Universität Trier

Optimal Aerodynamic Design under Uncertainties

Dissertation

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

Dem Fachbereich IV der Universität Trier vorgelegt von

Dipl.-Math. Claudia Schillings

Eingereicht am 21. Dezember 2010

Gutachter: Prof. Dr. Volker H. Schulz
Prof. Dr. Hermann G. Matthies

Abstract

Recently, optimization has become an integral part of the aerodynamic design process chain. However, because of uncertainties with respect to the flight conditions and geometrical uncertainties, a design optimized by a traditional design optimization method seeking only optimality may not achieve its expected performance. Robust optimization deals with optimal designs, which are robust with respect to small (or even large) perturbations of the optimization setpoint conditions. The resulting optimization tasks become much more complex than the usual single setpoint case, so that efficient and fast algorithms need to be developed in order to identify, quantize and include the uncertainties in the overall optimization procedure. In this thesis, a novel approach towards stochastic distributed aleatory uncertainties for the specific application of optimal aerodynamic design under uncertainties is presented.

In order to include the uncertainties in the optimization, robust formulations of the general aerodynamic design optimization problem based on probabilistic models of the uncertainties are discussed. Three classes of formulations, the worst-case, the chance-constrained and the semi-infinite formulation, of the aerodynamic shape optimization problem are identified. Since the worst-case formulation may lead to overly conservative designs, the focus of this thesis is on the chance-constrained and semi-infinite formulation. A key issue is then to propagate the input uncertainties through the systems to obtain statistics of quantities of interest, which are used as a measure of robustness in both robust counterparts of the deterministic optimization problem. Due to the highly nonlinear underlying design problem, uncertainty quantification methods are used in order to approximate and consequently simplify the problem to a solvable optimization task. Computationally demanding evaluations of high dimensional integrals resulting from the direct approximation of statistics as well as from uncertainty quantification approximations arise. To overcome the curse of dimensionality, sparse grid methods in combination with adaptive refinement strategies are applied. The reduction of the number of discretization points is an important issue in the context of robust design, since the computational effort of the numerical quadrature comes up in every iteration of the optimization algorithm. In order to efficiently solve the resulting optimization problems, algorithmic approaches based on multiple-setpoint ideas in combination with one-shot methods are presented. A parallelization approach is provided to overcome the amount of additional computational effort involved by multiple-setpoint optimization problems.

Finally, the developed methods are applied to 2D and 3D Euler and Navier-Stokes test cases verifying their industrial usability and reliability. Numerical results of robust aerodynamic shape optimization under uncertain flight conditions as well as geometrical uncertainties are presented. Further, uncertainty quantification methods are used to investigate the influence of geometrical uncertainties on quantities of interest in a 3D test case. The results demonstrate the significant effect of uncertainties in the context of aerodynamic design and thus the need for robust design to ensure a good performance in real life conditions. The thesis proposes a general framework for robust aerodynamic design attacking the additional computational complexity of the treatment of uncertainties, thus making robust design in this sense possible.

Zusammenfassung

Die numerische Strömungssimulation und Optimierung hat sich heutzutage als unverzichtbares Werkzeug für die Flugzeugentwicklung etabliert. Aufgrund von Unsicherheiten in den Flugbedingungen sowie der Flugzeuggeometrie wird jedoch die Ausnutzung des Potenzials numerischer Optimierungsverfahren bislang begrenzt. Um dennoch eine gute Performance der optimierten Entwürfe zu erzielen, werden robust optimale Entwürfe betrachtet, die auch bei kleinen Störungen von Eingangsparametern und Auslegungspunkten noch sehr gute Designs darstellen. Die resultierenden, hoch komplexen stochastischen Fragestellungen erfordern die Entwicklung von schnellen und leistungsfähigen Methoden, mit denen die auftretenden Unsicherheiten im numerischen Entwurf identifiziert, quantifiziert und in den Optimierungsalgorithmus einbezogen werden. Gegenstand dieser Arbeit ist die Entwicklung einer neuen und effizienten Optimierungsmethodik zur Behandlung von stochastisch verteilten, aleatorischen Unsicherheiten in der aerodynamischen Formoptimierung.

Um die inhärenten Unsicherheiten in ein behandelbares Optimierungskonzept zu integrieren, werden verschiedene robuste Modellierungskonzepte, basierend auf einer stochastischen Charakterisierung der unsicheren Größen, diskutiert: die worst-case, chance-constrained und semi-infinite Formulierung. Aufgrund der zu konservativen Robustheitsbewertung des worst-case Modells liegt der Fokus dieser Arbeit auf der chance-constrained und semi-infiniten Formulierung. Ein zentraler Punkt stellt die Analyse des Einflusses der Eingangsunsicherheiten auf relevante Zielgrößen zur Berechnung von Statistiken dar, die in beiden Formulierungen zur Bewertung der Robustheit des aerodynamischen Entwurfs dienen. Die hochgradig nichtlinearen Zusammenhänge werden mittels Methoden zur Quantifizierung von Unsicherheiten approximiert. Die direkte Berechnung von Statistiken sowie die Methoden zur Quantifizierung von Unsicherheiten erfordern die rechenaufwändige Auswertung hochdimensionaler Integrale. Hierzu werden Sparse Grid Techniken in Kombination mit adaptiven Verfeinerungstechniken eingesetzt, die eine effiziente Diskretisierung des Wahrscheinlichkeitsraumes gewährleisten. Die Reduktion der benötigten Diskretisierungspunkte stellt einen wichtigen Aspekt dar, da der Rechenaufwand in jeder Iteration der Optimierung anfällt und somit maßgeblich die Performance des Algorithmus beeinflusst. Zur Lösung der diskretisierten, robusten Optimierungsprobleme werden Verfahren, basierend auf Mehrzieloptimierungskonzepten und problemangepassten, parallelisierten One-shot Ansätzen, entwickelt und implementiert.

Zum Abschluss werden die industrielle Anwendbarkeit und Zuverlässigkeit der entwickelten Methoden anhand von 2D und 3D Euler und Navier-Stokes Testfällen nachgewiesen. Numerische Resultate der robusten Optimierung unter unsicheren Flugbedingungen sowie Geometrieunsicherheiten werden vorgestellt und hinsichtlich ihrer Performance mit deterministisch optimierten Profilen verglichen. Desweiteren wird der Einfluss von Geometrieunsicherheiten auf relevante Strömungsgrößen mittels Methoden der Quantifizierung von Unsicherheiten in einem 3D Testfall untersucht. Die Ergebnisse zeigen den signifikanten Effekt von Unsicherheiten und verdeutlichen zugleich die Notwendigkeit der robusten Optimierung in der aerodynamischen Formoptimierung. Die Arbeit stellt eine allgemeine, effiziente Methodik zur Behandlung von Unsicherheiten im aerodynamischen Entwurf bereit, so dass die Berechnung von robust optimalen Designs ermöglicht wird.

Acknowledgments

This work evolved during my PhD studies as a research assistant on the project MUNA at the University of Trier, Department of Mathematics, in the research group of Prof. Dr. Volker H. Schulz. I could not have succeeded in finishing this work without the guidance and support of many people I would like to thank at this point.

Foremost, I would like to thank my advisor Prof. Dr. Volker H. Schulz for his excellent support and for providing such a creative and stimulating atmosphere within his research group. Numerous inspiring discussions and his valuable suggestions as well as the opportunities to participate in various international conferences were decisive in completing the thesis. I also would like to thank Prof. Dr. Hermann G. Matthies for accepting the position as second referee.

Moreover, I would like to express my gratitude to the partners in the project MUNA, particularly the German Federal Ministry of Economics and Technology for the financial support of the project. In addition, I am indebted to the researchers at the German Aerospace Center in Braunschweig, especially to Dr. Bernhard Eisfeld, Prof. Dr. Nicolas Gauger, Caslav Ilic, Dr. Jochen Wild and Markus Widhalm for their excellent cooperation and help. Furthermore, I wish to thank Dr. Alexander Litvinenko for many fruitful discussions.

To all my colleagues at the Department of Mathematics I owe thanks for the cooperative and enjoyable working conditions. Deserving of particular mention are the coworkers in the research group of Prof. Dr. Volker H. Schulz, especially Dr. Stephan Schmidt, for numerous discussions and exchanging ideas on the topics of my research.

Additionally, I would like to thank Andre Lörx for his support and encouragement over the past years. Without his active part the completion of this thesis would not have been possible.

Finally and naturally, I would like to acknowledge my parents Anneliese and Wolfgang for their moral and financial support of my academic career and to thank them and my sister Bettina, her husband Michael and my brother Volker for their constant encouragement.

Contents

Chapter 1

Introduction

Recently, aerodynamic shape optimization is a very active research field facing the challenges from highly demanding computational fluid dynamics problems, from optimization with partial differential equations as constraints as well as from the proper treatment of uncertainties. Due to increasing computing power and advancing algorithms, the numerical flow simulation has reached a highly sophisticated level and is therefore established as an indispensable tool for the development of new and for the improvement of existing airplanes. Further, the introduction of gradient computation via adjoint approach allows to efficiently evaluate derivatives needed in gradient-based optimization (cf. [54, 60, 61, 139, 149]). Besides standard optimization routines, which on average require 20-40 flow simulations, fast optimization methods based on one-shot ideas are available, too (see e.g. [58, 63, 164, 172]). They have the potential to reduce the overall costs of the optimization to just a few flow simulations. The one-shot method is based on approximate reduced SQP iterations solving the necessary optimality conditions simultaneously. However, a deterministic approach ignores the fact that there are stochastic and unknown variations in the problem. The conditions of operation are not known exactly, i.e. variations of the macroscopic flight conditions like the Mach number or the angle of attack may occur. The geometry itself undergoes unknown operational changes due to wear and tear and manufacturing inaccuracies. All these deviations from assumed setpoints of the deterministic optimization may render the supposedly optimal solution worthless, since their conclusions are not realized in practice. The proper treatment of these uncertainties within a numerical context is a very important challenge. This thesis is devoted to the enhancement of highly efficient optimal design techniques by a robustness component, which tries to make the optimal design generated a still good design, if the setting of a specific design point is varied.

In order to formulate the robust design optimization problem, the boundary conditions and input parameters are analyzed to identify the uncertainties, which cannot be avoided at all before constructing an aircraft. We will focus on shape optimization problems influenced by aleatory uncertainties, which arise because of natural, unpredictable variations of the boundary conditions. Definitions and classification of errors and uncertainties in the aerodynamic framework can be found in [5, 134]. Additional knowledge cannot reduce aleatory uncertainties, but it may be useful in getting a better characterization of the variability. In this thesis, a stochastic approach in order to include the uncertainties in the optimization problem is chosen avoiding a parametrization of the uncertainties, which would lead to a reduction of the space of realizations. A key issue is then to propagate the input uncertainties through the systems to obtain statistics of quantities of interest, which are used as a measure of robustness in the optimization. Uncertainty quantification in the context of computational fluid dynamics is a fast growing area of research. Due to the high complexity of the underlying deterministic problems, efficient methods need to be developed to make the methods industrially usable. A detailed overview of existing methods in uncertainty quantification with applications in CFD is given in [101]. Further details on the state of the art on uncertainty quantification in CFD can be found in [44, 115, 116, 121]. Using these techniques in a robust optimization framework, one should be aware that statistics are needed in every iteration of the optimization algorithm; that means the computational effort propagating the input uncertainties through the system repeatedly comes up in every iteration. Hence, the objective of this thesis is to provide efficient methods reducing the additional computational complexity of the treatment of uncertainties, thus making robust design in this sense possible. Since the resulting robust optimization tasks become much more complex than the usual single-setpoint case, most of the techniques developed so far pertain to problems with a low degree of nonlinearity (cf. [80, 152]). In the aerodynamic framework, the first papers towards robust optimal design are based on a multipoint optimization approach in order to achieve a design, which shows a good drag performance for several given realizations of the uncertain parameter, see e.g. [187]. In more recent papers (cf. [76, 110]), stochastic information of the input uncertainties is taken into account resulting in a stochastic optimization problem. Most of the papers so far restrict the discussion to unconstrained shape optimization problems influenced by scalar-valued uncertainties (e.g. [76, 109, 136]). The proper treatment of additional constraints in the optimization problem poses a difficult task from the numerical point of view. There are two main strategies to involve constraints with uncertain input data: the chance-constrained formulation and the semi-infinite formulation. Considering chance-constrained formulations, the constraint is required to hold with a certain probability. To check the feasibility of a chance-constraint, the evaluation of integrals, which are analytically tractable only for some special cases, arises. On the other hand, the semi-infinite formulation aims at maintaining the feasibility of the constraint for all realizations of the random input data resulting in infinitely many constraints in the continuous case. In this thesis, we will investigate both formulations and suggest approximations to overcome the difficulties. Further, a numerical comparison of the resulting robust solutions considering a 2D industrial test case will be presented. Beside the scalar-valued uncertainties caused by operational uncertainties, function-valued uncertainties are present in the shape optimization problem due to geometrical uncertainties. The real shape may vary from the planned shape due to manufacturing tolerances, temporary factors like e.g. icing or fatigue of material. Since there are so many factors having effects on the shape, this uncertainty has to be considered in the optimization problem in order to produce a design, which is robust to small perturbations of the shape itself. In literature, there can be found only few papers on this topic investigating the influence of variations of the profile (cf. [65, 115, 116]). So far, the geometrical uncertainties are modeled by variations of characteristic quantities describing the shape, e.g. the camber and thickness. However, the parametrization of the geometrical uncertainties leads to a reduction of the space of realizations. This method does not allow to properly model arbitrary input uncertainties given e.g. by measurements. To overcome this drawback, a stochastic approach describing the uncertainties by a random field is suggested in this thesis. The stochastic model allows to adapt the robust optimization to new information of the uncertain parameter, e.g. if new measurements are available, so that a general framework of robust aerodynamic design is provided. To reduce the additional computational effort resulting from the stochastic model, efficient approximation and discretization techniques need to be elaborated.

This thesis gives insight into significant enhancements of robust gradient-based shape optimization methods aiming at removing the deficits of state of the art methods outlined above. The investigations presented here are part of the research effort MUNA.

1.1 The MUNA project

The project MUNA (2007 - 2010) was initiated as part of the current German Aeronautical Research Program LuFo IV to introduce uncertainties within the computational fluid dynamics simulation process. The acronym stands for Management and Minimization of Uncertainties in Numerical Aerodynamics. It is lead-managed by the DLR Institute of Aerodynamics and Flow Technology in Braunschweig with the participation of Airbus Germany, EADS-MAS, Eurocopter and eight university institutes, among these the University of Trier. Within the former national initiatives MEGAFLOW [97] and MEGADESIGN [98], the simulation tools FLOWer and TAU as well as solvers for the corresponding adjoint equations were developed and introduced into the aerodynamic design process. Building on the highly sophisticated tools and methods developed in these projects, the project MUNA is working on management and minimization of CFD uncertainties. The main objectives of the project are:

- Identification and quantification of uncertainty sources
- Analysis of sensitivities and computation of confidence intervals
- Development of methods and strategies to minimize the main uncertainties and errors
- Verification of the developed methods and demonstration of their reliability.

The University of Trier focuses on the proper treatment of irreducible uncertainties arising in the formulation of aerodynamic design tasks. More precisely, the aim is to compute optimal designs, which are robust with respect to small (or even large) perturbations of the optimization setpoint conditions. Existing simulation and optimization methods will be improved, so that numerical uncertainties are identified, quantized and included in the overall optimization procedure, thus making robust design in this sense possible.

1.2 Outline of this thesis

The thesis is structured as follows:

Chapter 2: Background on Aerodynamic Design This chapter gives an introduction to fluid dynamics in order to derive the governing equations. Basic properties of the solvers, which are used to solve the flow equations, are presented focusing mainly on the use within an optimization procedure. Further, the deterministic shape optimization problem is formulated serving as a basis for the robust formulations.

Chapter 3: The Nature of Uncertainties in Aerodynamic Design The first part of the chapter is devoted to the identification and classification of uncertainties in the aerodynamic framework. After a short overview of the main stochastic models in order to mathematically describe uncertain parameters, probabilistic models of the identified uncertainties are introduced.

Chapter 4: Robust Formulations of Aerodynamic Design Problems The proper formulation of objective functions and constraints with respect to the uncertainties is discussed in chapter 4. Since the term robustness is not clearly defined in literature, the chapter contains a brief overview of research activities on robust optimization. Three classes of robust formulations, the worst-case, the chance-constrained and the semi-infinite formulation, of the aerodynamic shape optimization problem are identified. Since the worst-case formulation may lead to overly conservative designs, the focus of this thesis is on the chance-constrained and semi-infinite formulation.

Chapter 5: Uncertainty Quantification Uncertainty quantification methods, which determine the effect of uncertainties in the input data on quantities of interest in the output of a simulation, can be used to efficiently compute statistics measuring the robustness of a design within an optimization. Several uncertainty quantification methods are discussed in this chapter and a general framework to approximate the statistics of the quantities of interest is suggested.

Chapter 6: Discretization of the Probability Space The computation of statistics of quantities depending on the current design vector and the uncertain parameters require the evaluation of multi-dimensional integrals, which cannot be solved analytically. In chapter 6, efficient discretization methods based on sparse grids in combination with adaptive strategies in order to approximate the resulting integrals are discussed.

Chapter 7: Fundamentals of Optimization Optimality conditions as well as basic algorithms with focus on methods used in the aerodynamic shape optimization framework, especially the one-shot method, can be found in chapter 7. As the discretization of the semi-infinite formulation results in a multiple-setpoint problem, the basic one-shot method is adapted to this multiple-setpoint case and a generalized version of the one-shot approach is suggested. Optimizing the chance-constrained formulation, gradients with respect to the design variables cannot be evaluated by the use of the available adjoint solvers, so that a gradient-free algorithm has to be considered additionally.

Chapter 8: Aerodynamic Applications and Numerical Results The first part of this chapter is devoted to the numerical comparison of the semi-infinite and the chance-constrained formulation considering the optimization of a 2D transonic profile in Euler flow under scalar-valued uncertainties. Further, the influence of different measures of robustness of the objective function and the interaction of two scalar-valued input uncertainties are investigated on the basis of the 2D test case. In a 2D Euler and Navier-Stokes test case, numerical results of robust aerodynamic shape optimization under geometrical uncertainties are presented. The stochastic model describing the geometrical uncertainties is approximated by a goal-oriented uncertainty quantification method to reduce the computational complexity. Two adaptive refinement strategies of sparse grids discretizing the underlying probability space are compared in order to provide an efficient way of computing statistics within the optimization procedure. At last, the influence of geometrical uncertainties in a 3D industrial test case is quantized by expanding the quantities of interest in a series of orthogonal polynomials by a non-intrusive polynomial chaos method. The surrogate function is then used to determine statistics of the solution like the mean or variance.

Chapter 9: Conclusions and Outlook The thesis concludes with a review of the developed methods and gives an outlook to extensions and further work on robust aerodynamic optimization.

Chapter 2

Background on Aerodynamic Design

The first part of this chapter will give a survey of the fundamentals in optimal aerodynamic design. We start with a short introduction to fluid mechanics in order to derive the flow equations, Euler and Navier-Stokes. The derivation of the equations is based on conservation laws, which we will shortly discuss. In order to solve the flow equations, the solvers FLOWer and TAU developed at the DLR are used for the prediction of viscous and inviscous flows around complex geometries from the low subsonic to the hypersonic flow regime. Some basic properties will be introduced, focusing mainly on the use within an optimization procedure.

The second part is devoted to the shape optimization problem. The aim of shape optimization in aerodynamics is to reduce the forces, which negatively affect the performance of an airplane. Proper objective functions and constraints will be defined in order to formulate the single-setpoint aerodynamic shape optimization problem serving as a basis for the robust formulations.

2.1 Flow equations

Fluid dynamics is a field of science, which examines the physical laws governing the flow of fluids under various conditions. The studies go back at least to the days of ancient Greece whereas the history of Computational Fluid dynamics (CFD) did not start until the early 1970's with the availability of computers combined with the development of efficient numerical methods for solving the flow equations on such computers. CFD is concerned with the numerical solution of differential equations describing moving fluids. Today, CFD finds extensive usage in basic and applied research, in design of engineering equipment and in calculation of environmental and geophysical phenomena [34]. Several CFD applications can be found in [36]. The development of highly sophisticated numerical software and high speed computers within the last 30 years enables the simulation of more and more complex simulation scenarios such as 3D test cases, transonic and turbulent flows. Nowadays, experiments performed using a wind tunnel can often be replaced by CFD simulations due to the improvements concerning the accuracy and speed of the simulations. Further, CFD allows to simulate scenarios, which cannot be realized in a wind tunnel, and provides observations of flow properties in the whole domain, which is not the case in wind tunnel experiments. Another important advantage of CFD in design optimization is the fact that changes in geometry can easily be included whereas in experiments a new model is required. Hence, the application of CFD in design processes can significantly reduce costs and time consumptions. But CFD is still an aid to other analysis and experimental tools like wind tunnel testing and is used in conjunction with them in design processes. An important open issue is still the real time computation, which has not yet been reached despite the increasing speed of computation available. An introduction to CFD can be found e.g. in [6, 19, 34, 47, 48].

CFD is fundamentally based on the governing equations of fluid dynamics. They represent the mathematical statements of the conservation laws of physics. We will now introduce physical basics in order to derive the flow equations.

The dynamical behavior of a fluid is determined by the following conservations laws:

1. the conservation of mass
2. the conservation of momentum (Newton's second law)
3. the conservation of energy.

Further, we assume that the fluid obeys the continuum hypothesis, which considers fluids to be continuous. In other words, the discrete molecular structure of the fluid is ignored.

Remark 2.1.1 (Hypothesis of Continuum)**.** Given a domain $\Omega \in \mathbb{R}^d,\ d = 2, 3$, there exists a well defined mass density $\rho(x, t), \forall x \in \Omega, t > 0$, such that the total mass $m(\Omega, t)$ in the domain Ω at time t is given by

$$m(\Omega, t) = \int_\Omega \rho(x, t)\,\mathrm{d}V\,. \tag{2.1}$$

In the following, we denote by W a subdomain of Ω, ∂W the boundary of W, n the unit normal vector and by $\mathrm{d}A$ the area element on $\partial\Omega$.

Lemma 2.1.2 (Conservation of Mass)**.** *The conservation of mass implies that the mass of a closed system will remain constant over time, which means the change of mass in a fixed subdomain W is equal to the flow of mass over the boundary ∂W into W:*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_W \rho\,\mathrm{d}V = -\int_{\partial W}\langle \rho v, n\rangle\,\mathrm{d}A\,, \tag{2.2}$$

where v is the velocity. This is equivalent to

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho v) = 0\,. \tag{2.3}$$

Proof. The proof can be found e.g. in [19]. □

Using Newton's second law, which implies that the change of momentum is equal to the sum of all active forces, we obtain the following equation.

Theorem 2.1.3 (Conservation of Momentum)**.** *Considering the volume forces*

$$\int_W \rho(x, t) g(x, t)\,\mathrm{d}V$$

with $g \in \mathbb{R}^3$ given volume forces and surface forces

$$\int_{\partial W} \langle \tau^s(x, t), n\rangle\,\mathrm{d}A\,,$$

where $\tau^s \in \mathbb{R}^{3\times3}$, $\tau^s = -p^{pr} I + \sigma^f$ *is the stress tensor describing the internal friction* $\sigma^f \in \mathbb{R}^{3\times3}$ *and pressure* $p^{pr} \in \mathbb{R}$*, one obtains the relation*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_W \rho v \, \mathrm{d}V = \int_W \rho g \, \mathrm{d}V + \int_W \operatorname{div} \tau^s \, \mathrm{d}V \tag{2.4}$$

with $(\operatorname{div} \tau^s)_i := \sum_{j=1}^3 \frac{\partial \tau^s_{ij}}{\partial x_j}$, $i = 1, 2, 3$.

Using the Reynolds transport theorem (cf. [19]), the conservation of momentum can be rewritten as

$$\frac{\partial}{\partial t}(\rho v_i) + \operatorname{div}(\rho v_i v) = \rho g_i + (div \tau^s)_i \,, \quad i = 1, 2, 3. \tag{2.5}$$

For the so called Newtonian fluids, to which air belongs, the viscosity σ^f *can be expressed in the viscosity coefficients* λ^{visc} *(volume viscosity) and* μ^{visc} *(shear viscosity)*

$$\sigma^f = \lambda^{visc}(\operatorname{div} v) I + 2\mu^{visc} D \,, \tag{2.6}$$

where $D = D_{ij} \in \mathbb{R}^{3\times3}$, $D_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$ *denotes the deformation tensor.*

Proof. The proof can be found e.g. in [19]. □

The last conservation law concerns the conservation of energy.

Theorem 2.1.4 (Conservation of Energy). *The change of energy is equal to the work performed by external forces plus heat supply*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_W \rho E \, \mathrm{d}V = \int_W \langle \rho g, v\rangle \, \mathrm{d}V + \int_{\partial W} \langle \kappa \nabla T, n\rangle \, \mathrm{d}A + \int_{\partial W} \langle \tau^s v, n\rangle \, \mathrm{d}A \tag{2.7}$$

with $E = \frac{\|v\|^2}{2} + e$ *total energy,* e *internal energy per unit of mass,* T *temperature and* κ *the heat conductivity. Applying the Reynolds transport theorem and the divergence theorem [19], the conservation of energy in integral form can be stated as*

$$\int_W \frac{\partial}{\partial t}(\rho E) + \operatorname{div}(\rho E v) \, \mathrm{d}V = \int_W (\langle \rho g, v\rangle - \operatorname{div}(-\kappa \nabla T - \tau^s v)) \, \mathrm{d}V \tag{2.8}$$

or in differential conservative form

$$\frac{\partial}{\partial t}(\rho E) = \langle \rho g, v\rangle - div(\rho \mathcal{H} v - \kappa \nabla T - \sigma^f v) \,, \tag{2.9}$$

where $\mathcal{H} = E + \frac{p^{pr}}{\rho}$ *is the enthalpy.*

Proof. The proof can be found e.g. in [19]. □

The conservation laws give five equations for the unknowns ρ, $v = (v_1, v_2, v_3)$, E, p^{pr}, T. In order to obtain a closed system, additional assumptions have to be made eliminating the pressure p^{pr} and the temperature T.

Remark 2.1.5 (Ideal Gas)**.** The remaining equations are algebraic material relations linking density, pressure, temperature and internal energy. We assume dealing with an ideal gas

- $$p^{pr} = R\rho T\,, \tag{2.10}$$

  where R is the gas constant. From thermodynamics, the gas constant can be related to the specific heats for a calorically perfect gas and a thermally perfect gas

  $$R = c_p - c_v\,, \tag{2.11}$$

  where c_p is the specific heat for a constant pressure and c_v is the specific heat for a constant volume [6].

- $$e = c_v T + const \tag{2.12}$$

  implies that the specific heat should be constant, if the volume is constant.

Equation (2.12) gives

$$p^{pr} = \frac{R}{c_v}\rho\left(E - \frac{\|v\|^2}{2}\right)\,,$$

which can be rewritten as

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

using the adiabatic exponent $\gamma = \frac{c_p}{c_v}$. Further, equation (2.10) eliminates the temperature from the unknowns

$$T = \frac{p^{pr}}{R\rho}\,.$$

The above additional assumptions and conservation equations give five equations for five unknown ρ, $v = (v_1, v_2, v_3)$, E, so that we can now introduce the Euler and Navier-Stokes equations.

2.1.1 Euler equations

If viscous effects due to internal friction are negligible, the Euler equations are used to describe the flow.

Definition 2.1.6 (Euler Equations)**.** *The behavior of a compressible, inviscid flow on the domain Ω is described by the Euler equations:*

$$\begin{aligned}
\frac{\partial\rho}{\partial t} + \mathrm{div}\,(\rho v) &= 0 && (2.13)\\
\frac{\partial}{\partial t}(\rho v_i) + \mathrm{div}\,(\rho v_i v) + \frac{\partial p^{pr}}{\partial x_i} &= \rho g_i\,, \quad i = 1,2,3 && (2.14)\\
\frac{\partial}{\partial t}(\rho E) + \mathrm{div}\,(\rho\mathcal{H} v) &= \langle\rho g, v\rangle\,. && (2.15)
\end{aligned}$$

A system of five equations (in the 3D case) is obtained. The first one results from the conservation of mass, the second to the fourth from the conservation of momentum and the last one from the *conservation of energy neglecting the viscous terms, that means the stress tensor reduces to $\tau^s = -p^{pr} I$. Equations (2.13-2.15) can be rewritten in conservative form as*

$$\frac{\partial y}{\partial t} + \sum_{j=1}^{3} \frac{\partial F_j}{\partial x_j} = \tilde{g} \tag{2.16}$$

with the vector of state variables, also referred to as conservative variables,

$$y = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \end{pmatrix},$$

the convective flux vectors

$$F_j = \begin{pmatrix} \rho v_j \\ \rho v_j v_1 + \delta_{j1} p^{pr} \\ \rho v_j v_2 + \delta_{j2} p^{pr} \\ \rho v_j v_3 + \delta_{j3} p^{pr} \\ \rho \mathcal{H} v_j \end{pmatrix}, \qquad j = 1, 2, 3$$

and the right hand side

$$\tilde{g} = \begin{pmatrix} 0 \\ \rho g_1 \\ \rho g_2 \\ \rho g_3 \\ \langle \rho g, v \rangle \end{pmatrix}.$$

Equation (2.16) can be reformulated using the primitive variables given by the vector

$$y_{primitive} = \begin{pmatrix} \rho \\ v_1 \\ v_2 \\ v_3 \\ E \end{pmatrix},$$

see e.g. [54].

