - \langle \nabla p_{0,\epsilon}, \tilde{r}_1 e_r \rangle \right] \left(r_\epsilon^2 + r_{\epsilon^2}' \right)^{\frac{1}{2}} \, d\varphi \\ & + \int_0^{2\pi} \frac{1}{2} \left(\frac{\partial \phi_\epsilon}{\partial \tau_\epsilon} - p_{0,\epsilon} \right)^2 \frac{r_\epsilon \tilde{r}_1 + r_\epsilon{}' \tilde{r}_1'}{\sqrt{r_\epsilon^2 + r_\epsilon^{2\prime}}} \, d\varphi \Bigg]. \end{aligned}$$

According to definition 3.4.1, the local shape derivatives are given by

$$\frac{d}{d\epsilon}\bigg|_{\epsilon=0} \langle \nabla \lambda_\epsilon, \nabla \phi_\epsilon \rangle = \langle \nabla \lambda'[\tilde{r}_2], \nabla \phi \rangle + \tilde{r}_2 e_r^T D^2 \lambda \nabla \phi + \langle \nabla \lambda, \nabla \phi'[\tilde{r}_2] \rangle + \tilde{r}_2 e_r^T D^2 \phi \nabla \lambda.$$

A straight forward standard differentiation results in the desired expression. □

The preliminary expression for the shape Hessian of the potential flow pressure matching problem, lemma (7.2.5), is astonishing complex for a problem based on the Laplacian. This complexity arises due to the objective function being a surface functional with strong dependence on geometric quantities such as the tangent τ. Also, the desired tangent derivative results in the need for an adjoint variable, which must be considered when deriving the Hessian. For this reason, the Stokes shape Hessian was considered first in chapter 5. Although considering a star-shaped domain allows a more straight forward differentiation, the resulting expression is of a complexity that is hardly applicable.

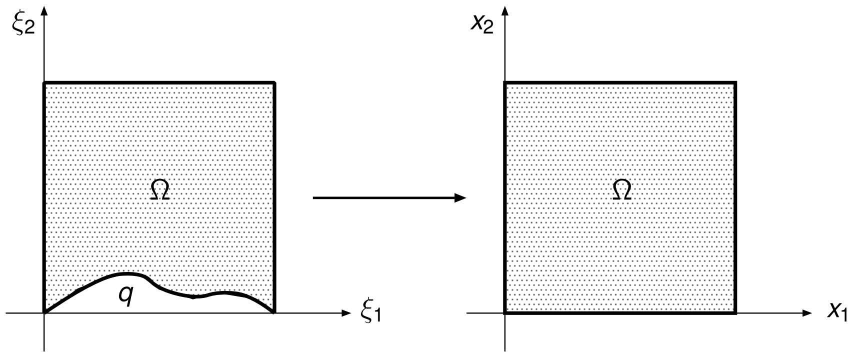


Figure 7.1: Method of Mapping

7.3 Method of Mapping and Fourier Analysis

In order to further study the Hessian, a slightly simplified problem is considered. The left part of figure 7.1 is imagined as the physical domain, which is thought of as being a part of the free air stream along a geometric detail of an airplane which is aligned with the ξ_1-axis. The shape of the boundary is supposed to be described by the function $q(\xi_1)$. In order to simplify the problem, the physical domain is mapped onto the computational domain on the right hand side of figure 7.1 by appropriately stretching it in the vertical direction, such that one can write

$$(x_1, x_2) = \left(\xi_1, \frac{\xi_2 - q(\xi_1)}{1 - q(\xi_1)}\right),$$

and the potential ϕ from definition 7.1.2 is thought of as $\phi(x(\xi))$. That means the Neumann condition in the physical plane at the shape boundary $\{(\xi_1, \xi_2) \mid \xi_2 = q(\xi_1)\}$

$$0 = \frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial \xi_1}\frac{\partial q}{\partial \xi_1} - \frac{\partial \phi}{\partial \xi_2}$$

is mapped to the boundary condition

$$0 = \frac{\partial q}{\partial \xi_1}\frac{\partial \phi}{\partial \xi_1} - \frac{1 + \left(\frac{\partial q}{\partial \xi_1}\right)^2}{1 - q^2}\frac{\partial \phi}{\partial x_2}. \tag{7.3}$$

For the purpose of simplification, the shape q is assumed to be almost a straight line, such that squared expressions like q^2 and $\left(\frac{\partial q}{\partial \xi_1}\right)^2$ can be neglected. Furthermore, the tangential velocity $\frac{\partial \phi}{\partial x_1}$ can be assumed to be constant, e.g. 1. This results in the approximative boundary condition

$$\frac{\partial \phi}{\partial x_2} = \frac{\partial q}{\partial x_1}. \tag{7.4}$$

The complete problem formulation in the computational domain is then

Definition 7.3.1 (Potential Flow Pressure Tracking with Method of Mapping)**.** *The potential flow inverse design problem using the method of mapping is given by*

$$\min_{(\phi,q)} J(\phi(q),q) := \int\limits_{\Gamma_0} \left(\frac{\partial \phi(x_1,x_2)}{\partial x_1} - p_0(x_1) \right)^2 dx_1 \tag{7.5}$$

subject to

$$\begin{aligned} -\Delta\phi &= 0 && \textit{in} \quad \Omega \\ \frac{\partial \phi}{\partial n} &= \frac{\partial q}{\partial x_1} && \textit{on} \quad \Gamma_0 \\ \phi &= \phi_0 && \textit{on} \quad \Gamma_+ \cup \Gamma_- . \end{aligned} \tag{7.6}$$

Strictly speaking, the above problem in definition 7.3.1 is no longer a shape optimization problem, and a detailed analysis of such problems can for example be found in [75]. However, it is analytically much more accessible and still closely related to the original shape optimization problem, lemma 7.1.3. For example, the resulting adjoint equation is almost the same as the adjoint equation for the proper shape optimization problem.

Lemma 7.3.2 (Derivative of Potential Flow Pressure Tracking with Method of Mapping)**.** *Choosing $\phi_0 = x_1$, the derivative of the potential flow pressure tracking using the method of mapping is given by*

$$\nabla_q J(\phi(q),q) = \frac{\partial \lambda(\phi(q))}{\partial x_1},$$

where λ solves the adjoint equation

$$\begin{aligned} -\Delta\lambda &= 0 && \textit{in} \quad \Omega \\ \frac{\partial \lambda}{\partial n} &= \frac{\partial}{\partial x_1}\left(\frac{\partial \phi}{\partial x_1} - p_0 \right) && \textit{on} \quad \Gamma_0 \\ \lambda &= 0 && \textit{on} \quad \Gamma_+ \cup \Gamma_- . \end{aligned}$$

Proof. The Lagrangian is given by

$$\begin{aligned} &\mathcal{L}(\phi,q,\lambda,\lambda_1,\lambda_2) \\ =&J(\phi,q) + \int\limits_{\Omega} \lambda(-\Delta\phi)\, dA + \int\limits_{\Gamma_0} \lambda_1 \left(\frac{\partial \phi}{\partial n} - \frac{\partial q}{\partial x_1} \right) dS + \int\limits_{\Gamma_+ \cup \Gamma_-} \lambda_2(x_1)(\phi(x_1) - x_1)\, dx_1 . \end{aligned}$$

Inserting the objective function and using Green's second identity provides

$$\begin{aligned} &\mathcal{L}(\phi,q,\lambda,\lambda_1,\lambda_2) \\ =&\int\limits_{\Gamma_0} \left(\frac{\partial \phi(x_1,x_2)}{\partial x_1} - p_0(x_1) \right)^2 dx_1 + \int\limits_{\Omega} (-\Delta\lambda)\phi\, dA \\ &+ \int\limits_{\partial\Omega} -\lambda \frac{\partial \phi}{\partial n} + \phi \frac{\partial \lambda}{\partial n}\, dS + \int\limits_{\Gamma_0} \lambda_1 \left(\frac{\partial \phi}{\partial n} - \frac{\partial q}{\partial x_1} \right) dS + \int\limits_{\Gamma_+ \cup \Gamma_-} \lambda_2(x_1)\left(\phi(x_1) - x_1\right)\, dx_1 . \end{aligned}$$

The adjoint boundary value problem is derived from the expression for perturbations $\tilde{\phi}$

$$\begin{aligned}0 =& \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathcal{L}(\phi + t \cdot \tilde{\phi}, q, \lambda, \lambda_1, \lambda_2)\\ =& \int\limits_{\Gamma_0} \left(\frac{\partial \phi}{\partial x_1} - p_0(x_1)\right) \frac{\partial \tilde{\phi}}{\partial x_1}\, dx_1 + \int\limits_{\Omega} (-\Delta\lambda)\tilde{\phi}\, dA\\ &+ \int\limits_{\partial\Omega} -\lambda \frac{\partial \tilde{\phi}}{\partial n} + \tilde{\phi}\frac{\partial \lambda}{\partial n}\, dS + \int\limits_{\Gamma_0} \lambda_1 \frac{\partial \tilde{\phi}}{\partial n}\, dS + \int\limits_{\Gamma_+ \cup \Gamma_-} \lambda_2 \cdot \tilde{\phi}\, dS.\end{aligned}$$

Integration by parts and re-grouping provides

$$\begin{aligned}0 =& \left[\left(\frac{\partial \phi}{\partial x_1} - p_0(x_1)\right)\tilde{\phi}\right]_0^1 + \int\limits_{\Gamma_0} \tilde{\phi}\left[-\frac{\partial}{\partial x_1}\left(\frac{\partial \phi}{\partial x_1} - p_0(x_1)\right) + \frac{\partial \lambda}{\partial n}\right] + (\lambda_1 - \lambda)\frac{\partial \tilde{\phi}}{\partial n}\, dx_1\\ &+ \int\limits_{\Omega} (-\Delta\lambda)\tilde{\phi}\, dx + \int\limits_{\Gamma_+ \cup \Gamma_-} -\lambda\frac{\partial \tilde{\phi}}{\partial n} + \tilde{\phi}\frac{\partial \lambda}{\partial n} + \lambda_2 \cdot \tilde{\phi}\, dS.\end{aligned}$$

On the boundary $\Gamma_+ \cup \Gamma_-$, not affected by the design q, the velocity potential ϕ is subject to a Dirichlet boundary condition. Thus, the state ϕ is fixed, and consequently $\tilde{\phi} = 0$ on $\Gamma_+ \cup \Gamma_-$, which simplifies the last integral in the equation above. Focussing on different perturbations $\tilde{\phi}$ results in the adjoint equations:

- A perturbation $\tilde{\phi}$ in Ω results in

$$-\Delta\lambda = 0 \quad \text{in} \quad \Omega.$$

- Perturbations with fixed Dirichlet value 0 and variable Neumann values give

$$\begin{aligned}\lambda &= \lambda_1 && \text{on} \quad \Gamma_0\\ -\lambda &= 0 && \text{on} \quad \Gamma_+ \cup \Gamma_-.\end{aligned}$$

- A perturbation in all respects with the exception of the corner points gives

$$\frac{\partial \lambda}{\partial n} = \frac{\partial}{\partial x_1}\left(\frac{\partial \phi}{\partial x_1} - p_0\right) \quad \text{on} \quad \Gamma_0.$$

Now, the adjoint boundary value problem reads completely

$$\begin{aligned}-\Delta\lambda &= 0 && \text{in} \quad \Omega\\ \frac{\partial \lambda}{\partial n} &= \frac{\partial}{\partial x_1}\left(\frac{\partial \phi}{\partial x_1} - p_0\right) && \text{on} \quad \Gamma_0\\ \lambda &= 0 && \text{on} \quad \Gamma_+ \cup \Gamma_-,\end{aligned}$$

which is exactly the same adjoint boundary condition as in the proper shape optimization problem, lemma 7.1.3.

For the design equation, e.g. the gradient, the Lagrangian is also perturbed in direction q, which means

$$\begin{aligned} 0 &= \left.\frac{d}{dt}\right|_{t=0} \mathcal{L}(\phi, q + t \cdot \tilde{q}, \lambda, \lambda_1, \lambda_2) = \int\limits_{\Gamma_1} -\lambda_1 \frac{\partial \tilde{q}}{\partial x_1}\, dS \\ &= -\left[\lambda_1 \tilde{q}\right]_0^1 + \int\limits_{\Gamma_1} \frac{\partial \lambda_1}{\partial x_1} \tilde{q}\, dx_1 = \int\limits_{\Gamma_1} \frac{\partial \lambda_1}{\partial x_1} \tilde{q}\, dx_1. \end{aligned}$$

Thus, the gradient on Γ_0 is given by

$$\nabla_q J(\phi(q), q) = \frac{\partial \lambda_1(\phi(q))}{\partial x_1} = \frac{\partial \lambda(\phi(q))}{\partial x_1}.$$

□

It is now possible to study the pseudo-differential operator nature of the Hessian similar to section 5.4.

Lemma 7.3.3 (Hessian Symbol of Potential Flow Pressure Tracking using Method of Mapping)**.** *The Hessian for potential flow pressure tracking using method of mapping is a second order differential operator with symbol*

$$\frac{\partial}{\partial q} \nabla_q J(\phi(q), q) = \frac{\partial}{\partial q} \left.\frac{\partial \lambda}{\partial x_1}\right|_{\Gamma_0} (x_1) = \pm \omega_q^2.$$

Proof. An arbitrary Fourier mode for q is assumed, i.e.

$$q = e^{i\omega_q x_1} \Rightarrow \frac{\partial q}{\partial x_1} = i\omega_q e^{i\omega_q x_1}.$$

Also, the solution of the forward problem is assumed to be of the form

$$\phi(x_1, x_2) = r e^{i\omega_1 x_1 + i\omega_2 x_2},$$

resulting in

$$\begin{aligned} \left.\frac{\partial \phi}{\partial n}\right|_{\Gamma_0} = \left.\frac{\partial \phi}{\partial x_2}\right|_{\Gamma_0} &= i\omega_2 r e^{i\omega_1 x_1} \\ \left.\frac{\partial^2 \phi}{\partial x_1^2}\right|_{\Gamma_0} &= -\omega_1^2 r e^{i\omega_1 x_1}. \end{aligned}$$

Using the boundary condition on Γ_0, design q and state ϕ can be linked together:

$$i\omega_q e^{i\omega_q x_1} = i\omega_2 r e^{i\omega_1 x_1},\ \forall x_1 \in \mathbb{R}.$$

For $x_1 = 0$ this means, in particular, $\omega_2 = \omega_q / r$ and thus also $\omega_1 = \omega_q$. The differential equation $-\Delta\phi = 0$ in Ω gives

$$0 = (\omega_1^2 + \omega_2^2) r e^{i\omega_1 x_1 + i\omega_2 x_2} \Leftrightarrow \frac{\omega_q^2}{r^2} + \omega_q^2 = 0,$$

which means $r = \pm i$. The adjoint solution is assumed to be of similar form

$$\lambda(x_1, x_2) = se^{i\theta_1 x_1 + i\theta_2 x_2} \Rightarrow \left.\frac{\partial\lambda}{\partial n}\right|_{\Gamma_0} = \left.\frac{\partial\lambda}{\partial x_2}\right|_{\Gamma_0} = i\theta_2 se^{i\theta_1 x_1},$$

and the Neumann boundary condition of the adjoint problem provides

$$i\theta_2 se^{i\theta_1 x_1} = \omega_1^2 re^{i\omega_1 x_1}, \ \forall x_1 \in \mathbb{R}.$$

With the same arguments as above, one also has

$$\begin{aligned} \theta_2 &= \frac{\omega_1^2 r}{is} = \pm\frac{\omega_q^2}{s} \\ \theta_1 &= \omega_1 = \omega_q. \end{aligned}$$

The adjoint differential equation in Ω results in

$$0 = \theta_1^2 + \theta_2^2 \Leftrightarrow \frac{\omega_q^4}{s^2} + \omega_q^2 = 0 \Leftrightarrow s = \pm i\omega_q.$$

Thus,

$$\left.\frac{\partial\lambda}{\partial x_1}\right|_{\Gamma_0}(x_1) = \pm\omega_q^2 e^{i\theta_1 x_1} = \pm\omega_q^2 q,$$

the symbol of a second order differential operator. □

Contrary to the Stokes case as described in lemma 5.4.2, where the Hessian was a pseudo-differential operator of order +1, the Hessian here is a differential operator of second order, and the Laplace–Beltrami preconditioner, equation (5.7), should provide a very good Hessian approximation.

7.4 Numerical Results

7.4.1 Panel Solver

For the optimizations the state equation is discretized by an aerodynamic panel solver, which is similar to a boundary element method. Using Green's second identity, the Laplace equation for the volume can be transformed into an integral equation on the boundary of the wing. Similar to finite elements, the potential ϕ is discretized as a finite linear combination of ansatz functions, resulting in a small, dense, and linear system to be solved. Any additional boundary conditions for the free stream velocity on $\Gamma_+ \cup \Gamma_-$ are embedded in the ansatz functions. To further ensure a physically meaningful solution and to allow a proper prediction of lift, the trailing edge stagnation point is prescribed by a slightly modified boundary condition, thus introducing lift via circulation similar to the Kutta–Joukowski theorem. For further information about the solver, the aerodynamics, and on how to ensure a physical solution that matches experimental measurements, see [43]. To avoid any discrepancies with the analytical adjoint equation as derived above, the adjoint of the state equation was taken as the transpose of the discrete linear state equation, i.e. adjoint in $\mathbb{R}^n$ with the standard scalar product.

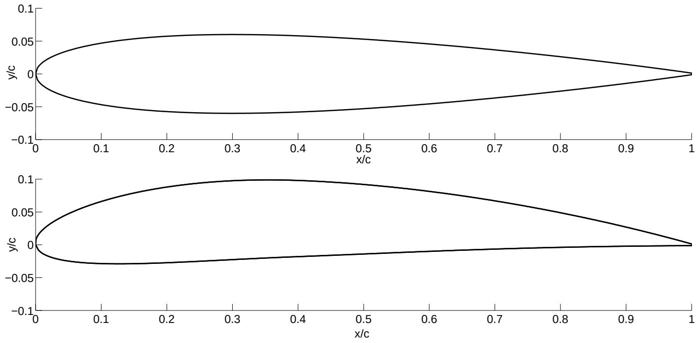


Figure 7.2: Initial NACA0012 airfoil and target NACA4412 airfoil

7.4.2 Numerical Pressure Fitting

The aim is to match pressure distributions generated by different NACA 4 digit airfoils. The initial shape is the symmetrical NACA0012 airfoil, and the pressure distribution generated by the cambered NACA4412 profile is to be matched. Figure 7.2 illustrates this. A detailed description of the NACA shapes can be found in [39]. Although local coordinates like star-shaped domains and the method of mapping have been discussed previously, the numerical results are again achieved by evaluating the shape derivative on each surface node. As an optimization procedure, the Laplace–Beltrami Hessian approximation, equation (5.7), is used in an approximative SQP method with Armijo linesearch. Convergence history and the effects of preconditioning can be seen in figure 7.3. The resulting optimal shapes are shown in figure 7.4.

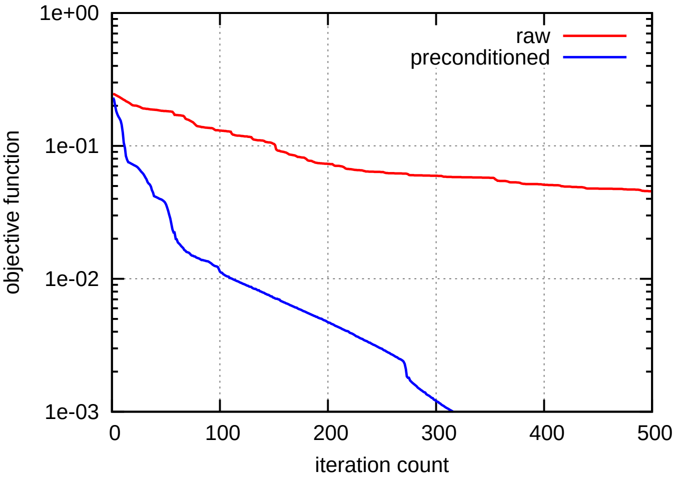


Figure 7.3: Optimization history for potential flow pressure tracking.

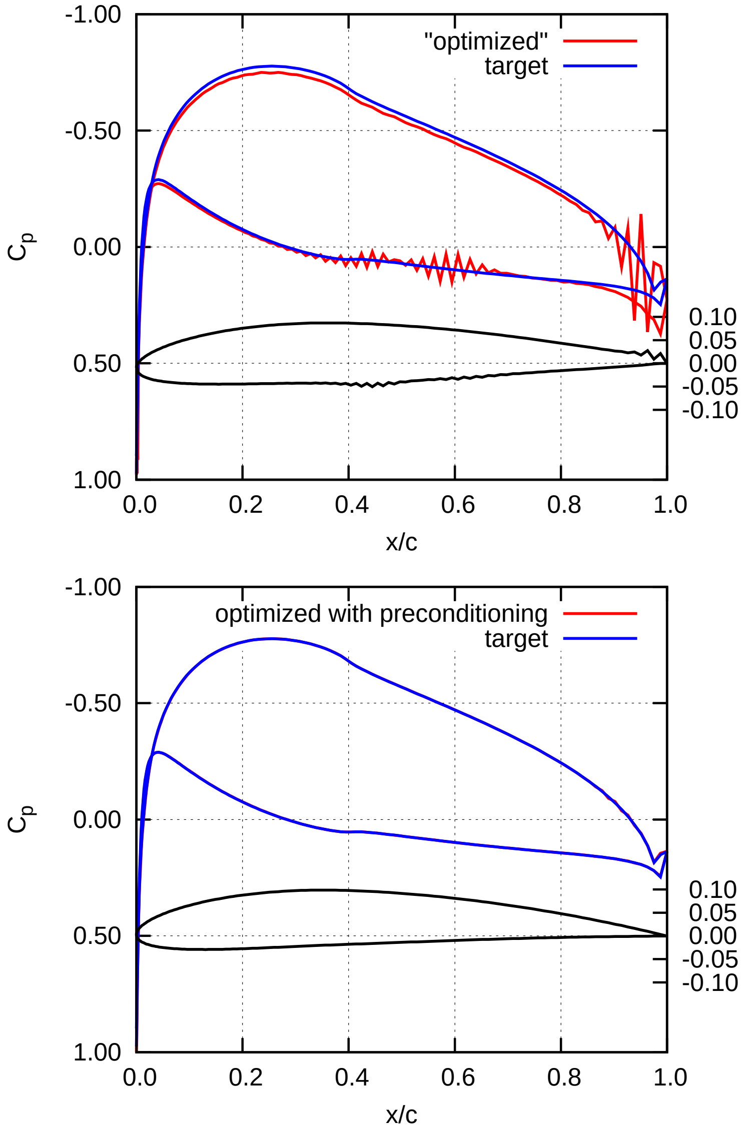
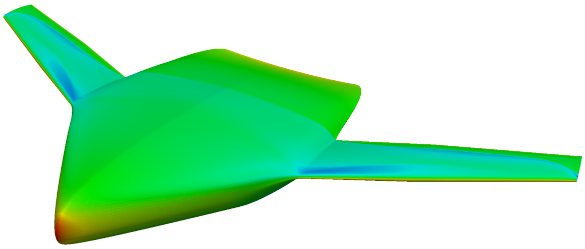


Figure 7.4: Resulting optimal shapes. Without preconditioning, the shapes and pressure distributions degenerate.



Chapter 8

Shape Optimization and Euler Equations

8.1 Introduction

The compressible Euler equations are of considerable importance when it comes to industry size application. Although the viscosity of the fluid is neglected, and consequently there are no boundary layers or turbulence, the compressibility of the gas is taken into full account. Since most flows of industrial importance, such as turbomachinery flow through turbines or compressors, external flow around an aircraft in cruise conditions, or flow around rockets and projectiles, are high speed flows where compressibility is dominating viscous effects, neglecting viscosity is acceptable. Due to the hyperbolic nature of the integral equations, shock waves often occur, requiring sophisticated numerical schemes.

Since problems of this kind are mostly large scale and feature complex geometries, the ease and straight forward applicability of the Hadamard formula of the shape derivative for almost any discretization resolution makes this an ideal application area for numerical procedures based on the Hadamard formula. Here, the focus lies on problems of the aerodynamic type, e.g. drag minimization of a fluid obstacle. However, with industry applicability in mind, the number of constraints to consider considerably increases: in addition of maintaining lift, the aircraft must also withstand the forces, resulting in constraints on the material stiffness, if not a consideration of a full fluid/structure coupling. Some two dimensional results of this chapter have already been published in [57, 58].

Definition 8.1.1 (Inviscid Fluid Forces)**.** *To handle variations in the angle of attack more easily, most flow solvers compute the forces in a rotated coordinate system. For the inviscid Euler equations, this leads to the following expression:*

$$F_a := \frac{1}{C_{\text{ref}}} \int\limits_{\Gamma_0} C_p \langle n, a \rangle \; dS,$$

where C_{ref} is the reference length, usually the airfoil chord, and C_p is the non-dimensional pressure, defined as

$$C_p = \frac{2(p - p_\infty)}{\gamma M_\infty^2 p_\infty}.$$

The subscript ∞ is used to denote the appropriate farfield value. Note that the pressure coefficient C_p should not be confused with the heat capacity at constant pressure c_p. Choosing a as

$$a_1 = (\cos\alpha, 0, \sin\alpha)^T \,,$$

where the first component is again chord wise, the second span wise, and the third wing thickness wise, results in the drag coefficient C_D, and choosing a as

$$a_2 = (-\sin\alpha, 0, \cos\alpha)^T$$

results in the lift coefficient C_L where α is the angle of attack.

Definition 8.1.2 (Euler Problem)**.** *Minimizing the drag coefficient for a lifting aircraft without body forces is given by*

$$\min_{(\rho,u,E,\Omega)} C_D(\rho, u, E, \Omega) \tag{8.1}$$

subject to

$$\begin{aligned}
\int\limits_\Omega \operatorname{div}\,(\rho u)\; dA &= 0\\
\int\limits_\Omega \sum_{j=1}^{3} \left[\frac{\partial(\rho u_i u_j)}{\partial x_j}\right] + \frac{\partial p}{\partial x_i}\; dA &= 0\\
\int\limits_\Omega \operatorname{div}\,(\rho H u)\;\; dA &= 0\\
\langle u, n \rangle &= 0 \quad \textit{on} \quad \Gamma_0\\
C_L &\geq C_{L_0}\\
L := \int\limits_{\Gamma_0} 1 \; dS &\leq L_0\\
I := \int\limits_{\Gamma_0} (y - y_c)^2 \;\; dS &\geq I_{x_0}\\
\text{Vol} &= V_0.
\end{aligned} \tag{8.2}$$

Here, the first three constraints are the Euler equations, and the pressure p and likewise C_p is linked to the conserved variables

$$U := (\rho, \rho u, \rho E)^T$$

by the perfect gas law

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2} \|u\|^2 \right).$$

The condition $\langle u, n \rangle = 0$ is the Euler slip boundary condition on the aircraft surface Γ_0. The farfield boundary conditions are usually more complex involving characteristics and are thought of as treated by the flow solver discretely. Condition $C_L \geq C_{L_0}$ means the optimal aircraft must maintain lift, condition $L \leq L_0$ penalizes the surface area and prevents an increase in perimeter, condition $I \geq I_{x_0}$ is a substitude model for the bending stiffness around the contour center of mass y_c above the x-axis, and constraint $\mathrm{Vol} = \mathrm{V}_0$ *is the usual volume constraint. Usually, not all of these constraints will be used at the same time, but without any of them, the solution will either degenerate into a flat line or will not be of any practical relevance. Also, the bending stiffness constraint is a substitude model valid in two dimensions only.*

For two dimensional applications there is also the additional constraint of the leading edge being fixed at $(0, 0)^T$ and the trailing edge being fixed at $(1, 0)^T$. Otherwise, the optimization changes the reference length C_{ref} of the airfoil, which would lead to a wrong non-dimensionalization of the flow quantities.

Intrigued by practical advantages, the shape derivative, although usually not called that way in this community, has long been sought after by both academia, [15] and [23], and industry [77, 78, 79, 80, 81]. Assuming a non-lifting body in a supersonic potential flow, the optimal shape, the so-called Haack or Sears–Haack body, is known to be ogive-like [33], making the non-lifting supersonic case a good test-case for validating numerics. When it comes to numerical application, the problem is seldom treated from a true shape optimization perspective, except in [3, 4] for pressure tracking or in [10]. None of the approaches above have so far been successfully applied on a large scale drag reduction problem.

8.2 First Order Calculus

Lemma 8.2.1 (Euler Shape Derivative)**.** *For an angle of attack α with corresponding rotation vector a, let p_a be given by $p_a := p \cdot a \in \mathbb{R}^3$. Then the derivative of the functional*

$$J(U, \Omega) = \int_{\Gamma_0} \langle p_a, n \rangle \; dS$$

is given by

$$
\begin{aligned}
& dJ(U,\Omega)[V] \\
&= \int_{\Gamma_0} \langle V,n\rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \operatorname{div}_\Gamma \, (p_a - \lambda U_H u) \right] \, dS && (8.3) \\
&= \int_{\Gamma_0} \langle V,n\rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \kappa \langle p_a, n\rangle \right] + \langle p_a - \lambda U_H u, dn[V] \rangle \, dS, && (8.4)
\end{aligned}
$$

where U are the conserved variables solving the Euler equations in the domain Ω, and p is linked to U by the perfect gas assumption. Additionally, U_H is the vector of conserved variables with the last component replaced by ρH. The adjoint variables λ are given as the solution of

$$
\begin{aligned}
-A_1^T \frac{\partial}{\partial x_1} \lambda - A_2^T \frac{\partial}{\partial x_2} \lambda - A_3^T \frac{\partial}{\partial x_3} \lambda &= 0 && \text{in} \quad \Omega \\
\langle (\lambda_2, \lambda_3, \lambda_4)^T, n \rangle &= -\langle a, n \rangle && \text{on} \quad \Gamma_0 .
\end{aligned}
$$

Here, A_k are the Euler flux Jacobians from remark 4.2.9. The adjoint boundary conditions on the farfield boundaries are assumed to be kept by the flow solver discretely. Based on the choice of a, this gradient expression can both be used for the drag or lift functional.

Proof. A formal differentiation according to lemma 3.3.13 results in

$$
dJ(U,\Omega)[V] = \int_{\Gamma_0} \langle V,n\rangle \left[\frac{\partial p_a}{\partial n} n + \kappa \langle p_a, n\rangle \right] + \langle p_a, dn[V] \rangle + \langle p_a'[V], n\rangle \, dS.
$$

The variation of the normal is given by remark 3.3.15, resulting in

$$
dJ(U,\Omega)[V] = \int_{\Gamma_0} \langle V,n\rangle \left[\frac{\partial p_a}{\partial n} n + (\operatorname{div}_\Gamma p_a) \right] + \langle p_a'[V], n\rangle \, dS. \tag{8.5}
$$

To remove the remaining local shape derivative of the pressure $p_a'[V]$, adjoint calculus will be conducted as in [23, 29]. Let the conserved variables U be given by

$$
U := (\rho, \rho u, \rho E)^T
$$

and the primitive variables U_p by

$$
U_\mathrm{p} := (\rho, u, E)^T.
$$

Almost all density based finite volume flow solvers for compressible fluid dynamics operate on conserved variables. Hence, a linearization of the Euler state equations results in a linearization in terms of conserved variables $U'[V]$:

$$
U'[V] = \left(\rho'[V], (\rho u)'[V], (\rho E)'[V] \right)^T,
$$

which are given as the solution of the linearized Euler equations. Using the non-conservative form of the Euler equations given by remark 4.2.9, the Euler state constraint can also be written by

$$\sum_{k=1}^{3} A_k \frac{\partial U}{\partial x_k} = 0.$$

The forward linearization of the above leads to

$$\sum_{k=1}^{3} \frac{\partial}{\partial x_k} \left(A_k U'[V] \right) = 0.$$

The boundary conditions of the local shape derivatives are described in lemma 3.4.4:

$$\langle u'[V], n \rangle = -\langle V, n \rangle \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \langle u, dn[V] \rangle \tag{8.6}$$

on the wing Γ_0. Multiplication by an arbitrary $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$ and integration by parts gives

$$\begin{aligned} 0 &= \int_{\Omega} \sum_{k=1}^{3} \lambda \frac{\partial}{\partial x_k} \left(A_k U'[V] \right) \, dA \\ &= -\int_{\Omega} \sum_{k=1}^{3} \left(\frac{\partial}{\partial x_k} \lambda \right) A_k U'[V] \, dA + \int_{\partial\Omega} \sum_{k=1}^{3} \lambda n_k A_k U'[V] \, dS \\ &= -\int_{\Omega} \sum_{k=1}^{3} A_k^T \left(\frac{\partial \lambda}{\partial x_k} \right) U'[V] \, dA + \int_{\partial\Omega} \sum_{k=1}^{3} \lambda n_k A_k U'[V] \, dS. \end{aligned} \tag{8.7}$$

Looking at (8.7), one can see that if λ solves the adjoint equation

$$-A_1^T \frac{\partial}{\partial x_1} \lambda - A_2^T \frac{\partial}{\partial x_2} \lambda - A_3^T \frac{\partial}{\partial x_3} \lambda = 0 \quad \text{in} \quad \Omega$$

then the volume integrals will vanish. According to [23, 29], the matrices

$$T := \begin{bmatrix} 0 & \frac{n_1}{\rho} & \frac{n_2}{\rho} & \frac{n_3}{\rho} & 0 \\ \frac{\gamma-1}{2}(u_1^2 + u_2^2 + u_3^2) & (1-\gamma)u_1 & (1-\gamma)u_2 & (1-\gamma)u_3 & \gamma - 1 \end{bmatrix}$$

and

$$T^* := \begin{bmatrix} \rho & \rho u_1 & \rho u_2 & \rho u_3 & \rho H \\ 0 & n_1 & n_2 & n_3 & 0 \end{bmatrix}$$

satisfy the condition

$$\sum_{k=1}^{3} n_k A_k = {T^*}^T T$$

on the boundary Γ_0 of the airfoil. The local shape derivative to remove is $p'[V]$, which must be linked to the local shape derivative of the conserved variables $U'[V]$. This is given by the relation

$$p'[V] = CU'[V], \tag{8.8}$$

where C is given by

$$C := \left(\frac{\gamma - 1}{2}(u_1^2 + u_2^2 + u_3^2),\ (1-\gamma)u_1,\ (1-\gamma)u_2,\ (1-\gamma)u_3,\ \gamma - 1 \right),$$

which is exactly the second row of T. Thus, using (8.8) one arrives at

$$TU'[V] = \begin{pmatrix} \sum_{k=1}^{3} n_k u_k'[V] \\ p'[V] \end{pmatrix} \tag{8.9}$$

and

$$\lambda {T^*}^T = (\lambda U_H, (\lambda_2, \lambda_3, \lambda_4)\, n)\,,$$

where

$$U_H = (\rho, \rho u, \rho H)^T$$

is the vector of conserved variables with the last component replaced by ρH. Using the above in equation (8.7) results in

$$\begin{aligned} 0 = \int_\Gamma \lambda \sum_{k=1}^{3} n_k A_k U'[V]\, dS &= \int_\Gamma \lambda {T^*}^T TU'[V]\, dS \\ &= \int_\Gamma \lambda U_H \langle u'[V], n\rangle + (\lambda_2, \lambda_3, \lambda_4) n p'[V]\, dS. \end{aligned}$$

The linearized Euler slip boundary condition on the airfoil is given by equation (8.6) and provides

$$\begin{aligned} 0 &= \int_\Gamma \lambda \sum_{k=1}^{3} n_k A_k U'[V]\, dS \\ &= \int_\Gamma \langle V, n\rangle \left[-\lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle \right] + \lambda U_H \langle u, dn[V]\rangle + (\lambda_2, \lambda_3, \lambda_4)\, np'[V]\, dS. \end{aligned}$$

Adding the above to the preliminary gradient (8.5) results in

$$\begin{aligned} dJ(U, \Omega)[V] = &\int_{\Gamma_0} \langle V, n\rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + (\operatorname{div}_\Gamma p_a) \right] \\ &+ \langle p_a'[V], n\rangle - \lambda U_H \langle u, dn[V]\rangle + (\lambda_2, \lambda_3, \lambda_4) n p'[V]\, dS \\ = &\int_{\Gamma_0} \langle V, n\rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + (\operatorname{div}_\Gamma p_a) \right] \\ &+ p'[V] \left[\langle a, n\rangle + (\lambda_2, \lambda_3, \lambda_4) n \right] - \lambda U_H \langle u, dn[V]\rangle\, dS. \end{aligned}$$

Thus, the pressure variation will vanish using the following adjoint boundary condition

$$\langle (\lambda_2, \lambda_3, \lambda_4)^T, n \rangle = -\langle a, n \rangle,$$

which means the gradient is given by

$$dJ(U, \Omega)[V] = \int\limits_{\Gamma_0} \langle V, n \rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + (\mathrm{div}_\Gamma\, p_a) \right] - \lambda U_H \langle u, dn[V] \rangle \; dS.$$

Using lemma 3.3.14 and remark 3.3.15 results in

$$dJ(U, \Omega)[V] = \int\limits_{\Gamma_0} \langle V, n \rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \kappa \lambda U_H \langle u, n \rangle + \mathrm{div}_\Gamma\; (p_a - \lambda U_H u) \right] \; dS.$$

The Euler slip boundary condition $\langle u, n \rangle = 0$ eliminates the curvature term, leading to

$$dJ(U, \Omega)[V] = \int\limits_{\Gamma_0} \langle V, n \rangle \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \mathrm{div}_\Gamma\; (p_a - \lambda U_H u) \right] \; dS,$$

which creates the first expression. Application of lemma 3.3.14 in reverse provides

$$\begin{aligned} dJ(U, \Omega)[V] = \int\limits_{\Gamma_0} \langle V, n \rangle & \left[\frac{\partial p_a}{\partial n} n - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \kappa \langle p_a - \lambda U_H u, n \rangle \right] \\ & + \langle p_a - \lambda U_H u, dn[V] \rangle \; dS, \end{aligned}$$

and the second expression is created due to $\langle u, n \rangle = 0$. □

Remark 8.2.2 (Signs)**.** When applying the Euler gradient formula from lemma 8.2.1, all signs will have to be checked discretely, as all of them depend on conventions, such as if the normals are fluid pointing or not, if the adjoint solver defines the Lagrangian using plus or minus, if the method chosen to compute curvature denotes left or right bends with different signs, etc.

8.3 Optimization Strategy

The Euler shape optimization problem is considerably more complex, not only because the objective function is a surface functional, but also because the lift constraint again depends on the solution of a partial differential equation. Hence, a simple projection approach as used previously for preserving the volume constraints is no longer feasible. Although penalty methods have been applied successfully, it was decided to switch to a full approximative reduced SQP method as in [27], which is based on partially reduced SQP methods established in [64, 65].

Remark 8.3.1 (SQP Method for Equality Constraint Optimization)**.** For a standard optimization problem

$$\min_{(u,q)} f(u, q) \tag{8.10}$$

$$\begin{aligned} &\text{subject to} \\ &c(u,q) = 0 \\ &d(u,q) = 0 \end{aligned} \tag{8.11}$$

the Lagrangian is defined as

$$\mathcal{L}(u,q,\mu,\nu) := f(u,q) + \langle \mu, c(u,q) \rangle + \langle \nu, d(u,q) \rangle.$$

Here, c symbolizes the PDE constraint, i.e. the Euler equations, and d symbolizes other constraints such as lift and volume. Under the assumption of linear independence constraint qualification (LICQ), i.e.

$$\{\nabla_{(u,q)} c_1, \cdots, \nabla_{(u,q)} c_{n_c}, \nabla_{(u,q)} d_1, \cdots, \nabla_{(u,q)} d_{n_d}\}$$

is linear independent, the necessary optimality conditions are given by

$$\begin{aligned} \nabla_u \mathcal{L} &= 0 \\ \nabla_q \mathcal{L} &= 0 \\ c(u,q) = \nabla_\mu \mathcal{L} &= 0 \\ d(u,q) = \nabla_\nu \mathcal{L} &= 0. \end{aligned}$$

Applying Newton's method on the above optimality conditions results in the SQP updates

$$\begin{bmatrix} H_{uu} & H_{uq} & (D_u c)^T & (D_u d)^T \\ H_{qu} & H_{qq} & (D_q c)^T & (D_q d)^T \\ D_u c & D_q c & 0 & 0 \\ D_u d & D_q d & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \mu \\ \Delta \nu \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \\ -d \end{pmatrix}, \tag{8.12}$$

where

$$(u_{k+1}, q_{k+1}, \mu_{k+1}, \nu_{k+1})^T = (u_k, q_k, \mu_k, \nu_k)^T + (\Delta u, \Delta q, \Delta \mu, \Delta \nu)^T.$$

Remark 8.3.2 (Reduced SQP Method)**.** Approximating the Hessian matrix in equation (8.12) by

$$\begin{bmatrix} H_{uu} & H_{uq} \\ H_{qu} & H_{qq} \end{bmatrix} \approx \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}$$

and assuming $(D_u c)^{-1}$ exists, a block elemination reduces (8.12) to

$$\begin{bmatrix} B & \tilde{D}_d \\ (\tilde{D}_d)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ \Delta \nu \end{pmatrix} = \begin{pmatrix} -\nabla_q \mathcal{L} + (D_q c)^T (D_u c)^{-T} \nabla_u \mathcal{L} \\ -d + (D_u d)(D_u c)^{-1} c \end{pmatrix},$$

where $\tilde{D}_d$ is given by

$$\tilde{D}_d := (D_q d)^T - (D_q c)^T (D_u c)^{-T} (D_u d)^T. \tag{8.13}$$

Replacing $\Delta \nu$ with $\nu_{k+1} = \nu_k + \Delta \nu$ results in the system

$$\begin{bmatrix} B & \tilde{D}_d \\ (\tilde{D}_d)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ \nu_{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla_q f + (D_q c)^T (D_u c)^{-T} \nabla_u f \\ -d + (D_u d)(D_u c)^{-1} c \end{pmatrix},$$

and the reduction operator (8.13) applied to the scalar function f provides

$$\tilde{D}_f = \nabla_q f - (D_q c)^T (D_u c)^{-T} \nabla_u f,$$

resulting in the system

$$\begin{bmatrix} B & \tilde{D}_d \\ (\tilde{D}_d)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ \nu_{k+1} \end{pmatrix} = \begin{pmatrix} -\tilde{D}_f \\ -d + (D_u d)(D_u c)^{-1} c \end{pmatrix}. \tag{8.14}$$

Remark 8.3.3 (rSQP Method for Aerodynamic Shape Optimization)**.** System (8.14) can be further transformed. Since the adjoint flow solver computes

$$(D_u c)^T \lambda_d = (D_u d)^T,$$

equation (8.14) becomes

$$\begin{bmatrix} B & \tilde{D}_d \\ (\tilde{D}_d)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ \nu_{k+1} \end{pmatrix} = \begin{pmatrix} -\tilde{D}_f \\ \lambda_d c - d \end{pmatrix}. \tag{8.15}$$

For further applications here, the columns of the reduced derivative operator $\tilde{D}$ consist of the discretized shape gradients of the respective constraint d_i, and the reduced Hessian B is approximated by the Laplace–Beltrami operator plus identity, equation (5.7), where k determines the amount of Sobolev smoothing. System (8.15) can either be solved directly, or after a block elemination:

$$\begin{aligned} \tilde{D}_d^T B^{-1} \tilde{D}_d \nu_{k+1} &= d - \lambda_d c - \tilde{D}_d^T B^{-1} \tilde{D}_f \\ B \Delta q &= -\tilde{D}_f - \tilde{D}_d \nu_{k+1}. \end{aligned}$$

Note that if the state equation is solved sufficiently, i.e. $c \approx 0$, the λ_d-term can be neglected.