Remark 2.1.7 (Boundary Conditions)**.** In order to obtain a well posed problem, additional conditions on the boundary of the computational domain Ω are needed. On the surface Γ of the considered aircraft in our application, the so called Euler slip condition is required

$$\langle v, n \rangle = 0, \qquad on\ \Gamma. \tag{2.17}$$

Considering inviscid flows, the fluid slips over the surface. Since there is no friction force, the velocity vector has to be tangent to the surface, which is equivalent to equation (2.17). As the flow around the airfoil is simulated on a bounded domain, additional boundary conditions, referred to as farfield conditions, have to be introduced ensuring two basic requirements: The truncation of the domain should have no notable effect on the solution compared to the infinite domain. No outgoing disturbances of the flow should be reflected back into the flow field. In [6, 19], a survey of different boundary conditions is given.

2.1.2 Navier-Stokes equations

If the effects of frictions are not negligible, the flow is described by the Navier-Stokes equations. Since we consider Newtonian fluids, the viscous stresses can be written as

$$\sigma^f_{ij} = \mu^{visc}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \delta_{ij}\lambda^{visc}\sum_{k=1}^{3}\frac{\partial v_k}{\partial x_k},$$

as already stated in equation (2.6). Invoking the Stokes hypothesis

$$\lambda^{visc} = -\frac{2}{3}\mu^{visc},$$

one obtains the following system of partial differential equations.

Definition 2.1.8 (Navier-Stokes Equations)**.** *The compressible Navier-Stokes equation describing a compressible, viscous flow in Ω are given by*

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho v) &= 0 && (2.18)\\ \frac{\partial}{\partial t}(\rho v_i) + \sum_{j=1}^{3}\left(\frac{\partial}{\partial x_j}\left(-\mu^{visc}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\operatorname{div} v\right)\right)\right) + \operatorname{div}(\rho v_i v) + \frac{\partial p^{pr}}{\partial x_i} &= \rho g_i, && (2.19)\\ & \quad i = 1,2,3 &&\\ \frac{\partial}{\partial t}(\rho E) + \operatorname{div}\left(\rho\mathcal{H}v - \kappa\nabla T - \mu^{visc}\left(\nabla v + (\nabla v)^\top\right)v + \frac{2}{3}\mu^{visc}(v \operatorname{div} v)\right) &= \langle \rho g, v\rangle && (2.20)\end{aligned}$$

resulting from the conservation of mass, the conservation of momentum and the conservation of energy including the viscous terms. Equations (2.18-2.20) can be rewritten as

$$\frac{\partial y}{\partial t} + \sum_{j=1}^{3}\frac{\partial F_j}{\partial x_j} = \sum_{j=1}^{3}\frac{\partial F^v_j}{\partial x_j} + \tilde{g} \qquad (2.21)$$

with the vector of state variables, also referred to as conservative variables,

$$y = \begin{pmatrix}\rho\\ \rho v_1\\ \rho v_2\\ \rho v_3\\ \rho E\end{pmatrix},$$

the convective flux vectors

$$F_j = \begin{pmatrix}\rho v_j\\ \rho v_j v_1 + \delta_{j1}p^{pr}\\ \rho v_j v_2 + \delta_{j2}p^{pr}\\ \rho v_j v_3 + \delta_{j3}p^{pr}\\ \rho\mathcal{H}v_j\end{pmatrix}, \qquad j = 1,2,3$$

and the right hand side of (2.18-2.20)

$$\tilde{g} = \begin{pmatrix} 0 \\ \rho g_1 \\ \rho g_2 \\ \rho g_3 \\ \langle \rho g, v \rangle \end{pmatrix}.$$

The left hand side of equation (2.21) is identical to the left hand side of equation (2.16), the only difference are the viscous terms F_j^v on the right hand side, which are neglected in the Euler equations:

$$F_j^v = \begin{pmatrix} 0 \\ \mu^{visc}\left(\frac{\partial v_j}{\partial x_1} + \frac{\partial v_1}{\partial x_j} - \frac{2}{3}\delta_{j1}(\text{div } v)\right) \\ \mu^{visc}\left(\frac{\partial v_j}{\partial x_2} + \frac{\partial v_2}{\partial x_j} - \frac{2}{3}\delta_{j2}(\text{div } v)\right) \\ \mu^{visc}\left(\frac{\partial v_j}{\partial x_3} + \frac{\partial v_3}{\partial x_j} - \frac{2}{3}\delta_{j3}(\text{div } v)\right) \\ \sum_{i=1}^{3} v_i \mu^{visc}\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} - \frac{2}{3}\delta_{ji}(\text{div } v)\right) + \kappa \frac{\partial T}{\partial x_j} \end{pmatrix}, \qquad j = 1, 2, 3.$$

Remark 2.1.9 (Boundary conditions)**.** As in the previous section, we will shortly discuss appropriate boundary conditions. The relative velocity between the surface of the airfoil and the fluid directly at the surface is assumed to be zero for viscous fluids

$$v_1 = v_2 = v_3 = 0\,, \qquad on\ \Gamma. \tag{2.22}$$

As mentioned before, farfield boundary conditions have to be considered ensuring that the flow is not affected by the finite domain. Further details can be found in [6, 19].

Turbulence models

The simulation of turbulent flows poses an extremely complicated problem, since turbulence causes the appearance of eddies with a wide range of length and time scales that interact in a dynamically complex way [175]. Despite the performance of modern supercomputers, the direct simulation of turbulence solving the time dependent Navier-Stokes equation (2.21), referred to as direct numerical simulation (DNS), is applicable only in a few simple flow problems, see for details e.g. [19]. Another possibility is the large eddy simulation (LES) tracking the behavior of the larger eddies. LES involves space filtering of the unsteady Navier-Stokes equation passing the larger eddies and rejecting the smaller ones. The demands on computing resources are smaller than using DNS, but even large due to the fact that unsteady flow equations have to be solved.

Turbulence models for the Reynolds-averaged Navier-Stokes equations (RANS) are the most used approach as they require considerably less effort for reasonably accurate flow computations than the other two approaches. The RANS method, which is the oldest approach to turbulence modeling, first introduced by Reynolds in 1895, is based on the decomposition of the flow variables into mean and fluctuating parts followed by time or ensemble averaging [19]. Extra terms, the so called Reynolds stress tensor, appear in the averaged flow equations due to the interactions between various turbulent fluctuations. These extra terms are modeled with classical turbulence models, which can be classified into first order and second order closures [175]. The second order closure models, which are the most complex but also the most flexible models, can be divided into Reynolds stress transport models (RST) and algebraic Reynolds stress (ARS) models [19]. The first order closures are based mostly on the eddy-viscosity hypothesis of Boussinesq assuming that the Reynolds stress tensor is related linearly to the mean flow straining field. There are several subcategories for the first order models depending on the number of additional equations: algebraic, one equation and multiple equation models. An overview of existing turbulence models can be found e.g. in [19].

2.1.3 Nondimensionalization

Solving the governing equations, it is convenient to use dimensionless quantities in order to obtain the characteristic properties of the system. First, we introduce the Mach number, which is commonly used to represent the speed of an object travelling at multiples of the speed of sound.

Remark 2.1.10 (Mach number)**.** The Mach number named after the Austrian physicist Ernst Mach is defined as the ratio of the fluid velocity to the local speed of sound

$$M = \frac{\|v\|}{c},$$

where c is the speed of sound, given by $\sqrt{\gamma \frac{p^{pr}}{\rho}}$ (for a perfect gas). If the Mach number is less than 1, the flow is called subsonic, if $M \approx 1$, the flow is called transonic and if $M > 1$, the flow is supersonic.

In the software we use to solve the flow equations, a unit system is chosen such that the farfield density ρ_∞, the pressure p^{pr}_∞ and the temperature T_∞ in the farfield are unity. Introducing the nondimensional quantities, an important nondimensional parameter closely related to the viscosity is the Reynolds number

$$Re = \frac{\rho v L}{\mu^{visc}},$$

where L denotes the characteristic length scale. The Reynolds number gives a measure of the ratio of inertia forces to viscous forces in the flow. Neglecting the viscous terms as in the Euler equations results in a Reynolds number equal to infinity, that means, the lower the Reynolds number the more viscous is the flow.

Consequently, we obtain the following reference quantities (the subscripts ∞ denote the value in the farfield)

L	characteristic length scale
$\rho_\infty = 1$	density
$p^{pr}_\infty = 1$	pressure
$T_\infty = 1$	temperature
$v_\infty = M_\infty \sqrt{\gamma}$	velocity

$$\mu_\infty^{visc} = \frac{M_\infty \sqrt{\gamma}}{Re_\infty} \qquad \text{viscosity}$$

$$\kappa_\infty = \frac{\gamma \mu_\infty^{visc}}{(\gamma - 1) Pr} \qquad \text{heat conductivity}$$

$$Re_\infty = \frac{\rho_\infty v_\infty L}{\mu_\infty^{visc}} \qquad \text{Reynolds number,}$$

where Pr is the dimensionless Prandtl number indicating the ratio of kinematic viscosity and thermal diffusivity. The Mach number M, Reynolds number Re, adiabatic exponent γ, the Prandtl number Pr and the reference length L are externally given as flow parameters.

2.1.4 Aerodynamic coefficients

In this section, we will introduce the forces acting on an airplane. We will see that the aim of the aerodynamic shape optimization will be the optimization of these forces. All the aerodynamic forces and moments are caused by only two basic sources:

- pressure distribution over the surface
- distribution of friction forces on the surface.

The angle of attack α is defined as the angle between the freestream velocity M_∞ and the chord length c, which is the linear distance from the leading edge to the trailing edge of the airfoil. The normal component of the local force on each point of the body is the pressure p^{pr} and its tangential component the shear stress σ^f. Integrating the force over the surface produces a resultant force F_R and a moment F_M, as sketched in figure 2.1.

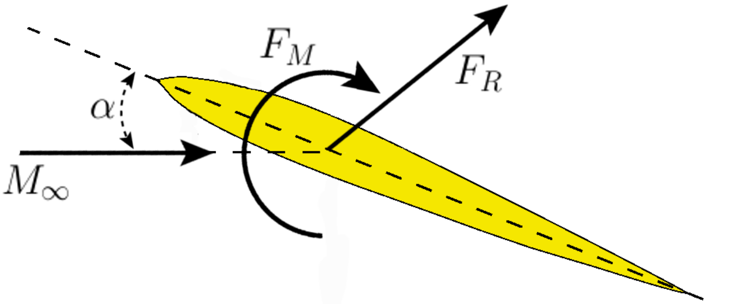


Figure 2.1: Resultant aerodynamic force and moment.

The resultant force F_R can be split up into two perpendicular components

- the drag F_D parallel to the freestream axes M_∞

- the lift F_L perpendicular to the freestream axes M_∞,

see figure 2.2. Since the drag force F_D acts in a direction opposite to the moving direction, an optimal aerodynamic design aims at optimizing the drag, the resistance to forward motion, and, depending on the gravity force, a given lift is required. The drag force is given by the following surface integral

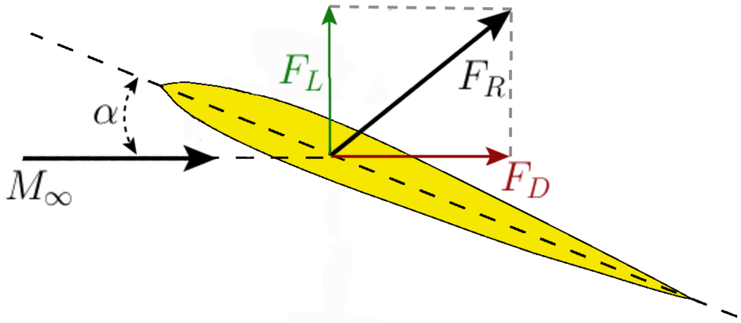


Figure 2.2: Resultant aerodynamic force and its components F_D, F_L.

$$F_D = \int_\Gamma (p^{pr} a)^\top n - (\sigma^f a)^\top n \, \mathrm{d}A\,, \tag{2.23}$$

where n denotes the unit vector normal to the profile Γ and a is defined by $a = (\cos\alpha, 0, \sin\alpha)^\top$. The lift force, which is perpendicular to the drag force, is then computed by

$$F_L = \int_\Gamma (p^{pr} a^\perp)^\top n - (\sigma^f a^\perp)^\top n \, \mathrm{d}A \tag{2.24}$$

with $a^\perp = (-\sin\alpha, 0, \cos\alpha)^\top$.

According to the previous section, we consider the dimensionless drag and lift coefficients given as

$$C_L = \frac{F_L}{q_\infty \mathcal{S}}\,,$$

$$C_D = \frac{F_D}{q_\infty \mathcal{S}}\,,$$

where $\mathcal{S}$ denotes the reference area and $q_\infty = \frac{1}{2} M_\infty^2 \gamma p_\infty^{pr}$ is the dynamic pressure. The moment coefficient is defined by

$$C_M = \frac{F_M}{q_\infty \mathcal{S} L}$$

with L reference length. For 2D bodies, the reference area $\mathcal{S}$ is simply the chord length and the reference length L is the chord length as well.

Further, the pressure and shear stress distributions causing the introduced forces can also be defined dimensionless

$$C_P = \frac{p^{pr} - p^{pr}_{\infty}}{q_{\infty}}$$

and the skin friction coefficient

$$C_f = \frac{\sigma^f}{q_{\infty}},$$

such that the drag and lift coefficients can be expressed in terms of C_P and C_f

$$C_D = \frac{1}{S} \int_{\Gamma} (C_p a)^{\top} n - (C_f a)^{\top} n \, \mathrm{d}A \tag{2.25}$$

and

$$C_L = \frac{1}{S} \int_{\Gamma} (C_p a^{\perp})^{\top} n - (C_f a^{\perp})^{\top} n \, \mathrm{d}A \, . \tag{2.26}$$

In the Euler case, the viscous terms are neglected in (2.25) and (2.26).

2.1.5 Flow solver

The numerical results of robust aerodynamic optimization presented in the thesis are all conducted using the flow solvers FLOWer and TAU developed primarily by the DLR within the German CFD initiative MEGAFLOW. A detailed presentation and demonstration of their capabilities for complex industrial applications can be found in [97].

Structured flow solver FLOWer

FLOWer calculates compressible flows either as inviscid based on the Euler equations or as viscous flows based on the RANS equations on structured grids. The solver provides various turbulence models, ranging from algebraic models over one- and two-equation models to algebraic stress models (cf. [97]). The numerical method is based on structured meshes, an example of a 2D structured grid used for the aerodynamic optimization with FLOWer is shown in figure 2.3. The method implemented to solve the governing equations is a finite volume method using either the cell vertex or the cell-centered approach. The convective fluxes are approximated by a central discretization scheme combined with scalar or matrix artificial viscosity, but several upwind schemes are also available. Time integration is carried out by an explicit hybrid Runge-Kutta scheme. In the case of steady-state calculations, the integration is accelerated by local time stepping and implicit or explicit residual smoothing [144]. An important feature for the optimization is the availability of adjoint solvers, which are needed to efficiently compute gradients with respect to the design variables describing the shape. We will discuss the general ideas of adjoint computation and derivative computation in section 2.2.3 and refer to [54] for a detailed discussion. The adjoint solver in FLOWer following the continuous adjoint formulation of the Euler equations can deal with the boundary conditions for drag, lift and pitching moment.

In order to handle also unstructured and hybrid meshes, the development of the flow solver TAU started about 15 years ago at the DLR. The main features will be presented in the following, a more detailed overview of the state of the art can be found in [70] or [166].

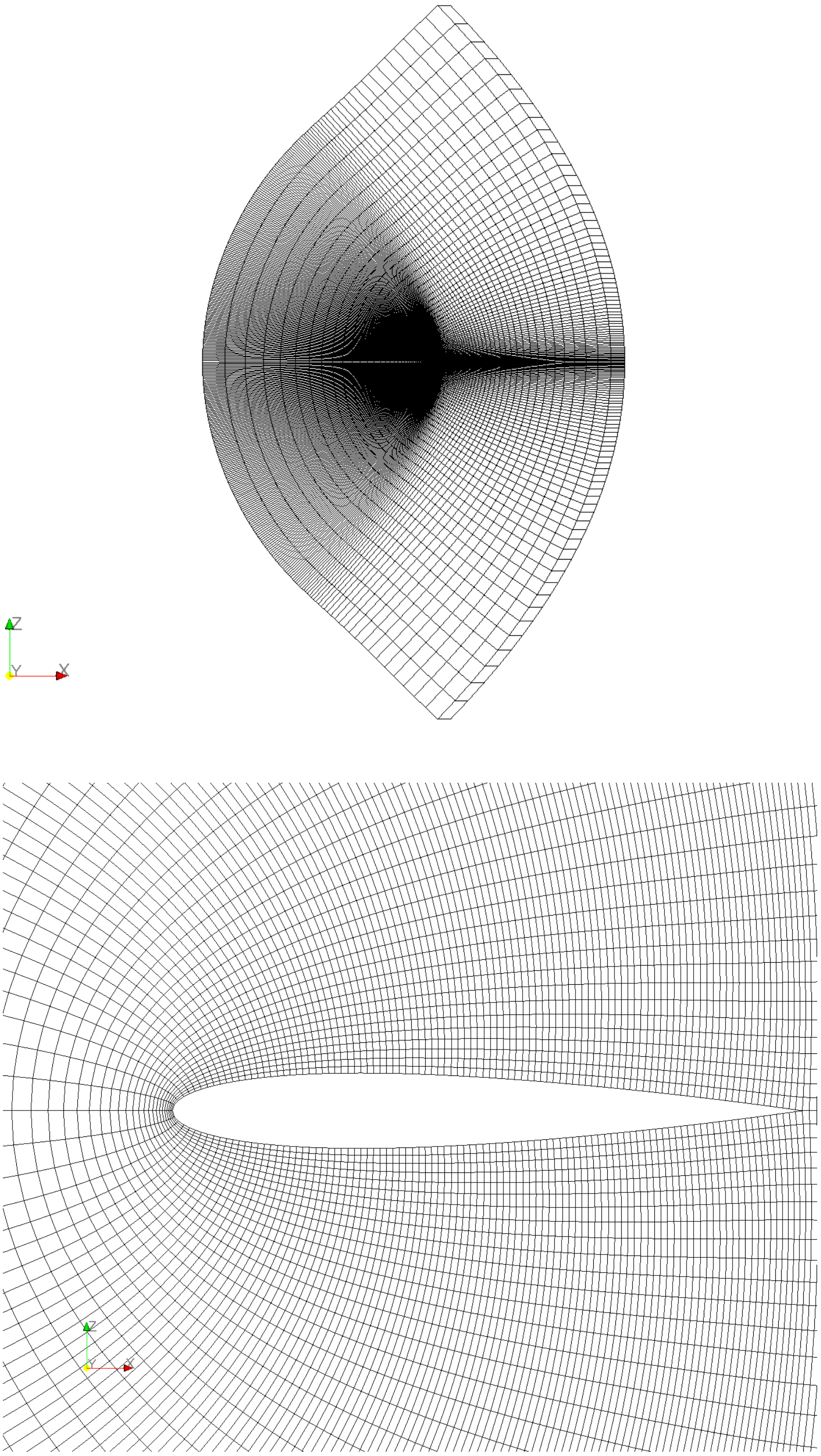


Figure 2.3: C-type grid for a supersonic airfoil (NACA0012): the total geometrical plane (above) and zoom around the airfoil (below).

Unstructured flow solver TAU

The TAU Code is a CFD software package for the prediction of viscous and inviscid flows around complex geometries from the low subsonic to the hypersonic flow regime employing hybrid unstructured grids. The initial unstructured or hybrid grid is input into the preprocessing module that computes dual grids using an edge-based data structure. Coarse grids for the multigrid algorithm are constructed recursively by agglomerating the control volumes at the finer grid level [144]. An example of an unstructured grid for the test case NACA0012 is depicted in figure 2.4. The standard solver module based on a finite volume scheme uses an edge based dual cell approach, where inviscid terms are computed employing either a second order central scheme or upwind schemes using linear reconstruction for second order accuracy. Viscous terms are computed with a second order central scheme. Scalar or matrix artificial dissipation may be chosen by the user. Beside an explicit Runge-Kutta multistage scheme in combination with an explicit residual smoothing and a multigrid method based on mesh agglomeration, an approximately factored implicit scheme (LU-SGS) has been implemented in order to improve the performance and robustness. As in the Euler code, several turbulence models are available. Further, the TAU software provides a continuous adjoint solver of the Euler equations and a discrete adjoint approach for the Euler and Navier-Stokes case, which is used in the optimization in order to compute gradients.

Both codes have been extended to allow geometry and mesh deformations for shape optimization and simulation of aeroelastic effects. Details on the deformation of the shape during the optimization will be discussed later on.

2.2 Optimal aerodynamic design

In the first part of this chapter, the basics of CFD were discussed in order to simulate the flow over a given geometry. The resulting flow is quantized by the dimensionless drag and lift coefficients C_D, C_L, which will be the primary quantities determining the optimality of a shape in our optimization problem. Before we can now formulate the aerodynamic optimization problem, we have to think about the representation of the shape, especially about the deformation during the optimization algorithm. The parametrization plays a key role in aerodynamic design, since a change in the parametrization may have a significant effect on the optimization problem.

2.2.1 Parametrization of the shape

Since parametrization of discrete surfaces is a fundamental and widely-used operation, it is an important application of geometry processing, which is a fast growing area of research. As mentioned before, the shape optimization algorithm finds the optimum shape for a given structural layout, so that the choice of parametrization has a strong influence on the whole optimization process. This influence has been confirmed e.g. in [126]. In the following, a survey of the most used parametrizations in CFD will be given, but since the parametrization of the airfoil and its influence on the solution are not the focus of this thesis, we will refer to the literature for more details on this topic (see e.g. [154, 169]).

One possible approach is the free-form deformation (FFD) originating from the computer graphics field. Instead of manipulating the surface of the airplane directly, the idea of FFD is to define a

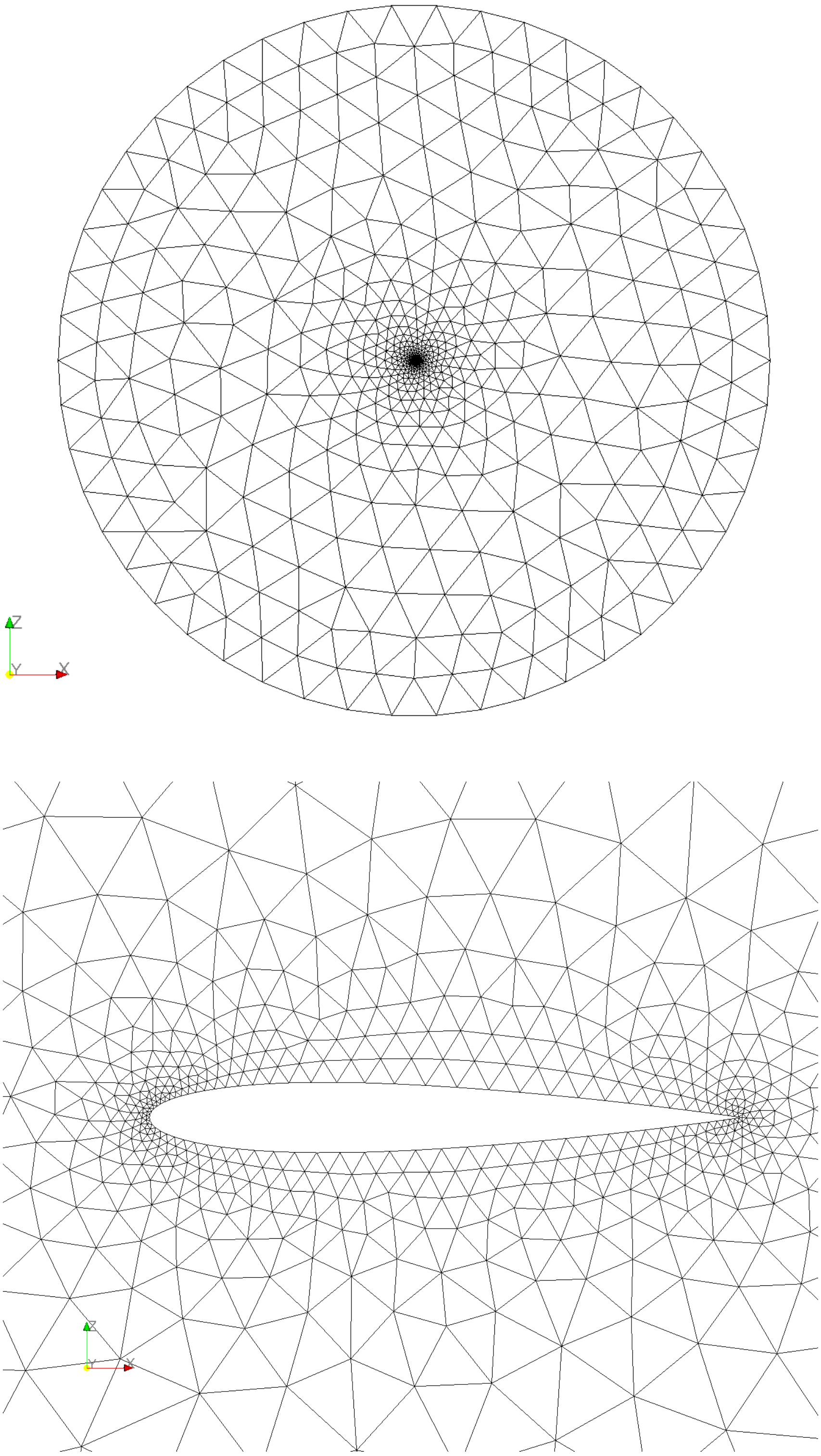


Figure 2.4: Unstructured grid for a supersonic airfoil (NACA0012): the total geometrical plane (above) and zoom around the airfoil (below).

deformation field over the space in which the object is embedded. By transforming the so called FFD control points in the space, the shape is deformed regardless of its geometrical description. One advantage of this approach is that the grid topology stays fixed during the optimization procedure. But on the other hand, large changes could produce unacceptable grids due to the fixed topology, so that the FFD is only suitable for small to medium geometry changes. Since the changes of the geometry in the aerodynamic optimization are typically small, this method becomes more and more popular, cf. e.g. [23, 155].

The most used parametrization in the aerodynamic shape optimization is the analytical approach introduced by Hicks and Henne [74]. The formulation is based on parametrizing the geometry using the weighted sum of a set of smooth (analytical) functions. In contrast to the spline interpolation, which uses piecewise polynomial approximations of the shape, the analytical approach leads to deformations that tend to be less wavy. This advantage has been investigated e.g. in [148]. No smoothing of the gradients is required, since the computed gradients always remain smooth, which ensures the smoothness of the optimized profile [29]. The Hicks-Henne functions $h_m : [0, 1] \rightarrow [0, 1]$ we use to deform the shape during the optimization are given by

$$h_m(x) = \left(\sin(\pi x^{\frac{\ln 0.5}{\ln x_m}})\right)^3 , \tag{2.27}$$

where x_m is defined as

$$x_m = \frac{m}{n_q + 4} , \qquad m \leq n_q, \ m \in \mathbb{N}$$

with n_q the number of functions used for the deformation. The “bump” functions h_m are then applied to the camberline *cam* of the shape

$$cam_{new}(x) = cam(x) + \sum_{i=1}^{n_p} p_i h_i(x) , \tag{2.28}$$

where the design variables are the coefficients p_i multiplying the various Hicks-Henne functions. Figure 2.5 illustrates the Hicks-Henne function of 5 design variables and the RAE2822 profile with the corresponding camberline, which will be deformed by the Hicks-Henne functions.