8.4 Discrete Differential Geometry

Since the drag objective function is a surface functional with dependence on the geometry, the discrete evaluation of the two equivalent versions of the Euler shape derivative, equation (8.3) and (8.4), requires approximating differential geometric quantities on discrete surface meshes. Similarly, the shape Hessian approximation by equation (5.7) in three dimensions also requires solving a PDE on a meshed, discrete, curved surface. Therefore, this section will discuss evaluating these quantities. Interestingly, much of the literature about these subjects stems from computer graphics and computer vision [22, 45] as well as signal and image processing [48] and not PDE constraint optimization.

Since there is much more literature avaliable concerning curvature computations on two dimensional unstructured surface meshes than methods for computing tangential divergence, formula (8.4) is discretized although strictly it does not fulfill the Hadamard form. However, as discussed later, the normal variation $dn[V]$ can be computed conveniently, and thus formula (8.4) was chosen over formula (8.3). Also, when evaluating the analytic expression of the shape gradient at each surface node, an uneven nodal distribution is not taken into account. Moving a vertex will create a much larger deformation of the shape the longer the connecting edges are. Thus, the analytic

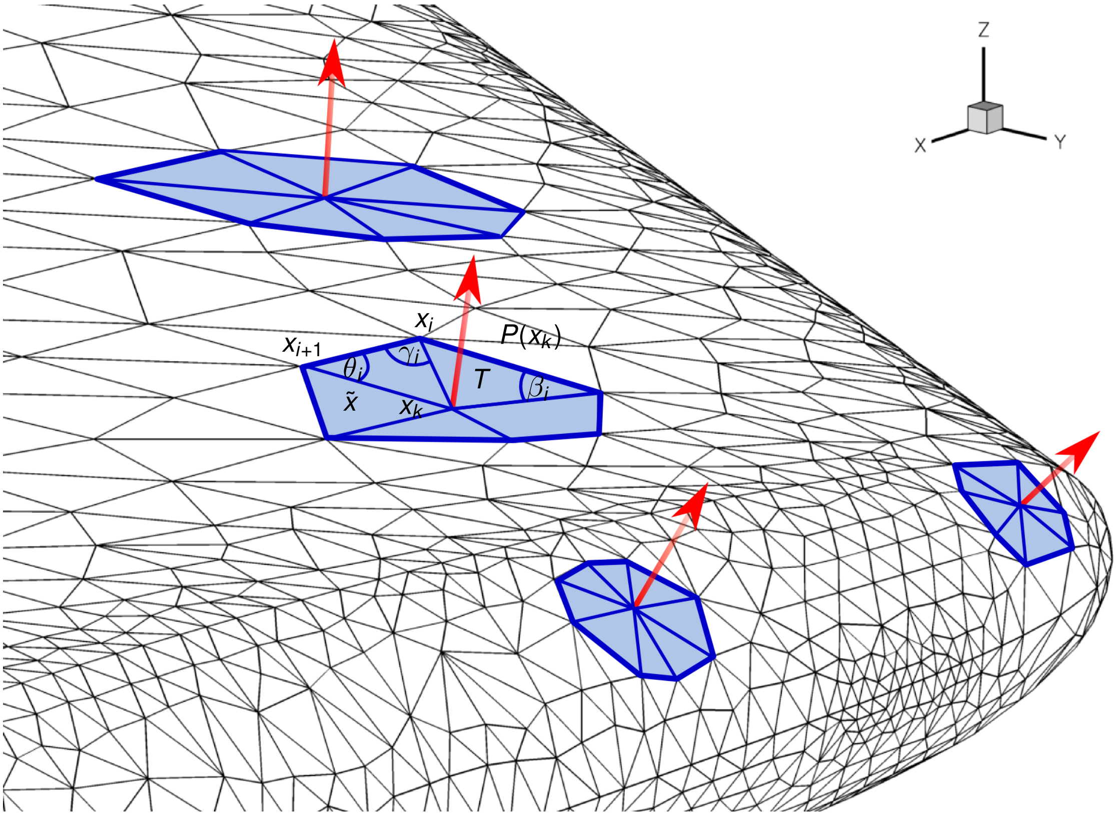


Figure 8.1: Leading edge wing tip of the Onera M6 wing. Surface triangle patches $P(x_k)$ around center nodes x_k shown in light blue. Patch P consists of triangles T with center nodes $\tilde{x}$. Vertices around x_k are labeled x_i counter-clockwise. Surface normals shown in red.

expression (8.4) must be appropriately discretized. For a triangulated surface the shape derivative is given by

$$dJ(U,\Omega)[V] = \int_{\Gamma_0} \langle V, n\rangle g \, dS = \sum_{T\in\Gamma_0} \int_T \langle V, n\rangle g \, dS.$$

In the discrete, a finite dimensional vector expression for the shape derivative is needed. Since each component of this vector is to be associated with a surface node, the above expression must be filtered into components using appropriate discrete perturbation directions V_k. A possible choice for V_k is a linear hat function over the surface triangle patch $P(x_k)$ given by center node x_k, i.e. $V_k(x_i) = 0$ and $V_k(x_k) = n(x_k)$ with linear interpolation in between. This provides for the k-th component of the discrete gradient vector $\nabla J(U)$

$$\begin{aligned}[\nabla J(U)]_k &= \sum_{T\in P(x_k)} \int_T \langle V, n\rangle g \, dS \\ &= \sum_{T\in\Gamma_0} \langle V_k(\tilde{x}_T), n(\tilde{x}_T)\rangle g(\tilde{x}_T)|T|,\end{aligned}$$

where the integral was replaced by a straight forward central node quadrature rule. Here, $|T|$ is the area of the triangle T. Because g also contains curvature, the expression $g(\tilde{x}_T)$ requires a curvature approximation at cell centers, which is surprisingly convenient.

8.4.1 Curvature Evaluation

In two dimensions, curvature can easily be computed according to its definition 2.1.9:

$$\kappa := \operatorname{div}_\Gamma n = \sum_{k=1}^{d-1} \left\langle \frac{\partial n}{\partial \tau_k}, \tau_k \right\rangle = \left\langle \frac{\partial n}{\partial \tau}, \tau \right\rangle .$$

Due to the integration procedure described above, curvature is required at the mid-point of a surface panel, where in two dimensions, surface edges and surface faces are the same and denoted panel. According to the definition above, curvature at the panel mid-point is approximated by

$$\kappa(\tilde{x}) \approx \left\langle \frac{n_{i+1} - n_i}{\|x_{i+1} - x_i\|}, \tau \right\rangle = \frac{\langle n_{i+1} - n_i, x_{i+1} - x_i \rangle}{\|x_{i+1} - x_i\|^2},$$

where n_i is the fluid pointing normal at x_i, and the panel is the line between x_i and x_{i+1}.

Theoretically, it is also possible to compute curvature similarly on an unstructured curved surface of a three dimensional problem. However, this would require reconstruction of the two tangent vectors, because usually the surface nodes will no longer form a basis of the tangent space. Instead, a similar strategy as in [56] is followed.

Definition 8.4.1 (Weingarten Matrix, Second Fundamental Tensor)**.** *Let τ^1 and τ^2 be the directions of an orthonormal coordinate system in the tangent plane of x. The Weingarten matrix, or second fundamental tensor,* **II** *is defined by*

$$\mathbf{II} := [D_{\tau^1} n, D_{\tau^2} n] = \begin{bmatrix} \frac{\partial n}{\partial \tau^1} \tau^1 & \frac{\partial n}{\partial \tau^2} \tau^1 \\ \frac{\partial n}{\partial \tau^1} \tau^2 & \frac{\partial n}{\partial \tau^2} \tau^2 \end{bmatrix} .$$

The derivative of the normal in any tangent direction τ is then given by

$$D_\tau n = \mathbf{II} \cdot \tau,$$

which should not be confused with the derivative of the normal under a shape perturbation $dn[V]$.

Remark 8.4.2 (Second Fundamental Tensor on Surface Triangle)**.** Using the notation as in figure 8.1, the second fundamental tensor **II** on a surface triangle T is given by

$$\begin{aligned} \mathbf{II} \begin{pmatrix} \langle e_1, \tau_1 \rangle \\ \langle e_1, \tau_2 \rangle \end{pmatrix} &= \begin{pmatrix} \langle n(x_i) - n(x_k), \tau_1 \rangle \\ \langle n(x_i) - n(x_k), \tau_2 \rangle \end{pmatrix} \\ \mathbf{II} \begin{pmatrix} \langle e_2, \tau_1 \rangle \\ \langle e_2, \tau_2 \rangle \end{pmatrix} &= \begin{pmatrix} \langle n(x_{i+1}) - n(x_i), \tau_1 \rangle \\ \langle n(x_{i+1}) - n(x_i), \tau_2 \rangle \end{pmatrix} \\ \mathbf{II} \begin{pmatrix} \langle e_3, \tau_1 \rangle \\ \langle e_3, \tau_2 \rangle \end{pmatrix} &= \begin{pmatrix} \langle n(x_k) - n(x_{i+1}), \tau_1 \rangle \\ \langle n(x_k) - n(x_{i+1}), \tau_2 \rangle \end{pmatrix}, \end{aligned} \tag{8.16}$$

where the edges are given by

$$\begin{aligned} e_1 &:= x_i \quad - x_k \\ e_2 &:= x_{i+1} - x_i \\ e_3 &:= x_k \quad - x_{i+1}. \end{aligned}$$

Remark 8.4.3 (Curvature Computation on Unstructured Triangular Meshes)**.** According to [56], curvature can be computed by the following algorithm, where contrary to [56] in the applications

Algorithm 1 Curvature Computation

```
for all Surface mesh nodes $x$ do
    Construct $(\tau^1, \tau^2)$ orthonormal in the tangent plane of $x$
end for
for all Faces $f$ do
    Compute edges and difference of normal $\Delta n$
    Solve equation (8.16) for $\mathbf{II}$ using least squares
    for all Vertex $p$ touching the face do
        Re-express $\mathbf{II}$ in terms of $(\tau^1, \tau^2)$
        Weight by $\omega_{f,p}$ and add to vertex curvature
    end for
end for
for all Vertices do
    Divide accumulated $\mathbf{II}$ by accumulated sum of weights
    Compute principal curvatures $\kappa_1$, $\kappa_2$ as eigenvalues of $\mathbf{II}$
    Compute mean curvature $\kappa$
end for
```

here, the weight $\omega_{f,p}$ is chosen as the area of the face f divided by the sum of the squares of the lengths of the two edges going into the point which is weighted:

$$\omega_{f,p} = \frac{A}{l_1^2 + l_2^2}.$$

Also, the normal difference Δn is weighted in the same way. This weightening is thought of to account for highly distorted surface triangles. Note that the above algorithm already provides a per face value of curvature.

8.4.2 Shape Derivative of the Normal

Due to the integration over mid-points, the variation of the face normal $dn_T[V_k]$ is required for computing the derivative information for node x_k.

Lemma 8.4.4. *For a face T, i.e. surface triangle consisting of the vertices x_k, x_i, and x_{i+1} (see figure 8.1), the variation $dn_T[V_k](x_k)$ of the face normal n_T in direction $V_k = n(x_k)$ is given by*

$$dn_T[V_k](x_k) = \frac{n_k \times (x_i - x_{i+1})}{|T|},$$

where $\times$ denotes the vector cross-product.

Proof. Not considering normalization, the initial face normal is given by

$$n_T := (x_k - x_{i+1}) \times (x_i - x_k) \,.$$

Using V_k as above, the patch center node x_k is moved in direction n_k, the normal at node x_k. This movement is reduced linearly to zero for x_i and x_{i+1}. Hence, a finite perturbation of x_k in direction V_k results in

$$x_k^\epsilon := x_k + \epsilon n_k,$$

and a new normal to the face T^ϵ is given by

$$n_T^\epsilon := \left(x_k^\epsilon - x_{i+1}\right) \times \left(x_i - x_k^\epsilon\right) .$$

Thus, the difference quotient is given by

$$\begin{aligned} \frac{n_T^\epsilon - n_T}{\epsilon} &= \frac{\left(x_k^\epsilon - x_{i+1}\right) \times \left(x_i - x_k^\epsilon\right) - (x_k - x_{i+1}) \times (x_i - x_k)}{\epsilon} \\ &= \frac{x_k^\epsilon \times x_i + x_{i+1} \times x_k^\epsilon - x_k \times x_i - x_{i+1} \times x_k}{\epsilon} \\ &= \frac{\epsilon\,(n_k \times x_i) + \epsilon\,(x_{i+1} \times n_k)}{\epsilon} \\ &= (n_k \times x_i) - (n_k \times x_{i+1}) \\ &= n_k \times (x_i - x_{i+1}) \,. \end{aligned}$$

The desired formula follows with normalization. □

Alternatively, the shape derivative of the normal can also be computed quite efficiently using finite differences, as neither a perturbation of the PDE state nor a disturbed surface or volume mesh is needed. Only the patch under consideration must be perturbed, which is very efficient.

8.4.3 Gradient Validation

In this section, the analytic gradient expressions are briefly compared with the classical approach using the standard formula

$$\frac{dJ}{dq}(u(q), q) = \frac{\partial J}{\partial q} - \lambda^T \frac{\partial c}{\partial q}, \tag{8.17}$$

where the mesh sensitivity Jacobian $\frac{\partial c}{\partial q}$ is computed via finite differences, or full finite differences involving the state equation PDE. The shape gradient is evaluated in each surface mesh node according to the numerical quadrature approach as described above, and the new shape is found by moving the surface node in normal direction accordingly. Other perturbation directions can be used by simply computing the projection $\langle V, n \rangle$. A comparison between the shape derivative and (8.17) can be seen in figure 8.2 for a two dimensional test-case. Supersonic flow at Mach 2.0 around a NACA0012 airfoil at zero angle of attack is considered. Shape derivative and classical Lagrangian based derivative using finite differences for the mesh sensitivity Jacobian are found to match extremely well.

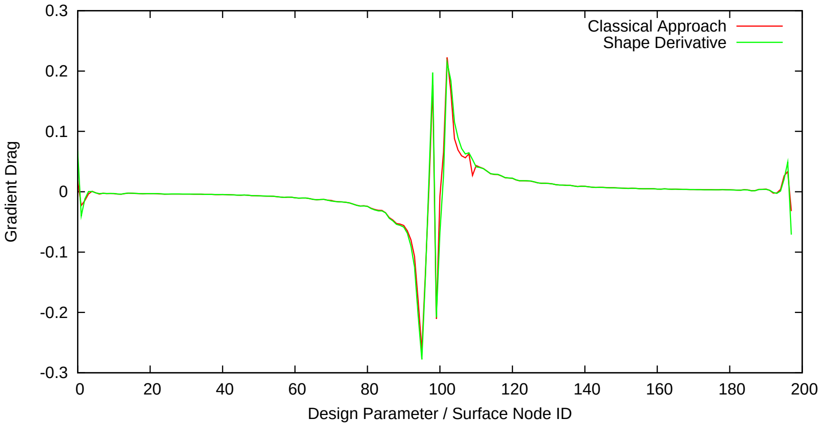


Figure 8.2: Comparison of the shape and classical reference derivative for a symmetric NACA0012 profile at Mach 2.0 and 0° angle of attack. Node 0 is lower side trailing edge, node 100 is leading edge, and node 200 is upper side trailing edge.

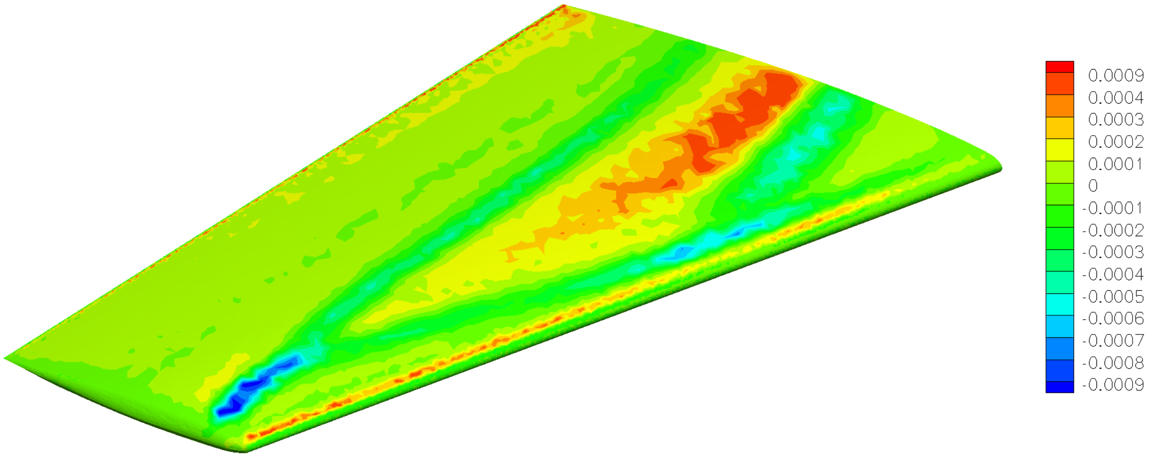
One of the standard test-cases for three dimensions is the Onera M6 wing. At the usual flow conditions of Mach 0.83 and 3.01° angle of attack, the wing develops two interacting shock waves on the upper side, and the shape derivative must be evaluated at discontinuous states. A comparison of the shape derivative and a validation with finite differences is shown in figure 8.3. Unlike the two dimensional validation, the reference solution here is created using complete or “full” finite differences, including the PDE state equation. Thus, for the finite difference gradient at all $18,285$ surface mesh nodes, a Core2Duo E6600 (single thread) processor needed 2.5 days computational time.

8.4.4 Laplace–Beltrami Operator

Before an actual optimization can be conducted, the Laplace–Beltrami preconditioner (5.7) needs to be computed. For two dimensional flow around airfoils, this is relatively straight forward: The surface of the airfoil is a planar graph, and the usual finite difference stencil for the Laplace problem can simply be applied. Left and right neighbors are given by the edges of the airfoil surface, and thus finite differencing is conducted in the tangent plane.

In three dimensions, a more sophisticated operator discretization using unstructured triangular surface meshes is needed. A detailed finite element analysis for the Laplace–Beltrami operator on arbitrary surfaces can be found in [14]. The strategy employed here corresponds to the Laplace–Beltrami operator discretization in [48]. Similar to other discrete differential geometry operators, construction of the discretized Laplace–Beltrami operator differs greatly depending on explicit or implicit representations of the surface. Since shapes for aerodynamic shape optimization are all explicitly given by the triangulated CFD mesh, the area averaged Laplace–Beltrami discretization for explicit surfaces from [48] is used. For a surface node x_k with surrounding triangle patch P, as shown in figure 8.1, the Laplace–Beltrami operator is approximated by

$$\Delta_\Gamma g(x_k) \approx \frac{1}{|P|} \int_P \Delta_\Gamma g(x)\, dS(x)$$



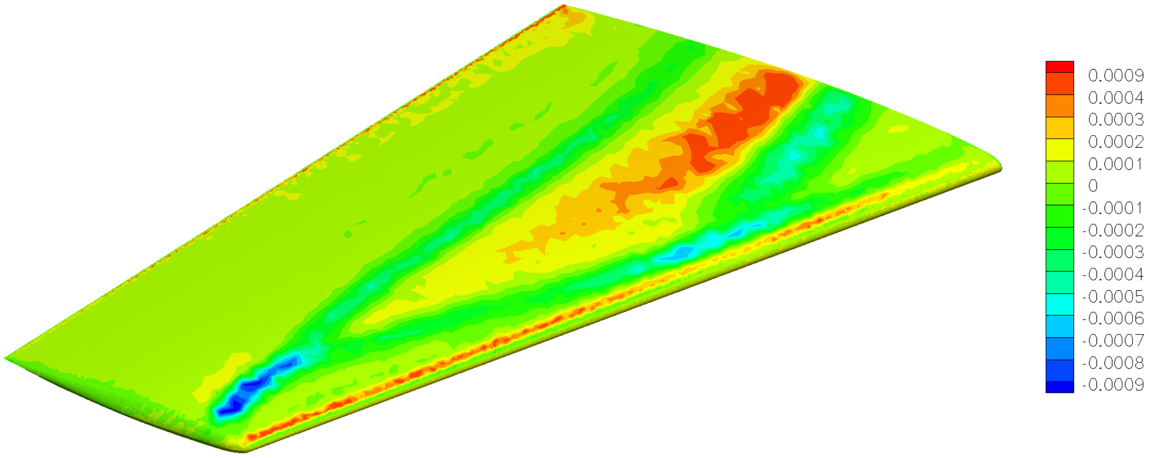


Figure 8.3: Validation of the shape derivative in three dimensions on an Onera M6 wing with 18, 285 design parameters. Inflow velocity Mach 0.83 and 3.01° angle of attack. Top finite differences, bottom shape derivative.

$$
\begin{aligned}
&= -\frac{1}{|P|}\int\limits_{P} \langle \nabla_\Gamma, \nabla_\Gamma g\rangle \, dS\\
&= -\frac{1}{|P|}\int\limits_{\partial P} \langle \nabla_\Gamma g, n_{\partial P}\rangle \, d\ell\\
&= -\frac{1}{|P|}\sum_{T\in P}\int\limits_{\partial P\cap T} \langle \nabla_\Gamma g, n_{\partial P}\rangle \, d\ell\\
&= -\frac{1}{|P|}\sum_{T\in P} \nabla_\Gamma g(x_k)\,[x_i - x_{i+1}]^\perp ,
\end{aligned}
$$

since $\nabla_\Gamma g$ is assumed constant in T. Also note that $\perp$—and likewise $n_{\partial P}$—means perpendicular to the boundary ∂P in the tangent space of the surface, as otherwise there would be an extra degree of freedom. As usual when employing finite elements, g is assumed to consist of nodal weights and linear ansatz functions:

$$
g\Big|_T(x) = g(x_k)\varphi_k(x) + g(x_i)\varphi_i(x) + g(x_{i+1})\varphi_{i+1}(x).
$$

Since the ansatz functions are linear, their contour lines in T are parallel to the opposite edge, and since the gradient is perpendicular to the contour lines, the following relation holds:

$$
\begin{aligned}
\nabla_\Gamma\varphi_k &= \frac{1}{2|T|}\,[x_{i+1} - x_i]^\perp\\
\nabla_\Gamma\varphi_i &= \frac{1}{2|T|}\,[x_k - x_{i+1}]^\perp\\
\nabla_\Gamma\varphi_{i+1} &= \frac{1}{2|T|}\,[x_i - x_k]^\perp .
\end{aligned}
$$

Thus, the tangential gradient $\nabla_\Gamma g$ is given by

$$
\begin{aligned}
&\nabla_\Gamma g(x)\Big|_T\\
&=\frac{1}{2|T|}\Big(g(x_k)\,[x_{i+1} - x_i]^\perp + g(x_i)\,[x_k - x_{i+1}]^\perp + g(x_{i+1})\,[x_i - x_k]^\perp\Big)\\
&=\frac{1}{2|T|}\Big(g(x_k)\,[x_{i+1} - x_k + x_k - x_i]^\perp + g(x_i)\,[x_k - x_{i+1}]^\perp + g(x_{i+1})\,[x_i - x_k]^\perp\Big)\\
&=\frac{1}{2|T|}\Big(-g(x_k)\,[x_k - x_{i+1}]^\perp - g(x_k)\,[x_i - x_k]^\perp + g(x_i)\,[x_k - x_{i+1}]^\perp + g(x_{i+1})\,[x_i - x_k]^\perp\Big)\\
&=\frac{1}{2|T|}\Big((g(x_i) - g(x_k))\,[x_k - x_{i+1}]^\perp + (g(x_{i+1}) - g(x_k))\,[x_i - x_k]^\perp\Big) ,
\end{aligned}
$$

a constant expression for all x in T. Thus, for the Laplace–Beltrami operator one has

$$
\begin{aligned}
&\Delta_\Gamma g(x_k)\\
&= -\frac{1}{|P|}\sum_{T\in P}\frac{1}{2|T|}\Big((g(x_i) - g(x_k))\,[x_k - x_{i+1}]^\perp + (g(x_{i+1}) - g(x_k))\,[x_i - x_k]^\perp\Big)\,[x_i - x_{i+1}]^\perp .
\end{aligned}
$$

Since the area of T is proportional to the sine of any angle of the triangle one has

$$\Delta_\Gamma g(x_k) = -\frac{1}{|P|}\sum_{T\in P}(g(x_i) - g(x_k))\cot(\theta_i) + (g(x_{i+1}) - g(x_k))\cot(\gamma_i).$$

Since $\gamma_i = \beta_{i+1}$, an index shift in the second part of the sum results in

$$\begin{aligned}\Delta_\Gamma g(x_k) &= -\frac{1}{|P|}\sum_{T\in P}(g(x_i) - g(x_k))\cot(\theta_i) + (g(x_i) - g(x_k))\cot(\beta_i)\\ &= \frac{1}{|P|}\sum_{T\in P}(\cot(\theta_i) + \cot(\beta_i))\,(g(x_k) - g(x_i))\,.\end{aligned}$$

The above equation is assembled into a matrix, and the discrete Laplace–Beltrami preconditioner (5.7) is factorized using a direct sparse linear solver. The effect on the gradient is illustrated in figure 8.4

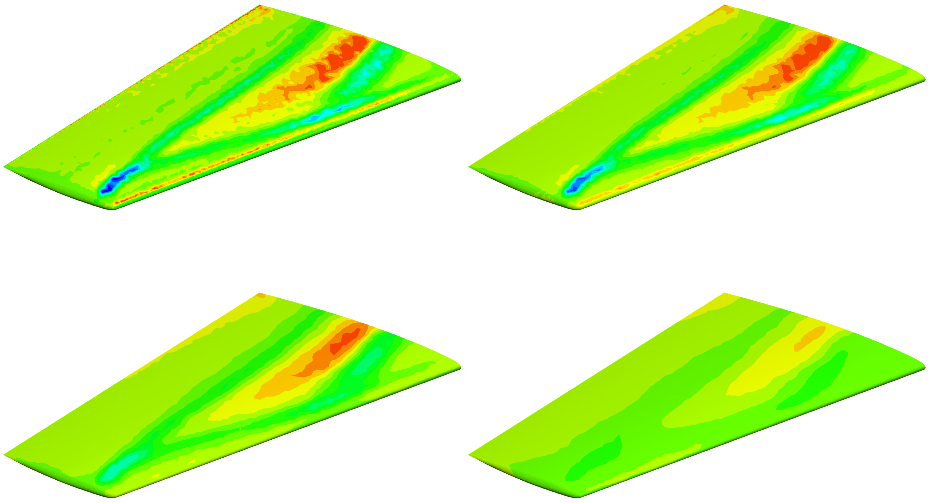


Figure 8.4: Effects of the Laplace–Beltrami preconditioner (5.7) on the drag gradient for $k = 0, 10^{-2}, 10^{-1}, 10^{0}$ on the Onera M6 wing.

for the Onera M6 wing at Mach 0.83 and angle of attack $\alpha = 3.01°$.

8.5 Airfoil Optimizations

All of the following two and three dimensional optimizations have been conducted using the DLR flow solver TAU, a hybrid unstructured vertex centered finite volume flow solver for viscid and inviscid compressible flow [21, 26, 68]. For inviscid computations, TAU features both a continuous and hand-discrete adjoint, while for viscous computations, only a discrete adjoint mode is avaliable. TAU is also one of the production codes of Airbus, making the following computations examples of real world applications. TAU also consists of several auxiliary routines for mesh partitioning for parallel computations and a mesh deformation routine. The auxiliary programs are connected using a common Python interface. The surface mesh is perturbed as discussed above, and the

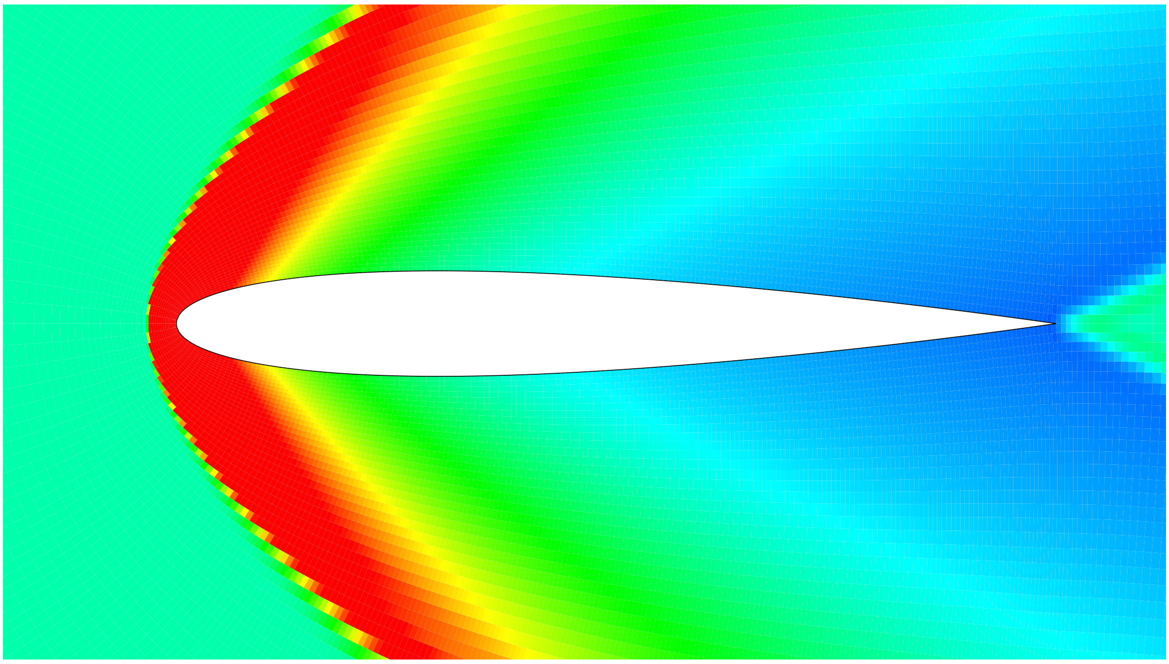


Figure 8.5: Initial NACA0012 airfoil at Mach 2.0. The blunt nose body produces a strong detached bow shock. Color denotes pressure.

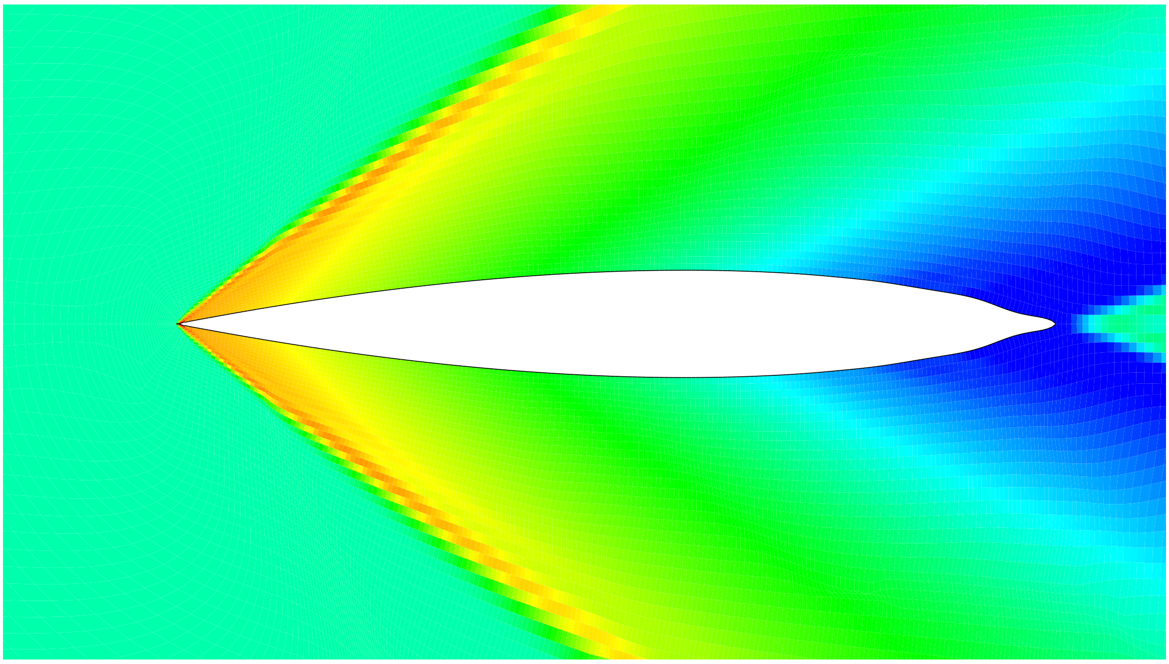


Figure 8.6: Optimized airfoil for Mach 2.0 at zero angle of attack. The blunt nose has become sharp, producing a much weaker attached shock wave. Color denotes pressure.

built-in TAU mesh deformation tool fits the volume cells accordingly. Unfortunately, the built-in mesh deformation tool is optimized for smooth deformations using few design parameters. Using the shape derivative to perturb all surface mesh nodes can occasionally overwhelm the TAU mesh deformation, sometimes resulting in zero volume or distorted cells. In such cases, re-meshing is advisable.

8.5.1 Supersonic NACA Airfoil

A simple two dimensional problem that demonstrates the features of Euler flow drag reduction is that of a symmetric airfoil in supersonic flow, e.g. the cross-section of a vertical fin or stabilizer. The supersonic setup, as opposed to subsonic, is advantageous for testing purposes for several reasons. Physical drag is always present and substantially higher than numerical drag, optimal shapes for the different constraints are known beforehand [33], and it is easier to capture drag correctly using coarser grids. In subsonic flow, physical drag may disappear, leading to many possible local optima, and good grid resolution is needed in the area of the weaker shock, the position of which changes during the optimization run. Also, the shock wave is detached from the airfoil, meaning although there is a discontinuity in the state, it is not at the surface where the shape derivative must be evaluated. The optimization history for different mesh resolutions is shown in figure 8.7, and the shapes are shown in figure 8.5 and figure 8.6. The geometry definition for standard four digit NACA airfoils can be found in [39]. On the finest level, the airfoil is discretized using 400 surface mesh nodes, of which 398 are variable. The leading edge and trailing edge are fixed, such that the airfoil chord remains at $(0, 0)$ and $(1, 0)$ and does not interfere with non-dimensionalization. Also, the volume of the airfoil is fixed at the initial NACA0012 value of $8.2160 \cdot 10^{-2}$. After initializing the forward solution with 278 iterations, the shape is optimized in a one-shot sense using three forward and adjoint flow solver iterations per optimization step. The optimal

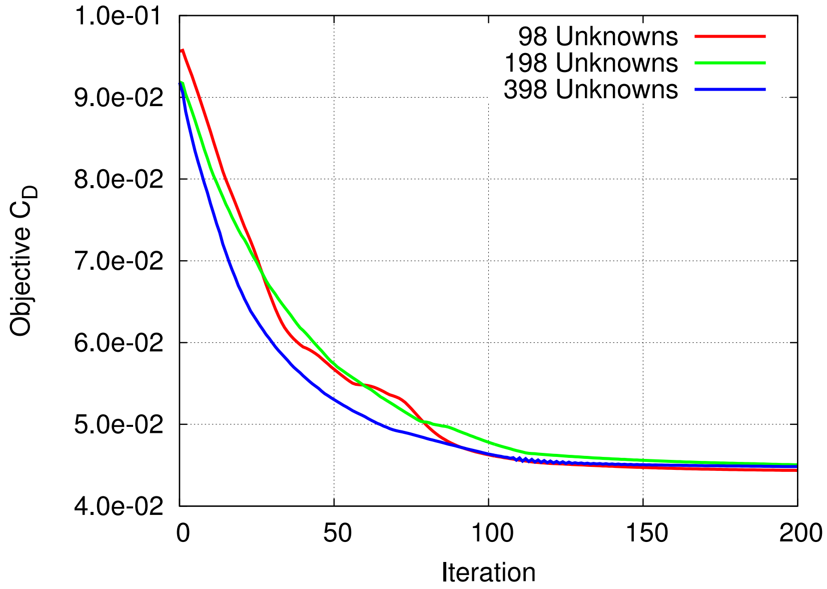


Figure 8.7: Optimization history for the supersonic NACA0012 optimization on different meshes.

shape is reached after about 150 iterations of the rSQP method as described in remark 8.3.3. The reduced Hessian B is approximated by the Laplace–Beltrami preconditioner (5.7).

8.5.2 Mesh Independence

Although the flow solver is not completely mesh independent when computing the state and adjoint variables, a careful selection of the various parameters such as the number of inner iterations, the optimization step length, and the amount of gradient smoothing can lead to a potentially mesh resolution independent optimization convergence behavior, as can be seen in figure 8.7. The precise settings used in the optimization are listed in table 8.1. The first column shows the number of flow solver iterations needed to reach a primal residual of 10^{-7}. The second column shows the number of adjoint flow solver iterations needed to reach an adjoint residual of 10^{-4} for the lift adjoint. Thus, the flow solver convergence rate is not mesh independent. Nevertheless, during the one-shot optimization, the number of inner flow solver iterations can be kept constant at three for both primal and adjoint solver. The smoothing factor for the two dimensional Laplace–Beltrami preconditioner, equation (5.7), had to be adapted, and the respective value for k is shown in the last column. Interestingly, the smoothing parameter here features the same monotonicity behavior as in the Stokes case, equation (5.8).

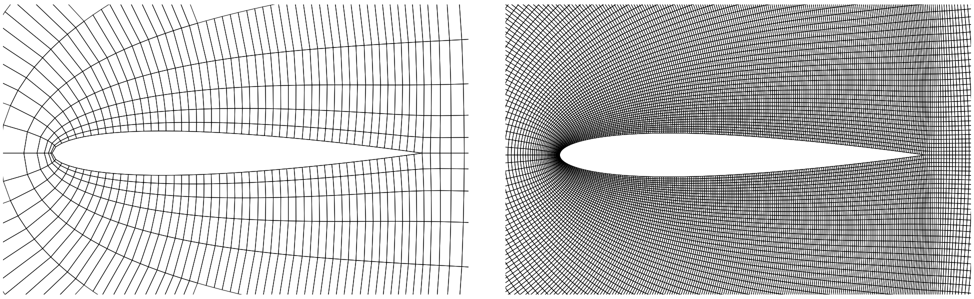


Figure 8.8: Coarse NACA0012 mesh and third refinement for testing mesh independent optimization convergence.

Unknowns	Initial Forward	Initial Adjoint	Inner Forward	Inner Adjoint	Smoothing
98	158	128	3	3	4.0
198	182	215	3	3	2.0
398	278	404	3	3	0.5

Table 8.1: Parameters used in the mesh independency study

Hence, the Laplace–Beltrami Hessian approximation has the potential for mesh independent convergence, which is highly desireable for the optimizations of wings and aircrafts in three dimensions, where the number of shape parameters increases significantly. Two of the initial meshes for the mesh independency study are shown in figure 8.8.

8.5.3 Transonic Lifting RAE2822 Airfoil

One of the classical examples for aerodynamic shape optimization is the optimization of the RAE-2822 airfoil in transonic cruise condition of Mach 0.73 and 2° angle of attack. The initial lift is to be maintained. Due to its popularity, this test-case is sometimes simply denoted by “case 9”. At these flow conditions, it is assumed the shock wave of the upper side can be completely removed by optimization. Thus a shock free solution is considered optimal. Unlike the other examples where the aim was a drastic deformation of the shape, here only small modifications to the airfoil surface are required. Usually for “case 9”, a mesh consisting of 128 surface nodes is used, which is deformed by Hicks–Henne functions [35] enforcing the airfoil thickness to remain constant at each chord point during the optimization. This test-case has been re-created using the shape derivative for all surface nodes. The final shape is shown in figure 8.9 and is shock free. After fixing the leading and trailing edge, 126 variable surface nodes remain. Instead of the constant thickness by Hicks–Henne parameterization, a constant volume is enforced. The initial lift of $C_L = 7.817 \cdot 10^{-1}$ is increased to $C_L = 7.831 \cdot 10^{-1}$, a very slight increase by 0.2%, while the drag is reduced by 49.4%, from $C_D = 6.547 \cdot 10^{-3}$ to $C_D = 3.295 \cdot 10^{-3}$. The optimization requires again 100 steps with 20 inner iterations for each of the flow solvers, i.e. the primal flow solver and the two adjoint flow solver for drag and lift.

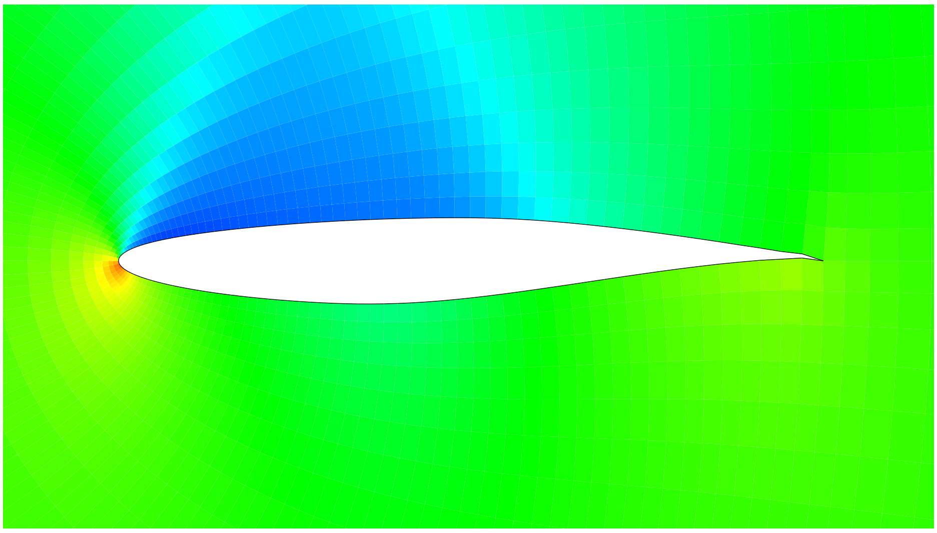


Figure 8.9: Optimal shape for the RAE2822 transonic cruise “case 9” optimization. The standard mesh is a C-type mesh using 128 surface nodes. The optimized solution is shock free on this mesh. Color denotes pressure.