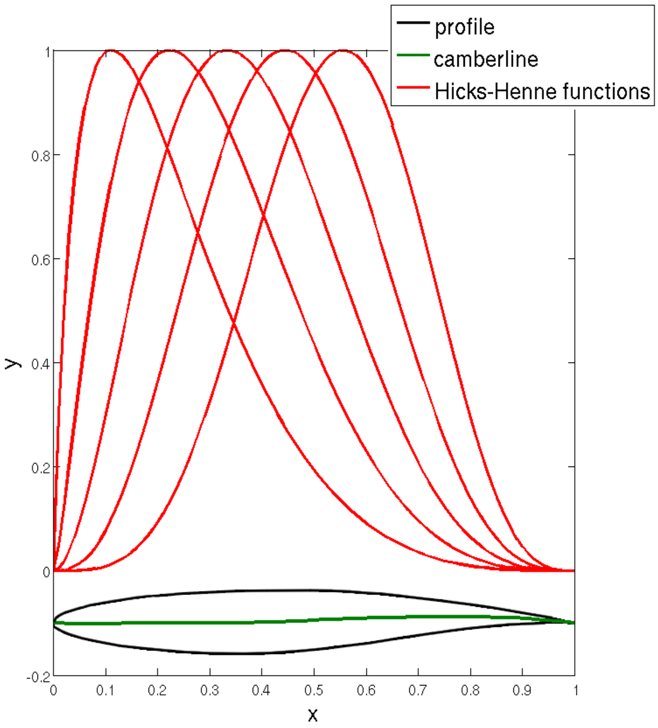


Figure 2.5: Hicks-Henne functions of 5 design variables (in red), the RAE2822 profile and the corresponding camberline (in green).

The different Hicks-Henne functions can be chosen such that only specific regions of the shape are deformed, that means this approach allows a local control of the deformation. Further, an important advantage concerning the optimization is the fact that constant thickness of the profile is automatically preserved by this approach when parametrizing just the camberline. Thus, additional geometric constraints can be avoided by this parametrization. The disadvantage of the Hicks-Henne functions is that they are not orthogonal and therefore unable to represent the complete set of continuous functions vanishing at $x = 0$ and $x = 1$, cf. [149]. Nevertheless, this approach has proved to be quite effective with only a small number of design variables. Since the Hicks-Henne deformation is well established in the optimization context at the DLR, this parametrization will be maintained in the robust optimization procedure.

Another possibility is the direct approach, which means the surface grid points are directly used in the deformation process. The main disadvantages of this method lie in the high number of design points and the non-smoothness of the resulting shape, which can be overcome by the use of shape derivatives and gradient preconditioning in the optimization, cf. [159, 160]. As the drawbacks of the parametrization by splines or Hicks-Henne functions, which lie in the limitation of the search space and the dependency of the computation of gradients on the number of design variables because of the so-called “mesh-sensitivities”, can be remedied, this direct approach provides an effective alternative to the common parametrizations. A combination of the robust approach proposed in this work and shape optimization based on the shape calculus is under current research.

2.2.2 The formulation of the aerodynamic shape optimization problem

The usual single-setpoint aerodynamic shape optimization problem can now be formulated in the following rather abstract form

$$\begin{aligned} & \min_{y,p} f(y,p) && (2.29)\\ \text{s.t.} \quad & c(y,p) = 0 && (2.30)\\ & h(y,p) \geq 0\,. && (2.31) \end{aligned}$$

The variable $y \in \mathbb{R}^{n_y}$ denotes the discrete state variable in conservative form, which means they indicate the conservative quantities in each grid point. The design variables $p \in \mathbb{R}^{n_p}$ describe the shape to be optimized. As mentioned before, we use Hicks-Henne functions deforming the camberline of the profile, so that the design variables are the multipliers of the various Hicks-Henne functions (cf. equation (2.28)). We think of equation (2.30) as the flow equation, more precisely $c : \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_y}$, represents the Euler or Navier-Stokes equation with appropriate boundary conditions. The objective in (2.29) $f : \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \to R$ is the drag coefficient to be minimized and the inequality constraint (2.31) represents the lift requirements, that means $h : \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \to \mathbb{R}$ computes the lift coefficient minus a target lift coefficient.

2.2.3 Adjoint equations and derivative computation

The efficient computation of derivatives with respect to the design variables is an important task in the aerodynamic shape optimization framework. The approximation of the derivatives using finite differences suffers from the high computational costs of one flow simulation, so that this approach is not an appropriate choice in our application. There exists another more efficient way to compute the derivatives, the so called adjoint approach, first introduced in the aerodynamic framework by Pironneau [139] and in the aerodynamic shape optimization context by Jameson [77, 78] and by Giles [60]. Below, we will give an introduction to the basics of the adjoint approach in the aerodynamic optimization framework based on [54, 61]. For a more in-depth discussion, we refer to [118]. First, the adjoint theory is discussed considering linear problems serving as a basis for the discrete adjoint CFD approach. Then, we will shortly present the extensions to operators defining PDEs as introduced by Jameson, see e.g. [77].

Discrete adjoint approach

Consider the following problem

$$\begin{aligned} \text{evaluate} \quad & o := r^\top x && (2.32)\\ \text{with x satisfying} \quad & Ax = b\,, && (2.33) \end{aligned}$$

where $A \in \mathbb{R}^{n \times m}$, $r \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, $x \in \mathbb{R}^m$. The system (2.32 - 2.33) is called the primal problem. Suppose that $\lambda \in \mathbb{R}^n$ solves

$$A^\top \lambda = r\,.$$

Then, it holds

$$o = r^\top x = (A^\top \lambda)^\top x = \lambda^\top Ax = \lambda^\top b$$

proving the equivalence of the primal problem (2.32 - 2.33) and the dual problem defined as

$$\begin{aligned} &\text{evaluate} && o := \lambda^\top b && (2.34)\\ &\text{with } \lambda \text{ satisfying} && A^\top \lambda = r\,, && (2.35) \end{aligned}$$

where $\lambda \in \mathbb{R}^m$ is called the dual or adjoint variable. So, the quantity of interest o can either be evaluated by solving the primal or the dual problem. If $m = n$, the computational effort will be exactly the same for both systems. The costs of the vector dot product in the objective is negligible compared to solving the linear system ($m = n \gg 1$), so that the dual approach will be much cheaper if one evaluates the objective for k different values of r and l different values of b where $k \ll l$. The adjoint variables are here considered as a mathematical construct reducing the computational effort, for physical interpretation of the variables, we refer to e.g. [54, 61].

Continuous adjoint approach

The introduced concept of duality in the linear setting can be extended to the continuous approach in the case of PDEs. Suppose that one wants to evaluate the functional

$$\langle r, x\rangle_\Omega := \int_\Omega r^\top x \,\mathrm{d}V$$

where $x : \Omega \to \mathbb{R}^m$ is the solution of the PDE

$$\mathcal{D}x = b$$

on the domain Ω subject to homogeneous boundary conditions on the boundary $\partial\Omega$. Then, the adjoint operator $\mathcal{D}^*$ is defined as

$$\mathcal{D}^* : \quad \langle \lambda, \mathcal{D}x\rangle_\Omega = \langle \mathcal{D}^*\lambda, x\rangle_\Omega\,, \quad \forall \lambda, x.$$

If the adjoint variable $\lambda : \Omega \to \mathbb{R}^n$ solves the adjoint PDE

$$\mathcal{D}^*\lambda = r$$

on the domain Ω with appropriate homogeneous adjoint boundary conditions, then it holds

$$\langle r, x\rangle_\Omega = \langle \mathcal{D}^*\lambda, x\rangle_\Omega = \langle \lambda, \mathcal{D}x\rangle_\Omega = \langle \lambda, b\rangle_\Omega\,. \tag{2.36}$$

Since the computation of the scalar product is usually much cheaper than the computation of the solution of a PDE, the discussion concerning the numerical efficiency of the discrete adjoint can be directly transferred to the continuous case.

Adjoints in aerodynamic shape optimization

As mentioned before, the introduced mathematical approach can be used to efficiently compute derivatives in the field of aerodynamic shape optimization, which is an important issue to make optimization computationally feasible even for complicated design tasks. The continuous adjoint method requires the computation of the adjoint operator $\mathcal{L}^*$ for the operator $\mathcal{L}$ defining the Euler or

Navier-Stokes equations. The derivation of the adjoint equations for inviscid and viscous compressible flow with appropriate adjoint boundary conditions can be found in [60]. The two flow solvers FLOWer and TAU both provide implemented continuous adjoint equations in the Euler case. The solution of the adjoint PDE is then computed by the flow solver using the same method as for the primal solution, so that the computational costs are nearly the same for the adjoint and primal solution. Using the continuous adjoint approach, one has to have in mind that the continuous approach yields a discrete approximation to the gradient of the analytic objective function with respect to each design variables. Hence, this will not exactly be equal to the gradient of the discrete approximation of the objective function and may cause problems in the optimization process [61]. It still remains an open question which approach of the two is the better one. If the solutions are smooth enough, the two different approaches should be consistent and converge to the correct gradient of the objective function. The continuous adjoint for the Euler case in FLOWer and TAU shows good results and provides gradients that agree with finite differences. In addition to the continuous one, the discrete approach is available for the Euler and Navier-Stokes flow equations within the solver TAU. The method consists of explicit construction of the exact Jacobians of the spatial discretization with respect to the unknown variables, allowing the adjoint equations to be formulated and solved [166]. So, the adjoint problem is just the solution of a linear system, whereas the forward problem is still solved using a pseudo-time integration scheme. The advantage of the discrete approach is the fact that the exact gradient of the discrete objective function is obtained. But on the other hand, the discrete approach requires much more memory compared to the continuous one, which could lead to problems in highly complicated optimization tasks, e.g. 3D test cases. The discrete as well as the continuous adjoints in FLOWer and TAU are well tested using several test cases comparing the gradients with the gradients computed by finite differences, see e.g. [41, 54, 166].

Derivative computation

According to the introduced shape optimization problem (2.29 - 2.31), we now consider the function $f(y,p) : \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \to \mathbb{R}$, where the variables y, p fulfill the additional constraint

$$c(y,p) = 0$$

representing the residuum of the discretized steady state flow equations. For each design variable $p_i,\ i = 1, \dots, n_p$, it holds

$$\begin{aligned}\frac{\mathrm{d}c}{\mathrm{d}p_i} &= \frac{\partial c}{\partial y}\frac{\partial y}{\partial p_i} + \frac{\partial c}{\partial p_i}\\ &=: Ax - b\\ &= 0\,.\end{aligned}$$

Applying the chain rule to f, one obtains

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}p_i} &= \frac{\partial f}{\partial y}\frac{\partial y}{\partial p_i} + \frac{\partial f}{\partial p_i}\\ &=: r^\top x + \frac{\partial f}{\partial p_i}\,.\end{aligned}$$

From the discussion above, it follows

$$r^\top x = \frac{\mathrm{d}f}{\mathrm{d}p_i} - \frac{\partial f}{\partial p_i} = \lambda^\top b = -\lambda^\top \frac{\partial c}{\partial p_i}\,,$$

where λ solves the adjoint equation

$$A^\top \lambda - r = \left(\frac{\partial c}{\partial y}\right)^\top \lambda - \left(\frac{\partial f}{\partial y}\right)^\top = 0\,. \tag{2.37}$$

An alternative description of the formulas above can be derived using the terminology of Lagrange multipliers associated with constrained optimization (cf. chapter 7), which can be found e.g. in [54, 61]. The final equations are exactly the same as those derived above.

The adjoint equation (2.37) is independent of i, which means independent of the number of design variables, hence this equation has to be solved only once. The derivative $\mathrm{d}f/\mathrm{d}p$ is then computed by matrix vector products, which are numerically much less expensive than solving linear systems. Thinking again of the introduced shape optimization problem (2.29 - 2.31), the adjoint approach solves the PDE only once and then computes the derivative $\mathrm{d}f/\mathrm{d}p$ by matrix-vector products, whereas finite differences require n_p additional PDE solves. Therefore, the adjoint approach provides a very efficient way computing derivatives for problems of the form (2.29 - 2.31).

Chapter 3

The Nature of Uncertainties in Aerodynamic Design

Uncertainties arise in all aspects of aerodynamic design. In order to identify and quantize the uncertainties, we first need to distinguish between errors and uncertainties. The following definition can be found in [5], which provides guidelines for the verification and validation of CFD simulations. Uncertainty is defined as:

"A potential deficiency in any phase or activity of the modeling process that is due to the lack of knowledge."

Error is defined as:

"A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge."

The key words in the definition of uncertainty are potential deficiency and lack of knowledge. Since the sources of the uncertainties are mostly unknown, the characterization of uncertainties is a difficult task. Hence, this definition of uncertainties suggests a stochastic approach. On the other hand, the identification of errors is a more straightforward task due to the knowledge of the error sources.

In order to identify and characterize the uncertainties in the aerodynamic framework, we now introduce the following classification of uncertainties which is generally accepted in literature [134]: epistemic uncertainties and aleatory uncertainties.

3.1 Epistemic vs. aleatory uncertainties

Epistemic uncertainty is an uncertainty due to lack of knowledge of quantities or processes of the system or the environment. Examples are the lack of experimental data to characterize material or poor understanding of coupled physics phenomena. These uncertainties are sometimes characterized as "state of knowledge" uncertainties, which means that the uncertainties can be reduced with more knowledge of the physical process. Aleatory uncertainty however is an inherent variation associated with the physical system or the environment, e.g. variations in the atmospheric conditions, also referred to as irreducible, stochastic uncertainties [133].

In the following, we will focus on aleatory uncertainties in aerodynamic design and develop a framework in order to include the identified uncertainties in the optimization.

3.2 Aleatory uncertainties in aerodynamic design

Aleatory uncertainties arise due to natural, unpredictable variations of parameter values, initial and boundary conditions, geometry, etc. Additional knowledge cannot reduce aleatory uncertainties, but it may be useful in getting a better characterization of the variability. In order to formulate the robust design optimization problem, we analyze the boundary conditions and input parameters identifying the uncertainties which cannot be avoided at all before constructing an aircraft [165].

In the following, we further distinguish two types of uncertainties: uncertainties with respect to the flight conditions and geometrical uncertainties.

The main characteristics of the macroscopic flight conditions are angle of incidence, the velocity (Mach number) of the plane, the density and the Reynolds number. The uncertainty of these parameters mostly results from atmospheric turbulences, which can occur during a flight. Gusts causes changes of the velocity in the range of $\pm 10\frac{m}{s}$. Measurements of the changes in the angle of attack and the density are not published so far, so they are assumed to be less than 10% of the setpoint. The variations of the Reynolds number will only effect the simulation, if the Reynolds number is in the range of $12 - 15 \cdot 10^6$, that means this uncertain parameter has not to be taken into account in our test cases.

On the other hand, we consider the shape itself as an uncertainty source. The real shape may vary from the planned shape due to manufacturing tolerances, temporary factors like icing or fatigue of material. Since there are so many factors having effects on the shape, this uncertainty has to be considered in the optimization problem in order to produce a design, which is robust to small perturbations of the shape itself. In literature, there can be found only a few papers on this topic investigating the influence of variations of the profile (cf. [66, 116]).

3.3 Stochastic models of aleatory uncertainties

Since we want to avoid a parametrization of the uncertainties, which would lead to a reduction of the space of realizations, we choose a stochastic approach in order to include the uncertainties in the optimization problem. Furthermore, this approach allows to adapt the robust optimization to new information of the uncertain parameter, e.g. if new measurements are available, so that a general framework of robust aerodynamic design can be developed.

The proper treatment of the uncertainties within a numerical context is a very important challenge, since simulation and also optimization under uncertainties is a fast growing field of research. As uncertainties pose problems for the reliability of numerical computations and their results in all technical contexts one can think of, there can be found various stochastic models depending on the application in literature (cf. [33, 59, 82, 138, 156]). In the following, a short overview of the main stochastic models is given. The background in probability theory can be found e.g. in [113, 114].

3.3.1 Random variables

Depending on the dimension of the uncertainty, the stochastic model is chosen. If the uncertain parameter is a scalar, which does not depend on other variables, e.g. time or space, the proper mathematical description will be a real-valued random variable. The outcomes of the random variable are the possible values of the uncertain parameter. Since the uncertain parameter may have an infinite number of possible realizations, the mathematical description by a random variable provides a suitable framework.

For all what follows, a given probability space $(\mathcal{O}, Y, \mathcal{P})$ is required.

Definition 3.3.1 (Measurable Space, Probability Space)**.** *A probability space is a triple* $(\mathcal{O}, Y, \mathcal{P})$ *consisting of*

- $\mathcal{O}$ *is a non-empty set*
- Y *is a family of subsets of* $\mathcal{O}$ *such that*
  1. $\emptyset \in Y$
  2. $\mathcal{O} \in Y$
  3. *if* $U \in Y$*, then* $\mathcal{O} \setminus U \in Y$
  4. *if* $U_n \in Y$, $\forall n \in \mathbb{N}$*, then* $\bigcup\limits_{n=1}^{\infty} U_n \in Y$.
- $\mathcal{P}$ *is a function on* Y *taking values in* $[0, 1]$ *such that*
  1. $\mathcal{P}(\emptyset) = 0$
  2. $\mathcal{P}(\mathcal{O}) = 1$
  3. *if* $\{U_n\}_{n=1}^{\infty}$ *is a countable, pairwise disjoint sequence of subsets in* Y*, then* $\mathcal{P}\left(\bigcup\limits_{n=1}^{\infty} U_n\right) = \sum\limits_{n=1}^{\infty} \mathcal{P}(U_n)$.

The pair $(\mathcal{O}, Y)$ *is called a measurable space and* Y *a* σ*-algebra.*

The set $\mathcal{O}$ consists of elementary elements, the σ-algebra Y is a collection of all events and the probability measure $\mathcal{P}$ indicates the probability of an event. Further information can be found in [39]. In order to introduce random variables, we first have to consider the concept of measurable functions.

Definition 3.3.2 (Measurable Function and Random Variable)**.** *Given two measurable spaces* $(\mathcal{O}_1, Y_1)$, $(\mathcal{O}_2, Y_2)$*, a function* $X : \mathcal{O}_1 \to \mathcal{O}_2$ *is said to be a measurable function, if*

$$X^{-1}(U) := \{\zeta \in \mathcal{O}_1 : X(\zeta) \in U\} \in Y_1\,, \qquad \forall U \in Y_2. \tag{3.1}$$

If $(\mathcal{O}_1, Y_1, \mathcal{P})$ *is a probability space, then the measurable function* X *is called a* $(\mathcal{O}_2, Y_2)$*-valued random variable.*

A real-valued random variable, which is used to describe the behavior of the uncertainty, is then defined as follows:

Definition 3.3.3 (Real-valued Random Variable)**.** *Let* $(\mathcal{O}, Y, \mathcal{P})$ *be a probability space and* $(\mathbb{R}, \mathcal{B})$ *a measurable space, where* $\mathcal{B}$ *denotes the Borel algebra. Then, the function* $X : \mathcal{O} \to \mathbb{R}$ *is a real-valued random variable on* $\mathcal{O}$ *to* $\mathbb{R}$*, if* X *is a measurable function, which means the inverse images under* X *of all Borel sets in* $\mathbb{R}$ *are events, which is equivalent to*

$$\{\zeta : X(\zeta) \leq r\} \in Y\,, \quad \forall r \in \mathbb{R}.$$

Remark 3.3.4 (Real-valued Random Vector)**.** A real-valued random vector is the generalization of random variables to the d-dimensional case. More precisely, a real-valued random vector $\boldsymbol{X} : \mathcal{O} \to \mathbb{R}^d$ is a measurable function defined on a probability space $(\mathcal{O}, Y, \mathcal{P})$ to a measurable space $(\mathbb{R}^d, \mathcal{B}_d)$.

The random variable modeling the uncertain parameter in the system can be described by its distribution function and the density function, respectively. The distribution function of real-valued random variable indicates the probability that the outcome of the random variable is less than or equal to a specific value x.

Definition 3.3.5 (Probability Distribution Function of a Real-valued Random Variable)**.** *The distribution function of a real-valued random variable X is $F_X : \mathbb{R} \to [0,1]$ such that*

$$F_X(x) := \mathcal{P}[X \leq x], \quad \forall x \in \mathbb{R}.$$

If the probability distribution function of X is continuous, X is a continuous random variable and we can define the probability density function as follows.

Definition 3.3.6 (Probability Densitiy Function of a Continous Real-Valued Random Variable)**.** *A probability density function of X is $f_X : \mathbb{R} \to \mathbb{R}_+$, such that*

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\,\mathrm{d}t, \quad \forall x \in \mathbb{R}.$$

An example of characterizing uncertain parameters by random variables can be found in [96, 143].

If the uncertain parameter also depends on additional deterministic variables, random fields are an appropriate choice in order to mathematically describe the uncertainty. Uncertainties of quantities that vary in time are usually modeled as random processes (cf. [120]). Random fields can be seen as a generalization of random processes, so that we will not discuss random processes here in detail and refer to [113, 114].

3.3.2 Random fields

In this section, the concept of random fields, which we use to model the aerodynamic uncertainties, will be shortly introduced. General introduction to the theory of random fields can be found in [3, 85, 113, 114]. The formal definition of a random field is the following (cf. [3]).

Definition 3.3.7 (Random Field)**.** *Let $(\mathcal{O}, Y, \mathcal{P})$ be a probability space and T a topological space. The measurable mapping $\gamma : \mathcal{O} \to \mathbb{R}^T$ (the space of all real-valued functions on T) is called a real-valued random field. Measurable mappings from $\mathcal{O}$ to $(\mathbb{R}^T)^d, d > 1$, are called vector-valued random fields. If $T \subset \mathbb{R}^N$, γ is a (N, d) random field, and if $d = 1$, γ is a N-dimensional random field.*

A one-dimensional random field is usually called a stochastic process. Random fields in two or three dimensions are encountered in a wide range of sciences and especially in earth sciences such as hydrology, agriculture and geology (cf. [69, 82, 86, 95, 112, 185]). In our application, it holds $T \subset \mathbb{R}^2$, $T \subset \mathbb{R}^3$, so that we denote the random field γ in the following by

$$\gamma : T, \mathcal{O} \to \mathbb{R}$$

for simplicity.

In literature, there can be found several special cases of random fields, but we will focus on Gaussian random fields as many natural processes tend to have this Gaussian distribution.

Gaussian random fields

First of all, a survey of characteristics of Gaussian random variables is given. A real-valued random variable X is said to be Gaussian ($X \sim \mathcal{N}(m, \sigma^2)$), if it has the density function

$$\varphi(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{(x-m)^2}{2\sigma^2}}, \quad x \in \mathbb{R} \tag{3.2}$$

for $m \in \mathbb{R}$ and $\sigma > 0$. A $\mathbb{R}^d$-valued random vector $\boldsymbol{X}$ is said to be multivariate Gaussian, if the real-valued variable $\sum_{i=1}^{d} \alpha_i \boldsymbol{X}_i, \quad \forall \alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{R}^d$ is Gaussian. Then, there exits a mean vector $m \in \mathbb{R}^d$ with $m_j = \mathbb{E}\left(\boldsymbol{X}_j\right)$ and a positive definite $d \times d$ covariance matrix Cov, $Cov_{ij} = \mathbb{E}\left((\boldsymbol{X}_i - m_i)\left(\boldsymbol{X}_j - m_j\right)\right)$, such that the probability density function of $\boldsymbol{X}$ is given by

$$\varphi(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |Cov|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-m)^\top Cov^{-1}(x-m)}, \tag{3.3}$$

where $|Cov| = \det(Cov)$ is the determinant of Cov [3]. From (3.3), one can see that the Gaussian variable is completely determined by the specification of the mean and the Covariance function. Another important property, which we will need later on, is the fact that uncorrelated Gaussian variables are independent and linear combinations of Gaussian random variables are again Gaussian.

Definition 3.3.8 (Gaussian Random Field). *A Gaussian random field is a random field γ on T, where all the finite dimensional distributions of $(\gamma(t_1), \dots, \gamma(t_N))$ are multivariate Gaussian for all $1 \le N < \infty$, $(t_1, \dots, t_N) \in T^N$.*

The expectation can be arbitrarily chosen, but the covariance function must be positive definite to ensure that all the finite-dimensional distributions exist [3, 64]. In order to model the uncertainties by a random field, the second order statistics usually have to be extracted from measurements. Since there are no statistical information about the uncertainties available in our application, we choose a covariance model from the literature, which could be adapted or replaced in the optimization algorithm, if more knowledge about the uncertainty is at hand. A short survey of the most common covariance functions is given in the following.

Covariance functions

A common simplifying assumption is to impose the requirement that the random field is stationary and isotropic, which means that the mean $m(t) = m$, $\forall t \in T$ and the covariance function $cov : T \times T \to \mathbb{R}$ only depends on the distance between points $cov(t, s) = cov(\tau)$, $\tau = \|t - s\|$, $\forall s, t \in T$. These properties are suitable in order to describe the identified uncertainties in the aerodynamic framework, so that we only introduce covariance functions matching these assumptions.

Exponential Covariance

$$cov(t, s) = \sigma^2 \cdot e^{-\frac{\|t-s\|}{l}}, \quad \sigma, l > 0$$

The Matérn class (cf. [43]) includes as a special case the exponential covariance function. One can show that random fields with a exponential covariance function are not differentiable in mean square.

Squared Exponential Covariance

$$cov\,(t,s) = \sigma^2 \cdot e^{-\frac{\|t-s\|^2}{l^2}}\,, \quad \sigma, l > 0$$

The squared exponential covariance function is the most widely used covariance function to describe uncertainties in a physical application (cf. [22, 171, 184]). Since the covariance function is infinitely many times differentiable, the corresponding random field is very smooth [145]. Due to this property, the geometrical uncertainties will be also described by this covariance function.

Rational Quadratic Covariance

$$cov\,(t,s) = \sigma^2 \cdot \left(1 + \frac{\|t-s\|^2}{\alpha l^2}\right)^{-\alpha}\,, \quad \alpha, \sigma, l > 0$$

The rational quadratic covariance function can be seen as a scale mixture of squared exponential covariance functions with different length-scales [145].

Remark 3.3.9. Besides the stochastic approach, other possibilities modeling uncertainties can also be found in literature: fuzzy sets and interval analysis are the most common ones. In the theory of fuzzy sets, the boundaries of the sets are not precise and the membership to a fuzzy set is a matter of degree. An overview and introduction to modeling uncertainty with fuzzy logic can be found in [30]. The interval analysis approach characterizes the uncertain parameter by mostly convex sets to which the parameters belong (cf. [179]). These approaches are applied to problems, if uncertainty is recognized, but cannot be quantified in statistical terms. Typical applications are problems including risks far into the future, e.g. radioactive waste repositories, or risks aggregated across sectors and over the world [7].

The uncertainties in the aerodynamic framework can be quantified by measurements, so that we choose the stochastic approach in order to develop a robust optimization algorithm, which is able to include more knowledge about the uncertainties, if it will be available in the future.

3.4 Mathematical description of the identified uncertainties

Again, we distinguish between the uncertainties with respect to the flight conditions, the scalar-valued uncertainties, and the geometrical uncertainties, the function-valued uncertainties.

3.4.1 Scalar-valued uncertainties

The scalar-valued uncertainties, e.g. the Mach number, are modeled as real-valued, continuous random variables

$$s : \mathcal{O} \to \mathbb{R}\,, \tag{3.4}$$

defined on a given probability space $(\mathcal{O}, Y, \mathcal{P})$. They are characterized by a given probability density function

$$\varphi_{truncated} : \mathbb{R} \longrightarrow \mathbb{R}_+ . \tag{3.5}$$

We assume (mainly due to the lack of statistical data) a truncated normal distribution of the perturbations ensuring that the realizations lie in between the given bounds. Furthermore, the mean value of the random variable corresponds with the value of the deterministic model. These assumptions are widely used in order to describe uncertainties in CFD (cf. [116]). Nevertheless, the model needs to be adapted to measurements, if available.

The following figure shows the probability density function of the Mach number in the test case RAE2822 (Euler flow).

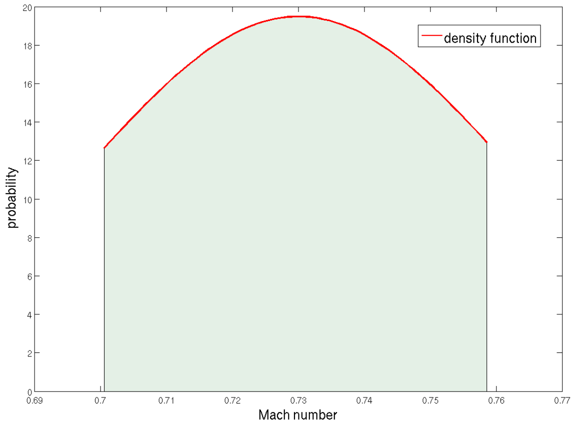


Figure 3.1: Truncated normal distributed Mach number.