Additionally, a much finer mesh with 512 surface nodes is also considered. Initial flow field and optimized flow field are shown in figure 8.10 and figure 8.11. The optimization history is shown in figure 8.12. Using a proper tuning of all parameters, the optimal solution can be found in a mere 28 optimization steps with 40 inner iterations for each of the flow solvers. Although the optimal shape on this finer mesh features a very weak shock wave on the upper side, the computed drag value of $C_D = 2.721 \cdot 10^{-3}$ is lower than the optimal solution on the coarser mesh. In total, the drag has been reduced by 62.9%, while lift has increased to $C_L = 7.828 \cdot 10^{-1}$, an increase by

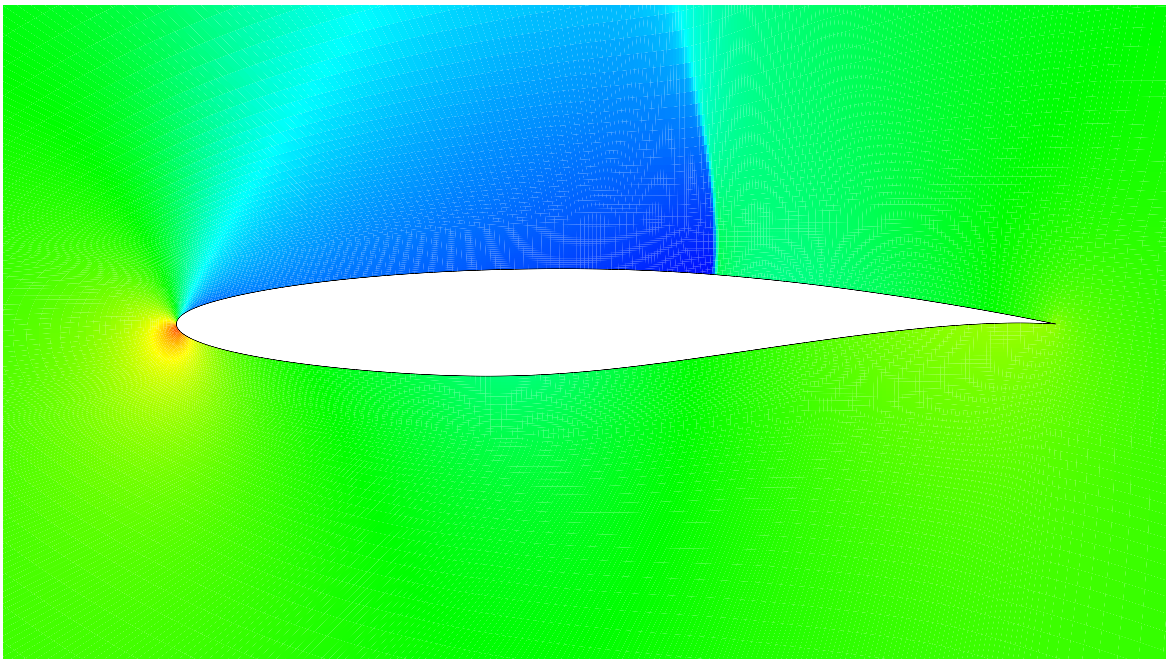


Figure 8.10: Initial RAE2822 airfoil at Mach 0.73 and 2° angle of attack. Under these cruise conditions, the airfoil features a strong shock wave on the upper side. Color denotes pressure.

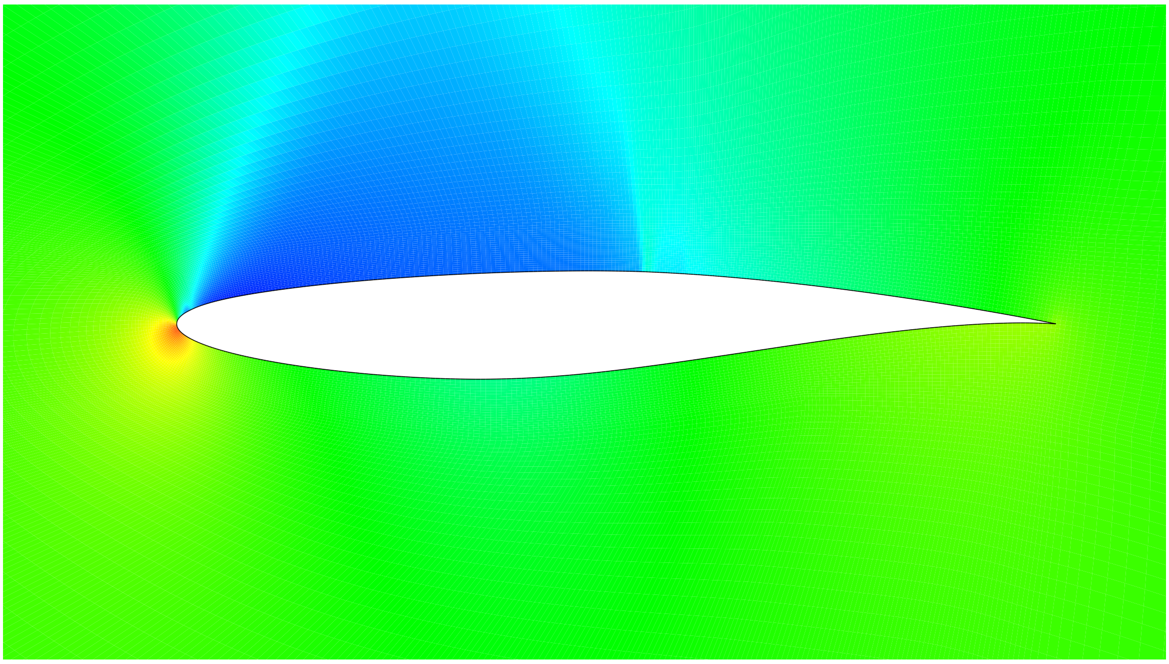


Figure 8.11: Optimized RAE2822 airfoil. The upper side shock wave is greatly reduced. Color denotes pressure.

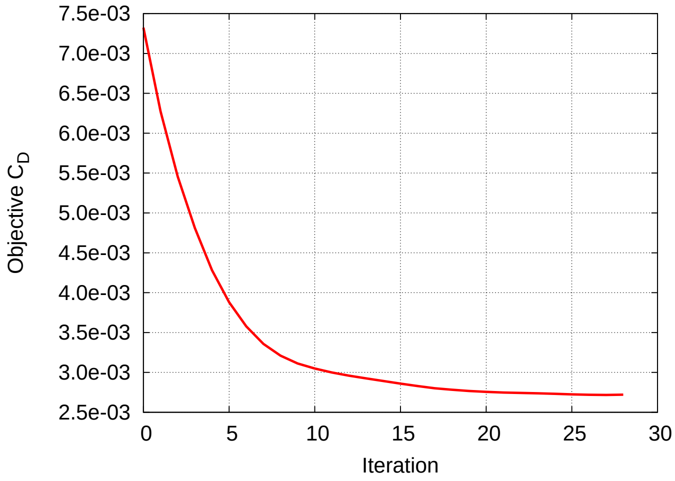


Figure 8.12: Optimization history for the transonic RAE2822 optimization with 510 variable surface mesh nodes.

0.1%. Before the one-shot optimization is started, the primal solution is initialized by 580 iterations. The adjoint solution for the drag is initialized by 945 adjoint flow solver iterations. Initializing the lift adjoint requires 1348 iterations.

8.6 Onera M6 Wing Optimization

One of the standard test-cases in three dimensions is the Onera M6 wing. In cruise condition of

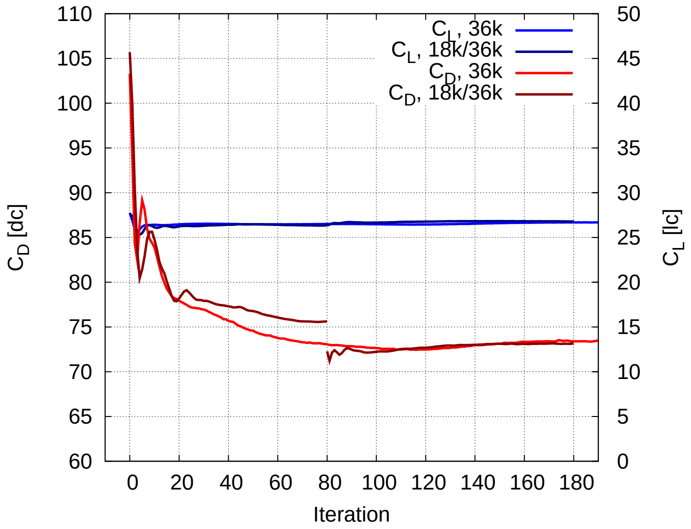


Figure 8.13: Multilevel optimization history for the Onera M6 wing. One drag count dc is a scaling of 10^{-4} and one lift count lc is a scaling of 10^{-2}.

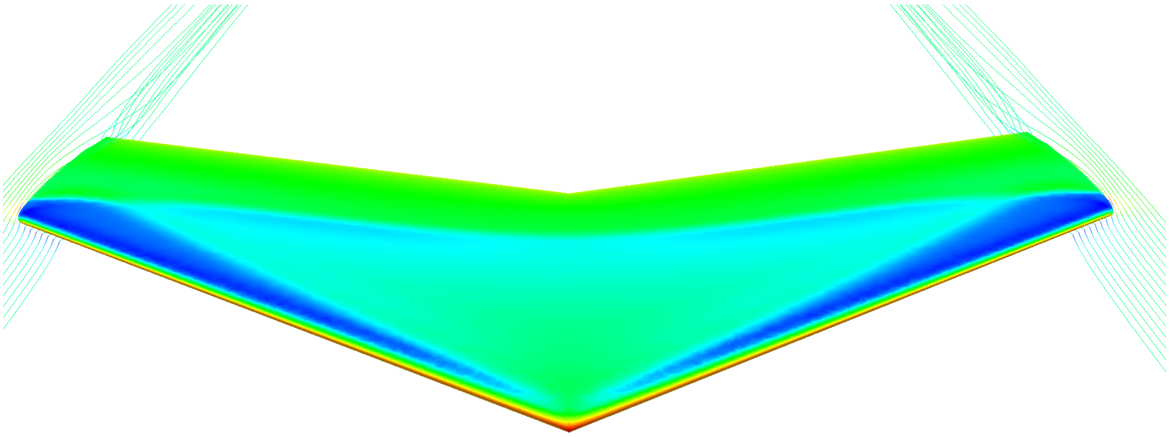


Figure 8.14: Initial Onera M6 wing at Mach 0.83 and 3.01° angle of attack. The wing develops two interacting shock waves on the upper side, so called λ-shock waves. The wing surface consists of $18,285$ nodes. Color denotes pressure.

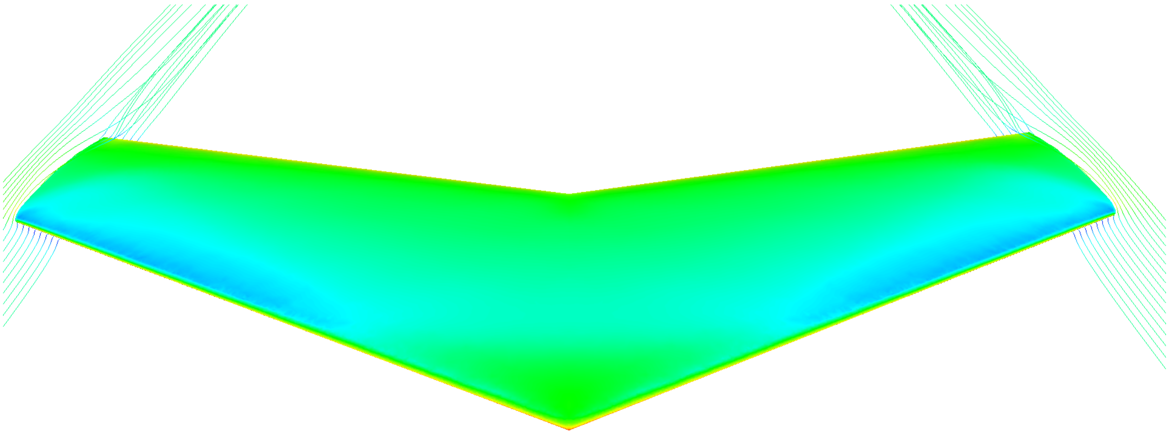


Figure 8.15: Optimized Onera M6 wing. The shock waves are completely removed. Color denotes pressure.

Mach 0.83 and 3.01° angle of attack, the wing develops two interacting shock waves on the upper side, sometimes also called λ-shock due to the visual shape of the merging shock waves. A multilevel optimization is considered, and the coarser mesh consists of $18,285$ surface nodes. In total, this mesh results in $541,980$ state unknowns for the Euler equations. After refinement, the next finer mesh features $36,806$ surface nodes and $1,486,315$ Euler state unknowns. Simulation and optimization is conducted only on one half of the wing, the entire wing is then created by mirroring at the symmetry plane. Initial and optimal wing are shown in figure 8.14 and figure 8.15. Starting on the coarser mesh, an initial drag value of $C_D = 7.52 \cdot 10^{-3}$ and lift value of $C_L = 2.65 \cdot 10^{-1}$ is computed. Lift and the initial volume of $V_0 = 63.6$ are used as a constraint. A multi-level optimization is beneficial, because usually mesh refinement is based on better resolving shock waves, which are removed due to optimization. The planform is considered fixed, because otherwise, the TAU mesh deformation tool for the volume mesh very frequently produces meshes of bad quality with inverted or almost zero cells. The optimization history for the multilevel optimization is shown in figure 8.13. The respective meshes are shown in figure 8.16.

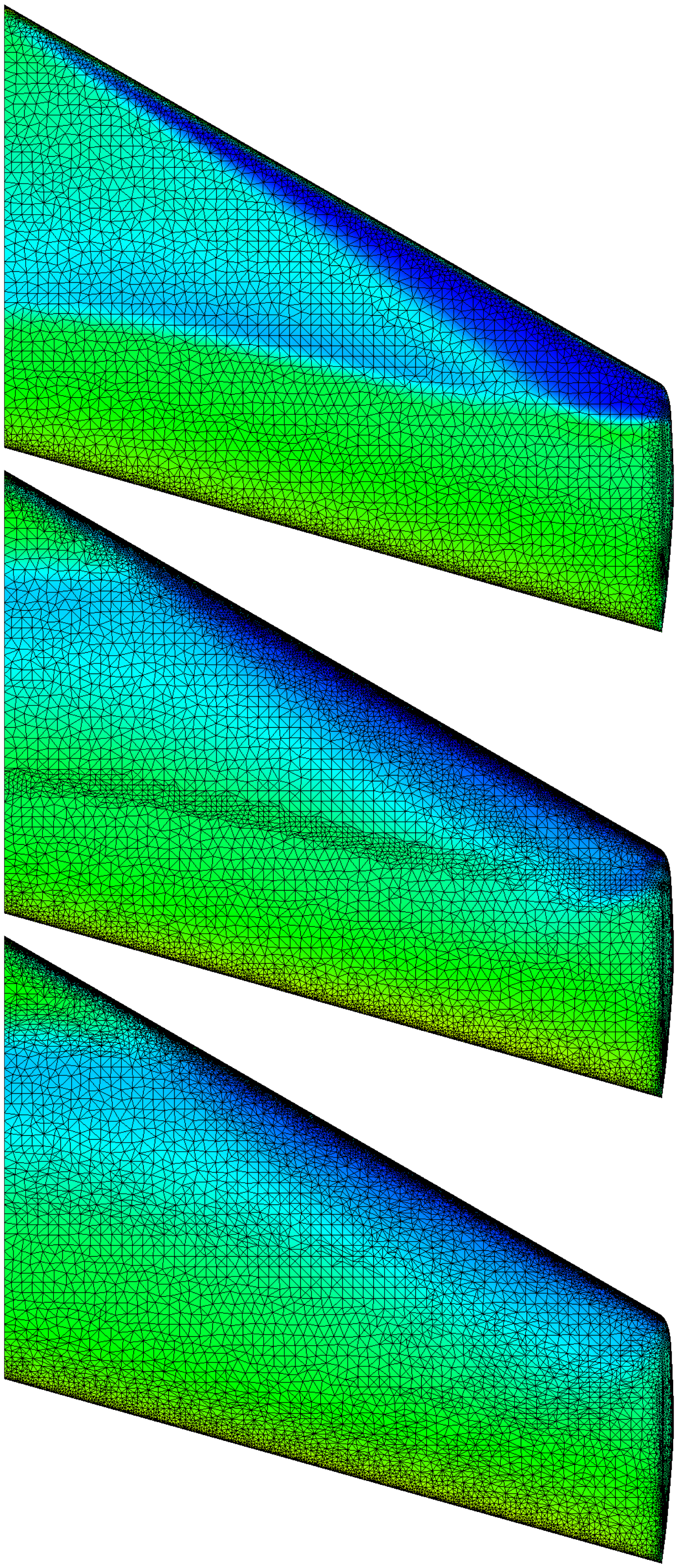
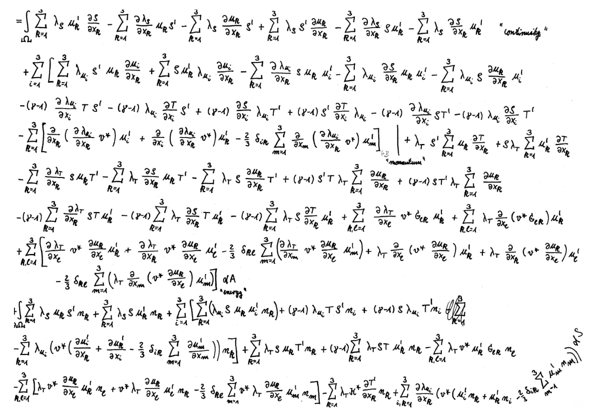


Figure 8.16: Initial solution, optimal solution with refinement based on initial solution, and optimal solution with refinement during optimization.



Chapter 9

Compressible Navier–Stokes Equations

9.1 Introduction

After considering viscous incompressible fluids in chapter 6 and inviscid compressible fluids in chapter 8, this chapter seeks to combine both and considers the compressible Navier–Stokes equations, i.e. a viscous and compressible fluid. Presently, the compressible Navier–Stokes equations are considered the ultimate fluid model. They are assumed to correctly model all single phase fluid flows of matching stress tensor. Usually, the often used perfect gas assumption is the limiting factor in applicability, as at hypersonic speeds or in chemically active fluids, the chemistry must be modeled also. Contrary to the Euler equations considered in chapter 8, the compressible Navier–Stokes equations reproduce fluid boundary layers, flow separation, and consequently turbulence. Unfortunately, they are correspondingly complex and both analytically inaccessible and numerically very difficult to solve. Since high Reynolds number fluid phenomena are of very different time- and space scales, e.g. fast, small-scale turbulent motion versus large-scale, almost steady mean flow, resolving all fluid phenomena would require a prohibitively fine discretization. This is also called direct numerical simulation (DNS) and is almost always used to calibrate substitute turbulence models in simple domains. Since almost all flows of industrial importance are high Reynolds number flows, the compressible Navier–Stokes equations are almost always averaged in time and space. Usually, this is accomplished by the so-called Reynolds averaging, which ignores density fluctuations, but other averaging approaches including density variations exist, such as Favre averaging. Unfortunately, averaging introduces new unknowns in the eddy viscosity, and the resulting compressible Reynolds averaged Navier–Stokes equations (RANS) require additional turbulence modeling. Most turbulence models turn out to be analytically non-differentiable, such as the wall boundary condition of the k-ω model, or introduce excessively complex expressions. For this reason, a shape differentiation for the RANS equations is conducted only around the mean flow ignoring the eddy viscosity and Sutherland's law. This is sometimes also called frozen viscosity approach, meaning the derivative is made for the original compressible Navier–Stokes equations in a DNS fashion, but during application the averaged, i.e. turbulent, values are used.

Definition 9.1.1 (RANS Equations)**.** *The original compressible Navier–Stokes equations are given by definition 4.1.11. To simplify differentiation, they will be re-written in differential form using the variables* (ρ, u, T)*. Note that this is different from both the standard conserved variables* $(\rho, \rho u, \rho E)$ *and the standard primitive variables* (ρ, u, p)*. Not considering external body forces g, they are given by*

$$\sum_{k=1}^{3}\left(u_k\frac{\partial\rho}{\partial x_k}+\rho\frac{\partial u_k}{\partial x_k}\right)=0 \tag{9.1}$$

$$\left(\rho\sum_{k=1}^{3}u_k\frac{\partial u_i}{\partial x_k}\right)+(\gamma-1)\frac{\partial}{\partial x_i}(\rho T)-\sum_{k=1}^{3}\frac{\partial}{\partial x_k}(\mu^*\sigma_{ik})=0 \tag{9.2}$$

$$\left(\rho\sum_{k=1}^{3}u_k\frac{\partial T}{\partial x_k}\right)+(\gamma-1)\rho T\sum_{k=1}^{3}\frac{\partial u_k}{\partial x_k}-\sum_{j,k=1}^{3}\mu^*\frac{\partial u_k}{\partial x_j}\sigma_{jk}-\sum_{k=1}^{3}\frac{\partial}{\partial x_k}\left(\kappa^*\frac{\partial T}{\partial x_k}\right)=0. \tag{9.3}$$

The transition from definition 4.1.11 is straight forward, and more information can be found in [9]. The continuity equation is given by (9.1)*, momentum conservation is given by* (9.2) *in the three spatial dimensions* $i = 1, 2, 3$*, and* (9.3) *denotes the energy equation. For the Newtonian fluids considered here, the strain tensor* σ *is given by*

$$\sigma_{kj}=\sigma_{jk}=\frac{\partial u_k}{\partial x_j}+\frac{\partial u_j}{\partial x_k}-\left(\frac{2}{3}\sum_{m=1}^{3}\frac{\partial u_m}{\partial x_m}\right)\delta_{kj}, \tag{9.4}$$

see also definition 4.1.5, and the pressure is given by

$$p = R\rho T.$$

Using an appropriate non-dimensionalization, the pressure is more conveniently given by

$$p = (\gamma-1)\rho T. \tag{9.5}$$

Here, values marked by $*$ *denote the total quantities, i.e. the mean and averaged turbulent value, such as the total eddy viscosity*

$$\mu^* = \mu + \mu_{turb},$$

and the total heat conduction

$$\kappa^* := \frac{\gamma}{Re}\left(\frac{\mu}{Pr}+\frac{\mu_{turb}}{Pr_{turb}}\right),$$

where Re is the Reynolds number and Pr is the Prandtl number. A formula for these quantities depend on the chosen turbulence model, which links these turbulent quantities to the remaining variables and the geometry of the domain by some additional algebraic or differential equations.

9.2 First Order Calculus

In this section, the shape derivative for the force optimization problem in a compressible Navier–Stokes fluid is constructed. Special attention is paid to finding an expression that can easily be evaluated. For more theoretical details, especially concerning existence and uniqueness, see [51, 52, 54]. Especially in [51], a general framework for the analysis of inhomogeneous elliptic-hyperbolic equations such as the compressible Navier–Stokes equations is established, and an analysis is performed for small perturbations of the so-called approximate solutions, which are determined from Stokes problems. Furthermore, the existence of optimal shapes of the isothermal Navier–Stokes equations is shown in [52] using generalized solutions for the Navier–Stokes equations. Also, from a standard, non shape calculus point of view, the problem has also been studied in [29] for a thin shear-layer approximation of the viscous stress tensor or in [41].