3.4.2 Function-valued uncertainties

The geometrical uncertainties also depend on the geometry itself, so they are modeled as a Gaussian random field

$$\psi : \Gamma, \mathcal{O} \rightarrow \mathbb{R} , \tag{3.6}$$

defined on a probability space $(\mathcal{O}, Y, \mathcal{P})$ and on the shape of the airfoil Γ. In each point x of the shape Γ, the uncertainty is described by a normally distributed random variable $\psi(x, \cdot) : \mathcal{O} \rightarrow \mathbb{R}$. Additionally, the second order statistics, the mean value and the covariance function, are given to fully describe the random field. According to the scalar-valued uncertainties, the mean value of the random field ψ is equal to 0, since we expect no perturbations and the squared exponential covariance function describes the interaction between the random variables on the shape:

$$\mathbb{E}\left(\psi\left(x, \zeta\right)\right) = \psi_0\left(x\right) = 0 , \qquad \forall x \in \Gamma \tag{3.7}$$

$$cov\left(x, y\right) = \sigma^2 \cdot \exp\left(-\frac{\|x-y\|^2}{l^2}\right) , \qquad \forall x, y \in \Gamma \tag{3.8}$$

The parameter l determines how quickly the covariance falls off and σ controls the magnitude of the bumps. A squared exponential covariance function is chosen, since the resulting perturbed geometry is smooth due to the smoothness of the random field as mentioned in section 3.3.2.

Then, a perturbed geometry is given as

$$v(x,\zeta) = x + \psi(x,\zeta) \cdot n(x) , \qquad \forall x \in \Gamma, \zeta \in \mathcal{O}, \tag{3.9}$$

where n is the unit vector in x normal to the profile Γ. Figure 3.2 shows a realization of the random field ψ and the corresponding, resulting perturbed shape of the RAE2822 profile is depicted in figure 3.3.

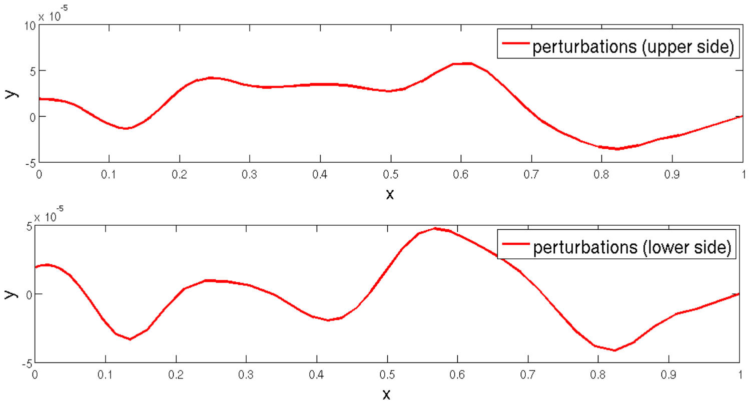


Figure 3.2: One realization of the random field ψ: perturbations on the upper side of the profile (above) and on the lower side (below).

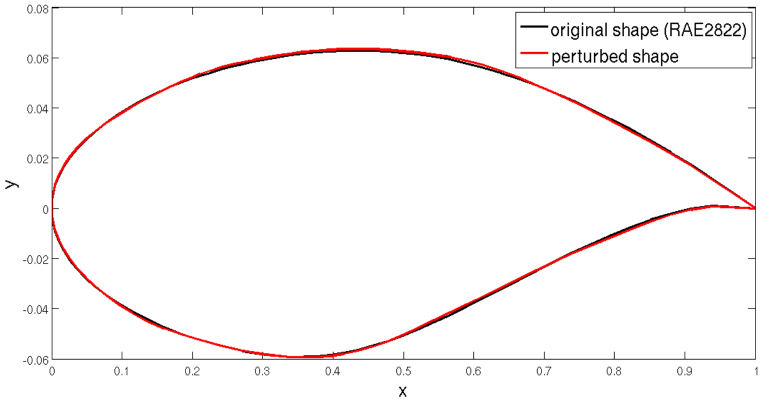


Figure 3.3: Resulting perturbed geometry (in red) compared with the original shape (in black).

As we need to compute statistics of the flow depending on the uncertainty in our optimization algorithm, we have to approximate and discretize the probability spaces. In the following, we will introduce the Karhunen-Loève expansion, which provides an approximation of the random field ψ for the numerical evaluation of such statistics and efficient discretization techniques of the probability space.

Chapter 4

Robust Formulations of Aerodynamic Design Problems

There is a variety of ways how optimization problems under uncertainties can be formulated. We will discuss several robust formulations of the general aerodynamic design optimization problem (2.29 - 2.31) based on probabilistic models of the uncertainties introduced in chapter 3.

In a first naive approach, we consider the general deterministic problem formulation (2.29 - 2.31) influenced by stochastic perturbations. For the sake of simplicity, the discussion below is restricted to random variables. The generalization of the formulations to random vectors and random fields is straightforward. We assume that the dependence of the uncertain parameter can arise in all aspects, i.e. a naive stochastic variant might be rewritten as

$$\min_{y,p} f(y, p, s(\zeta)) \tag{4.1}$$

$$\text{s.t.} \quad c(y, p, s(\zeta)) = 0 \tag{4.2}$$

$$h(y, p, s(\zeta)) \geq 0\,, \tag{4.3}$$

where $s : \mathcal{O} \to \mathbb{R}$ is a real-valued, continuous random variable defined on a given probability space $(\mathcal{O}, Y, P)$ and characterized by a given probability density function $\varphi : \mathbb{R} \to \mathbb{R}$. This formulation still treats the uncertain parameter as an additional fixed parameter. As the flow solution and consequently the lift and drag are dependent of the random input variable, these quantities itself become random vectors and random variables, respectively. Further, the state vector y also depends on the uncertain quantity s, so that there is a different y_s for each realization of the random variable s:

$$\min_{y(s(\zeta)),p} f(y(s(\zeta)), p, s(\zeta)) \tag{4.4}$$

$$\text{s.t.} \quad c(y(s(\zeta)), p, s(\zeta)) = 0 \tag{4.5}$$

$$h(y(s(\zeta)), p, s(\zeta)) \geq 0\,. \tag{4.6}$$

Assuming that the Jacobian

$$C_y = \frac{\partial c}{\partial y}$$

is invertible, the implicit function theorem leads to

$$\min_{p} f(\Theta(p, s(\zeta)), p, s(\zeta)) \tag{4.7}$$

$$\text{s.t.} \quad h(\Theta(p, s(\zeta)), p, s(\zeta)) \geq 0\,, \tag{4.8}$$

where $\Theta : \mathbb{R}^{n_p} \times \mathcal{O} \to \mathbb{R}^{n_y}$ such that $y = \Theta(p, s(\zeta))$ and $c(\Theta(p, s(\zeta)), p, s(\zeta)) = 0$.

The goal of robust optimization is now to find an optimal solution of problem (4.1 - 4.3), which is stable with respect to stochastic variations in s. Hence, the main task is to define an appropriate measure of robustness to reformulate problem (4.1 - 4.3) and make it mathematically tractable. In literature, there can be found various definitions of robustness. We will first give a short overview of the main results concerning robust optimization and then introduce three reformulations of (4.1 - 4.3).

4.1 A brief overview of literature on robust optimization

Recently, robust optimization has become a very active area of research. Concepts of robustness and robust design optimization have been developed independently in different scientific disciplines, such that the term robust optimization is not clearly defined. One interpretation of robustness is to require that the solution stays optimal under any realization. The main contributors of this worst-case philosophy are Soyster [170], Ben-Tal, Nemirovski et al. [11, 12, 13], El Ghaoui et al. [26, 42, 57] and Bertsimas and Sim [15, 16]. Since the worst-case counterparts will usually turn underlying linear or quadratic problems into nonlinear ones, the major challenge of recent research is to identify conditions, so that the robust counterpart is computationally tractable. The main results can be summarized as follows. The robust counterpart of linear programs (LP) are second-order cone programs (SOCP) and of SOCP are semi-definite programs (SDP) [186]. Nonlinear optimization problems are considered so far only in a few papers. In [37], a linearization of the uncertainty set reducing the dependence of the constraints on the uncertain parameters is proposed in order to compute robust solutions of nonlinear problems. [186] provides a general robust optimization approach for a nonlinear programming setting with parameter uncertainties involving equality and inequality constraints leading to solutions, which are robust to the first order.

An alternative approach to address optimization under uncertainty is stochastic optimization. A good overview and introduction to stochastic programming can be found e.g. in [18] or in [152]. In contrast to the worst-case approach, stochastic optimization involves additional probabilistic information of the uncertain parameters. Whereas the previous approach seeks for an optimal design with the best worst-case performance, the stochastic optimization allows to define further measures of robustness. This flexibility is able to overcome the conservatism of the worst-case philosophy, which might be in some applications a disadvantage. Various risk measures depending on the application can be found in literature, see [17, 119, 153]. Beside the flexibility in the choice of the objective function, the stochastic approach also offers the possibility to relax the requirements in the constraints by so called chance-constrained formulations, cf. [31]. The chance-constrained optimization problems were extensively studied by Prékopa [142].

A comprehensive survey of existing methods in robust optimization can be found in [11, 17] and the papers provided by the Committee on Stochastic Programming (COSP), see e.g. [38, 127].

In the aerodynamic framework, the first papers towards robust aerodynamic design consider a multipoint optimization in order to achieve a design, which shows a good drag performance for several given realization of the uncertain parameter, see e.g. [187]. Based on this, recent papers include stochastic information of the uncertain parameter leading to a stochastic optimization problem [76, 110]. The worst-case approach is only discussed in very few papers due to the resulting overly conservative designs and the required high computational effort to solve the robust counterparts, see e.g. [40, 109]. Most of the papers on robust aerodynamic design treat the angle of attack as a variable parameter, which is adjusted to reach the required lift. Since we model the lift requirements as an additional constraint in the optimization problem (cf. (4.3)), the corresponding robust counterpart of the constraint has to be formulated. Chance-constrained approaches are so far only considered in the field of structural design, e.g. [1].

4.2 Worst-case formulation

The following worst-case formulation is the most common one in literature:

$$\min_{y(s),p} \max_{s \in \mathcal{S}} f(y(s), p, s) \tag{4.9}$$

$$\text{s.t.} \quad c(y(s), p, s) = 0\,, \qquad \forall s \in S \tag{4.10}$$

$$h(y(s), p, s) \geq 0\,, \qquad \forall s \in S, \tag{4.11}$$

where $\mathcal{S}$ denotes the set of all possible realizations of the uncertain parameter s. This kind of accounting for uncertainties is also referred to as robust regularization [17]. The formulation (4.9 - 4.11) requires no stochastic description of the uncertain parameter, which can be of advantage, if no knowledge on the stochastic behavior of the parameter is at hand. If the probability density function of the uncertain parameter is not available, this approach could potentially be an attractive strategy. Otherwise, this formulation will ignore problem specific information, if it is at hand. Furthermore, this approach can also handle problems with uncertain parameters, which are not of a stochastic nature. This is the case, if we think of epistemic uncertainties, e.g. measurement or estimation errors [11]. The constraints of the optimization problem (4.10 - 4.11) are hard, that means no violations of the constraints are tolerated. This strict formulation of the constraints is typical in the context of robust optimization. We will later discuss a model which leaves more flexibility to the constraints by requiring the feasibility only with a given probability.

The concept of worst-case optimization leads to conservative designs, since the solution of (4.9 - 4.11) is optimal with respect to the worst-case. Hence, an always feasible solution is obtained, which shows an optimal objective value in the worst realizations of the uncertain parameter with regard to the objective function. A typical application of the introduced measure of robustness is structural design. When designing a bridge e.g., the engineers are interested in conservative designs with regard to safety related quantities. In this kind of problems, the design should show still a good performance even in the worst possible scenario.

In the context of robust aerodynamic optimization, this approach will lead to overly conservative designs. If one considers e.g. the optimization under Mach number uncertainties caused by turbulences, the worst-case formulation will ignore the additional information, how probable the worst-case realization is. So, the optimization could lead to a design which is optimal with respect to a Mach number, which occur only during extreme turbulences. Hence, the optimized design results in higher drag in much more frequent flight conditions leading to a higher fuel consumption over the whole flight. Due to this property, the worst-case formulation is not an appropriate approach in the aerodynamic framework and we will not treat this formulation.

4.3 Semi-infinite formulation

Here, the robust version of the nonlinear programming problem is written in the form of a semi-infinite optimization problem

$$\begin{aligned} &\min_{y(s(\zeta)),p} \mathbb{E}(f(y(s(\zeta)), p, s(\zeta))) && (4.12)\\ \text{s.t.} \quad & c(y(s(\zeta)), p, s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} && (4.13)\\ & h(y(s(\zeta)), p, s(\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}. && (4.14) \end{aligned}$$

The semi-infinite reformulation (4.12 - 4.14) aims at optimizing the average objective function but maintaining the feasibility with respect to the constraints everywhere. Thus, it aims at an average optimal and always feasible robust solution. This definition of robustness in the aerodynamic framework can also be found e.g. in [76, 110]. Compared to the worst-case formulation (4.9 - 4.11), the semi-infinite formulation (4.12 - 4.14) minimizes the expected value of the objective function f as a measure of robustness. This approach leads to less conservative designs, which fulfill again the constraints for all realizations of the uncertain parameter as in the worst-case formulation (cf. (4.11) and (4.14)). The computation of the mean value in (4.12) depends on the given probability distribution of the uncertain parameter, so this formulation allows to include additional information in the optimization problem. In the context of robust optimization, it is often proposed to add an additional term to the objective function associated with variability of f in order to compute designs, which are more conservative (cf. [76, 152, 180]). A compromise between the average and a risk associated with variability of the drag f can be reached by adding the variance $\mathbb{V}ar(f)$ to the objective (4.12), see [152]. Such choice of the variability measure was first suggested by Markowitz [119]. Hence, we obtain the following problem

$$\begin{aligned} &\min_{y(s(\zeta)),p} \mathbb{E}(f(y(s(\zeta)), p, s(\zeta))) + \theta \cdot \mathbb{V}ar(f(y(s(\zeta)), p, s(\zeta))) && (4.15)\\ \text{s.t.} \quad & c(y(s(\zeta)), p, s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} && (4.16)\\ & h(y(s(\zeta)), p, s(\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}. && (4.17) \end{aligned}$$

The coefficient θ indicates the risk aversion, that means a large coefficient θ results in a more conservative design of the underlying optimization problem. The mean-variance approach is popular e.g. in financial planning [127]. Higher moments of f are usually ignored.

The evaluation of the objective function (4.15) requires calculation of the corresponding integrals

$$\mathbb{E}(f(y(s(\zeta)), p, s(\zeta))) = \int_{\mathcal{O}} f(y(s(\zeta)), p, s(\zeta))\, \mathrm{d}\mathcal{P}(\zeta)$$

and

$$\mathbb{V}ar(f(y(s(\zeta)), p, s(\zeta))) = \int_{\mathcal{O}} (f(y(s(\zeta)), p, s(\zeta)) - \mathbb{E}(f(y(s(\zeta)), p, s(\zeta)))^2\ \mathrm{d}\mathcal{P}(\zeta)\,.$$

The integrals can be written in a closed form only in rather simple cases, so that we have to evaluate it numerically. The generalization of quadrature formulas to the high dimensional case is not an appropriate technique to evaluate the integrals due to the so called curse of dimensionality. In the aerodynamic setting, there are mainly two efficient approaches, the use of surrogate models of $f(y(s(\zeta)), p, s(\zeta))$ and the sparse grid techniques, which can circumvent the high computational costs of the direct integration by tensor grids. In the following chapters, we will introduce and discuss the techniques to efficiently evaluate these statistics.

4.4 Chance-constrained formulation

Chance constraints leave some flexibility with respect to the inequality restrictions (cf. [142]). In the context of structural optimization (which is typically a bilinear problem), this formulation is also called reliability-based design optimization [17]. The inequality constraints are only required to hold with a certain probability $\mathcal{P}_0$

$$\min_{y(s(\zeta)),p} \int_{\mathcal{O}} f(y(s(\zeta)), p, s(\zeta))\, \mathrm{d}\mathcal{P}(\zeta) \tag{4.18}$$

$$\text{s.t.} \qquad c(y(s(\zeta)), p, s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{4.19}$$

$$\mathcal{P}(\{\zeta \mid h(y(s(\zeta)), p, s(\zeta)) \geq 0\}) \geq \mathcal{P}_0\,. \tag{4.20}$$

Historically, the first paper introducing chance-constrained formulations was written by Charnes et al. [31]. The general chance-constrained formulation, where the probabilistic constraints are taken jointly for the stochastic constraints and the input random variables can be stochastically dependent, was initiated by Prékopa [142].

So far, chance constraints are used mainly for weakly nonlinear optimization problems (cf. [71, 80]). There are two primary reasons for the difficulties of chance-constrained problems [4]:

- The check for feasibility for a given design vector $p \in \mathbb{R}^{n_p}$ involves the evaluation of the integral $\int_{\{\zeta \mid h(y(s(\zeta)),p,s(\zeta))\geq 0\}} \mathrm{d}\mathcal{P}$. The computation of this integral is analytically tractable only for some special cases.
- The feasible set defined by the chance constraint (4.20) is generally not convex, which can lead to problems concerning the optimization procedure.

Solution methods for problems, where one of the two difficulties are absent, can be found e.g. in [142, 151, 152]. In more recent papers [28, 108, 130, 143], several approaches dealing with both difficulties have been proposed.

We will discuss an approximation of the function $h(y(s(\zeta)), p, s(\zeta))$ in the following chapter, which is based on a Taylor series approximation. Due to this simplification, the distribution of the lift constraint $h(y(s(\zeta)), p, s(\zeta))$ will be known, so that the probability $\mathcal{P}(\{\zeta \mid h(y(s(\zeta)), p, s(\zeta)) \geq 0\})$ can be evaluated and thus the feasibility for a given design $p \in \mathbb{R}^{n_p}$ can be checked.

Remark 4.1. Robust Optimization in Practice

The introduced robust counterparts of (4.1 - 4.3) are so far computationally not tractable due to the high nonlinearity of the underlying shape optimization problem. The next chapters will give a survey of methods simplifying and approximating the robust optimization problems in order to reduce or transform the problem to a solvable optimization task. In chapter 5, the focus is on uncertainty quantification methods providing approaches in order to efficiently determine statistical information of quantities of interest depending on the uncertain parameter. Further, efficient discretization techniques of the probability space will be discussed. Since we will see that the evaluation of such statistics is required in each iteration of the optimization algorithm, the main difficulty lies in the compromise between the approximation or simplification of the robust optimization problems and the preservation of accuracy.

Chapter 5

Uncertainty Quantification

The purpose of uncertainty quantification methods is the determination of the effect of uncertainties in the input data on quantities of interest in the output of a simulation. Uncertainty quantification methods are employed if statistical information like the density function of the solution, the mean or the variance are needed to completely describe the random field resulting from the solution of a equation with random input data. In the following, a general framework in order to determine all statistical quantities of interest of the solution will be discussed. Uncertainty quantification in CFD computations is a fast growing field of research in the last years, since classical methods are often inadequate for application to CFD problems due to their high amount of computational effort. Beside the CFD simulations, uncertainty quantification techniques are widely applied to large scale problems (e.g. deformation processes [2], chemical processes [32, 105]). The proposed methods in literature can be classified into two classes: the first one directly applies statistical methods to the simulation (e.g. Monte Carlo methods, response surface methods,...), whereas the second one solves the stochastic equation (e.g. polynomial chaos, stochastic operator expansion,...). The first class of methods performs a large number of random realizations of the uncertainty in order to determine statistics of the solution. The main advantage of these methods is the straightforward implementation, they belong to the so called non-intrusive methods, that means the CFD code to solve the deterministic PDE does not have to be changed. However, its main disadvantage is the requirement of large computational efforts, especially for large scale problems, so that we will concentrate on the second class of uncertainty quantification methods which represent the uncertainties by a spectral approximation that allows high order representations [56]. The most common non sampling technique is the perturbation or moment methods [121], which expand the random field in a Taylor series approximation around their mean. Typically, the series is truncated at second order due to the computational complexity of higher order terms. We apply the moment method to the chance-constrained optimization problem (4.18 - 4.20) in order to perform a numerical comparison of the introduced robust formulations. So, this method will be shortly introduced and discussed in section 5.1. Another possibility is the Karhunen-Loève expansion, which is the most effective approach concerning the convergence rate. The Karhunen-Loève expansion is based on a spectral decomposition of the covariance kernel, hence the covariance function of the output has to be known which is not the case in most of the applications. Due to this fact, the Karhunen-Loève expansion is not a suitable choice in order to compute the statistics of the solution in our application. But this approach can be used to approximate the input random field describing the geometrical uncertainties in a finite number of random variables.

Polynomial chaos and stochastic collocation techniques are the most used methods to propagate uncertainties in numerical simulations due to their computational efficiency and strong mathematical basis [44]. Both methods represent the stochastic output in a finite series of polynomials in the stochastic variable

$$
\begin{aligned}
f(p, \boldsymbol{X}(\zeta)) &\approx \sum_{i=1}^{N_c} \tilde{f}(p)\, \mathbf{P_i}(\boldsymbol{X}(\zeta)) \\
&= \sum_{i_1=1}^{m_{i_1}} \dots \sum_{i_d=1}^{m_{i_d}} \tilde{f}(p) \left(P_{i_1}^{j_1} \otimes \dots \otimes P_{i_d}^{j_d} \right) (\boldsymbol{X}(\zeta))) \,,
\end{aligned}
$$

where N_P denotes the truncation order of the expansion, **P** are the multivariate polynomials. The dependency of the objective function f on the state variables y is disregarded for simplicity reasons, assuming that the implicit function theorem can be applied (cf. formulation (4.7 - 4.8)). Polynomial chaos and stochastic collocation methods split the deterministic part $\tilde{f}$ and the stochastic part **P** and differ in the choice of the polynomial basis and the computation of the deterministic coefficients $\tilde{f}$. Stochastic collocation expansion is formed as a sum of multi-dimensional Lagrange interpolation polynomials and the coefficients are determined by a collocation method. The polynomial chaos method uses polynomials, which are orthogonal with respect to the probability density function of the uncertain parameters. Stochastic Collocation and (non-intrusive) polynomial chaos allow to use the flow solver as a black box, which is an important feature concerning our application. We will focus on polynomial chaos methods in the following, since this approach shows satisfactory results in the numerical examples. A detailed discussion with application of the Stochastic Collocation approach in the CFD framework can be found e.g. in [116].

Remark 5.0.1. The uncertainty quantification methods will be exemplarily discussed considering the scalar-valued objective function f in this chapter. The generalization to the lift constraints h and to spatially distributed quantities, e.g. the pressure coefficient C_P over the profile Γ is straightforward. Furthermore, we restrict the discussion to one scalar-valued uncertain parameter in most of the cases for ease of notation. The methods can be naturally extended to the multi-dimensional case.

5.1 Moment methods

Moment methods estimate moments of the output from the moments of the uncertain input parameters. The moments of the output are calculated from truncated Taylor series expansions at the mean value of the input [46]. Considering multiple random input variables, the generalization of the moment methods, which is then often referred to as perturbation method, is straightforward.

Given the input random variable X with mean value $m_X = \mathbb{E}(X)$ and standard deviation $\sigma_X = \sqrt{\mathbb{V}ar(X)}$ and variance $\mathbb{V}ar(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$, first- and second-order Taylor series approximations are given by

$$
f(X) \doteq f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) \tag{5.1}
$$

and

$$
f(X) \doteq f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) + \frac{1}{2} \left.\frac{\partial^2 f}{\partial X^2}\right|_{m_X} (X - m_X)(X - m_X)\,. \tag{5.2}
$$

For simplicity reasons, the dependency of the objective function on additional deterministic variables is disregarded, since there is no effect on the Taylor series approximation.

A survey of the most popular moment methods will be discussed in the next section.

5.1.1 First order first moment method (FOFM)

The first order first moment method (FOFM) estimates the mean of the output f evaluating the function using the first moment m_X of the input random variable X, that means

$$\mathbb{E}(f(X)) \doteq \mathbb{E}\left(f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) \right) = f(m_X)\,. \tag{5.3}$$

In special cases, e.g. if the random variable X is Gaussian distributed and the function f is linear, this approximation of the mean value computes the exact value. But in most of the applications, the FOFM method only provides a poor approximation of the exact value.

5.1.2 First order second moment method (FOSM)

According to the FOFM method, the FOSM method approximate the second moment, the variance, using the first order Taylor series expansion of f in (5.1). Hence, the approximation of the variance leads to

$$\begin{aligned} \mathbb{V}ar(f(X)) &\doteq \mathbb{V}ar\left(f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) \right) && (5.4)\\ &= \mathbb{E}\left((f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X))^2 \right) - \left(\mathbb{E}(f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X)) \right)^2 && (5.5)\\ &= \left(\left.\frac{\partial f}{\partial X}\right|_{m_X} \right)^2 \mathbb{V}ar(X)\,. && (5.6) \end{aligned}$$

Therefore, the variance of f is equal to the variance of the random variable X multiplied by the square of the sensitivity derivative evaluated at the mean of the input variable [46].

5.1.3 Second order first moment method (SOFM)

The second order first moment method (SOFM) approximates the mean value of the output f by adding the second order terms to the Taylor series expansion. This approach gives

$$\begin{aligned} \mathbb{E}(f(X)) &\doteq \mathbb{E}\left(f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) + \frac{1}{2} \left.\frac{\partial^2 f}{\partial X^2}\right|_{m_X} (X - m_X)(X - m_X) \right) && (5.7)\\ &= f(m_X) + \frac{1}{2} \left.\frac{\partial^2 f}{\partial X^2}\right|_{m_X} \mathbb{V}ar(X)\,. && (5.8) \end{aligned}$$

The additional terms may result in a large improvement of the approximation of the mean value, especially in nonlinear applications.

5.1.4 Second order second moment method (SOSM)

Compared to the FOSM method, the second order second moment method (SOSM) adds a higher order correction term involving the square of the second order derivative evaluated at the mean value of the input

$$\mathbb{V}ar(f(X)) \doteq \mathbb{E}\left((f(m_X) + \left.\frac{\partial f}{\partial X}\right|_{m_X} (X - m_X) - (f(m_X) + \frac{1}{2}\left.\frac{\partial^2 f}{\partial X^2}\right|_{m_X} \mathbb{V}ar(X)))^2 \right) \tag{5.9}$$

$$= \left(\left.\frac{\partial f}{\partial X}\right|_{m_X} \right)^2 \mathbb{V}ar(X) + \frac{1}{4}\left(\left.\frac{\partial^2 f}{\partial X^2}\right|_{m_X} \mathbb{V}ar(X) \right)^2 . \tag{5.10}$$

The SOSM method usually relies on a linear approximation of the objective function and the second order approximation of the mean (cf. equation (5.8)).

The moment method is used for the numerical comparison of the introduced robust shape optimization problems. The main disadvantage of perturbation methods is the fact that they can only be used to capture the local behavior of the solution under small variations of the random input data that means the variance of the uncertainty has to be small. As we will see later on in the numerical results, even the second order approach cannot capture the (nonlinear) behavior of the objective function, so that we will discuss in the following methods, which one will have the potential of higher approximation quality.

First, we are looking for an approximation of the input random field describing the geometrical uncertainties (cf. section 3.4.2) leading to representation in a Fourier-type form of

$$\psi(x, \zeta) = \sum_{i=1}^{\infty} c_i \kappa_i(x) X_i(\zeta) ,$$

where κ_i is an orthonormal set of deterministic functions, c_i are constants and X_i are random variables. Hence, the deterministic part is separated from the stochastic part. Two of the best-known representations in the context of uncertainty quantification are polynomial chaos methods and the Karhunen-Loève expansion. We will now concentrate on the Karhunen-Loève expansion due to its favorable properties. Further, we introduce an improvement of this approach to additionally reduce the number of random variables, which means to reduce the complexity of the robust optimization problem.

5.2 Karhunen-Loève expansion

The Karhunen-Loève expansion, also known as Proper Orthogonal Decomposition, represents the random field as a infinite linear combination of orthogonal functions chosen as the eigenfunctions of the covariance function. There are also other possibilities approximating the random field in a finite number of random variable, e.g. polynomial chaos methods which are discussed in detail in section 5.4, but the Karhunen-Loève expansion is the most used approximation due to its favorable features [82, 157]:

- The approximation of the random field using a truncated Karhunen-Loève expansion is in the mean-square sense optimal, that means the special choice of the random variables minimizes the mean-square error resulting from the truncation.

- If one considers a Gaussian random field as in our case, the random variables approximating the random field are independent, which is an assumption of the discretization approach we will discuss in section 6.3.

The Karhunen-Loève expansion of a stationary and isotropic Gaussian random field $\psi : \Gamma \times \mathcal{O} \to \mathbb{R}$ is given as:

$$\psi\left(x, \zeta\right) = \psi_0\left(x\right) + \sum_{i=1}^{\infty} \sqrt{\varsigma_i^{KL}} z_i^{KL}\left(x\right) X_i\left(\zeta\right) \ , \tag{5.11}$$

where $\varsigma_1^{KL} \geq \varsigma_2^{KL} \geq ... \geq \varsigma_i^{KL} \geq ... \geq 0$ and z_i^{KL} are the eigenvalues and eigenfunctions of the covariance function *cov* which is symmetric and positive definite by definition, X_i are random variables and ψ_0 is the mean value of the random field. Since we expect no perturbations (cf. section 3.4), the Karhunen-Loève expansion can be reduced to:

$$\psi\left(x, \zeta\right) = \sum_{i=1}^{\infty} \sqrt{\varsigma_i^{KL}} z_i^{KL}\left(x\right) X_i\left(\zeta\right) \ , \qquad x \in \Gamma, \ \zeta \in \mathcal{O}.$$

The deterministic eigenfunctions z_i^{KL} are obtained from the spectral decomposition of the covariance function via solution of

$$\int_\Gamma cov\left(x, y\right) z_i^{KL}\left(y\right) = \varsigma_i^{KL} z_i^{KL}\left(x\right) \ . \tag{5.12}$$

Having the eigenpairs, the uncorrelated Gaussian random variables X_i in equation (5.11) can be expressed as

$$X_i\left(\zeta\right) = \frac{1}{\sqrt{\varsigma_i^{KL}}} \int_\Gamma \psi\left(x, \zeta\right) z_i^{KL}\left(x\right) \mathrm{d}x \, , \qquad j = 1, 2, ... \tag{5.13}$$

with zero mean and unit variance, i.e. $\mathbb{E}\left(X_i\right) = 0$ and $\mathbb{E}\left(X_i X_j\right) = \delta_{ij}$ for $j = 1, 2, ...$ (cf. [22]). In the special case of a Gaussian random field, uncorrelated implies independent, which is an important property that we will need for the sparse grid as mentioned before.

In [56], it is shown that the eigenfunction basis $\{z_i^{KL}\}$ is optimal in the sense that the mean-square error resulting from the truncation after the d-th term is minimized.

Lemma 5.2.1 (Optimality of the Eigenfunction Basis)**.** *The basis defined by the eigenfunctions of the covariance kernel is optimal in the sense that the mean-square error resulting from the finite representation of the random field $\psi(x, \zeta)$ is minimized.*

Proof. Given a complete orthonormal set of functions $\kappa_i(x)$, the random field $\psi(x, \zeta)$ can be represented in a convergent series of the form

$$\psi(x, \zeta) = \sum_{i=0}^{\infty} c_i \kappa_i(x) X_i(\zeta) \, . \tag{5.14}$$

The error of the truncation of (5.14) at the N-th term is then given as

$$\epsilon_N = \sum_{i=N+1}^{\infty} c_i \kappa_i(x) X_i(\zeta) \, .$$

Multiplying equation (5.14) by $\kappa_m(X)$ and intregating throughout gives

$$X_m(\zeta) = \frac{1}{c_m} \int_\Gamma \psi(x,\zeta)\kappa_m(x)\,\mathrm{d}x \tag{5.15}$$

due to the orthonormality of $\kappa_i(X)$. The mean-square error ϵ_N^2 using (5.15) can be written as

$$\begin{aligned}
\epsilon_N^2(x) &= \mathbb{E}\left(\psi(x,\zeta) - \sum_{i=0}^{N} c_i\kappa_i(x)X_i(\zeta)\right)^2 \\
&= \mathbb{E}\left(\sum_{i=N+1}^{\infty} c_i\kappa_i(x)X_i(\zeta)\right)^2 \\
&= \mathbb{E}\left(\sum_{i=N+1}^{\infty} c_i\kappa_i(x)\frac{1}{c_i}\int_\Gamma \psi(x,\zeta)\kappa_i(x)\,\mathrm{d}x\right)^2 \\
&= \sum_{i=N+1}^{\infty}\sum_{j=N+1}^{\infty} \kappa_i(x)\kappa_j(x)\int_\Gamma\int_\Gamma \mathbb{E}\left(\psi(x_1,\zeta)\cdot\psi(x_2,\zeta)\right)\kappa_i(x_1)\kappa_j(x_2)\,\mathrm{d}x_1\mathrm{d}x_2 \\
&= \sum_{i=N+1}^{\infty}\sum_{j=N+1}^{\infty} \kappa_i(x)\kappa_j(x)\int_\Gamma\int_\Gamma Cov(x_1,x_2)\kappa_i(x_1)\kappa_j(x_2)\,\mathrm{d}x_1\mathrm{d}x_2.
\end{aligned}$$

Due to the orthonormality, it holds

$$\int_\Gamma \epsilon_N^2(x)\,\mathrm{d}x = \sum_{i=N+1}^{\infty}\int_\Gamma\int_\Gamma Cov(x_1,x_2)\kappa_i(x_1)\kappa_i(x_2)\,\mathrm{d}x_1\mathrm{d}x_2\,.$$

The solution of min $\int_\Gamma \epsilon_N^2(x)\,\mathrm{d}x$ subject to $\int_\Gamma \kappa_i(x)\cdot\kappa_j(x)\,\mathrm{d}x = \delta_{ij}$ also minimizes

$$\sum_{i=N+1}^{\infty}\int_\Gamma\int_\Gamma Cov(x_1,x_2)\kappa_i(x_1)\kappa_i(x_2)\,\mathrm{d}x_1\mathrm{d}x_2 - c_i\left(\int_\Gamma \kappa_i(x)\kappa_i(x)\,\mathrm{d}x - 1\right)\,. \tag{5.16}$$

Differentiating (5.16) with respect to $\kappa_m(x)$, we obtain

$$\int_\Gamma\left(\int_\Gamma Cov(x_1,x_2)\kappa_m(x_1)\,\mathrm{d}x_1 - c_m\kappa_m(x_2)\right)\mathrm{d}x_2\,,$$

which equals to zero, if

$$\int_\Gamma Cov(x_1,x_2)\kappa_m(x_2)\,\mathrm{d}x_2 = c_m\kappa_m(x_1)\,.$$

□

Further, one can show the uniqueness of the expansion, that means the random variables in (5.14) are orthonormal if and only if the orthonormal functions $\kappa_i(x)$ and the constants c_i are the eigenfunctions and the eigenvalues of the covariance kernel. The proof can be found in [56].

Truncating now the Karhunen-Loève expansion after a finite number of terms, we obtain the approximation of the random field ψ

$$\psi_d\left(x,\zeta\right) = \sum_{i=1}^{d} \sqrt{\varsigma_i^{KL}} z_i^{KL}\left(x\right) X_i\left(\zeta\right) \ , \qquad x \in \Gamma,\ \zeta \in \mathcal{O}. \tag{5.17}$$

The corresponding covariance function is given by

$$cov_d\left(x,y\right) = \sum_{i=1}^{d} \varsigma_i^{KL} z_i^{KL}\left(x\right) z_i^{KL}\left(y\right) \ .$$

If cov is continuous, $cov_d\left(x,y\right) = \sum_{i=1}^{d} \varsigma_i^{KL} z_i^{KL}\left(x\right) z_i^{KL}\left(y\right)$ converges uniformly and absolutely to cov on $\Gamma \times \Gamma$ by Mercer's theorem [123]. Then, it follows that ψ_d converges to ψ in variance uniformly, i.e. in $L^{\infty}(\Gamma) \times L^2(\mathcal{O})$

$$\lim_{d\to\infty} \left\{ \sup_{\Gamma} \int\limits_{\mathcal{O}} \left(\psi - \psi_d\right)^2 \mathrm{d}\mathcal{P}\left(\zeta\right) \right\} = \lim_{d\to\infty} \left\{ \sup_{\Gamma} \left(\sum_{j=d+1}^{\infty} \varsigma_j^{KL} (z_j^{KL})^2 \right) \right\} = 0\,. \tag{5.18}$$

So, ψ_d may provide a suitable approximation of ψ, if the eigenvalues decay sufficiently fast and d is large enough, see [22].

If one assumes a Gaussian covariance function (cf. (3.8)), the eigenvalues will subexponentially decay towards zero, i.e.

$$0 < \varsigma_m^{KL} \lesssim \sigma^2 \frac{(\frac{1}{\Gamma})^{m^{\frac{1}{\dim(\Gamma)}}+2}}{\Gamma_f(0.5\ m^{\frac{1}{\dim(\Gamma)}})} \ , \qquad \forall m \geq 1, \tag{5.19}$$

where Γ_f is the gamma function, $l^2 = \gamma^2 \cdot \Lambda^2$, $\gamma > 0$ and Λ is the diameter of the domain Γ and l the parameter of the covariance function (cf. (3.8)). The proof of this behavior of the eigenvalues can be found e.g. in [49].

5.2.1 Discrete Karhunen-Loève expansion

Analytic solutions of the eigenvalue problem exist only in a few special cases, so that equation (5.12) needs to be solved numerically. Detailed closed form solutions for exponential covariance functions for one-dimensional random fields can be found in [56]. The majority of existing numerical methods for solving (5.12) transform the integral eigenvalue problem into an algebraic eigenvalue problem, hence an additional approximation error, which has to be taken into account, arises [157]. A Galerkin-type procedure is the most common discretization technique of the operator in (5.12) (cf. e.g. [56, 84, 162]), which leads to the following eigenvalue problem

$$W^{Galerkin} z_i^{KL} = \varsigma_i^{KL} M^{Galerkin} z_i^{KL} \ ,$$

where the matrix $W^{Galerkin}$ and mass matrix $M^{Galerkin}$ results from the Galerkin approach. More details can be found in [56, 84].

In our application, the surface grid of the test case describes the shape Γ, on which the random field ψ of perturbations is defined. As there is no parametrization of the shape Γ at hand, the discretization given by the grid is used to solve the eigenvalue problem (5.12), which thus simplifies to

$$Cov^h z_i^{KL} = \varsigma_i^{KL} z_i^{KL} \,, \tag{5.20}$$

where $Cov^h_{ij} = \sigma^2 \cdot \exp\left(-\frac{\|x_i-x_j\|^2}{l^2}\right)$, $\forall 1 \leq ... \leq i,j \leq ... \leq N_{\Gamma^h}$ with $\Gamma^h = \{x_1, \dots, x_{N^h_\Gamma}\}$ the discretized surface.

5.2.2 Solving the large eigenvalue problem

The resulting eigenvalue problem (5.20) depends on the size of the surface grid in our application. In the two-dimensional test cases, the number of surface grid points is less than 200 points, so that a direct eigensolver can compute the eigenvalues and eigenfunctions in an acceptable cpu time. So, the eigensolver provided by MATLAB is used to determine the Karhunen-Loève expansion in the RAE2822 test case.

In the three-dimensional test case, the number of grid points is in the range of 80.000 points, so that an iterative eigensolver needs to be used due to the problem size. Since the Karhunen-Loève expansion is a common approach in the field of uncertainty quantification and robust optimization, several methods solving the large eigenvalue problem are proposed in literature. In [49], a fast algorithm based on a kernel independent fast multipole method to compute the Karhunen-Loève approximation is introduced. Another approach to solve the large eigenvalue problem arising from the Karhunen-Loève expansion can be found in [84]. They propose a Krylov subspace method with a sparse matrix approximation using sparse hierarchical matrix techniques to solve it. Further, [92] suggests an algorithm of the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method for symmetric eigenvalue problems based on a local optimization of a three-term recurrence, which we employ in order to compute the Karhunen-Loève expansion due to its speed of convergence and memory requirements. In the following, a short introduction to this method is given. More details can be found in [92].

Locally optimal block preconditioned conjugate gradient method (LOBPCG)

In this section, we consider a generalized symmetric definite eigenvalue problem of the form

$$(A - \varsigma B)z = 0 \tag{5.21}$$

with real symmetric $n \times n$ matrices A, B, A positive definite, following closely [92]. The generalized eigenvalue problem (5.21) has all real eigenvalues ς_i with orthogonal eigenvectors z_i satisfying $(A - \varsigma_i B)z_i = 0$, and $\langle z_i, Az_j \rangle = \langle z_i, Bz_j \rangle$, $i \neq j$. The algorithm provides the smallest m eigenvalues of (5.21).

The discrete Karhunen-Loève expansion (5.20) leads to the eigenvalue problem of the form

$$(Cov^h - \varsigma^{KL} I)z^{KL} = 0 \,. \tag{5.22}$$

Since we need to compute the largest eigenvalues, the problem (5.22) can be transformed to

$$(-I + \vartheta Cov^h)z^{KL} = 0 \,, \tag{5.23}$$

where $\vartheta_i = \frac{1}{\varsigma_i^{KL}}$, so that the method gives the largest eigenvalues of (5.22).

First, a preconditioner P for the eigenvalue problem (5.21), $(B - \vartheta A)z = 0$ respectively, to accelerate the convergence is discussed. Similarly to the preconditioning of linear systems, symmetric positive definite preconditioners P with $P \sim A^{-1}$ are constructed, such that positive constants $\delta_2 \geq \delta_1 > 0$ exist with

$$\delta_1 \langle P^{-1}x, x\rangle \leq \langle Ax, x\rangle \leq \delta_2 \langle P^{-1}x, x\rangle \,.$$

The ratio of the constants δ_1, δ_2 indicates the approximation quality of P to A^{-1}. In the software package BLOPEX [92] including the eigensolver LOBPCG, one has to provide the preconditioner in the form of a matrix vector product Pz as input. In our tests, the preconditioner is simply set to I, but in literature, there are proposed several preconditioners showing good results (cf. [91, 93] or [125]).

Then, the preconditioned conjugate gradient method can be described by the following three-term recurrence formula:

$$\begin{aligned} z^{i+1} &= w^i + c_1^i z^i + c_2^i z^{i-1} && (5.24)\\ w^i &= P(Bz^i - \vartheta^i A z^i) && (5.25)\\ \vartheta^i &= \vartheta(z^i) && (5.26) \end{aligned}$$

with properly chosen scalar iteration parameters c_1, c_2. The approximation ϑ^i in (5.26) to an eigenvalue of $(B - \vartheta A)z = 0$ is computed by the Rayleigh quotient

$$\vartheta(x^i) = \frac{\langle z^i, Bz^i\rangle}{\langle z^i, Az^i\rangle} \tag{5.27}$$

for a given eigenvector approximation z^i. In [90], the scalar iteration parameters are chosen using an idea of local optimality, that means, the parameters, which maximize the Rayleigh quotient (5.27) by using the Rayleigh-Ritz method, are used. In order to increase the stability of the algorithm, an orthogonalization step before the Rayleigh-Ritz procedure is introduced, which leads to the mathematically equivalent algorithm:

$$\begin{aligned} z^{i+1} &= w^i + c_1^i z^i + c_2^i pr^i\\ w^i &= P(Bz^i - \vartheta^i A z^i)\\ \vartheta^i &= \vartheta(z^i)\\ pr^{i+1} &= w^i + c_2^i pr^i,\ pr^0 = 0\,. \end{aligned}$$

The block version of the preconditioned conjugate gradient method is the straightforward generalization of the introduced single-vector method.

Algorithm 5.1 Locally optimal block preconditioned conjugate gradient method (LOBPCG)

```
function LOBPCG
    Input:  matrix vector products $Az$, $Bz$ and $Pz$ and the vector inner product $\langle\cdot,\cdot\rangle$
    Output: the approximations of the m largest eigenvalues $m_j$ and eigenvectors $x_j$
    select starting vectors $z_j^0$, $j = 1, \dots, m$
    $pr_j^0 \leftarrow 0$, $j = 1, \dots, m$
    for $i = 0, \dots,$ until convergence do
        $\vartheta_j^i \leftarrow \langle z_j^i, Bz_j^i \rangle / \langle z_j^i, Az_j^i \rangle$, $j = 1, \dots, m$
        $r_j^i \leftarrow Bz_j^i - \vartheta_j^i Az_j^i$, $j = 1, \dots, m$
        $w_j^i \leftarrow Pr_j$, $j = 1, \dots, m$
        use the Rayleigh-Ritz method for $B - \vartheta A$ on the trial subspace
        $span\{w_1^i, \dots, w_m^i, z_1^i, \dots, z_m^i, pr_1^i, \dots, pr_m^i\}$, such that the column-vector
        $(\alpha_1^i, \dots, \alpha_m^i, (c_1)_1^i, \dots, (c_1)_1^i, (c_2)_1^i, \dots, (c_2)_1^i)^\top$
        is the $j$-th eigenvector corresponding to the $j$-th eigenvalue of the
        $3m \times 3m$ eigenvalue problem of the Rayleigh-Ritz method
        $z_j^{i+1} \leftarrow \sum_{k=1}^m \alpha_k^i w_k^i + (c_1)_k^i z_k^i + (c_2)_k^i pr_k^i$, $j = 1, \dots, m$
        $pr_j^{i+1} \leftarrow \sum_{k=1}^m \alpha_k^i w_k^i + (c_2)_k^i pr_k^i$
    end for
end function
```

The main advantage of the algorithm 5.1 is the so called matrix free property, which means that the multiplication of a vector by the matrices of the eigenproblem and the application of the preconditioner to a vector are needed only as functions. In our application, this feature is indispensable due to the problem size of the eigensystem.

5.3 Goal-oriented Karhunen-Loève expansion

The evaluation of the objective function in the robust optimization problem (4.12) requires the computation of the mean, i.e. the computation of the integral of the random field with respect to its probability measure. Applying the introduced Karhunen-Loève approximation, the objective function can be written as the following d-dimensional integral

$$\mathbb{E}\left(f(p, \psi(x, \zeta))\right) \doteq \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} f(p, \psi_d(x_1, \dots, x_d))\varphi_1(x_1) \cdots \varphi_1(x_d)\ \mathrm{d}x_1 \cdots \mathrm{d}x_d\,,$$

where φ_1 is the one-dimensional Gaussian measure. So, one term more in the truncated Karhunen-Loève expansion to increase the approximation accuracy results in an integral of one dimension higher. In order to reduce the computational effort, the orthogonal basis functions $\{z_i^{KL}\}$ will be chosen goal-oriented, i.e. the individual impact of the eigenvectors on the target functional will be taken into account. This method is well established in the model reduction methods of dynamic systems and the adaptive mesh refinement (cf. [10]). Motivating this approach, figure 5.1 shows the influence of the first ten eigenvectors on the target functional in the 2D test case RAE2822 (Euler flow). The third eigenvector has no impact on the objective function, hence it can be rejected from the Karhunen-Loève basis and the dimension of the probability space can be reduced.

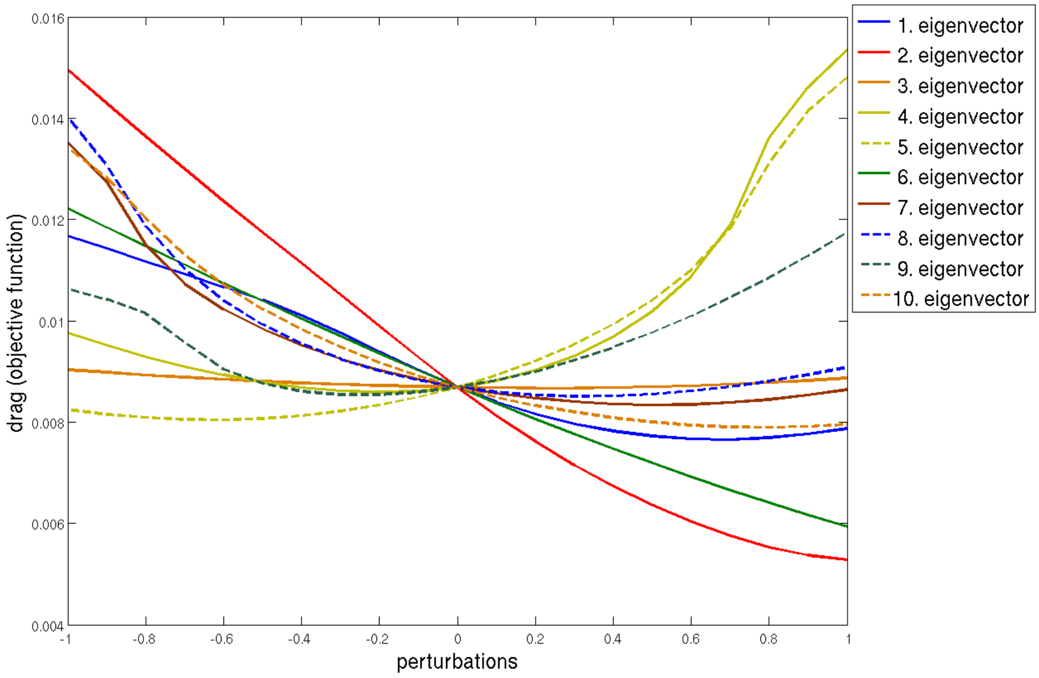


Figure 5.1: Influence of the first ten eigenvectors of the given random field ψ on the target functional (RAE2822 Euler flow).

Thus, the idea is to develop an error indicator for the individual eigenvectors reflecting the influence on the drag, following closely [124]. The introduced error analysis of the Karhunen-Loève expansion in section 5.2 only gives the approximation error of the random field ψ, but not of the function of interest f. We propose to use sensitivity information to capture the local behavior of the drag with respect to the eigenvectors

$$\eta_i := \frac{\mathrm{d}f}{\mathrm{d}z_i^{KL}} = -\lambda^{\top} \frac{\partial c}{\partial z_i^{KL}} + \frac{\partial f}{\partial z_i^{KL}}, \qquad \forall i = 1, \dots, d, \tag{5.28}$$

where λ solves the adjoint equation. The adjoint equation is independent of i, hence it has to be solved only once and the indicator η_i is numerically cheap to evaluate. Now, the reduced basis $\hat{z}^{KL}$ can be automatically selected, the eigenvector z_i^{KL} with a large value η_i has to be kept in the reduced basis, whereas a small value indicates that the basis vector can be rejected from the basis.

5.4 Polynomial chaos

Polynomial chaos expands the solution nonlinearly depending on the random vector $\boldsymbol{X}$ in a series of orthogonal polynomials with respect to the distribution of the random input vector $\boldsymbol{X}$

$$f(p, \boldsymbol{X}(\zeta)) = \sum_{i=1}^{\infty} \tilde{f}_i(p) \cdot \boldsymbol{\Phi}_i \left(\boldsymbol{X}(\zeta)\right) \tag{5.29}$$

with $\mathbf{\Phi}_i$ orthogonal polynomials, $\tilde{f}_i(p)$ deterministic coefficient functions. Again, the dependency of the objective function f on the state variables y is disregarded for simplicity reasons (cf. formulation (4.7 - 4.8)). As equation (5.29) indicates, the method allows a separation of the deterministic and stochastic part of the solution. In practice, one truncates the infinite expansion at a finite number of random variables and computes the statistics approximately by

$$f(p, \boldsymbol{X}(\zeta)) \approx \sum_{i=1}^{N_{PC}} \tilde{f}_i(p) \cdot \mathbf{\Phi}_i\left(\boldsymbol{X}(\zeta)\right) . \tag{5.30}$$

First, we will shortly discuss the theory and computation of the orthogonal polynomials $\mathbf{\Phi}$, since they serve as a basis to approximate the stochastic outcome depending on the random input.

5.4.1 Orthogonal Polynomials in the Askey scheme

The Askey scheme classifies the hypergeometrical orthogonal polynomials, which satisfy some type of differential or difference equation. The optimality of these basis in the context of polynomial chaos derives from the orthogonality with respect to density functions of the random variables. First, we will discuss some basic properties of orthogonal polynomials.

Definition 5.4.1 (Orthogonal System of Polynomials)**.** *A system of polynomials $\{\Phi_n : n \in I\}$, where Φ_n is a polynomial of exact degree n and $I = \mathbb{N}$ or $I = \{0, 1, 2, \dots, N\}$, is an orthogonal system of polynomials with respect to some real positive weighting function ω, if it holds*

$$\langle \Phi_n, \Phi_m \rangle = \int_S \Phi_n(x)\Phi_m(x)\omega(x)\,\mathrm{d}x \begin{cases} = 0, & \text{if } n \neq m \\ \neq 0, & \text{if } n = m \end{cases}, \qquad \forall m, n \in I, \tag{5.31}$$

where S denotes the support of the weighting function ω. The system $\{\Phi_n : n \in I\}$ is called orthonormal, if $\langle \Phi_n, \Phi_n \rangle = 1$, $\forall n \in I$.

Given a probability space $(\mathcal{O}, Y, P)$, a real-valued random variable on $X : \mathcal{O} \to \mathbb{R}$ with probability density function $f_X : \mathbb{R} \to \mathbb{R}_+$, then a system of polynomials is orthogonal with respect to f_X, if

$$\begin{aligned} \langle \Phi_n, \Phi_m \rangle &= \int_{\mathcal{O}} \Phi_n(X(\zeta))\Phi_m(X(\zeta))\,\mathrm{d}\mathcal{P}(\zeta) \\ &= \int_{\mathbb{R}} \Phi_n(x)\Phi_m(x)f_X(x)\,\mathrm{d}x \begin{cases} = 0, & \text{if } n \neq m \\ \neq 0, & \text{if } n = m \end{cases}, \qquad \forall m, n \in I. \end{aligned}$$

All orthogonal polynomials $\{\Phi_n\}$ fulfill the three term recurrence relation

$$-x\Phi_n(x) = b_n\Phi_{n+1}(x) + d_n\Phi_n(x) + c_n\Phi_{n-1}(x)\,, \qquad n \geq 1,$$

where $b_n, c_n \neq 0$ and $c_n/b_{n-1} > 0$ (cf. [161]).

In the generalized polynomial chaos theory, the basis of orthonormal polynomials, in which the solution of the stochastic PDE is expanded, is constructed with respect to the probability density function as weighting function. The Askey scheme provides a classification of orthogonal polynomials with respect to different weighting functions, some of them identical to standard probability density functions, and indicates the limit relations between them. The polynomials are classified according to the generalized hypergeometric series defined by

$$
{}_rF_s(a_1, \dots, a_r; b_1, \dots, b_s; z) := \sum_{k=0}^{\infty} \frac{(a_1)_k \cdots (a_r)_k z^k}{(b_1)_k \cdots (b_s)_k k!}\,, \tag{5.32}
$$

where $b_i \neq 0, -1, -2, \dots, \ \forall i = 1, \dots, s$ ensuring the non-negativity of the denominator factors, $(\cdot)_n$ is the Pochhammer symbol given by

$$
(a)_n = \begin{cases} 1, & \text{if } n = 0 \\ a(a+1)\dots(a+n-1), & \text{if } n = 1, 2, 3, \dots \end{cases}.
$$

The hypergeometric series becomes finite and hence a hypergeometric polynomial, if one of the numerator parameter a_i is a negative integer. The Askey scheme which is presented in figure 5.2 classifies the polynomials according to the classes ${}_rF_s$, where the number in parentheses is the number of free parameters. The lines connecting different polynomials denote the limit transition relationship between them, that means the polynomials at the end of a line can be computed by taking the limit of the polynomials on the upper part. The following table 5.1 shows for some standard probability functions the corresponding hypergeometric polynomials of the Askey scheme [44]. More details can be found in [94, 173].