Definition 9.2.1 (Compressible Navier–Stokes Shape Optimization Problem)**.** *As a shape optimization problem, the aerodynamic or fluid forces on a solid body are again considered. Contrary to the Euler problem, definition 8.1.1, the viscous part of the forces, i.e. the skin friction, must now be included in the objective function, resulting in the following shape optimization problem*

$$\min_{(u,p,\Omega)} J(u,p,\Omega) := \int_{\Gamma_0} \sum_{i=1}^{3} a_i p n_i - \sum_{i,k=1}^{3} a_k \mu^* \sigma_{ki} n_i \, dS = \int_{\Gamma_0} \langle pa, n\rangle - \langle \mu^* \sigma a, n\rangle \, dS \tag{9.6}$$

subject to

$$\begin{aligned} \sum_{k=1}^{3}\left(u_k \frac{\partial \rho}{\partial x_k} + \rho \frac{\partial u_k}{\partial x_k}\right) &= 0 \\ \left(\rho \sum_{k=1}^{3} u_k \frac{\partial u_i}{\partial x_k}\right) + (\gamma-1)\frac{\partial}{\partial x_i}(\rho T) - \sum_{k=1}^{3} \frac{\partial}{\partial x_k}(\mu^* \sigma_{ik}) &= 0 \qquad (9.7) \\ \left(\rho \sum_{k=1}^{3} u_k \frac{\partial T}{\partial x_k}\right) + (\gamma-1)\rho T \sum_{k=1}^{3} \frac{\partial u_k}{\partial x_k} - \sum_{j,k=1}^{3} \mu^* \frac{\partial u_k}{\partial x_j}\sigma_{jk} - \sum_{k=1}^{3} \frac{\partial}{\partial x_k}\left(\kappa^* \frac{\partial T}{\partial x_k}\right) &= 0. \end{aligned}$$

The equality in (9.6) *holds due to symmetry of σ. Additionally, the following boundary conditions are considered:*

$$\begin{aligned} u &= a && \text{on } \Gamma_+ && (9.8) \\ u &= 0 && \text{on } \Gamma_0 && (9.9) \\ T = T_\infty &= \frac{1}{\gamma(\gamma-1)M_\infty^2} && \text{on } \Gamma_+ && (9.10) \\ T = T_B &= T_\infty \left(1 + \frac{\gamma-1}{2M_\infty^2}\right) && \text{on } \Gamma_0 && (9.11) \\ \langle \nabla T, n\rangle &= 0 && \text{on } \Gamma_0 && (9.12) \\ \rho &= 1 && \text{on } \Gamma_+, && (9.13) \end{aligned}$$

where M_∞ is the free stream Mach number. Equation (9.8) *is the usual normalized inflow boundary condition in a rotated coordinate system, and equation* (9.9) *is the standard viscous no-slip* *boundary condition. Boundary condition* (9.11) *is the isothermal wall temperature condition, i.e. fixed Dirichlet boundary condition for the temperature on the wing. An alternative to the isothermal temperature wall boundary condition is the adiabatic wall temperature boundary condition* (9.12), *specifying zero temperature flux over the aircraft surface. Isothermal and adiabatic wall temperature boundary conditions are mutually exclusive and never used at the same time.*

Lemma 9.2.2 (Linearized Continuity Equation)**.** *Linearizing the continuity equation* (9.1) *results in the following expression*

$$
\begin{aligned}
0 = &\int\limits_{\Omega} \sum_{k=1}^{3} \left(\lambda_\rho u_k'[V] \frac{\partial \rho}{\partial x_k} - \rho'[V] \frac{\partial}{\partial x_k} \left(\lambda_\rho u_k \right) + \lambda_\rho \rho'[V] \frac{\partial u_k}{\partial x_k} - u_k'[V] \frac{\partial}{\partial x_k} (\lambda_\rho \rho) \right) \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{k=1}^{3} \left(\lambda_\rho u_k \rho'[V] n_k + \lambda_\rho \rho u_k'[V] n_k \right) \, dS,
\end{aligned}
$$

where λ_ρ is a sufficiently smooth arbitrary multiplicator.

Proof. A linearization of the continuity equation (9.1) results in

$$
\begin{aligned}
&\operatorname{div}\left(\rho u'[V]\right) + \operatorname{div}\left(\rho'[V] u\right) \\
&= \sum_{k=1}^{3} \left(u_k'[V] \frac{\partial \rho}{\partial x_k} + u_k \frac{\partial \rho'[V]}{\partial x_k} + \rho'[V] \frac{\partial u_k}{\partial x_k} + \rho \frac{\partial u_k'[V]}{\partial x_k} \right) = 0,
\end{aligned}
$$

and multiplication of the linearized continuity equation with an arbitrary λ_ρ and integration by parts results in the desired expression. □

Lemma 9.2.3 (Linearized Strain Tensor, Volume)**.** *For sufficiently smooth arbitrary multiplicators λ_{u_i}, the following relation holds*

$$
\begin{aligned}
&\int\limits_{\Omega} \sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} \sigma_{ik}'[V] \, dA \\
&= \int\limits_{\Omega} - \sum_{i,k=1}^{3} \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) u_i'[V] \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{i,k=1}^{3} \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) u_i'[V] \, dS.
\end{aligned}
$$

Proof. The definition of the strain tensor, equation (9.4), results in

$$\begin{aligned}
&\int\limits_{\Omega} \sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} \sigma'_{ik}[V]\, dA \\
&= \int\limits_{\Omega} \sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} \left(\frac{\partial u'_i[V]}{\partial x_k} + \frac{\partial u'_k[V]}{\partial x_i} - \frac{2}{3}\delta_{ik} \sum_{m=1}^{3} \frac{\partial u'_m[V]}{\partial x_m} \right) dA \\
&= \int\limits_{\Omega} \left[\sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} \left(\frac{\partial u'_i[V]}{\partial x_k} + \frac{\partial u'_k[V]}{\partial x_i} \right) \right] - \frac{2}{3} \sum_{i,m=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_i} \frac{\partial u'_m[V]}{\partial x_m}\, dA.
\end{aligned}$$

Integration by parts and regrouping provides

$$\begin{aligned}
&\int\limits_{\Omega} \sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} \sigma'_{ik}[V]\, dA \\
&= \int\limits_{\Omega} \left[-\sum_{i,k=1}^{3} \frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} u'_i[V] + \frac{\partial^2 \lambda_{u_i}}{\partial x_k \partial x_i} u'_k[V] \right] + \frac{2}{3} \sum_{i,m=1}^{3} \frac{\partial^2 \lambda_{u_i}}{\partial x_i \partial x_m} u'_m[V]\, dA \\
&+ \int\limits_{\partial\Omega} \left[\sum_{i,k=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_k} n_k u'_i[V] + \frac{\partial \lambda_{u_i}}{\partial x_k} n_i u'_k[V] \right] - \frac{2}{3} \sum_{i,m=1}^{3} \frac{\partial \lambda_{u_i}}{\partial x_i} n_m u'_m[V]\, dS \\
&= \int\limits_{\Omega} \left[\sum_{i,k=1}^{3} \left(-\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} - \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} \right) u'_i[V] \right] + \frac{2}{3} \sum_{i,k=1}^{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} u'_i[V]\, dA \\
&+ \int\limits_{\partial\Omega} \left[\sum_{i,k=1}^{3} \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k \right) u'_i[V] \right] - \frac{2}{3} \sum_{i,k=1}^{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i u'_i[V]\, dS \\
&= \int\limits_{\Omega} -\sum_{i,k=1}^{3} \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) u'_i[V]\, dA \\
&+ \int\limits_{\partial\Omega} \sum_{i,k=1}^{3} \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) u'_i[V]\, dS.
\end{aligned}$$

□

Lemma 9.2.4 (Linearized Strain Tensor, Surface)**.** *For sufficiently smooth arbitrary multiplicators λ_{u_i}, the following relation holds*

$$\begin{aligned}
&\int\limits_{\partial\Omega} \sum_{i,k=1}^{3} \lambda_{u_i} \sigma'_{ik}[V] n_k\, dS \\
&= \int\limits_{\partial\Omega} \sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i - \frac{2}{3}\delta_{ik} \sum_{m=1}^{3} \lambda_{u_m} n_m \right) \frac{\partial u'_i[V]}{\partial x_k}\, dS.
\end{aligned}$$

Proof. Inserting equation (9.4) and regrouping results in

$$
\begin{aligned}
&\int_{\partial\Omega} \sum_{i,k=1}^{3} \lambda_{u_i} \sigma'_{ik}[V] n_k \, dS \\
&= \int_{\partial\Omega} \left[\sum_{i,k=1}^{3} \left(\lambda_{u_i} \frac{\partial u'_i[V]}{\partial x_k} n_k + \lambda_{u_k} \frac{\partial u'_i[V]}{\partial x_k} n_i \right) \right] - \frac{2}{3} \sum_{i,k,m=1}^{3} \delta_{ik} \lambda_{u_i} \frac{\partial u'_m[V]}{\partial x_m} n_k \, dS \\
&= \int_{\partial\Omega} \left[\sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i \right) \frac{\partial u'_i[V]}{\partial x_k} \right] - \frac{2}{3} \sum_{i,m=1}^{3} \lambda_{u_m} n_m \frac{\partial u'_i[V]}{\partial x_i} \, dS \\
&= \int_{\partial\Omega} \left[\sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i \right) \frac{\partial u'_i[V]}{\partial x_k} \right] - \frac{2}{3} \sum_{i,k,m=1}^{3} \delta_{ik} \lambda_{u_m} n_m \frac{\partial u'_i[V]}{\partial x_k} \, dS \\
&= \int_{\partial\Omega} \sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i - \frac{2}{3} \delta_{ik} \sum_{m=1}^{3} \lambda_{u_m} n_m \right) \frac{\partial u'_i[V]}{\partial x_k} \, dS.
\end{aligned}
$$

□

Lemma 9.2.5 (Linearized Momentum Equation)**.** *Linearizing the momentum equation* (9.2) *under the frozen viscosity assumption results in the following expression*

$$
\begin{aligned}
0 = &\int_{\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} u_k \frac{\partial u_i}{\partial x_k} \right) + (\gamma - 1) \left(\lambda_{u_i} \frac{\partial T}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} T) \right) \right] \rho'[V] \, dA \\
+ &\int_{\Omega} \sum_{i,k=1}^{3} \left[-\mu^* \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) + \lambda_{u_k} \rho \frac{\partial u_k}{\partial x_i} - \frac{\partial}{\partial x_k} (\lambda_{u_i} \rho u_k) \right] u'_i[V] \, dA \\
+ &\int_{\Omega} \sum_{i=1}^{3} (\gamma - 1) \left[\lambda_{u_i} \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\lambda_{u_i} \rho \right) \right] T'[V] \, dA \\
+ &\int_{\partial\Omega} \sum_{i,k=1}^{3} \left[\mu^* \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) + \lambda_{u_i} \rho u_k n_k \right] u'_i[V] \, dS \\
+ &\int_{\partial\Omega} \sum_{i=1}^{3} (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \, dS \\
+ &\int_{\partial\Omega} -\mu^* \sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i - \frac{2}{3} \delta_{ik} \sum_{m=1}^{3} \lambda_{u_m} n_m \right) \frac{\partial u'_i[V]}{\partial x_k} \, dS,
\end{aligned}
$$

where λ_{u_i} are sufficiently smooth arbitrary multiplicators.

Proof. The linearized momentum equation (9.2) is given by

$$
\begin{aligned}
0 = &\sum_{k=1}^{3} \left(\rho'[V] u_k \frac{\partial u_i}{\partial x_k} + \rho u_k'[V] \frac{\partial u_i}{\partial x_k} + \rho u_k \frac{\partial u_i'[V]}{\partial x_k} \right) \\
&+ (\gamma - 1) \left(\frac{\partial \rho'[V]}{\partial x_i} T + \frac{\partial \rho}{\partial x_i} T'[V] + \rho'[V] \frac{\partial T}{\partial x_i} + \rho \frac{\partial T'[V]}{\partial x_i} \right) - \sum_{k=1}^{3} \mu^* \frac{\partial}{\partial x_k} \sigma_{ik}'[V].
\end{aligned}
$$

A multiplication of the momentum equation with λ_{u_i} for $i = 1, 2, 3$ and integration by parts provides

$$
\begin{aligned}
0 = &\int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_{u_i} \rho'[V] u_k \frac{\partial u_i}{\partial x_k} + \lambda_{u_i} \rho u_k'[V] \frac{\partial u_i}{\partial x_k} - u_i'[V] \frac{\partial}{\partial x_k} \left(\lambda_{u_i} \rho u_k \right) \right] \, dA \\
&+ \int\limits_{\Omega} (\gamma - 1) \left[-\rho'[V] \frac{\partial}{\partial x_i} (\lambda_{u_i} T) + \lambda_{u_i} \frac{\partial \rho}{\partial x_i} T'[V] + \lambda_{u_i} \rho'[V] \frac{\partial T}{\partial x_i} - T'[V] \frac{\partial}{\partial x_i} (\lambda_{u_i} \rho) \right] \, dA \\
&+ \int\limits_{\Omega} \sum_{k=1}^{3} \mu^* \sigma_{ik}'[V] \frac{\partial \lambda_{u_i}}{\partial x_k} \, dA \\
&+ \int\limits_{\partial\Omega} \left(\sum_{k=1}^{3} \lambda_{u_i} \rho u_k u_i'[V] n_k \right) + (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \, dS \\
&+ \int\limits_{\partial\Omega} - \sum_{k=1}^{3} \mu^* \lambda_{u_i} \sigma_{ik}'[V] n_k \, dS.
\end{aligned}
$$

Summing the momentum equations over $i = 1, 2, 3$ and regrouping leads to

$$
0 = \int\limits_{\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} u_k \frac{\partial u_i}{\partial x_k} \right) + (\gamma - 1) \left(\lambda_{u_i} \frac{\partial T}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} T) \right) \right] \rho'[V] \, dA \tag{9.14}
$$

$$
+ \int\limits_{\Omega} \sum_{i,k=1}^{3} \lambda_{u_i} \rho \frac{\partial u_i}{\partial x_k} u_k'[V] \, dA \tag{9.15}
$$

$$
+ \int\limits_{\Omega} \sum_{i,k=1}^{3} \left[-\frac{\partial}{\partial x_k} \left(\lambda_{u_i} \rho u_k \right) \right] u_i'[V] \, dA \tag{9.16}
$$

$$
+ \int\limits_{\Omega} \sum_{i=1}^{3} (\gamma - 1) \left(\lambda_{u_i} \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} \rho) \right) T'[V] \, dA \tag{9.17}
$$

$$
+ \int\limits_{\Omega} \sum_{i,k=1}^{3} \mu^* \sigma_{ik}'[V] \frac{\partial \lambda_{u_i}}{\partial x_k} \, dA \tag{9.18}
$$

$$
+ \int\limits_{\partial\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} \rho u_k n_k \right) u_i'[V] + (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \right] \, dS \tag{9.19}
$$

$$+\int_{\partial\Omega} -\sum_{i,k=1}^{3} \mu^* \lambda_{u_i} \sigma'_{ik}[V] n_k \, dS. \tag{9.20}$$

Very similar to the incompressible case, there is again a summation over two different indices i and k in (9.15) and (9.16) for the local shape derivatives of the velocity. In the incompressible case, equations (6.15) and (6.16), this has been dealt with by another integration by parts, which is convenient there, since the resulting additional terms are removed due to div $u'[V] = 0$. Since this is no longer true in the compressible case, the second integration by parts is omitted. Exchanging indices in equation (9.15) and regrouping results in

$$\begin{aligned}
0 =& \int_{\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} u_k \frac{\partial u_i}{\partial x_k} \right) + (\gamma - 1) \left(\lambda_{u_i} \frac{\partial T}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} T) \right) \right] \rho'[V] \, dA \\
&+ \int_{\Omega} \sum_{i,k=1}^{3} \left[\lambda_{u_k} \rho \frac{\partial u_k}{\partial x_i} - \frac{\partial}{\partial x_k} (\lambda_{u_i} \rho u_k) \right] u'_i[V] \, dA \\
&+ \int_{\Omega} \sum_{i=1}^{3} (\gamma - 1) \left[\lambda_{u_i} \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} \rho) \right] T'[V] \, dA \\
&+ \int_{\Omega} \sum_{i,k=1}^{3} \mu^* \sigma'_{ik}[V] \frac{\partial \lambda_{u_i}}{\partial x_k} \, dA \\
&+ \int_{\partial\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} \rho u_k n_k \right) u'_i[V] + (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \right] dS \\
&+ \int_{\partial\Omega} -\sum_{i,k=1}^{3} \mu^* \lambda_{u_i} \sigma'_{ik}[V] n_k \, dS.
\end{aligned}$$

Contrary to the Euler case, the linearized strain tensor $\sigma'[V]$ contains another set of derivatives, such that an additional integration by parts is needed. Applying lemma 9.2.3 provides

$$\begin{aligned}
0 =& \int_{\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} u_k \frac{\partial u_i}{\partial x_k} \right) + (\gamma - 1) \left(\lambda_{u_i} \frac{\partial T}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} T) \right) \right] \rho'[V] \, dA \\
&+ \int_{\Omega} \sum_{i,k=1}^{3} \left[-\mu^* \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) + \lambda_{u_k} \rho \frac{\partial u_k}{\partial x_i} - \frac{\partial}{\partial x_k} (\lambda_{u_i} \rho u_k) \right] u'_i[V] \, dA \\
&+ \int_{\Omega} \sum_{i=1}^{3} (\gamma - 1) \left[\lambda_{u_i} \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\lambda_{u_i} \rho \right) \right] T'[V] \, dA \\
&+ \int_{\partial\Omega} \sum_{i,k=1}^{3} \left[\mu^* \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) + \lambda_{u_i} \rho u_k n_k \right] u'_i[V] \, dS
\end{aligned}$$

$$
\begin{aligned}
&+ \int\limits_{\partial\Omega} \sum_{i=1}^{3} (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \, dS \\
&+ \int\limits_{\partial\Omega} - \sum_{i,k=1}^{3} \mu^* \lambda_{u_i} \sigma'_{ik}[V] n_k \, dS.
\end{aligned}
$$

It remains to remove the strain tensor variation on the boundary $\partial\Omega$ from the first integration by parts. Applying lemma 9.2.4 results in

$$
\begin{aligned}
0 = &\int\limits_{\Omega} \sum_{i=1}^{3} \left[\left(\sum_{k=1}^{3} \lambda_{u_i} u_k \frac{\partial u_i}{\partial x_k} \right) + (\gamma - 1) \left(\lambda_{u_i} \frac{\partial T}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_{u_i} T) \right) \right] \rho'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{i,k=1}^{3} \left[-\mu^* \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) + \lambda_{u_k} \rho \frac{\partial u_k}{\partial x_i} - \frac{\partial}{\partial x_k} (\lambda_{u_i} \rho u_k) \right] u_i'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{i=1}^{3} (\gamma - 1) \left[\lambda_{u_i} \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\lambda_{u_i} \rho \right) \right] T'[V] \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{i,k=1}^{3} \left[\mu^* \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) + \lambda_{u_i} \rho u_k n_k \right] u_i'[V] \, dS \\
&+ \int\limits_{\partial\Omega} \sum_{i=1}^{3} (\gamma - 1) \left(\lambda_{u_i} \rho'[V] T n_i + \lambda_{u_i} \rho T'[V] n_i \right) \, dS \\
&+ \int\limits_{\partial\Omega} -\mu^* \sum_{i,k=1}^{3} \left(\lambda_{u_i} n_k + \lambda_{u_k} n_i - \frac{2}{3} \delta_{ik} \sum_{m=1}^{3} \lambda_{u_m} n_m \right) \frac{\partial u_i'[V]}{\partial x_k} \, dS,
\end{aligned}
$$

creating the desired expression. □

Lemma 9.2.6 (Linearized Energy Equation)**.** *Linearizing the energy equation* (9.3) *under the frozen viscosity assumption results in the following expression*

$$
\begin{aligned}
0 = &\int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_T u_k \frac{\partial T}{\partial x_k} + (\gamma - 1) \lambda_T T \frac{\partial u_k}{\partial x_k} \right] \rho'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_T \rho \frac{\partial T}{\partial x_k} - (\gamma - 1) \frac{\partial}{\partial x_k} (\lambda_T \rho T) \right] u_k'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{j,k=1}^{3} \mu^* \left[\frac{\partial}{\partial x_j} (\lambda_T \sigma_{jk}) + \frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_j}{\partial x_k} \right) + \frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_k}{\partial x_j} \right) - \frac{2}{3} \frac{\partial}{\partial x_k} \left(\lambda_T \frac{\partial u_j}{\partial x_j} \right) \right] u_k'[V] \, dA \\
&+ \int\limits_{\Omega} \left[-\kappa^* \Delta \lambda_T + \sum_{k=1}^{3} \left((\gamma - 1) \lambda_T \rho \frac{\partial u_k}{\partial x_k} - \frac{\partial}{\partial x_k} (\lambda_T \rho u_k) \right) \right] T'[V] \, dA
\end{aligned}
$$

$$+\int_{\partial\Omega}\sum_{k=1}^{3}(\gamma-1)\lambda_T\rho T n_k u_k'[V]\,dS$$

$$+\int_{\partial\Omega}-\mu^*\sum_{j,k=1}^{3}\left(\lambda_T\sigma_{jk}n_j+\lambda_T\frac{\partial u_j}{\partial x_k}n_j+\lambda_T\frac{\partial u_k}{\partial x_j}n_j-\frac{2}{3}\lambda_T\frac{\partial u_j}{\partial x_j}n_j\right)u_k'[V]\,dS$$

$$+\int_{\partial\Omega}\left[\kappa^*\frac{\partial\lambda_T}{\partial n}+\sum_{k=1}^{3}\lambda_T\rho u_k n_k\right]T'[V]-\kappa^*\lambda_T\frac{\partial T'[V]}{\partial n}\,dS,$$

where λ_T is a sufficiently smooth arbitrary multiplicator.