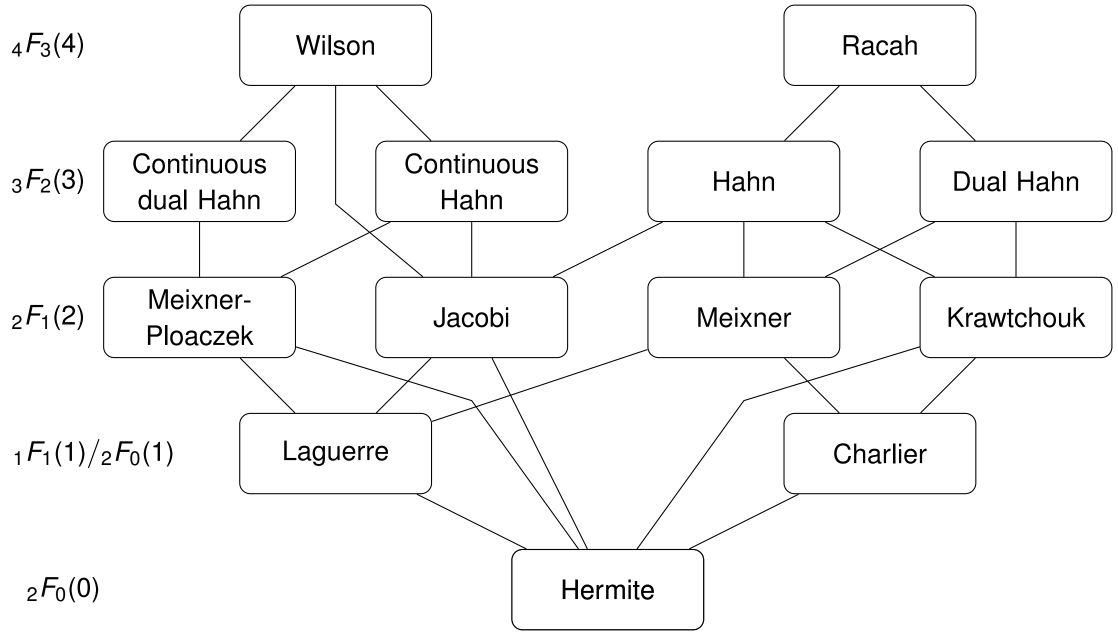


Figure 5.2: Askey scheme.

distribution	density function	hypergeometric polynomial	weight function
Gauß	$\frac{1}{\sqrt{2\pi}}\exp(\frac{-x^2}{2})$	Hermite	$\exp(\frac{-x^2}{2})$
Exponential	$\exp(-x)$	Laguerre	$\exp(-x)$
Beta	$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du}$	Jacobi	$(1-x)^{\alpha}(1+x)^{\beta}$

Table 5.1: Standard probability functions and the corresponding hypergeometric polynomials.

Considering d independent random variables, the generalization from the one-dimensional to the multi-dimensional case of orthogonal polynomials is straightforward. Given probability density functions f_{X_j} of d independent real-valued random variables $X_1, \dots, X_d$, the joint probability density function of $\boldsymbol{X} = (X_1, \dots, X_d)$ can thus be written as

$$f_{\boldsymbol{X}}(x_1, \dots, x_d) = \prod_{j=1}^{d} f_{X_j}(x_j)\,.$$

Using multi-indices $\mathbf{i} = (i_1, \dots, i_d) \in I^d$, an orthogonal system of multi-dimensional polynomials $\{\boldsymbol{\Phi}_{\mathbf{i}} : \mathbf{i} \in I^d\}$ is defined as

$$\boldsymbol{\Phi}_{\mathbf{i}}(\boldsymbol{X}(\zeta)) = \prod_{j=1}^{d} \Phi^{j}_{i_j}(X_j(\zeta))\,,$$

where $\{\Phi^j_n : n \in I\}$ is an orthogonal system of one-dimensional polynomials of X_j. It holds

$$\begin{aligned}\langle \boldsymbol{\Phi}_{\mathbf{i}}, \boldsymbol{\Phi}_{\mathbf{j}} \rangle &= \int_{\mathbb{R}^d} (\prod_{k=1}^{d} \Phi^k_{i_k}(x_k)\Phi^k_{j_k}(x_k))(\prod_{k=1}^{d} f_{X_k}(x_k))\, \mathrm{d}(x_1, \dots, x_d) \\ &= \prod_{k=1}^{d} \langle \Phi^k_{i_k}, \Phi^k_{j_k} \rangle \begin{cases} = 0, & \text{if } \mathbf{i} \neq \mathbf{j} \\ \neq 0, & \text{if } \mathbf{i} = \mathbf{j} \end{cases}.\end{aligned}$$

Since the geometrical uncertainties are modeled as a Gaussian random field (cf. section 3.4.2), the finite number of random variables resulting from the Karhunen-Loève expansion are independent Gaussian random variables. Hence, the stochastic outcome is expanded in a series of Hermite polynomials. The first few multi-dimensional Hermite polynomials for a two-dimensional case are given as

$$
\begin{aligned}
\mathbf{H}_{(0,0)}(X_1, X_2) &= H_0(X_1)H_0(X_2) = 1\\
\mathbf{H}_{(1,0)}(X_1, X_2) &= H_1(X_1)H_0(X_2) = X_1\\
\mathbf{H}_{(0,1)}(X_1, X_2) &= H_0(X_1)H_1(X_2) = X_2\\
\mathbf{H}_{(1,1)}(X_1, X_2) &= H_1(X_1)H_1(X_2) = X_1X_2\\
\mathbf{H}_{(2,0)}(X_1, X_2) &= H_2(X_1)H_0(X_2) = X_1^2 - 1\\
\mathbf{H}_{(0,2)}(X_1, X_2) &= H_0(X_1)H_2(X_2) = X_2^2 - 1\\
&\vdots
\end{aligned}
$$

The theory of polynomial chaos was first developed for the analysis of stochastic functions depending on Gaussian random variables and later generalized to arbitrary distribution functions. In the following, we will concentrate on the so called homogeneous polynomial chaos and refer to the literature concerning the generalized polynomial chaos theory [105, 176, 183].

5.4.2 Homogeneous polynomial chaos

Homogeneous polynomial chaos expanding a nonlinear function depending on Gaussian random variables in a series of Hermite polynomials was originally introduced by Wiener [178]. As shown in the last section, the multi-dimensional Hermite polynomials, also referred to as Wick-polynomials, defined as

$$
\mathbf{H}_{\mathbf{i}}(X_1, \dots, X_d) = \prod_{k=1}^{d} H_{i_k}(X_k) \tag{5.33}
$$

provide an orthogonal system.

The following theorem by Cameron and Martin [27] ensures the convergence of the Hermite expansion in the L^2 sense.

Lemma 5.4.2 (Cameron/Martin)**.** *Given a functional $R : \mathbb{R}^d \to \mathbb{R}$ in L^2 with respect to the Gaussian density function $\varphi_{\boldsymbol{X}}$, that means*

$$
\int_{\mathbb{R}^d} R(x)^2 \varphi_{\boldsymbol{X}}(x)\,\mathrm{d}x < \infty\,. \tag{5.34}
$$

Then it holds

$$
\lim_{M\to\infty} \int_{\mathbb{R}^d} \left(R(x) - \sum_{k=0}^{M} \sum_{|\boldsymbol{i}|=k} \tilde{r}_{\boldsymbol{i}} \boldsymbol{H}_{\boldsymbol{i}}(x) \right)^2 \varphi_{\boldsymbol{X}}(x)\,\mathrm{d}x = 0\,, \tag{5.35}
$$

where

$$
\tilde{r}_{\boldsymbol{i}} = \frac{1}{\int_{\mathbb{R}^d} \boldsymbol{H}_{\boldsymbol{i}}(x)^2 \varphi_{\boldsymbol{X}}(x) dx} \int_{\mathbb{R}^d} R(x) \boldsymbol{H}_{\boldsymbol{i}}(x) \varphi_{\boldsymbol{X}}(x)\,\mathrm{d}x\,. \tag{5.36}
$$

The proof can be found in [27].

It directly follows that the Hermite polynomials form an orthogonal basis of L^2. Considering a random variable $Y := R(\mathbf{X})$, it holds in the L^2 sense

$$Y(\boldsymbol{X}(\zeta)) = \sum_{k=0}^{\infty} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}} \, \mathbf{H}_{\mathbf{i}}(\boldsymbol{X}(\zeta)) \, . \tag{5.37}$$

The truncation of the polynomial chaos expansion (5.37)

$$Y(\boldsymbol{X}(\zeta)) \doteq \sum_{k=0}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}} \, \mathbf{H}_{\mathbf{i}}(\boldsymbol{X}(\zeta)) \tag{5.38}$$

is then called a stochastic discretization.

If the polynomial chaos coefficients $\tilde{r}_{\mathbf{i}}$ are known, the mean of Y can be approximated by

$$\begin{aligned} \mathbb{E}(Y) &\doteq \int_{\mathbb{R}^d} \sum_{k=0}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}} \, \mathbf{H}_{\mathbf{i}}(x) \varphi_{\boldsymbol{X}}(x) \, \mathrm{d}x \\ &= \tilde{r}_{\mathbf{0}} \end{aligned}$$

using the orthogonality $\mathbf{H}_{\mathbf{i}}$. The variance can be approximated by

$$\begin{aligned} \mathbb{V}ar(Y) &= \mathbb{E}(Y^2) - (\mathbb{E}(Y))^2 \\ &\doteq \int_{\mathbb{R}^d} \left(\sum_{k=0}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}} \mathbf{H}_{\mathbf{i}}(x) \right)^2 \varphi_{\boldsymbol{X}}(x) \, \mathrm{d}x - \tilde{r}_{\mathbf{0}}^2 \\ &= \sum_{k=0}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}} \sum_{l=0}^{M} \sum_{|\mathbf{j}|=l} \tilde{r}_{\mathbf{l}} \int_{\mathbb{R}^d} \mathbf{H}_{\mathbf{i}}(x) \mathbf{H}_{\mathbf{j}}(x) \varphi_{\boldsymbol{X}}(x) \, \mathrm{d}x - \tilde{r}_{\mathbf{0}}^2 \\ &= \sum_{k=0}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}}^2 \int_{\mathbb{R}^d} \mathbf{H}_{\mathbf{i}}^2(x) \varphi_{\boldsymbol{X}}(x) \, \mathrm{d}x - \tilde{r}_{\mathbf{0}}^2 \\ &= \sum_{k=1}^{M} \sum_{|\mathbf{i}|=k} \tilde{r}_{\mathbf{i}}^2 \langle \mathbf{H}_{\mathbf{i}}^2 \rangle \, . \end{aligned}$$

Higher moments and the covariance can also efficiently be determined using the truncated polynomial chaos expansion. The main advantage of the polynomial chaos approach in contrast to direct integration methods, like Monte-Carlo or sparse grid techniques, is due to the fact that the computation of statistics of the solution are very cheap and results are highly accurate.

As mentioned before, we deal with Gaussian random fields in our application, hence the focus of this chapter is on polynomial chaos expansion using Hermite polynomials. Nevertheless, a short survey of further developments of the homogeneous polynomial chaos will be given in the next section in order to emphasize the wide applicability and well-founded theory of polynomial chaos in uncertainty quantification tasks.

5.4.3 Further developments of the Homogeneous Chaos

Xiu and Karniadakis [182] introduce the generalized polynomial chaos, also called Askey Chaos, proposing a more general framework of polynomial chaos employing the classes of orthogonal polynomials of the Askey scheme (cf. figure. 5.2), including the Hermite chaos as a subset. Due to the closed connection between the orthogonal polynomials in the Askey scheme to the probability density function of certain random distributions (cf. table 5.1), the generalized polynomial chaos allows to represent many non-Gaussian stochastic processes, including some discrete processes. The convergence in the L^2 sense can be proved by a generalization of the theorem 5.4.2 (further details can be found in [27, 135, 183]).

Furthermore, Wan and Karniadakis propose a multi-element generalized polynomial chaos representation, which is able to deal with arbitrary probability measures [176]. The approach is based on a decomposition of the probability space and a locally constructed generalized polynomial chaos expansion in each subdomain.

In order to overcome discontinuities, Maître et al. suggest a combination of the polynomial chaos approach with wavelets [104, 103]. The proposed method relies on an orthogonal projection of the uncertain data and solution variables onto a multi-wavelet basis. Due to the high amount of computational effort, the approach is combined with an adaptive refinement strategy of multi-wavelet basis.

5.4.4 Polynomial chaos efficiency

The convergence of the polynomial chaos expansion is ensured by the theorem of Cameron and Martin, cf. section 5.4.2, and the generalized version respectively. In several papers, the convergence rate depending on the order of the polynomials used in the polynomial chaos expansion is numerically examined. [8] and [182] numerically show exponential convergence of the expansion, but one can also show applications, where an adaptivity strategy or problem-dependent improvements of the expansion are needed to achieve a good performance of the polynomial chaos expansion (cf. [50]). The total number of expansion terms increases fast for large dimensional problems, thus even with exponential convergence, the polynomial chaos method is computationally effective, only if the number of input random variables is not too large. Techniques reducing the number of random variables can circumvent this limitation. As discussed in section 5.3, methods investigating the influence of the individual random variables on the output have the potential of reducing the dimension of the probability space, cf. [122, 124].

In the last section, a general framework using polynomial chaos expansion was discussed. We will now concentrate on the polynomial chaos approach solving stochastic PDEs, which we will use later on to determine statistics of the flow under uncertain conditions.

5.5 Polynomial chaos solving SDEs

The basic idea of polynomial chaos representing the stochastic output of a differential equation with random input data is to reformulate the SDE replacing the solution and the right hand side of the PDE by a polynomial chaos expansion. The deterministic polynomial chaos coefficients can then be computed using different approaches which we will introduce later on. Given a stochastic differential equation of the form

$$\mathcal{D}(x,t,X(\zeta);y) = g(x,t,X(\zeta))\,, \qquad x \in G,\ t \in [0,T],\ \zeta \in \mathcal{O}, \tag{5.39}$$

where $y = y(x,t,X(\zeta))$ is the solution and $g(x,t,X(\zeta))$ is the right hand side. The operator $\mathcal{D}$ can be nonlinear and appropriate initial and boundary conditions are assumed. Replacing pointwise the solution $y = y(x,t,X(\zeta))$ of equation (5.39) by the polynomial chaos expansion leads to

$$y(x,t,X(\zeta)) \doteq \sum_{k=1}^{M} \tilde{y}_k(x,t)\Phi_k(X(\zeta))\,,$$

where $\tilde{y}_k(x,t)$ are the deterministic polynomial chaos coefficients, which need to be determined. Furthermore, the right hand side $g(x,t,X(\zeta))$ will also be expanded by a polynomial chaos expansion

$$g(x,t,X(\zeta)) = \sum_{k=1}^{M} \tilde{g}_k(x,t)\Phi_k(X(\zeta))$$

and the deterministic coefficients are given by

$$\tilde{g}_k(x,t) = \frac{\langle g(x,t,\cdot),\Phi_k\rangle}{\langle\Phi_k^2\rangle}$$

according to section 5.4.2.

In literature, two different classes of methods determining the unknown coefficients $\tilde{y}_k(x,t)$ are proposed: non-intrusive and intrusive polynomial chaos, see reviews [81, 128, 181]. An intrusive approach calculates the coefficients by projecting the resulting equations onto basis functions for different modes. As the name indicates, the intrusive methods require a modification of the existing code solving the deterministic PDE (if the Operator $\mathcal{D}$ is nonlinear) [75]. Non-intrusive methods overcome this drawback and are therefore investigated by many researchers.

5.5.1 Intrusive polynomial chaos solving SDEs

A common approach evaluating the deterministic polynomial chaos coefficients $\tilde{y}_k(x,t)$ is a stochastic Galerkin method [82, 89, 122]. First, the polynomial chaos variables y, g are replaced by their polynomial chaos expansion in the stochastic differential equation (5.39)

$$\mathcal{D}(x,t,X(\zeta);\sum_{k=0}^{M}\tilde{y}_k\Phi_k) = \sum_{k=0}^{M}\tilde{g}_k\Phi_k\,, \qquad x \in G,\ t \in [0,T],\ \zeta \in \mathcal{O}. \tag{5.40}$$

Then, a Galerkin projection of equation (5.40) onto each orthogonal polynomial Φ_j leads to

$$\langle\mathcal{D}(x,t,X(\zeta);\sum_{k=0}^{M}\tilde{y}_k\Phi_k),\Phi_j\rangle = \langle\sum_{k=0}^{M}\tilde{g}_k\Phi_k,\Phi_j\rangle\,, \quad j = 0,\dots,M \tag{5.41}$$

ensuring the orthogonality of the error to the functional space spanned by $\{\Phi_j\}_{j=0}^{M}$. Using the orthogonality of the polynomial system $\{\Phi_j\}_{j=0}^{M}$, the resulting equations are a set of coupled deterministic differential equations which can be solved by standard methods. In most cases (if $\mathcal{D}$

is nonlinear), the resulting deterministic differential equations differ from the original deterministic differential equation, so that existing solvers cannot be applied. In [82], a convergence proof of the Galerkin approximation using the usual theory (of the deterministic Galerkin method) is given. Results concerning the convergence rate of the stochastic Galerkin method can be found in [8] and [14]. In [89], a general overview of intrusive polynomial chaos applications in CFD is provided, especially the coupled deterministic PDEs resulting from the stochastic Navier-Stokes equation are derived. Further applications of intrusive polynomial chaos are discussed e.g. in [102, 106, 121].

Several papers show the efficiency of intrusive methods in CFD, but the application of stochastic Galerkin methods to the Navier-Stokes equations requires the modification of the existing flow solver. Since we use a highly sophisticated solver developed at the DLR within several years, this drawback of the method forces us to consider non-intrusive polynomial chaos methods, which allow to use the flow solver as a black-box.

5.5.2 Non-intrusive polynomial chaos solving SDEs

The main objective of non-intrusive polynomial chaos methods is to obtain approximations of the deterministic coefficients $\tilde{y}_k(x, t, \cdot)$ without making any modifications to the existing solver of the deterministic differential equation. All non-intrusive approaches are based on the definition of the deterministic coefficients in the theorem of Cameron and Martin

$$\begin{aligned} \tilde{y}_k(x,t) &= \frac{\langle y(x,t,\cdot), \Phi_k \rangle}{\langle \Phi_k^2 \rangle} && (5.42) \\ &= \frac{1}{\langle \Phi_k^2 \rangle} \int_{\mathbb{R}^d} y(x,t,\nu)\Phi_k(\nu) f_X(\nu)\,\mathrm{d}\nu\,, \qquad x \in G,\ t \in [0,T],\ k = 1,\dots,M. && (5.43) \end{aligned}$$

The unknown coefficients can now be approximated by numerically evaluating the integral of equation (5.43). In the following chapter 6, several methods discretizing the probability space will be discussed.

Chapter 6

Discretization of the Probability Space

Statistics of quantities depending on the current design vector and the uncertain parameters are required in each iteration of the robust optimization. Since the multi-dimensional integrals (cf. (4.12) and (5.43)) cannot be solved analytically, we have to approximate them in an appropriate, efficient way. In literature, several possibilities in order to numerically compute the required integrals are proposed. The most common are: Monte Carlo simulation, respectively general Sampling methods, full tensor grid interpolation and sparse grid interpolation. Their efficiency depends on the dimension d of the probability space $\mathcal{O}$ and on the properties of the integrand. Exemplarily, we will consider the numerical computation of the mean value of the drag $\mathbb{E}(f)$ depending on geometrical uncertainties approximated by d random variables, which is the objective as e.g. in the semi-infinite formulation (4.12 - 4.14). The introduced methods can further be used to determine the variance or higher moments and the deterministic coefficients of the PC representation (cf. (5.43)).

Each of the discretization methods provides an approximation $\mathbb{E}_N$ of the mean value $\mathbb{E}(f)$ by evaluating the function f in N integration points $\psi_d^1, \cdots, \psi_d^N$ and summing up the results $f(y_i, p, \psi_d^i)$ multiplied with weights $\omega_1, \cdots, \omega_N$;

$$\mathbb{E}_N = \sum_{i=1}^{N} \omega_i \cdot f(y_i, p, \psi_d^i)\,.$$

In this chapter, a survey of existing discretization methods will be given and afterwards, we will focus on sparse grid methods as an efficient discretization of the probability space for the fast numerical evaluation of the statistics.

6.1 Sampling methods

The Sampling methods randomly select realizations of the uncertainties in the given probability space and take some kind of average of the function values at these points, which converges to the exact value of the integral due to the law of large numbers. The advantage of this approach consists of the straightforward implementation, the algorithm only needs the underlying integration space as input and function evaluations at the randomly selected points. But on the other hand, the expected convergence rate $O(N^{-\frac{1}{2}})$ requires a large number of function evaluations to ensure a given error tolerance. For this reason, several improvements of this method have been developed, which can be classified into the following two classes: variance reduction techniques and quasi-random or low-discrepancy sampling [107].

The main techniques for variance reduction are importance sampling and conditional Monte Carlo. The idea behind the variance reduction methods is to use additional information in order to generate more accurate estimators reducing the variance of the integrand. The key is to find another function, whose integrand is equal to the integrand of f, but whose variance is smaller than the one of f. In addition to the smaller variance, the methods often lead to an improvement in the efficiency, that means the variance and the computation time will be reduced. Details can be found in [68, 107, 150].

On the other hand, the quasi Monte Carlo methods seek to construct nodes that perform significantly better than the average [131], which means that the nodes are deterministically chosen, such that a small error is guaranteed. The error bound of the method can be improved to $O(N^{-1}(log\ N)^{d-1})$, if the nodes are properly chosen [131]. The main difference between Monte Carlo methods and quasi Monte Carlo methods is that quasi Monte Carlo methods are completely deterministic, thus the error bounds are also deterministic. Moreover, the quasi Monte Carlo method can significantly reduce the computational effort. Detailed information about these methods can be found in [62, 68, 131] or [111].

In our application, one function evaluation is very expensive, since the solution of the flow equation, Euler or Navier-Stokes equation, is needed. The Sampling methods, even the improved methods, are not an appropriate choice in our case to compute statistics of the flow, since they do not offer the possibility to improve the approximation quality by additional knowledge of the underlying function, e.g. smoothness information.

6.2 Full tensor grids

Another possibility to obtain the objective value is the full tensor grid quadrature derived from the full tensor product of the one-dimensional interpolation formulas. Constructing the multi-dimensional interpolation, we first consider the following one-dimensional interpolation formula in order to approximate a function $g : [-1, 1] \to \mathbb{R}$:

$$\mathcal{I}^i(g) = \sum_{j=1}^{m_i} g(Y_j^i) \cdot a_j^i$$

with the set of interpolation points $G^i = \{Y_j^i | Y_j^i \in [-1, 1],\ j = 1, 2, \dots, m_i\}$, m_i is the number interpolation points and $a_j^i \equiv a_j(Y_j^i)$ are the interpolation functions. The full tensor grid interpolation of a d-dimensional function is then given by

$$(\mathcal{I}^{i_1} \otimes \cdots \otimes \mathcal{I}^{i_d})(f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_d=1}^{m_d} f(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}) \cdot (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d})\,. \tag{6.1}$$

This generalization of the one-dimensional formula to the full tensor interpolation (6.1) provides an approximation of $f : [-1, 1]^d \to \mathbb{R}$ by evaluating the function f on the regular mesh $G^{i_1} \times \cdots \times G^{i_d}$. Considering the difference formulas defined by

$$\begin{aligned} \Delta^i &:= \mathcal{I}^{i+1} - \mathcal{I}^i \\ \mathcal{I}^0 &:= 0 \end{aligned}$$

with $G^{\Delta i} = G^{i+1} \setminus G^{i}$, (6.1) can be reformulated as

$$(\mathcal{I}^{i_1} \otimes \cdots \otimes \mathcal{I}^{i_d})(f) = \sum_{i_1,\dots,i_d \leq k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f)\ . \tag{6.2}$$

The mean value is then derived from the following equation:

$$\mathbb{E}_N(f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_d=1}^{m_d} f(Y^{i_1}_{j_1}, \dots, Y^{i_d}_{j_d}) \cdot \int_{[-1,1]^d} (a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_d}_{j_d})(\mathbf{Y})\, \mathrm{d}\mathbf{Y}\,.$$

Directly using quadrature formulas leads to

$$\begin{aligned} \mathbb{E}_N(f) &= (\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_d})(f) \\ &= \sum_{i_1,\dots,i_d \leq k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f)\ , \end{aligned}$$

where $\Delta^{i} := \mathcal{Q}^{i+1} - \mathcal{Q}^{i}$, $\mathcal{Q}^{0} := 0$. The approximation error for functions with bounded derivatives up to order k has a behavior of $O(N^{-\frac{k}{d}})$, cf. [35]. The resulting full tensor grids using linear, Clenshaw-Curtis and Gauß-Hermite abscissas are shown in the figure 6.1.

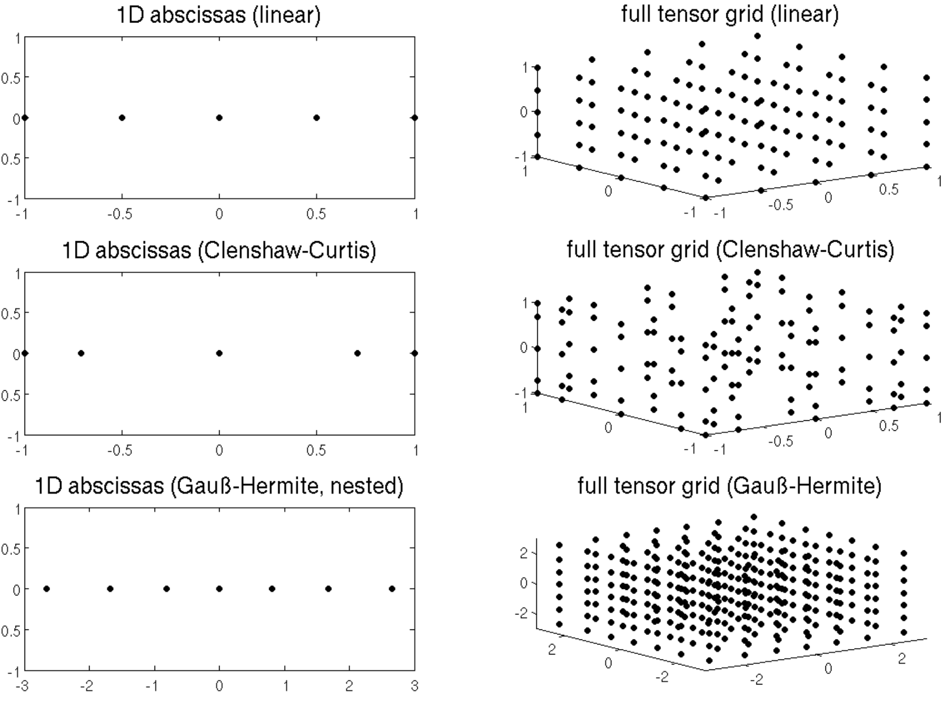


Figure 6.1: 1D abscissas of the linear, Clenshaw-Curtis and Gauß-Hermite formulas and the corresponding full tensor grids in 3D.

Due to the exponential growth of the effort with increasing dimension, this method is not suitable for high stochastic dimensions, which is the case in our application. To circumvent this curse of dimensionality, we introduce a sparse grid method in order to preserve the accuracy of the tensor grid quadrature, but avoiding the exponential growth of interpolation nodes.