Proof. The linearization of the energy equation (9.3) is given by

$$\begin{aligned}0=&\left[\sum_{k=1}^{3}\left(\rho'[V]u_k\frac{\partial T}{\partial x_k}+\rho u_k'[V]\frac{\partial T}{\partial x_k}+\rho u_k\frac{\partial T'[V]}{\partial x_k}\right)\right]\\&+(\gamma-1)\left[\sum_{k=1}^{3}\left(\rho'[V]T\frac{\partial u_k}{\partial x_k}+\rho T'[V]\frac{\partial u_k}{\partial x_k}+\rho T\frac{\partial u_k'[V]}{\partial x_k}\right)\right]\\&-\left[\sum_{j,k=1}^{3}\left(\mu^*\frac{\partial u_k'[V]}{\partial x_j}\sigma_{jk}+\mu^*\frac{\partial u_k}{\partial x_j}\sigma_{jk}'[V]\right)\right]-\kappa^*\Delta T'[V].\end{aligned}$$

Multiplication by λ_T, integration by parts, and Green's second identity lead to

$$\begin{aligned}0=&\int_{\Omega}\sum_{k=1}^{3}\left[\lambda_T\rho'[V]u_k\frac{\partial T}{\partial x_k}+\lambda_T\rho u_k'[V]\frac{\partial T}{\partial x_k}-T'[V]\frac{\partial}{\partial x_k}(\lambda_T\rho u_k)\right]dA\\&+\int_{\Omega}(\gamma-1)\sum_{k=1}^{3}\left[\lambda_T\rho'[V]T\frac{\partial u_k}{\partial x_k}+\lambda_T\rho T'[V]\frac{\partial u_k}{\partial x_k}-u_k'[V]\frac{\partial}{\partial x_k}(\lambda_T\rho T)\right]dA\\&+\int_{\Omega}\left[\sum_{j,k=1}^{3}\left(\mu^*u_k'[V]\frac{\partial}{\partial x_j}(\lambda_T\sigma_{jk})-\mu^*\lambda_T\frac{\partial u_k}{\partial x_j}\sigma_{jk}'[V]\right)\right]-\kappa^*T'[V]\Delta\lambda_T\,dA\\&+\int_{\partial\Omega}\left(\sum_{k=1}^{3}\lambda_T\rho u_k T'[V]n_k\right)+(\gamma-1)\left(\sum_{k=1}^{3}\lambda_T\rho Tu_k'[V]n_k\right)-\sum_{j,k=1}^{3}\mu^*\lambda_T u_k'[V]\sigma_{jk}n_j\,dS\\&+\int_{\partial\Omega}\kappa^*\left(T'[V]\frac{\partial\lambda_T}{\partial n}-\lambda_T\frac{\partial T'[V]}{\partial n}\right)dS=0.\end{aligned}$$

Regrouping results in

$$0=\int_{\Omega}\sum_{k=1}^{3}\left[\lambda_T u_k\frac{\partial T}{\partial x_k}+(\gamma-1)\lambda_T T\frac{\partial u_k}{\partial x_k}\right]\rho'[V]\,dA$$

$$
\begin{aligned}
&+ \int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_T \rho \frac{\partial T}{\partial x_k} - (\gamma - 1) \frac{\partial}{\partial x_k} (\lambda_T \rho T) + \left(\sum_{j=1}^{3} \mu^* \frac{\partial}{\partial x_j} (\lambda_T \sigma_{jk}) \right) \right] u_k'[V] \, dA \\
&+ \int\limits_{\Omega} \left[-\kappa^* \Delta \lambda_T + \sum_{k=1}^{3} \left((\gamma - 1) \lambda_T \rho \frac{\partial u_k}{\partial x_k} - \frac{\partial}{\partial x_k} (\lambda_T \rho u_k) \right) \right] T'[V] \, dA \\
&+ \int\limits_{\Omega} - \sum_{j,k=1}^{3} \mu^* \lambda_T \frac{\partial u_k}{\partial x_j} \sigma_{jk}'[V] \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{k=1}^{3} \left[(\gamma - 1) \lambda_T \rho T n_k - \mu^* \sum_{j=1}^{3} \lambda_T \sigma_{jk} n_j \right] u_k'[V] \, dS \\
&+ \int\limits_{\partial\Omega} \left[\kappa^* \frac{\partial \lambda_T}{\partial n} + \sum_{k=1}^{3} \lambda_T \rho u_k n_k \right] T'[V] - \kappa^* \lambda_T \frac{\partial T'[V]}{\partial n} \, dS.
\end{aligned}
$$

Since the strain tensor linearization still contains derivatives, lemma 9.2.3 must be used again, providing

$$
\begin{aligned}
&\int\limits_{\Omega} \sum_{j,k=1}^{3} \lambda_T \frac{\partial u_k}{\partial x_j} \sigma_{jk}'[V] \, dA \\
= &\int\limits_{\Omega} \sum_{j,k=1}^{3} \left[-\frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_j}{\partial x_k} \right) - \frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_k}{\partial x_j} \right) + \frac{2}{3} \frac{\partial}{\partial x_k} \left(\lambda_T \frac{\partial u_j}{\partial x_j} \right) \right] u_k'[V] \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{j,k=1}^{3} \left[\lambda_T \frac{\partial u_j}{\partial x_k} n_j + \lambda_T \frac{\partial u_k}{\partial x_j} n_j - \frac{2}{3} \lambda_T \frac{\partial u_j}{\partial x_j} n_j \right] u_k'[V] \, dS.
\end{aligned}
$$

Thus, the linearized energy equation becomes after regrouping

$$
\begin{aligned}
0 = &\int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_T u_k \frac{\partial T}{\partial x_k} + (\gamma - 1) \lambda_T T \frac{\partial u_k}{\partial x_k} \right] \rho'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{k=1}^{3} \left[\lambda_T \rho \frac{\partial T}{\partial x_k} - (\gamma - 1) \frac{\partial}{\partial x_k} (\lambda_T \rho T) \right] u_k'[V] \, dA \\
&+ \int\limits_{\Omega} \sum_{j,k=1}^{3} \mu^* \left[\frac{\partial}{\partial x_j} (\lambda_T \sigma_{jk}) + \frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_j}{\partial x_k} \right) + \frac{\partial}{\partial x_j} \left(\lambda_T \frac{\partial u_k}{\partial x_j} \right) - \frac{2}{3} \frac{\partial}{\partial x_k} \left(\lambda_T \frac{\partial u_j}{\partial x_j} \right) \right] u_k'[V] \, dA \\
&+ \int\limits_{\Omega} \left[-\kappa^* \Delta \lambda_T + \sum_{k=1}^{3} \left((\gamma - 1) \lambda_T \rho \frac{\partial u_k}{\partial x_k} - \frac{\partial}{\partial x_k} (\lambda_T \rho u_k) \right) \right] T'[V] \, dA \\
&+ \int\limits_{\partial\Omega} \sum_{k=1}^{3} (\gamma - 1) \lambda_T \rho T n_k u_k'[V] \, dS
\end{aligned}
$$

$$+\int\limits_{\partial\Omega} -\mu^* \sum_{j,k=1}^{3} \left(\lambda_T\sigma_{jk}n_j + \lambda_T\frac{\partial u_j}{\partial x_k}n_j + \lambda_T\frac{\partial u_k}{\partial x_j}n_j - \frac{2}{3}\lambda_T\frac{\partial u_j}{\partial x_j}n_j\right) u_k'[V]\, dS$$

$$+\int\limits_{\partial\Omega}\left[\kappa^*\frac{\partial\lambda_T}{\partial n} + \sum_{k=1}^{3} \lambda_T\rho u_k n_k\right] T'[V] - \kappa^*\lambda_T\frac{\partial T'[V]}{\partial n}\, dS,$$

which is the desired expression. □

Lemma 9.2.7 (Shape Derivative for the Compressible Navier–Stokes Equations)**.** *The shape derivative of the fluid forces acting on an immersed obstacle is given by*

$$\begin{aligned}
&dJ(\rho, u, T, \Omega)[V] \\
&= \int\limits_{\Gamma_0} \langle V, n\rangle \left[\sum_{i=1}^{3}\left(a_i\frac{\partial p}{\partial n}n_i - \lambda_\rho\rho\frac{\partial u_i}{\partial n}n_i\right) - \sum_{i,k=1}^{3} a_i\mu^*\frac{\partial\sigma_{ik}}{\partial n}n_k\right] dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \operatorname{div}_\Gamma \left(pa - \mu^*\sigma a\right)\, dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \sum_{i,k=1}^{3}\left[-\mu^*\left(\frac{\partial\lambda_{u_i}}{\partial x_k}n_k + \frac{\partial\lambda_{u_k}}{\partial x_i}n_k - \frac{2}{3}\frac{\partial\lambda_{u_k}}{\partial x_k}n_i\right)\right]\frac{\partial u_i}{\partial n}\, dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle\kappa^*\frac{\partial\lambda_T}{\partial n}\frac{\partial(T_B - T)}{\partial n}\, dS
\end{aligned}$$

in case of isothermal walls, $T = T_B$*, and*

$$\begin{aligned}
&dJ(\rho, u, T, \Omega)[V] \\
&= \int\limits_{\Gamma_0} \langle V, n\rangle \left[\sum_{i=1}^{3}\left(a_i\frac{\partial p}{\partial n}n_i - \lambda_\rho\rho\frac{\partial u_i}{\partial n}n_i - \lambda_T p\frac{\partial u_i}{\partial n}n_i\right) - \sum_{i,k=1}^{3} a_i\mu^*\frac{\partial\sigma_{ik}}{\partial n}n_k\right] dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \operatorname{div}_\Gamma \left(pa - \mu^*\sigma a\right)\, dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \sum_{i,k=1}^{3}\left[-\mu^*\left(\frac{\partial\lambda_{u_i}}{\partial x_k}n_k + \frac{\partial\lambda_{u_k}}{\partial x_i}n_k - \frac{2}{3}\frac{\partial\lambda_{u_k}}{\partial x_k}n_i\right)\right]\frac{\partial u_i}{\partial n}\, dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \sum_{i,k=1}^{3}\mu^*\left(\lambda_T\sigma_{ik}n_i + \lambda_T\frac{\partial u_i}{\partial x_k}n_i + \lambda_T\frac{\partial u_k}{\partial x_i}n_i - \frac{2}{3}\lambda_T\frac{\partial u_i}{\partial x_i}n_i\right)\frac{\partial u_k}{\partial n}\, dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle\kappa^*\lambda_T\left[\frac{\partial^2 T}{\partial n^2} - \kappa\langle\nabla_\Gamma T, n\rangle + \Delta_\Gamma T\right] dS,
\end{aligned}$$

in case of adiabatic walls, $\frac{\partial T}{\partial n} = 0$*. Here,* κ^* *is the total thermal conductivity of the fluid, and* κ *is the curvature. In either case, the adjoint velocity viscous wall boundary condition is*

$$\lambda_{u_i} = -a_i \quad \text{on} \quad \Gamma_0,$$

where a is given by $a = (\cos\alpha, 0, \sin\alpha)^T$ with α being the angle of attack. For isothermal walls Γ_0, the adjoint temperature boundary condition is

$$\lambda_T = 0 \quad on \quad \Gamma_0,$$

and for adiabatic walls Γ_0 it is given by

$$\frac{\partial \lambda_T}{\partial n} = 0 \quad on \quad \Gamma_0.$$

For prescribed density, velocity, and pressure in the farfield, the adjoint farfield boundary conditions are given by

$$\begin{aligned} \lambda_\rho &= 0 \\ \lambda_{u_i} &= 0 \\ \lambda_T &= 0. \end{aligned}$$

Proof. The objective function is of the type scalar product of vector times normal, and a formal differentiation according to lemma 3.3.14 immediately results in

$$dJ(\rho, u, T, \Omega)[V] = \int\limits_{\Gamma_0} \langle V, n \rangle \left[\sum_{i=1}^{3} a_i \frac{\partial p}{\partial n} n_i - \sum_{i,k=1}^{3} a_i \mu^* \frac{\partial \sigma_{ik}}{\partial n} n_k \right] \, dS \tag{9.21}$$

$$+ \int\limits_{\Gamma_0} \langle V, n \rangle \operatorname{div}_\Gamma \left(pa - \mu^* \sigma a \right) \, dS \tag{9.22}$$

$$+ \int\limits_{\Gamma_0} \sum_{i=1}^{3} a_i p'[V] n_i - \sum_{i,k=1}^{3} a_i \mu^* \sigma'_{ik}[V] n_k \, dS. \tag{9.23}$$

Due to the frozen viscosity assumption and not including Sutherland's law, viscosity variations $\mu^{*\prime}[V]$ are not considered. Adjoint calculus is again needed to transform the last part of the above equation into Hadamard form. The results from lemmas 9.2.2, 9.2.5, and 9.2.6 are made such that the adjoint equations in Ω can easily be read from them. Thus, when $(\lambda_\rho, \lambda_{u_i}, \lambda_T)^T$ fulfills the following adjoint equations in Ω

$$
\begin{aligned}
0 =& \sum_{j=1}^{3} \left[-\frac{\partial}{\partial x_j} \left(\lambda_\rho u_j \right) + \lambda_\rho \frac{\partial u_j}{\partial x_j} + \left(\sum_{k=1}^{3} \lambda_{u_j} u_k \frac{\partial u_j}{\partial x_k} \right) \right.\\
&\left. +(\gamma - 1) \left(\lambda_{u_j} \frac{\partial T}{\partial x_j} - \frac{\partial}{\partial x_j} (\lambda_{u_j} T) \right) + \lambda_T u_j \frac{\partial T}{\partial x_j} + (\gamma - 1) \lambda_T T \frac{\partial u_j}{\partial x_j} \right]\\
0 =& \lambda_\rho \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_i} (\lambda_\rho \rho) + \lambda_T \rho \frac{\partial T}{\partial x_i} - (\gamma - 1) \frac{\partial}{\partial x_i} (\lambda_T \rho T)\\
&+ \sum_{k=1}^{3} \left[-\mu^* \left(\frac{\partial^2 \lambda_{u_i}}{\partial x_k^2} + \frac{\partial^2 \lambda_{u_k}}{\partial x_i \partial x_k} - \frac{2}{3} \frac{\partial^2 \lambda_{u_k}}{\partial x_k \partial x_i} \right) + \lambda_{u_k} \rho \frac{\partial u_k}{\partial x_i} - \frac{\partial}{\partial x_k} (\lambda_{u_i} \rho u_k) \right]\\
&+ \sum_{k=1}^{3} \mu^* \left[\frac{\partial}{\partial x_i} (\lambda_T \sigma_{ik}) + \frac{\partial}{\partial x_i} \left(\lambda_T \frac{\partial u_i}{\partial x_k} \right) + \frac{\partial}{\partial x_i} \left(\lambda_T \frac{\partial u_k}{\partial x_i} \right) - \frac{2}{3} \frac{\partial}{\partial x_k} \left(\lambda_T \frac{\partial u_i}{\partial x_i} \right) \right]\\
0 =& - \kappa^* \Delta \lambda_T + \sum_{j=1}^{3} \left[(\gamma - 1) \left(\lambda_T \rho \frac{\partial u_j}{\partial x_j} + \lambda_{u_j} \frac{\partial \rho}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\lambda_{u_j} \rho \right) \right) - \frac{\partial}{\partial x_j} (\lambda_T \rho u_j) \right],
\end{aligned}
$$

where $i = 1, 2, 3$, all volume integrals will vanish. Hence, only the remaining boundary parts have to be considered. For prescribed farfield density, velocity, and temperature, one has

$$
\begin{aligned}
\rho'[V] &= 0\\
u_i'[V] &= 0\\
T'[V] &= 0
\end{aligned}
$$

in the farfield $\Gamma_+ \cup \Gamma_-$. Consequently, the adjoint farfield boundary conditions can be read from lemmas 9.2.2, 9.2.5, and 9.2.6:

$$
\begin{aligned}
\lambda_\rho &= 0\\
\lambda_{u_i} &= 0\\
\lambda_T &= 0.
\end{aligned}
$$

It remains to consider variations on the solid wall Γ_0. According to equation (9.5), the pressure is given by

$$p = (\gamma - 1) \rho T.$$

Consequently, the pressure variation is given by

$$p'[V] = (\gamma - 1) \rho'[T] T + (\gamma - 1) \rho T'[V]. \tag{9.24}$$

Since there are many different ways of how flow solvers handle the farfield boundary conditions, the farfield adjoint boundary conditions will not be considered here. Instead, they are assumed to be implemented correctly in whatever adjoint flow solver is used. Merging the boundary integrals of lemmas 9.2.2, 9.2.5, and 9.2.6 with the preliminary gradient (9.21) – (9.23) and inserting the no-slip boundary condition $u = 0$ on Γ_0 together with the pressure variation (9.24) results in

$$
\begin{aligned}
&dJ(\rho, u, T, \Omega)[V] \\
&= \int_{\Gamma_0} \langle V, n\rangle \left[\sum_{i=1}^{3} a_i \frac{\partial p}{\partial n} n_i - \sum_{i,k=1}^{3} a_i \mu^* \frac{\partial \sigma_{ik}}{\partial n} n_k \right] dS \\
&+ \int_{\Gamma_0} \langle V, n\rangle \operatorname{div}_\Gamma \left(pa - \mu^* \sigma a\right) dS \\
&+ \int_{\Gamma_0} \sum_{i=1}^{3} a_i p'[V] n_i - \sum_{i,k=1}^{3} a_i \mu^* \sigma'_{ik}[V] n_k \, dS \\
&+ \int_{\Gamma_0} \sum_{k=1}^{3} \lambda_\rho \rho u'_k[V] n_k \, dS \\
&+ \int_{\Gamma_0} \sum_{i,k=1}^{3} \left[\mu^* \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) \right] u'_i[V] \, dS \\
&+ \int_{\Gamma_0} \sum_{i=1}^{3} \lambda_{u_i} n_i p'[V] - \sum_{i,k=1}^{3} \mu^* \lambda_{u_i} \sigma'_{ik}[V] n_k \, dS \\
&+ \int_{\Gamma_0} \sum_{k=1}^{3} (\gamma - 1) \lambda_T \rho T n_k u'_k[V] \, dS \\
&+ \int_{\Gamma_0} -\mu^* \sum_{i,k=1}^{3} \left(\lambda_T \sigma_{ik} n_i + \lambda_T \frac{\partial u_i}{\partial x_k} n_i + \lambda_T \frac{\partial u_k}{\partial x_i} n_i - \frac{2}{3} \lambda_T \frac{\partial u_i}{\partial x_i} n_i \right) u'_k[V] \, dS \\
&+ \int_{\Gamma_0} \kappa^* \frac{\partial \lambda_T}{\partial n} T'[V] - \kappa^* \lambda_T \frac{\partial T'[V]}{\partial n} \, dS.
\end{aligned}
$$

Note that for contributions from the momentum equations, the preliminary expression containing the strain tensor variation $\sigma'[V]$ was used, which is more convenient in this case due to the same expression already being present in the preliminary gradient. Also, just as in the incompressible case, it becomes obvious that arbitrary objective functions will not always lead to consistent adjoint boundary conditions. Here, due to equation (9.24), the objective function matches the state linearization in such a way that the pressure variation is created on the boundary, and the multiplicators λ_{u_i} can be chosen such that both the inviscid pressure contribution and the viscous contributions are eliminated at the same time by the boundary condition

$$\lambda_{u_i} = -a_i.$$

For badly chosen objective functions, there is the potential danger that density variations $\rho'[V]$ and velocity variations $u'[V]$ cannot be removed by the same multipliers. The no-slip boundary condition for the velocities is a Dirichlet boundary condition, and according to lemma 3.4.3 the velocity variation is given by

$$u_i'[V] = -\langle V, n\rangle \frac{\partial u_i}{\partial n}.$$

Taking everything together, the preliminary gradient becomes

$$\begin{aligned}
&dJ(\rho, u, T, \Omega)[V] \\
&= \int\limits_{\Gamma_0} \langle V, n\rangle \left[\sum_{i=1}^{3} \left(a_i \frac{\partial p}{\partial n} n_i - \lambda_\rho \rho \frac{\partial u_i}{\partial n} n_i - \lambda_T p \frac{\partial u_i}{\partial n} n_i \right) - \sum_{i,k=1}^{3} a_i \mu^* \frac{\partial \sigma_{ik}}{\partial n} n_k \right] dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \operatorname{div}_\Gamma \left(pa - \mu^* \sigma a \right) dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \sum_{i,k=1}^{3} \left[-\mu^* \left(\frac{\partial \lambda_{u_i}}{\partial x_k} n_k + \frac{\partial \lambda_{u_k}}{\partial x_i} n_k - \frac{2}{3} \frac{\partial \lambda_{u_k}}{\partial x_k} n_i \right) \right] \frac{\partial u_i}{\partial n} dS \\
&+ \int\limits_{\Gamma_0} \langle V, n\rangle \sum_{i,k=1}^{3} \mu^* \left(\lambda_T \sigma_{ik} n_i + \lambda_T \frac{\partial u_i}{\partial x_k} n_i + \lambda_T \frac{\partial u_k}{\partial x_i} n_i - \frac{2}{3} \lambda_T \frac{\partial u_i}{\partial x_i} n_i \right) \frac{\partial u_k}{\partial n} dS \\
&+ \int\limits_{\Gamma_0} \kappa^* \frac{\partial \lambda_T}{\partial n} T'[V] - \kappa^* \lambda_T \frac{\partial T'[V]}{\partial n} dS,
\end{aligned}$$

which almost fulfills the Hadamard form except for the temperature variations $T'[V]$. In case of an isothermal wall, the temperature is subject to a Dirichlet boundary condition

$$T = T_B \quad \text{on} \quad \Gamma_0.$$

Thus, lemma 3.4.3 applies again, resulting in

$$T'[V] = \langle V, n\rangle \frac{\partial (T_B - T)}{\partial n}.$$

Consequently, the boundary condition for λ_T is

$$\lambda_T = 0 \quad \text{on} \quad \Gamma_0.$$

In case of an adiabatic wall, the temperature is subject to a Neumann boundary condition

$$\frac{\partial T}{\partial n} = 0 \quad \text{on} \quad \Gamma_0,$$

and lemma 3.4.5 applies, resulting in

$$\int\limits_{\Gamma_0} \frac{\partial T'[V]}{\partial n} = \int\limits_{\Gamma_0} \langle V, n\rangle \left[-\frac{\partial^2 T}{\partial n^2} + \kappa \langle \nabla_\Gamma T, n\rangle - \Delta_\Gamma T \right] dS,$$

where κ is the curvature. Consequently, the adjoint boundary condition is given by

$$\frac{\partial \lambda_T}{\partial n} = 0,$$

which creates the desired expressions. □

Remark 9.2.8. The shape derivative for the drag reduction problem, lemma 9.2.7, can be adapted to lift instead of drag simply by replacing the incident vector *a* accordingly. Also, the shape derivative requires knowledge of the normal derivative of the strain tensor

$$\frac{\partial \sigma_{ik}}{\partial n},$$

which consists of velocity derivatives of second order. Accurately computing second order derivatives of flow unknowns can be problematic for certain finite volume schemes.