6.3 Sparse grids

The idea of sparse grids is to combine quadrature formulas of high order in some dimensions with quadrature formulas of lower order in the other dimensions in such a way that the interpolation error is nearly the same as for full tensor products. The method was originally developed for the solution of partial differential equations and is now successfully used for integral equations, interpolation and approximation problems [52]. The underlying idea of sparse grids was originally found by the Russian mathematician Smolyak [168]. The sparse integrand is given as [137]:

$$\mathcal{S}(k,d)(f) = \sum_{k-d+1\leq|\mathbf{i}|\leq k} (-1)^{k-|\mathbf{i}|} \cdot \binom{d-1}{k-|\mathbf{i}|} \cdot (\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_d})(f) \tag{6.3}$$

with $k \geq d$, $\mathbf{i} \in \mathbb{N}^d$ multi-index and $|\mathbf{i}| = \sum_{j=1}^{d} i_j$. The index i_j indicates the order in the j-th dimension, so the algorithm combines only those one-dimensional quadrature formulas, whose indices fulfill the constraint that the total sum across all dimensions is greater or equal $k-d+1$ and smaller or equal k. Using incremental interpolation formulas $\Delta^{\mathbf{i}}$, (6.3) can be transformed to

$$\begin{aligned} \mathcal{S}(k,d)(f) &= \sum_{|\mathbf{i}|\leq k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) & (6.4)\\ &= \mathcal{S}(k-1,d)(f) + \sum_{|\mathbf{i}|=k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) & (6.5)\end{aligned}$$

with $\Delta^i = \mathcal{Q}^{i+1} - \mathcal{Q}^i$, $\mathcal{Q}^0 \equiv 0$ and $\mathcal{S}(d-1,d) \equiv 0$. The collection of all the interpolation points

$$\mathcal{H}(k,d) = \bigcup_{k-d+1\leq|\mathbf{i}|\leq k} \left(G^{i_1} \times \cdots \times G^{i_d}\right) \tag{6.6}$$

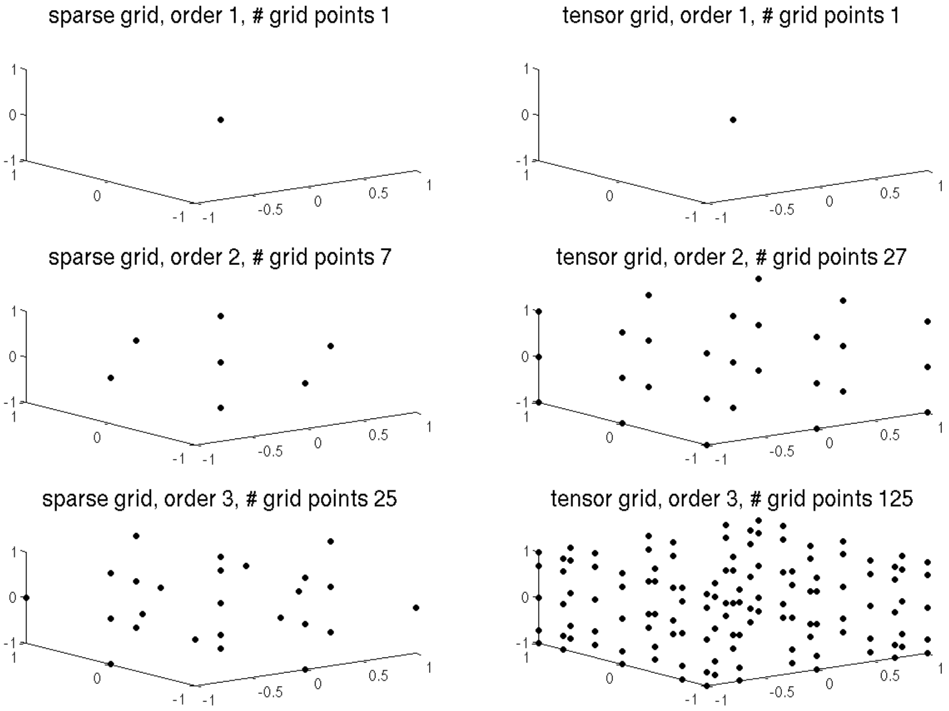
is called a sparse grid of level k. If we compare the two incremental formulas (6.2) and (6.5), the difference between the full tensor grid and the sparse grid interpolation becomes clearer. Whereas the full tensor interpolation combines the one-dimensional interpolation function in each dimension up to order k, the Smolyak algorithm constructs the sparse grid using only those functions which sum up to order k in all dimensions. The derivation of the sparse grid suggests the use of nested interpolation functions due to the recursive construction. To compute the sparse grid $\mathcal{S}(k,d)$, one only needs to add to the old sparse grid $\mathcal{S}(k-1,d)$ the function evaluations at the new points $\Delta\mathcal{H}(k,d) = \bigcup_{|\mathbf{i}|=k}\left(G^{i_1}_{\Delta} \times \cdots \times G^{i_d}_{\Delta}\right)$, where $G^j_{\Delta} = G^j \backslash G^{j-1}$ are the differential points of the one-dimensional interpolation functions. Therefore, the Smolyak approximation, which employs nested points, requires less function evaluations than the corresponding formula with non-nested points. In literature, the most popular choice of the collocation points is the Clenshaw-Curtis grid at the non-equidistant extrema of the Chebyshev polynomials. The corresponding underlying interpolation formula is the Chebyshev-Gauß-Lobatto formula. These knots are given by

$$\begin{aligned} Y_j^i &= -cos\left(\frac{\pi(j-1)}{m_i-1}\right), \qquad j = 1, \dots, m_i, \qquad &\text{if } m_i > 1 \\ Y_1^i &= 0, &\text{if } m_i = 1 \end{aligned}$$

with $m_1 = 1$ and $m_i = 2^{i-1} + 1$, for $i > 1$. The corresponding basis functions are given by:

$$a_j^i = \begin{cases} 1, & \text{for } i = 1 \\ \prod\limits_{\substack{k=1 \\ k \neq j}}^{m_i} \frac{Y - Y_k^i}{Y_j^i - Y_k^i}, & \text{for } i > 1 \text{ and } j = 1, \dots, m_i \end{cases} .$$

A comparison of the full tensor grid and the sparse grid using Clenshaw-Curtis formulas is shown in figure 6.2.


sparse grid, order 1, # grid points 1

tensor grid, order 1, # grid points 1

sparse grid, order 2, # grid points 7

tensor grid, order 2, # grid points 27

sparse grid, order 3, # grid points 25

tensor grid, order 3, # grid points 125

Figure 6.2: Comparison of the full tensor grid and sparse grid using Clenshaw-Curtis formulas up to order 3.

To exhibit the efficiency of the isotropic sparse grid derived from equation (6.3), we briefly discuss the convergence rate compared to the full tensor grid below.

6.3.1 Error bounds

In order to derive error bounds for sparse grids of dimension $d > 1$, the known results for the one-dimensional case serve as a basis. The proofs of the following results and further error analysis can be found in [9, 55] and [100].

The error bound for one-dimensional interpolatory quadrature formulas with positive weights integrating a function $f \in \mathcal{C}^k([-1,1])$ is given by

$$\epsilon(N) = O\left(N^{-k}\right) , \tag{6.7}$$

where N is the number of quadrature nodes (cf. [35]).

For $d > 1$, function classes $\mathcal{W}_d^k$ with bounded mixed derivatives up to order k

$$\mathcal{W}_d^k = \{f : [-1,1]^d \to \mathbb{R}, \| \frac{\partial^{|\mathbf{s}|} f}{\partial x_1^{s_1} \ldots \partial x_d^{s_d}} \|_\infty < \infty,\ s_i \leq k\}$$

with $|\mathbf{s}| = \sum_{i=1}^d s_i$ are considered. Using one-dimensional interpolatory quadrature formulas with positive weights as basis for the Smolyak algorithm, the approximation quality of the resulting sparse grid is of order

$$\epsilon(N) = O\left(N^{-k} \cdot (\log N)^{(k+1)(d-1)}\right) \tag{6.8}$$

with N number of sparse grid points (cf. [177]). The sparse grid approach outperforms the conventional full tensor grids, which achieve an approximation quality of $O(N^{-\frac{k}{d}})$. Hence, the sparse grid approach overcomes the curse of dimensionality and is an appropriate choice for higher dimensional problems. Especially in combination with adaptive refinement strategies, the sparse grid method has the potential to reach a high approximation quality with less grid points than the other discussed sampling methods.

6.4 Adaptive sparse grids

Since the function evaluations are very expensive in our application, we introduce in this section an adaptive sparse grid strategy in order to further reduce the number of grid points but conserving the approximation quality. The presented isotropic Smolyak algorithm is effective for problems whose input data uniformly depend on all dimensions. But the convergence rate deteriorates for highly anisotropic problems, such as those appearing, when the input random variables come from a Karhunen-Loève expansion as in our application [45]. The reduction of computational effort can be achieved by using spatially adaptive or dimension adaptive refinement [55, 117]. In order to develop adaptive schemes during the cubature process, the interpolation error can be used as an adaptivity indicator. Therefore, nested cubature formulas are useful, since they allow the error evaluation based on the difference of two subsequent formulas. Due to the fact that in our application the mean value is computed by the sparse grid interpolation, this target value is also used as an error indicator for the adaptivity. The dimension adaptive quadrature method tries to find important dimensions and adaptively refines those dimensions with respect to given error estimators. This leads to an approach, which is based on generalized sparse grid index sets [55]. This strategy allows to employ arbitrary interpolation formulas, so they can be chosen problem dependent, e.g. in our application depending on the distribution of the random variables. On the other hand, the locally refined sparse grid gives more flexibility in the adaptive procedure, but requires equidistant support nodes. In the following, we will discuss both strategies and compare the resulting sparse grids in the numerical results later on.

6.4.1 Dimension adaptive sparse grids

The main advantage of the dimension adaptive refinement strategy is the fact that one can use problem dependent quadrature formulas in order to construct the adaptive sparse grid. In our application, the objective function, the drag, is multiplied by the Gaussian density function, so that Gaussian Hermite polynomials are optimal with respect to the weighting function.

First, a generalization of sparse grids will be introduced, which allows to weight the dimensions according to their importance on the target functional. The idea of generalized sparse grids and especially dimension adaptive sparse grid can be found in [25, 53, 55] and [87]. The original sparse grid of order k combines all the incremental functions, which sum up to order k, that means only those indices are considered, which are contained in the unit simplex $|\mathbf{i}| \leq k$. [51] and [55] suggest to allow a more general index set, which can be adaptively chosen with respect to the importance of each dimension.

An index set $\mathbf{I}$ is called *admissible* if $\forall \mathbf{i} \in \mathbf{I}$

$$\mathbf{i} - e_j \in \mathbf{I}, \qquad \forall 1 \leq j \leq d,\ i_j > 1,$$

where $e_j \in \mathbb{R}^d$ is the j-th unit vector. The generalized index set $\mathbf{I}$ contains for an index $\mathbf{i}$ all indices, which have smaller entries in one or more dimensions. Due to this fact, the incremental sparse grid formula (6.5) is still well defined for the new index sets and is given as

$$\mathcal{S}(k,d)(f) = \sum_{\mathbf{i}\in\mathbf{I}} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f)\ . \tag{6.9}$$

The generalized definition of sparse grids includes the original sparse grid and the full tensor grid definition (cf. (6.5), (6.2)). Further, equation (6.9) particularly leaves more flexibility to the choice of the grids and therefore allows to handle anisotropic problems, which the following example of an admissible index set in $\mathbb{R}^2$ illustrates:

$$\mathbf{I} = \left\{ \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}4\\1\end{pmatrix}, \begin{pmatrix}5\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix} \right\}.$$

This example of an admissible index set $\mathbf{I}$ shows the feasibility of a refinement in only one dimension (here in the first dimension), which is the required feature for the adaptivity.

If a priori knowledge of the underlying function is available, an admissible index set with respect to the additional information can be chosen. Since this is not the case in our application, an algorithm is introduced in the following, which automatically computes an admissible index set in a dimension adaptive way (cf. [51, 55]). Therefore, we start with the coarsest sparse grid, that means $\mathbf{I} = \{(0, \dots, 0)^\top\}$, and then successively add new indices, such that

- the new index set remains admissible
- the approximation error is reduced.

For the second point, an error indicator is needed. Taking a look at the difference formula (6.9), the term

$$\Delta_{\mathbf{i}}(f) = \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) \tag{6.10}$$

indicates the reduction in the approximated integral for each new added index. [55] suggests to further involve the number of function evaluations to avoid a too early stopping. Since equation (6.10) shows good results in our application, we directly use $\Delta_{\mathbf{i}}$ as an error indicator for the adaptivity.

Therefore, the algorithm starts with the coarse grid, computes new admissible indices and adds the index with the largest error to the grid. In order to compute admissible indices, one has to consider two index sets $\mathbf{I}_{old}$ and $\mathbf{I}_{active}$. The old index set $\mathbf{I}_{old}$ contains all indices added to the grid during the procedure. If an index $\mathbf{k}$ is added to $\mathbf{I}_{old}$, its forward neighborhood defined as $\{\mathbf{o} : \mathbf{o} = \mathbf{k} + e_j,\ 1 \leq j \leq d\}$ is checked for admissible indices and added to the active index set $\mathbf{I}_{active}$. That means, an index $\mathbf{k}$ can only be added to $\mathbf{I}_{active}$, if all its backward neighbors $\{\mathbf{o} : \mathbf{o} = \mathbf{k} - e_j,\ 1 \leq j \leq d\}$ are in the old index set and whose forward neighbors are not considered yet. In the next iteration, the index $\mathbf{k} \in \mathbf{I}_{active}$ with the largest error indicator is added to the old index set $\mathbf{I}_{old}$ and the admissible forward neighbors are added to $\mathbf{I}_{active}$. This procedure is repeated until a given error tolerance is reached. The concept of dimension adaptive refinement is outlined in algorithm 6.1.

Algorithm 6.1 Dimension adaptive sparse grid

```
function DIMASG
    $\mathbf{i} \leftarrow (0, \dots, 0)$
    $\mathbf{I}_{old} \leftarrow \emptyset$
    $\mathbf{I}_{active} \leftarrow \{\mathbf{i}\}$
    $w_{\mathbf{i}} \leftarrow |\Delta_{\mathbf{i}}(f)|$
    $\mathbf{w} \leftarrow |\Delta_{\mathbf{i}}(f)|$
    $\mathbb{E} \leftarrow 0$
    while $\mathbf{w} > tol$ do
        $\mathbf{I}_{active} \leftarrow \mathbf{I}_{active} \setminus \{\mathbf{i}\}$ with $\mathbf{i} : w_{\mathbf{i}} \geq w_{\mathbf{j}},\ \forall j \in \mathbf{I}_{active}, j \neq i$
        $\mathbf{I}_{old} \leftarrow \mathbf{I}_{old} \cup \{\mathbf{i}\}$
        $\mathbb{E} \leftarrow \mathbb{E} + \Delta_{\mathbf{i}}(f)$
        $\mathbf{w} \leftarrow \mathbf{w} - w_{\mathbf{i}}$
        for $j = 1, \dots, d$ do
            $\mathbf{o} \leftarrow \mathbf{i} + e_j$
            if $\mathbf{o} - e_m \in \mathbf{I}_{old},\ \forall 1 \leq m \leq d$ then
                $\mathbf{I}_{active} \leftarrow \mathbf{I}_{active} \cup \{\mathbf{o}\}$
                $w_{\mathbf{o}} \leftarrow |\Delta_{\mathbf{o}}(f)|$
                $\mathbf{w} \leftarrow \mathbf{w} + |\Delta_{\mathbf{o}}(f)|$
            end if
        end for
    end while
end function
```

The introduced algorithm differs from [51] in the return value. During the optimization loop, the grid induced by the old index set $\mathbf{I}_{old}$ is used, such that the approximated integral is also computed on the grid defined by the old index set $\mathbf{I}_{old}$ during the generation of the dimension adaptive sparse grid. The algorithm stops, if the sum of the error indicators of the new index set $\mathbf{I}_{active}$ is less than a given error tolerance. The mean value is then computed during the optimization by the following algorithm 6.2.

Algorithm 6.2 Computation of the mean using the dimension adaptive sparse grid defined by $\mathbf{I}_{old}$

```
function MEANDIMASG
Input: $\mathbf{I}_{old}$
    $int \leftarrow 0$
    for all $\mathbf{o} \in \mathbf{I}_{old}$ do
        $w_\mathbf{o} \leftarrow |\Delta_\mathbf{o}(f)|$
        $int \leftarrow int + w_\mathbf{o}$
    end for
end function
```

As mentioned before, the main advantage of the dimension adaptive refinement is the fact that the quadrature formulas can be chosen problem dependent. Considering geometrical uncertainties in the robust optimization, the Karhunen-Loève expansion leads to the following objective function

$$\mathbb{E}\left(f(p,\psi\left(x,\zeta\right))\right) \doteq \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} f(p,\psi_d\left(x_1,\dots,x_d\right))\varphi_1\left(x_1\right)\cdots\varphi_1\left(x_d\right)\ \mathrm{d}x_1\cdots\mathrm{d}x_d\,,$$

so that the Gauß-Hermite formulas are an appropriate choice for the quadrature.

The one-dimensional Hermite polynomials are orthogonal polynomials over $(-\infty,\infty)$ with the weighting function $\omega(x) = \exp(-x^2)$. The Gauß-Hermite quadrature belongs to the class of Gauß formulas, which are constructed by choosing both, the points and the weights, with the goal to exactly integrate as many polynomials as possible. The Gauß formulas achieve the polynomial exactness of $2n-1$, where n is the number of abscissas of the quadrature rule. In Gauß-Hermite quadrature, the integral of the form $\int_{-\infty}^{\infty} f(x)\exp(-x^2)\,\mathrm{d}x$ is approximated by

$$\int_{-\infty}^{\infty} f(x)\exp(-x^2)\,\mathrm{d}x \approx \sum_{i=1}^{m} \omega_i f(x_i)\,,$$

where the abscissas x_i are zeros of the m-th Hermite polynomial and the ω_i are the corresponding weights. The one-dimensional Hermite polynomials are defined as

$$H_n(x) = (-1)^n \exp(x^2)\frac{\mathrm{d}^n}{\mathrm{d}x^n}\exp(-x^2) \tag{6.11}$$

and the weights

$$\omega_i = \frac{2^{n-1}n!\sqrt{\pi}}{n^2 H_{n-1}(x_i)^2}\,. \tag{6.12}$$

The Gauß-Hermite quadrature formulas are weakly nested, that means the rules of odd order all include the abscissa 0. Since the nesting is a favorable feature constructing the sparse grid, this property will be taken into account. Figure 6.3 shows the abscissas of the Gauß-Hermite quadrature of order $1, 3, 7, 15$, which we will use to construct the dimension adaptive sparse grid. As an

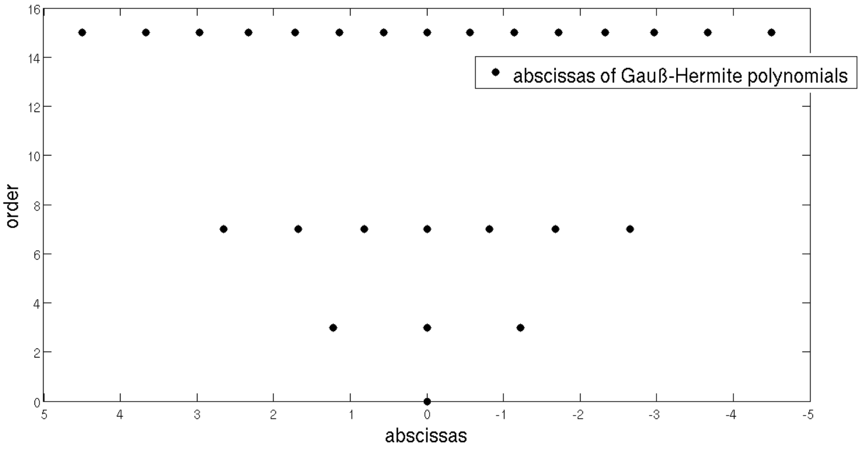


Figure 6.3: Abscissas of Gauß-Hermite polynomials of order 1, 3, 7, 15.

example, we consider the integration of the following function

$$\int_{(-\infty,\infty)} \int_{(-\infty,\infty)} \frac{1}{2}(\exp(x_1) + x_2) \exp(-x_1^2) \exp(-x_2^2) \, \mathrm{d}x_1 \mathrm{d}x_2 \,. \tag{6.13}$$

Figure 6.4 shows that the algorithm detects the nonlinearity in the first dimension and automatically refines until the given error tolerance 10^{-3} is reached.

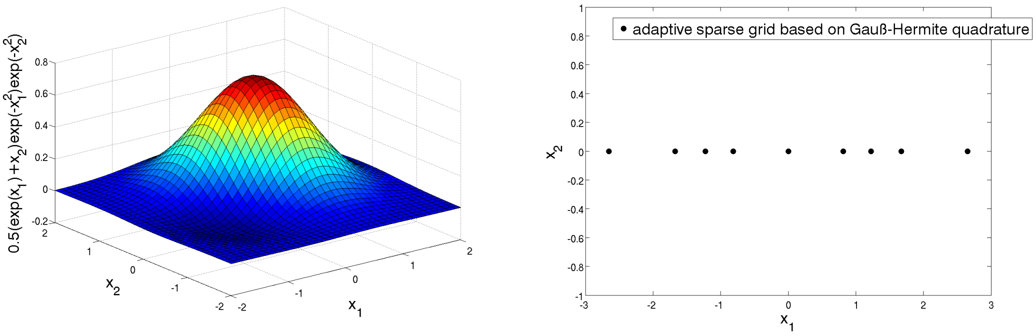


Figure 6.4: Demonstration of the dimension adaptive refinement strategy evaluating $\int_{[-\infty,\infty]^2} \frac{1}{2}(\exp(x_1) + x_2) \exp(-x_1^2) \exp(-x_2^2) \, \mathrm{d}x_1 \mathrm{d}x_2$.

Next, we will discuss the second refinement approach based on linear interpolation functions and compare the two algorithms in section 8 considering 2D Euler and Navier-Stokes test cases.

6.4.2 Locally refined sparse grids

Below, we introduce a locally adaptive hierarchical sparse grid approach using piecewise multilinear hierarchical basis functions following closely [88, 117]. Due to the straightforward implementation of the refinement, we choose the linear hat functions as interpolation basis functions, which are also well established in the adaptive mesh refinement. The support nodes of the one-dimensional basis function are given by

$$Y_j^i = \begin{cases} 0, & \text{for } j = 1, m_i = 1 \\ 2 \cdot \frac{j-1}{m_i-1} - 1, & \text{for } j = 1, \dots, m_i,\ m_i > 1 \end{cases}$$

with

$$m_i = \begin{cases} 1, & \text{for } i = 1 \\ 2^{i-1} + 1, & \text{for } i > 1 \end{cases} .$$

Hence, the interpolation formulas are defined by

$$a_j^i\,(Y) = \begin{cases} 1 - \frac{1}{2}\,(m_i - 1) \cdot \left|Y - Y_j^i\right|, & \text{if } \left|Y - Y_j^i\right| < \frac{2}{m_i-1} \\ 0, & \text{otherwise} \end{cases} . \tag{6.14}$$

Figure 6.5 shows the defined linear ansatz functions.

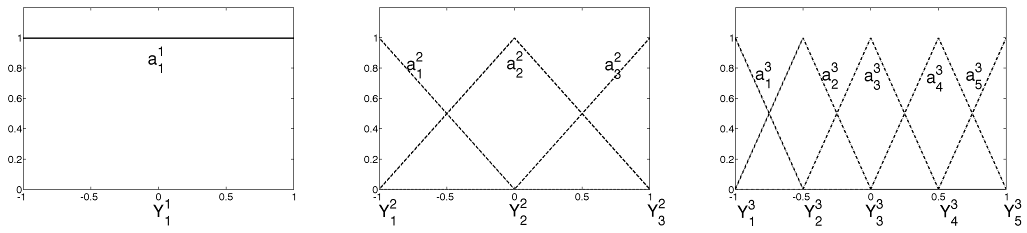


Figure 6.5: 1D linear ansatz functions.

The discussed univariate nodal basis functions (6.14) are now transformed into multivariate hierarchical basis functions, which are fundamental for the adaptive sparse grid. Considering once again the one-dimensional difference formula

$$\Delta^i(g) = \mathcal{I}^{i+1}(g) - \mathcal{I}^i(g) \tag{6.15}$$

with

$$\mathcal{I}^i(h) = \sum_{Y_j^i \in G^i} a_j^i \cdot g\left(Y_j^i\right) ,$$

we obtain due to the fact that the support nodes are nested (e.g. $G^i \subset G^{i+1}$) and accordingly $\mathcal{I}^{i-1}\,(g) = \mathcal{I}^i\left(\mathcal{I}^{i-1}\,(g)\right)$ the following representation of (6.15)

$$\begin{aligned} \Delta^i(g) &= \sum_{Y_j^i \in G^i} a_j^i \cdot g\left(Y_j^i\right) - \sum_{Y_j^i \in G^i} a_j^i \cdot \mathcal{I}^{i-1}\,(g)\left(Y_j^i\right) \\ &= \sum_{Y_j^i \in G^i} a_j^i \cdot \left(g\left(Y_j^i\right) - \mathcal{I}^{i-1}\,(g)\left(Y_j^i\right)\right) \\ &= \sum_{Y_j^i \in G_\Delta^i} a_j^i \cdot \left(g(Y_j^i) - \mathcal{I}^{i-1}\,(g)\left(Y_j^i\right)\right) , \end{aligned}$$

since $g(Y_j^i) - \mathcal{I}^{i-1}(g)(Y_j^i) = 0, \quad \forall Y_j^i \in G^{i-1}$. Renumbering the elements in $G^i_\Delta = G^i \setminus G^{i-1}$, with $m_i^\Delta = \#G^i_\Delta = m_i - m_{i-1}$, leads to

$$\Delta^i(g) = \sum_{j=1}^{m_i^\Delta} a_j^i \cdot \left(g(Y_j^i) - \mathcal{I}^{i-1}(g)(Y_j^i)\right) . \tag{6.16}$$

We define $w_j^i = g(Y_j^i) - \mathcal{I}^{i-1}(g)(Y_j^i)$ as the 1D hierarchical surplus [24], which is the difference between the current and previous interpolation level. If the 1D grid of level k interpolates the function g exactly, w_j^k is equal to zero for all j. So, this value can be used as an error indicator for each inserted grid point, since the hierarchical surpluses tend to zero as the level k tends to infinity (for continuous functions). Considering now again the multi-dimensional case, we obtain the sparse grid (6.5) in hierarchical form applying the derived formula (6.16) of Δ^i.

$$\begin{aligned} \mathcal{S}(k,d)(f) &= \mathcal{S}(k-1,d)(f) + \sum_{|\mathbf{i}|=k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) \\ &= \mathcal{S}(k-1,d)(f) + \Delta\mathcal{S}(k,d)(f) \end{aligned}$$

with

$$\begin{aligned} \Delta\mathcal{S}(k,d)(f) = \sum_{|\mathbf{i}|=k} \sum_{\mathbf{j} \in I^{\mathbf{i}}_{hier}} & \underbrace{\left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}\right)}_{a^{\mathbf{i}}_{\mathbf{j}}} \cdot \\ & \cdot \underbrace{\left(f\left(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}\right) - \mathcal{S}(k-1,d)(f)\left(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}\right)\right)}_{w^{\mathbf{i}}_{\mathbf{j}}}, \end{aligned} \tag{6.17}$$

where $I^{\mathbf{i}}_{hier} := \{\mathbf{j} \in \mathbb{N}^d : Y_{j_l}^{i_l} \in G^{i_k}_\Delta \text{ for } j_l = 1, \dots, m^{i_l}_\Delta, k = 1, \dots, d\}$ is a new set of multi-indices consistent with the multivariate hierarchical basis $\{a^{\mathbf{i}}_{\mathbf{j}} : \mathbf{j} \in I^{\mathbf{o}}_{hier}, \mathbf{o} \leq \mathbf{i}\}$. Figure 6.6 shows the resulting 1D hierarchical ansatz functions.

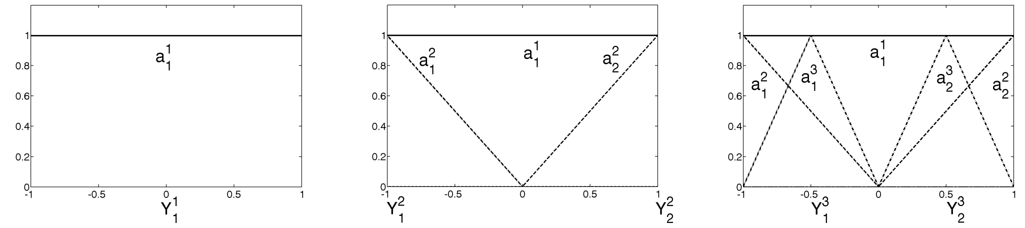


Figure 6.6: 1D hierarchical linear ansatz functions.

Thus, the objective function in our application can be approximated by the following rather abstract expression:

$$f(p, \psi_d(\zeta)) = \sum_{|\mathbf{i}| \leq k} \sum_{\mathbf{j} \in I^{\mathbf{i}}_{hier}} w^{\mathbf{i}}_{\mathbf{j}}(p) \cdot a^{\mathbf{i}}_{\mathbf{j}}(\zeta) .$$

The mean value of the objective function can be then computed as:

$$\mathbb{E}_N\left(f\left(p,\psi_d(\zeta)\right)\right) = \sum_{|\mathbf{i}|\leq k}\sum_{\mathbf{j}\in I^{\mathbf{i}}_{hier}} w^{\mathbf{i}}_{\mathbf{j}}(p)\cdot\int_{\mathcal{O}} a^{\mathbf{i}}_{\mathbf{j}}\left(\psi_d(\zeta)\right)\mathrm{d}\mathcal{P}(\zeta)\ ,$$

where

$$\int_{-1}^{1} a^{i}_{j}(Y)\,\mathrm{d}Y = \begin{cases} 1\cdot l, & \text{if } i=1 \\ \frac{1}{2}\cdot l, & \text{if } i=2 \\ 2^{1-i}\cdot l, & \text{otherwise} \end{cases}\ ,$$

where l denotes the length of the given 1D interval, that means in our example $l=2$. Instead of using the hierarchical surplus $w^{\mathbf{i}}_{\mathbf{j}}$ as an error indicator for the adaptivity (cf. [88, 117]), we suggest to adapt the grid checking the following expression:

$$\tilde{w}^{\mathbf{i}}_{\mathbf{j}} := w^{\mathbf{i}}_{\mathbf{j}}\cdot\int_{\mathcal{O}} a^{\mathbf{i}}_{\mathbf{j}}\left(\psi_d(\zeta)\right)\mathrm{d}\mathcal{P}(\zeta)\ . \tag{6.18}$$

Since it is not necessary to exactly interpolate the drag depending on the uncertainty in the optimization loop in our application, the introduced adaptivity indicator $\tilde{w}^{\mathbf{i}}_{\mathbf{j}}$ (cf. (6.18)) only measures the difference between the value of the mean inserting a new point $Y^{\mathbf{i}}_{\mathbf{j}}$ of the current level of interpolation and the corresponding value of the mean at the previous interpolation level. The resulting algorithm 6.3 in order to construct the sparse grid, which is then used for the optimization, slightly differs from [88, 117] due to the modification in the adaptivity indicator.