9.3 Primal and Adjoint Variables

The construction of the shape derivative in lemma 9.2.7 was conducted using the primal variables ρ, u, and T, and consequently, the corresponding adjoint variables are λ_ρ, λ_u, and λ_T. However, most flow solvers operate on the conserved variables ρ, ρu, and ρE, and when run in adjoint mode, it is not immediately clear how to use their respective adjoint variables λ_ρ, $\lambda_{\rho u}$, and $\lambda_{\rho E}$. However, since an adjoint variable corresponds to a primal equation, and not a primal variable, the set of adjoint variables is actually independent of the primal unknowns.

Lemma 9.3.1 (Primal and Adjoint Variables). *The shape derivative is independent of the choice of primal variables.*

Proof. Let the conserved and primitive variables in two dimensions be given by

$$\begin{aligned} U &= (\rho, \rho u, \rho E)^T \\ U_\mathrm{p} &= (\rho, u, p)^T \\ U_\mathrm{T} &= (\rho, u, T)^T. \end{aligned}$$

Furthermore, let M_1 and M_2 be given by

$$\begin{aligned} M_1 &:= \frac{\partial U_\mathrm{p}}{\partial U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{u_1}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{u_2}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{(\gamma-1)\|u\|^2}{2} & (1-\gamma)u_1 & (1-\gamma)u_2 & (\gamma-1) \end{bmatrix} \\ M_2 &:= \frac{\partial U_\mathrm{p}}{\partial U_\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ (\gamma-1)T & 0 & 0 & (\gamma-1)\rho \end{bmatrix}, \end{aligned}$$

such that the following relation holds

$$M_2 U_\mathrm{T}'[V] = M_1 U'[V] = U_\mathrm{p}'[V],$$

since

$$p'[V] = (\gamma-1)\rho'[V]T + (\gamma-1)\rho T'[V].$$

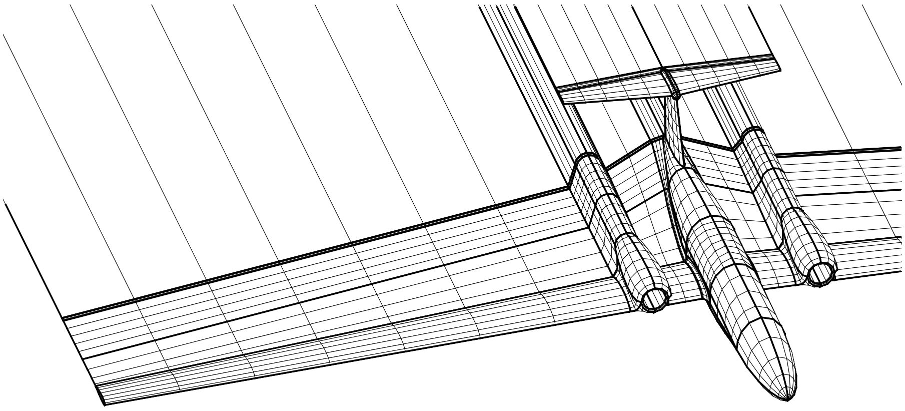
Let λ solve the formal adjoint system used in the shape derivative, lemma 9.2.7, based on U_T

$$\frac{\partial J}{\partial U_\mathrm{T}} + \lambda^T \frac{\partial c}{\partial U_\mathrm{T}} = 0.$$

Consequently, the adjoint system corresponding to a primal formulation in U is constructed by multiplication with $M_3 := \frac{\partial U_\mathrm{T}}{\partial U} = M_2^{-1} M_1$

$$\begin{aligned} \frac{\partial J}{\partial U_\mathrm{T}} M_3 + \lambda^T \frac{\partial c}{\partial U_\mathrm{T}} M_3 &= 0 \\ \Leftrightarrow \qquad \frac{\partial J}{\partial U} + \lambda^T \frac{\partial c}{\partial U} &= 0, \end{aligned}$$

which is equivalently solved by the same set of variables λ. □



Chapter 10

Conclusions and Outlook

10.1 Summary

The aim of this work was to conduct a shape sensitivity analysis followed by a study of the shape Hessian and an actual numerical optimization for a variety of large scale shape optimization problems in computational fluid dynamics. The Hadamard form of the shape gradient played a crucial part in numerical structure exploitation for applied shape optimization. Being able to express the gradient on the surface bypassed all of the problematic partial derivatives, i.e. mesh sensitivity Jacobians, of a shape optimization problem that is considered as a standard optimal control problem by parameterization. Since the one-shot approach achieves a major speed-up by trading few exact descent steps for many inexact ones, the discretized Hadamard gradient also greatly supported the one-shot approach. Furthermore, the gradient evaluation is truly independent of the number of design parameters, which made using all mesh surface nodes in the optimization possible, enabling large scale deformations and morphing of shapes.

When using a parameterization, the resulting shapes are usually smooth by construction, meaning that for any choice of design parameters, the resulting shapes are in a desired regularity class. Unfortunately, this is no longer true in case of a non-parametric shape optimization based on the Hadamard formula where one needs to make sure the updates maintain regularity of the shape. Interpreting the gradient as the Riesz representative of the derivative, one has to compute this representative in the desired regularity space. Sometimes also called Sobolev gradient method, such a gradient smoothing essentially turns a steepest descent method into an approximative Newton method. Thus, the shape Hessian of the problem is a very good candidate for the re-smoothing procedure. Fortunately, the Hadamard form of the shape gradient also allows a better analysis of the Hessian, which would otherwise be camouflaged by parameterization. The Hessian of a PDE constraint shape optimization problem usually is a pseudo-differential operator, and this work studied Fourier analysis to determine the symbol of the Hessian. A Hessian approximation based on its symbol could be used to greatly accelerate a wide variety of shape optimization problems in fluid dynamics, up to the adaptively refined multi-level optimization of a complete Onera M6 wing with more than $36,000$ design parameters using an unstructured surface mesh and the compressible Euler equations. Special attention was also given to the correct evaluation of the Hadamard form and the surface shape Hessian approximation, as this usually requires discrete differential geometry on unstructured, triangulated surfaces. The work concludes with a formal extension to the compressible Navier–Stokes equations using a frozen eddy viscosity approach.

10.2 Future Work

New possibilities created by using the Hadamard form of the shape gradient are very profound. Being able to conduct large scale morphing of shapes very effectively opens new and fascinating opportunities for optimization. Transition from the Onera M6 wing to a complete aircraft, i.e. wing-body-nacelle configurations, is straight forward. After defining new constraints, basically only the input mesh must be changed. Optimization under uncertainties and aero-structure coupling can also be envisioned. Preliminary studies for aero-structure interaction are considered in [60], where the application of new and very fast single instruction multiple data graphics processing units (GPUs) for numerical optimization was also studied.

Naturally, the formal considerations for the compressible Navier–Stokes equations should also be applied in an actual numerical optimization. However, for perfecting large scale morphing of shapes, more robust volume mesh deformations are needed. Since a re-meshing in the volume after most of the surface updates is infeasible, the procedure to match the volume mesh to the new surface must be able to create quality meshes for large deformations, even though the mesh deformation is no longer part of the derivative chain. With the present industrially applied mesh deformation tools, problems with bad quality and zero or negative volume cells are frequently encountered when the planform changes too dramatically. The sharp trailing edge also frequently creates troubles with volume mesh deformation. It can be expected that this behavior worsens when the anisotropic boundary layer cells are included in a viscous and compressible fluid such as the compressible Navier–Stokes equations. Automatic mesh refinement and boundary layer discretization is also problematic in this context.

Another open question is turbulence. While an analytic derivation for the RANS equations is possible around mean flow, i.e. frozen eddy viscosity, the shape derivation of most—if not all—turbulence models is rather problematic. Algebraic and one equation models usually have non-differentiable switchings in the volume, and two equation models usually have problematic boundary conditions, especially the k-ω model. In general, turbulence models often have discrete expressions in the viscous wall boundary conditions which is highly contra productive for the all analytic shape differentiation and the Hadamard theorem. In fact, some turbulent boundary conditions are only defined discretely. One remedy could be making automatic or algorithmic differentiation better aware of shape optimization problems. This probably requires more studies of the gap between discrete, i.e. parameterized, and analytic shape optimization.

Additionally, free surface flows could be considered. When designing ship hulls, surface wave patterns need to be considered, and the interaction between the free fluid surface and the shape of the immersed body can lead to a fascinating coupling. Including buoyancy also results in a fluid-structure interaction to be considered during optimization. Furthermore, shape optimization in general can be considered a free surface problem, thus coupled approaches with free interface problems can lead to considerable synergies.

Bibliography

[1] C. Amrouche, Š. Nečasová, and J. Sokolowski. Shape sensitivity analysis of the Neumann problem of the Laplace equation in the half-space. Technical Report preprint 2007-12-20, Institute of Mathematics, AS CR, Prague, 2007.

[2] E. Arian. Analysis of the Hessian for aeroelastic optimization. Technical Report 95-84, Institute for Computer Applications in Science and Engineering (ICASE), 1995.

[3] E. Arian and S. Ta'asan. Analysis of the Hessian for aerodynamic optimization: Inviscid flow. Technical Report 96-28, Institute for Computer Applications in Science and Engineering (ICASE), 1996.

[4] E. Arian and V. N. Vatsa. A preconditioning method for shape optimization governed by the Euler equations. Technical Report 98-14, Institute for Computer Applications in Science and Engineering (ICASE), 1998.

[5] K. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*. Texts in Applied Mathematics 39. Springer, second edition, 2007.

[6] M. Berggren. A unified discrete-continuous sensitivity analysis method for shape optimization. In review, 2008.

[7] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001.

[8] S. Boisgérault and J.-P. Zolésio. Shape derivative of sharp functionals governed by Navier–Stokes flow. In W. Jäger, J. Necas, O. John, K. Najzar, and J. Stará, editors, *Partial Differential Equations: Theory and Numerical Solution*, pages 49–63. Chapman & Hall/CRC Reseach Notes in Mathematics, 1993.

[9] S. Boivin. A finite element method to solve the compressible Navier–Stokes equations with turbulence modelling. Technical Report RR-0923, INRIA, 1988.

[10] C. Castro, C. Lozano, F. Palacios, and E. Zuazua. Systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA*, 45(9):2125–2139, 2007.

[11] M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM Philadelphia, 2001.

[12] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[13] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[14] G. Dziuk. Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations*, volume 1357/1988 of *Lecture Notes in Mathematics*, pages 142–155. Springer Berlin / Heidelberg, 1988.

[15] O. Enoksson. *Shape Optimization in Compressible Inviscid Flow*. Licentiate thesis, Linköpings Universitet, S-581 83 Linköping, Sweden, 2000.

[16] K. Eppler. Second derivatives and sufficient optimality conditions for shape functionals. *Control and Cybernetics*, 29:485–512, 2000.

[17] K. Eppler. *Efficient Shape Optimization Algorithms for Elliptic Boundary Value Problems*. Habilitation thesis, Chemnitz University of Technology, Chemnitz, Germany, 2007.

[18] K. Eppler and H. Harbrecht. A regularized Newton method in electrical impedance tomography using shape Hessian information. *Control and Cybernetics*, 34(1):203–225, 2005.

[19] K. Eppler, S. Schmidt, V. Schulz, and C. Ilic. Preconditioning the pressure tracking in fluid dynamics by shape Hessian information. *Journal of Optimization Theory and Applications*, 141(3):513–531, 2009.

[20] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 2002.

[21] M. Galle. Ein Verfahren zur numerischen Simulation kompressibler, reibungsbehafteter Strömungen auf hybriden Netzen. Technical Report DLR-FB 99-04, Deutsches Zentrum für Luft- und Raumfahrt e.V., 1999.

[22] R. V. Garimella and B. K. Swartz. Curvature estimation for unstructured triangulations of surfaces. Technical Report LA-UR-03-8240, Los Alamos National Laboratory, 2003.

[23] N. Gauger. *Das Adjungiertenverfahren in der aerodynamischen Formoptimierung*. PhD thesis, Braunschweig University of Technology, Braunschweig, Germany, 2003.

[24] N. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm. Automatic differentiation of an entire design chain for aerodynamic shape optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 96:454–461, 2007.

[25] N. Gauger, A. Walther, E. Özkaya, and C. Moldenhauer. Efficient aerodynamic shape optimization by structure exploitation. Technical Report Preprint SPP1253-15-05, TU Dresden, 2008. submitted (Optimization and Engineering).

[26] T. Gerhold. Overview of the hybrid RANS TAU-code. In N. Kroll and J. K. Fassbender, editors, *MEGAFLOW — Numerical Flow Simulation for Aircraft Design*, volume 89 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer, 2005.

[27] I. Gherman. *Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems*. PhD thesis, University of Trier, Trier, Germany, 2008.

[28] I. Gherman and V. Schulz. Preconditioning of one-shot pseudo-timestepping methods for shape optimization. *PAMM*, 5(1):741–742, 2005.

[29] M. B. Giles and N. A. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA*, 97-1850, 1997.

[30] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier–Stokes Equations*. Springer, 1986.

[31] A. Griewank. Projected Hessians for preconditioning in one-step one-shot design optimization. *Nonconvex Optimization and its Application*, 83:151–172, 2006.

[32] M.D. Gunzburger. *Perspectives in Flow Control and Optimization*. Advances in Design and Control. SIAM Philadelphia, 2003.

[33] W. Haack. Geschoßformen kleinsten Wellenwiderstandes. *Bericht der Lilienthal-Gesellschaft*, 136(1):14–28, 1941.

[34] S. B. Hazra and V. Schulz. Simultaneous pseudo-timestepping for PDE-model based optimization problems. *Bit Numerical Mathematics*, 44(3):457–472, 2004.

[35] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[36] M. Hintermüller and W. Ring. An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *Journal of Mathematical Imaging and Vision*, 20:19–42, 2004.

[37] M. Hintermüller and W. Ring. A second order shape optimization approach for image segmentation. *SIAM Journal on Applied Mathematics*, 64(2):442–467, 2004.

[38] K. Ito, K. Kunisch, G. Gunther, and H. Peichl. Variational approach to shape derivatives. *Control, Optimisation and Calculus of Variations*, 14:517–539, 2008.

[39] E. N. Jacobs, K. E. Ward, and R. Pinkerton. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Technical Report 460, NACA, 1933.

[40] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.

[41] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10:213–237, 1998.

[42] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In *AIAA 14th Fluid and Plasma Dynamic Conference*, 1981-1259, June 23-25 1981.

[43] J. Katz and A. Plotkin. *Low-Speed Aerodynamics — From Wing Theory to Panel Models, International Edition*. McGraw–Hill, 1991.

[44] R. J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[45] A. M. McIvor and R. J. Valkenburg. A comparison of local surface geometry estimation methods. *Machine Vision and Applications*, 10(1):17–26, 1997.

[46] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Numerical Mathematics and Scientific Computation. Clarendon Press Oxford, 2001.

[47] D. I. Papadimitriou and K. C. Giannakoglou. A continous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. *Computers & Fluids*, 36:325–341, 2007.

[48] L. D. López Pérez. *Régularisation d'images sur des surfaces non planes*. PhD thesis, Université de Nice–Sophia Antipolis, 2006.

[49] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973.

[50] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[51] P. I. Plotnikov, E. V. Ruban, and J. Sokolowski. Inhomogeneous boundary value problems for compressible Navier–Stokes equations: Well-posedness and sensitivity analysis. *Journal on Mathematical Analysis*, 40(3):1152–1200, 2008.

[52] P. I. Plotnikov and J. Sokolowski. On compactness, domain dependence and existence of steady state solutions to compressible isothermal Navier–Stokes equations. *Journal of Mathematical Fluid Mechanics*, 7(4):529–573, 2005.

[53] P. I. Plotnikov and J. Sokolowski. Domain dependence of solutions to compressible Navier–Stokes equations. *SIAM Journal on Control and Optimization*, 45(4):1165–1197, 2006.

[54] P. I. Plotnikov and J. Sokolowski. Stationary boundary value problems for compressible Navier–Stokes equations. *Handbook of Differential Equations*, 6:313–410, 2008.

[55] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. In *33rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 9–12 1995. AIAA Paper 95-0123.

[56] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, 2004.

[57] S. Schmidt, C. Ilic, N. Gauger, and V. Schulz. Shape gradients and their smoothness for practical aerodynamic design optimization. Technical Report Preprint SPP1253-10-03, DFG-SPP 1253, 2008. (submitted to OPTE).

[58] S. Schmidt, C. Ilic, V. Schulz, and N. Gauger. Fast non-parametric large scale aerodynamic shape optimization. In M. Heinkenschloss, R.H.W. Hoppe, and V. Schulz, editors, *Numerical Techniques for Optimization Problems with PDE Constraints*, number 4/2009 in Oberwolfach Reports, pages 69–71. Mathematisches Forschungszentrum Oberwolfach, 2009.

[59] S. Schmidt and V. Schulz. Pareto-curve continuation in multi-objective optimization. *Pacific Journal of Optimization*, 4(2):243–257, 2008.

[60] S. Schmidt and V. Schulz. A 2589 line topology optimization code written for the graphics card. Technical Report Preprint SPP1253-068, DFG-SPP 1253, 2009. (submitted to CVS).

[61] S. Schmidt and V. Schulz. Impulse response approximations of discrete shape Hessians with application in CFD. *SIAM Journal on Control and Optimization*, 48(4):2562–2580, 2009.

[62] S. Schmidt and V. Schulz. Shape derivatives for general objective functions and the incompressible Navier–Stokes equations. Technical Report Preprint SPP1253-10-05, DFG-SPP 1253, 2009. (submitted to Control and Cybernetics).

[63] R. Schneider. *Applications of the Discrete Adjoint Method in Computational Fluid Dynamics*. PhD thesis, The University of Leeds School of Computing, 2006.

[64] V. Schulz. A direct PRSQP method for path planning of satellite mounted robots. *Zeitschrift für angewandte Mathematik und Mechanik*, 76(S3):17–20, 1996.

[65] V. Schulz. *Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 1996.

[66] V. Schulz and I. Gherman. One-shot methods for aerodynamic shape optimization. In N. Kroll, D. Schwamborn, K. Becker, H. Rieger, and F. Thiele, editors, *MEGADESIGN and MegaOpt — Aerodynamic Simulation and Optimization in Aircraft Design*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2008.

[67] V. Schulz and C. Schillings. On the nature and treatment of uncertainties in aerodynamic design. Technical Report Trierer Forschungsberichte Nr. 07-3, University of Trier, 2007. submitted (AIAA Journal).

[68] D. Schwamborn and T. Gerhold. The DLR TAU-code: Recent application in research and industry. In P. Wessling, E. Oñate, and J. Périaux, editors, *ECCOMAS CFD 2006*, Egmond aan Zee, the Netherlands, September 5-8 2006.

[69] S. Shankaran and A. Jameson. Adjoint formulations for topology, shape and discrete optimization. In *45th AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, January 8–11 2007. AIAA Paper 2007-0055.

[70] J. Sokolowski and J.-P. Zolésio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer, 1992.

[71] M. Spivak. *Calculus on Manifolds*. W. A. Benjamin, Inc., New York, Amsterdam, 1965.

[72] S. Ta'asan. Pseudo-time methods for constrained optimization problems governed by PDE. Technical Report 95-32, Institute for Computer Applications in Science and Engineering (ICASE), 1995.

[73] S. Ta'asan, G. Kuruvila, and M.D. Salas. Aerodynamic design and optimization in one shot. In *30th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 92-0025*, 1992.

[74] T. Tiihonen. Shape optimization and trial methods for free boundary problems. *RAIRO Model. Math. Anal. Numér*, 31, 1996.

[75] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*. Vieweg+Teubner Verlag, 2005.

[76] R. Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO Anal. Numér.*, 18:175–182, 1984.

[77] P. Weinerfeld. Aerodynamic optimization using control theory and surface mesh points as control variables. Technical Report FAU-97.044, SAAB Aerospace, Linköping, 1997.

[78] P. Weinerfeld. Alternative gradient formulation for aerodynamic shape optimization based on the Euler equations. Technical Report FF-2001-0042, SAAB Aerospace, Linköping, 2001.

[79] P. Weinerfeld. Gradient formulation for aerodynamic shape optimization based on the Navier–Stokes equations. Technical Report FF-2001-0043, SAAB Aerospace, Linköping, 2001.

[80] P. Weinerfeld. Gradient formulations for aerodynamic shape optimization based on Euler equations. Technical Report FF-2001-0041, SAAB Aerospace, Linköping, 2001.

[81] P. Weinerfeld. Some theorems related to the variation of metric terms and integrals. Technical Report FF-2001-0040, SAAB Aerospace, Linköping, 2001.

[82] P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer, 2000.