The algorithm starts with the sparse grid at level $k=0$ and tests the introduced adaptivity criterion for all grid points in the old index set. If the test fails, the region around this point will locally be refined, which means the $2d$ neighbor points will be added to the active set $\mathbf{I}_{active}$. Therefore, the algorithm successively adds those points to the Grid $\mathcal{H}_{adaptive}$, which are generated by the local refinement procedure. The output of the algorithm is the computed adaptive grid $\mathcal{H}_{adaptive}$ and an index vector indicating the number of points in each sparse grid level. In order to approximate the mean value in each optimization step, an additional function 6.4 is needed. It computes the hierarchical surpluses of equation (6.17), because the weights are not explicitly given as in the non-adaptive case (6.3).

Algorithm 6.3 Locally refined sparse grid

```
function LASG
    Initialize the sparse grid at level $k = 0$:
        Calculate the function value at $(0,\dots,0)^\top \in \mathbb{R}^d$
        $\mathcal{S}_{adaptive}(k,d) = f((0,\dots,0)^\top)$, $\mathcal{H}_{adaptive}(k,d) \leftarrow \{(0,\dots,0)^\top\}$
        Add the $2d$ neighbor points to the active index set $\mathbf{I}_{active}$
        Initialize the old index set $\mathbf{I}_{old} \leftarrow \emptyset$
        $k \leftarrow k+1$
    while $k \leq k_{max}$ and $\mathbf{I}_{active} \neq \emptyset$ do
        $\mathbf{I}_{old} \leftarrow \mathbf{I}_{old} \cup \mathbf{I}_{active}$, $\mathbf{I}_{active} \leftarrow \emptyset$
        for all $Y_{\mathbf{j}}^{\mathbf{i}} \in \mathbf{I}_{old}$ do
            Calculate
                $\tilde{w}_{\mathbf{j}}^{\mathbf{i}} = \left( f\left(Y_{j_1}^{i_1},\dots,Y_{j_d}^{i_d}\right) - \mathcal{S}_{adaptive}(k-1,d)(f)\left(Y_{j_1}^{i_1},\dots,Y_{j_d}^{i_d}\right)\right) \cdot$
                $\cdot \int_{\mathcal{O}} a_{\mathbf{j}}^{\mathbf{i}}(\psi_d(\zeta))\,\mathrm{d}\mathcal{P}(\zeta)$ ,    $\forall Y_{\mathbf{j}}^{\mathbf{i}} \in \mathbf{I}_{old}$
            if then $\left|\tilde{w}_{\mathbf{j}}^{\mathbf{i}}\right| > \epsilon$
                Add the $2d$ neighbor points of the current point $Y_{\mathbf{j}}^{\mathbf{i}}$ to the active index set $\mathbf{I}_{active}$
            end if
        end for
        $\mathcal{H}_{adaptive}(k,d) \leftarrow \mathcal{H}_{adaptive}(k-1,d) \cup \mathbf{I}_{old}$, $\mathbf{I}_{old} \leftarrow \emptyset$
        $k \leftarrow k+1$
    end while
end function
```

Algorithm 6.4 Computation of the mean value using the locally refined adaptive sparse grid $\mathcal{H}_{adaptive}$

```
function MEANLASG
    Input: $\mathcal{H}_{adaptive}$, index vector $v_{index}(1,\dots,maxlevel_{\mathcal{H}_{adaptive}})$, $v_{index}(k) = \#Y_{\mathbf{j}}^{\mathbf{i}}$, $|\mathbf{i}| = k$,
        $Y_{\mathbf{j}}^{\mathbf{i}} \in \mathcal{H}_{adaptive}$,
    $\mathcal{S}(0,d) \leftarrow 0$
    for $k = 1$ to $maxlevel_{\mathcal{H}_{adaptive}}$ do
        for $l = 1$ to $v_{index}(k)$ do
            Compute $w_{\mathbf{j}}^{\mathbf{i}} = \left( f\left(Y_{j_1}^{i_1},\dots,Y_{j_d}^{i_d}\right) - \mathcal{S}_{adaptive}(k-1,d)(f)\left(Y_{j_1}^{i_1},\dots,Y_{j_d}^{i_d}\right)\right)$ ,
                $\forall Y_{\mathbf{j}}^{\mathbf{i}}$, $|\mathbf{i}| = k$, $Y_{\mathbf{j}}^{\mathbf{i}} \in \mathcal{H}_{adaptive}$
        end for
        $\mathcal{S}_{adaptive}(k,d)(f) = \sum_{|\mathbf{i}|=k} \sum_{Y_{\mathbf{j}}^{\mathbf{i}} \in \mathcal{H}_{adaptive}} w_{\mathbf{j}}^{\mathbf{i}} \cdot a_{\mathbf{j}}^{\mathbf{i}}(Y_{\mathbf{j}}^{\mathbf{i}}) + \mathcal{S}_{adaptive}(k-1,d)(f)$
    end for
    $\mathbb{E} \leftarrow \sum_{|\mathbf{i}| \leq maxlevel} \sum_{Y_{\mathbf{j}}^{\mathbf{i}} \in \mathcal{H}_{adaptive}} w_{\mathbf{j}}^{\mathbf{i}}(p) \cdot \int_{\mathcal{O}} a_{\mathbf{j}}^{\mathbf{i}}(\psi_d(\zeta))\,\mathrm{d}\mathcal{P}(\zeta)$
end function
```

Now, we shortly consider the following sample function

$$f : \mathbb{R}^2 \to \mathbb{R}, \quad f(x) = \frac{1}{2}(\exp(x_1) + x_2)$$

in order to demonstrate the adaptive refinement procedure. We are looking for the value of the integral

$$\int_{[-1,1]^2} \frac{1}{2}(\exp(x_1) + x_2)\, dx$$

for the given error tolerance $\epsilon = 10^{-3}$. The resulting adaptive sparse grid is shown in figure 6.7.

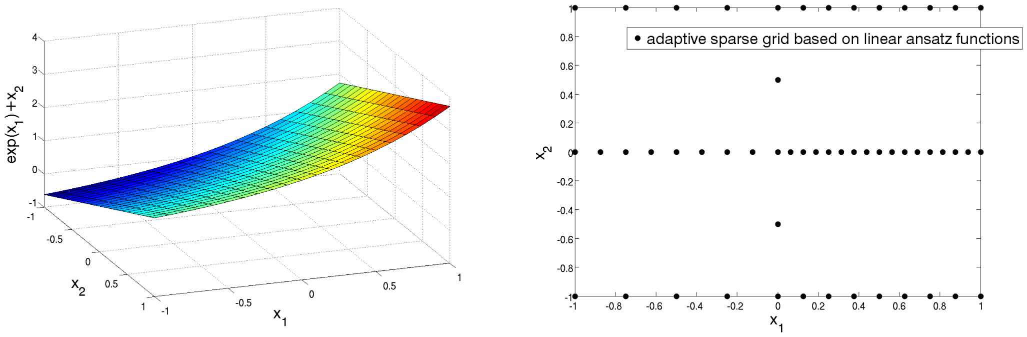


Figure 6.7: Demonstration of the adaptive refinement strategy evaluating $\int_{[-1,1]^2} \frac{1}{2}(\exp(x_1) + x_2)\, dx$.

It shows that the algorithm is able to recognize the linearity in the second dimension, since the refinement only takes place in the first dimension, especially for values $x_1 \geq 0$. So, this example reflects the capability of reducing the number of grid points using the local refinement approach for anisotropic problems. The reduction of grid points due to the antisymmetric behavior of the objective function can also be observed in our application (cf. chapter 8).

Chapter 7

Fundamentals of Optimization

The following chapter will give a brief introduction to constrained optimization and basic algorithms. The main focus lies on methods used in the aerodynamic shape optimization framework, especially the one-shot method, which we will use in the robust optimization. First, we introduce a general formulation of a constrained nonlinear optimization problem of the form

$$\min_{x} f(x) \tag{7.1}$$

$$\text{s.t.} \quad c(x) = 0 \tag{7.2}$$

$$h(x) \geq 0\,, \tag{7.3}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a scalar-valued function and $c : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{eq}}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{in}}$. All functions are assumed to be sufficiently smooth, which means twice continuously differentiable in this context. We denote by $\mathcal{F}$ the feasible set of points x satisfying the constraints, i.e.

$$\mathcal{F} = \left\{ x \in \mathbb{R}^n | c(x) = 0, h(x) \geq 0 \right\}. \tag{7.4}$$

Therefore, the problem (7.1 - 7.3) can then be rewritten as

$$\min_{x \in \mathcal{F}} f(x)\,. \tag{7.5}$$

Considering the aerodynamic shape optimization problem (2.29 - 2.31), the variable x can be divided into two parts,

$$x = \begin{pmatrix} y \\ p \end{pmatrix} \begin{matrix} \in \mathbb{R}^{n_y} \\ \in \mathbb{R}^{n_p} \end{matrix}\,, \qquad n_y + n_p = n\,,$$

where y denotes the state vector and p are the design variables. We will discuss the special framework of separable problems later on in this chapter, and first concentrate on conditions to characterize solutions of constrained optimization problems of the form (7.1 - 7.3).

7.1 Theory of constrained optimization

The following definitions distinguish two types of solutions of (7.1 - 7.3).

Definition 7.1.1 (Global Solution)**.** *A vector $x^\star$ is a global solution of the problem (7.1 - 7.3), if $x^\star$ is feasible, i.e. $x^\star \in \mathcal{F}$, and $f(x^\star) \leq f(x)$ for all $x \in \mathcal{F}$.*

Definition 7.1.2 (Local Solution)**.** *A vector $x^\star$ is a local solution of the problem (7.1 - 7.3), if $x^\star \in \mathcal{F}$ and there is a neighborhood $\mathcal{U}$ of $x^\star$ such that $f(x^\star) \leq f(x)$ for all $x \in \mathcal{U} \cap \mathcal{F}$.*

Except in some special cases, only a local solution instead of a global one of the optimization problem can be found by the algorithms, which we will discuss below. In the application of interest, one is interested in a design which leads to better drag performance than the initial one, which means a design minimizing the objective function. But on the other hand, the difference of the initial and optimized design should be small due to manufacturing reasons, so that the local behavior of the algorithms can be of advantage in the aerodynamic design task.

Necessary and sufficient optimality conditions are now introduced in order to characterize the solutions of the constrained optimization problem (7.1 - 7.3). The definitions and theorems follow essentially those in [132].

7.1.1 Necessary and sufficient optimality conditions

The Lagrangian for the constrained optimization problem (7.1 - 7.3) is defined as

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top c(x) + \mu^\top h(x)\,. \tag{7.6}$$

The active set $\mathcal{AS}(x)$ of inequality constraints at any feasible x is defined as

$$\mathcal{AS}(x) = \{i \in \{1, \dots, m_{in}\} | h_i(x) = 0\}\,. \tag{7.7}$$

Definition 7.1.3 (LICQ)**.** *Given a feasible point $x^\star \in \mathcal{F}$ and the corresponding active set $\mathcal{AS}(x^\star)$ defined by (7.7), the linear independence constraint qualification (LICQ) holds, if the set*

$$\{\nabla c_1(x^\star), \dots, \nabla c_{m_{eq}}(x^\star), \nabla h_{i_1}(x^\star), \dots, \nabla h_{i_s}(x^\star)\}$$

of active constraint gradients is linearly independent, with $\{i_1, \dots, i_s\} = \mathcal{AS}(x^\star)$.

Theorem 7.1.4 (First Order Necessary Conditions)**.** *Suppose $x^\star$ is a local solution of (7.1 - 7.3) and that the LICQ holds at $x^\star$. Then, there exist Lagrange multipliers $\lambda^\star \in \mathbb{R}^{m_{eq}}, \mu^\star \in \mathbb{R}^{m_{in}}$, such that the following conditions are satisfied*

$$\begin{aligned}
\nabla_x \mathcal{L}(x^\star, \lambda^\star, \mu^\star) &= 0 && (7.8)\\
c(x^\star) &= 0 && (7.9)\\
h(x^\star) &\geq 0 && (7.10)\\
\mu_i^\star &\leq 0\,, \qquad \forall i \in \{1, \dots, n_{in}\} && (7.11)\\
(\mu^\star)^\top h(x^\star) &= 0. && (7.12)
\end{aligned}$$

The conditions (7.8 - 7.12) are known as Karush-Kuhn-Tucker conditions (KKT conditions).

Remark 7.1.5. The aerodynamic shape optimization problem (2.29 - 2.31) can be transformed to an equality constrained problem due to physical reasons. So, the KKT conditions can be simplified to:

$$\begin{aligned}
\nabla_x \mathcal{L}(x^\star, \lambda^\star) &= 0\\
\nabla_\lambda \mathcal{L}(x^\star, \lambda^\star) = c(x^\star) &= 0\,.
\end{aligned}$$

In order to formulate second order conditions, the following set of directions needs to be defined.

Definition 7.1.6. *Given a point $x^\star$ and the active constraint set $\mathcal{AS}(x^\star)$ defined by (7.7), the set $\mathcal{F}_1$ is defined by*

$$\mathcal{F}_1(x^\star) = \left\{ \alpha d \mid \alpha > 0, \quad \begin{array}{ll} d^\top \nabla c_i(x^\star) = 0, & \forall i \in \{1, \dots, m_{eq}\} \\ d^\top \nabla h_i(x^\star) \geq 0, & \forall i \in \mathcal{AS}(x^\star) \end{array} \right\} . \tag{7.13}$$

If the LICQ holds at $x^\star$, $\mathcal{F}_1(x^\star)$ is the tangent cone to the feasible set at $x^\star$.

Given the Lagrange multipliers $\lambda^\star, \mu^\star$, such that $x^\star, \lambda^\star, \mu^\star$ satisfy the KKT conditions (7.8 - 7.12), we define a subset $\mathcal{F}_2(x^\star, \lambda^\star, \mu^\star)$ of $\mathcal{F}_1(x^\star)$ by

$$\mathcal{F}_2(x^\star, \lambda^\star, \mu^\star) = \{ w \in F_1 | \nabla h_i(x^\star)^\top w = 0, \ \forall i \in \mathcal{AS}(x^\star) \ \text{ with } \ \mu_i^\star < 0 \} . \tag{7.14}$$

Equivalently,

$$w \in \mathcal{F}_2(x^\star, \lambda^\star, \mu^\star) \Leftrightarrow \begin{cases} \nabla c_i(x^\star)^\top w = 0, & 1 \leq i \leq m_{eq} \\ \nabla h_i(x^\star)^\top w = 0, & \forall i \in \mathcal{AS}(x^\star) \text{ with } \mu_i^\star < 0 \\ \nabla h_i(x^\star)^\top w \geq 0, & \forall i \in \mathcal{AS}(x^\star) \text{ with } \mu_i^\star = 0 \end{cases} .$$

Theorem 7.1.7 (Second Order Necessary Conditions)**.** *Suppose that $x^\star$ is a local solution of (7.1 - 7.3) and that the LICQ condition holds at $x^\star$. Given the Lagrange multipliers $\lambda^\star, \mu^\star$ such that $x^\star, \lambda^\star, \mu^\star$ satisfy the KKT conditions (7.8 - 7.12) and $\mathcal{F}_2(x^\star, \lambda^\star, \mu^\star)$ is defined by (7.14). Then,*

$$w^\top \nabla_{xx} \mathcal{L}(x^\star, \lambda^\star, \mu^\star) w \geq 0 , \qquad \forall w \in \mathcal{F}_2(x^\star, \lambda^\star, \mu^\star). \tag{7.15}$$

Theorem 7.1.8 (Second Order Sufficient Conditions)**.** *Suppose that for some feasible point $x^\star$, there exist Lagrange multipliers $\lambda^\star \in \mathbb{R}^{m_{eq}}$ and $\mu^\star \in \mathbb{R}^{m_{in}}$ such that the KKT conditions (7.8 - 7.12) are satisfied. Suppose further that*

$$w^\top \nabla_{xx} \mathcal{L}(x^\star, \lambda^\star, \mu^\star) w > 0 , \qquad \forall w \in \mathcal{F}_2(x^\star, \lambda^\star, \mu^\star), w \neq 0. \tag{7.16}$$

Then, $x^\star$ is a local solution for (7.1 - 7.3).

7.2 Optimization algorithms

7.2.1 Sequential quadratic programming (SQP)

Sequential Quadratic Programming (SQP) is one of the most popular and effective methods for nonlinear constrained optimization. We will shortly discuss the basic approach and extensions in order to introduce the optimization strategy in the aerodynamic framework as in [58].

For simplicity reasons, we will begin by considering the following equality constrained problem

$$\min_x f(x) \tag{7.17}$$

$$\text{s.t.} \quad c(x) = 0 . \tag{7.18}$$

From (7.6), the Lagrangian function of the problem (7.17 - 7.18) is $\mathcal{L}(x,\lambda) = f(x) + \lambda^\top c(x)$. By applying the first order KKT conditions, we obtain the system

$$\begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ \nabla_\lambda \mathcal{L}(x,\lambda) \end{bmatrix} = \begin{bmatrix} \nabla f(x) + C(x)^\top \lambda \\ c(x) \end{bmatrix} = 0\,, \tag{7.19}$$

where $C(x)^\top = [\nabla c_1(x), \nabla c_2(c), \dots, \nabla c_m(x)]$. The system of nonlinear equations (7.19) can be solved by using Newton's method. The Jacobian of (7.19) is given by

$$\begin{bmatrix} H(x,\lambda) & C(x)^\top \\ C(x) & 0 \end{bmatrix}\,,$$

where $H(x,\lambda)$ denotes the Hessian of the Lagrangian,

$$H(x,\lambda) = \nabla^2_{xx} \mathcal{L}(x,\lambda)\,. \tag{7.20}$$

Thus, we obtain the full Newton step

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \end{bmatrix}\,,$$

where Δx_k and $\Delta \lambda_k$ solve the KKT system

$$\begin{bmatrix} H(x,\lambda) & C(x)^\top \\ C(x) & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) - C(x_k)^\top \lambda_k \\ -c(x_k) \end{bmatrix}\,. \tag{7.21}$$

Remark 7.2.1.

1. Ensuring the nonsingularity of the KKT matrix, the following assumptions have to be made:
   - The Jacobian $C(x_k)$ has full rank, which is equivalent to LICQ.
   - The matrix $H(x_k, \lambda_k)$ is positive definite on the tangent space of the constraints. If (x, λ) is close to the solution $(x^\star, \lambda^\star)$ of problem (7.17 - 7.18) and the second order sufficient condition holds at the solution, this second assumption will be fulfilled.

2. The Newton iteration can be shown to be quadratically convergent under the above-mentioned assumptions, see [132] for more details.

3. Obviously, the linear system (7.21) is equivalent to the quadratic problem

$$\min_{\Delta x_k \in \mathbb{R}^n} \quad \tfrac{1}{2}(\Delta x_k)^\top H_k \Delta x_k + (\nabla f_k)^\top \Delta x_k \tag{7.22}$$

$$\text{s.t.} \qquad C_k \Delta x_k + c_k = 0\,. \tag{7.23}$$

   The Langrange (adjoint) variable μ_k of the quadratic problem (7.22 - 7.23) can be identified as the new estimate for λ_{k+1}. The quadratic program can be interpreted as a quadratic approximation of the Lagrangian function and a linear approximation of the constraints. The advantage of this alternative point of view is the straightforward extension of the method to the inequality constrained case.

4. As the exact computation of the Hessian of the Lagrangian is mostly not practicable, a quasi-Newton approximation B_k is applied. The idea of quasi-Newton methods is to compute B_{k+1} from B_k by using update formulas. Here, we will concentrate on BFGS update formulas and refer to [132] for more details. Defining the vectors

   $$s_k = x_{k+1} - x_k \,, \qquad \gamma_k^{BFGS} = \nabla f_{k+1} - \nabla f_k \,,$$

   the BFGS update is given by

   $$B_{k+1} = B_k - \frac{B_k s_k (s_k)^\top B_k}{(s_k)^\top B_k s_k} + \frac{\gamma_k^{BFGS} (\gamma_k^{BFGS})^\top}{(\gamma_k^{BFGS})^\top s_k} \,. \tag{7.24}$$

   In order to ensure the applicability of the introduced update formulas in large optimization problems, we will briefly discuss a modification of the BFGS approach, the limited-memory BFGS method, which reduces memory and computational requirements. The main idea of this method is to use curvature information from only the most recent iterations to compute the approximation of the Hessian. Instead of storing the fully dense $n \times n$ approximation from the previous iteration, which is required as input in (7.24), the limited-memory BFGS updates construct the Hessian approximation using information contained in the most recent N vectors s_i, γ_i^{BFGS} for a given N.

   Beside the computational aspects, the BFGS updates further show some favorable properties:

   - The BFGS updates generate a sequence of positive definite matrices, if B_0 is positive definite and $(\gamma_k^{BFGS})^\top s_k > 0$ for all k. Since the Hessian of the Lagrangian is only positive definite on the tangent space of the constraints, a Powell-relaxation can be applied guaranteeing the positive definite updates.
   - The update formulas can be directly formulated for the update of the inverse matrix of H_{k+1}.
   - The high amount of memory can be overcome by storing only a certain number of the most recent gradient information s_i, γ_i^{BFGS}.
   - If the Hessian of the Lagrangian is positive definite, superlinear convergence can be shown, otherwise linear convergence of the method.

5. If the Hessian is approximated by the identity, the linearized gradient projection method, outlined in algorithm 7.1 is obtained, which is originally based on the gradient method for unconstrained optimization.

Algorithm 7.1 Linearized gradient projection method

```
function GPM
    Choose an initial iterate $x^0$
    $k \leftarrow 0$
    repeat
        Choose step length $\tau_k$
        $x^{k+1} \leftarrow x^k - \tau_k \nabla f(x^k) - C_k^\top (C_k C_k^\top)^{-1} (c_k - C_k \tau_k \nabla f(x^k))$
        $k \leftarrow k + 1$
    until convergence
end function
```

There are several possibilities for the choice of the step length τ_k, we refer to [132] for more details.

The SQP method in its basic form is stated in algorithm 7.2.

Algorithm 7.2 Basic SQP algorithm

```
function SQP
    Choose an initial iterate $x^0$
    $k \leftarrow 0$
    repeat
        Evaluate $\nabla f(x_k)$ and $H(x_k, \lambda_k)$ (or approximation of $H_k$)
        Solve
            $$\min_{\Delta x_k \in \mathbb{R}^n} \quad \frac{1}{2}(\Delta x_k)^\top H_k \Delta x_k + (\nabla f_k)^\top \Delta x_k$$
            $$\text{s.t.} \quad C_k \Delta x_k + c_k = 0$$
        to obtain $\Delta x_k$ and $\mu_k$
        $x_{k+1} \leftarrow x_k + \Delta x_k$
        $\lambda_{k+1} \leftarrow \mu_k$
        $k \leftarrow k + 1$
    until convergence
end function
```

Reduced SQP method

We consider the separable case now again, that means

$$x = \begin{pmatrix} y \\ p \end{pmatrix} \begin{matrix} \in \mathbb{R}^{n_y} \\ \in \mathbb{R}^{n_p} \end{matrix} \qquad \text{with } n_y + n_p = n$$

as in the aerodynamic shape optimization framework, cf. (7.1 - 7.3), and the optimization problem

$$\min_{y,p} f(y,p) \tag{7.25}$$

$$\text{s.t.} \quad c(y,p) = 0\,. \tag{7.26}$$

The Jacobian

$$C_y = \frac{\partial c}{\partial y}$$

is assumed to be invertible. Applying the implicit function theorem, problem (7.25 - 7.26) can be transformed to

$$\min_p \hat{f}(p)\,, \tag{7.27}$$

where $\hat{f}(p) = f(\Theta(p), p)$ and $\Theta : \mathbb{R}^{n_p} \to \mathbb{R}^{n_y}$ such that $y = \Theta(p)$ and $c(\Theta(p), p) = 0$. The derivatives of $\hat{f}$ are given as

$$\nabla \hat{f}(p) = T(y,p)^\top \begin{pmatrix} \nabla_y f(y,p) \\ \nabla_p f(y,p) \end{pmatrix} = \nabla_p f(y,p) + C_p(y,p)^\top \lambda \tag{7.28}$$

and

$$\frac{\partial^2}{\partial p^2}\hat{f}(p) = T(y,p)^\top \begin{bmatrix} \frac{\partial^2(f(y,p)+\lambda^\top c(y,p))}{\partial y^2} & \frac{\partial^2(f(y,p)+\lambda^\top c(y,p))}{\partial y \partial p} \\ \frac{\partial^2(f(y,p)+\lambda^\top c(y,p))}{\partial p \partial y} & \frac{\partial^2(f(y,p)+\lambda^\top c(y,p))}{\partial p^2} \end{bmatrix} T(y,p)\,, \tag{7.29}$$

where

$$T(y,p) := \begin{pmatrix} -\left(\frac{\partial c}{\partial y}(y,p)\right)^{-1} \frac{\partial c}{\partial p}(y,p) \\ I \end{pmatrix}, \qquad \lambda := -R(y,p)^\top \begin{pmatrix} \nabla_y f(y,p) \\ \nabla_p f(y,p) \end{pmatrix}, \tag{7.30}$$

$$R(y,p) := \begin{pmatrix} \left(\frac{\partial c}{\partial y}(y,p)\right)^{-1} \\ 0 \end{pmatrix}, \tag{7.31}$$

and y satisfies $c(y,p) = 0$. The proof can be found in [163]. The derivatives (7.28) and (7.29) of the function $\hat{f}$ are referred to as reduced derivatives. The variable λ defined in (7.30) is exactly the adjoint variable of problem (7.25 - 7.26) with the Lagrangian $\mathcal{L}(y,p,\lambda) = f(y,p) + \lambda^\top c(y,p)$ at the solution $\nabla_y \mathcal{L}(y,p,\lambda) = \nabla_y f(y,p) + \frac{\partial c}{\partial y}^\top \lambda = 0$.

Using the implicit function theorem, the reduced SQP methods correspond to a Newton/quasi-Newton method for the unconstrained problem

$$\min_p f(\Theta(p), p)$$

with only one Newton step solving the nonlinear equation $c(y, p_k) = 0$ in each iteration. Further, the rSQP method can be interpreted as a SQP method of the form

$$\begin{pmatrix} 0 & 0 & C_y^\top \\ 0 & T^\top HT & C_p^\top \\ C_y & C_p & 0 \end{pmatrix} \begin{bmatrix} \Delta y \\ \Delta p \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla_y f \\ -\nabla_p f \\ -c \end{bmatrix}. \tag{7.32}$$

The rSQP approach is outlined in algorithm 7.3.

Algorithm 7.3 rSQP algorithm

```
function RSQP
    Choose an initial iterate $y_0, p_0$
    $k \leftarrow 0$
    repeat
        Compute the reduced gradient $\gamma_k^{red} = T_k^\top \begin{pmatrix} \nabla_y f_k \\ \nabla_p f_k \end{pmatrix}$
         and some approximation $B_k$ of $T_k^\top H_k T_k$
        Solve $B_k \Delta p_k = -\gamma_k^{red}$
        $\Delta y_k \leftarrow -C_y^{-1} C_p \Delta p_k - C_y^{-1} c_k$
        $y_{k+1} \leftarrow y_k + \Delta y_k$
        $p_{k+1} \leftarrow p_k + \Delta p_k$
        $\lambda_{k+1} \leftarrow -R_k^\top \begin{pmatrix} \nabla_y f_k \\ \nabla_p f_k \end{pmatrix}$
        $k \leftarrow k + 1$
    until convergence
end function
```

More details on the algorithm and theoretical background can be found in [163].

Partially reduced SQP method

The partially reduced SQP approach combines the advantages of SQP and rSQP methods. On the one hand, the SQP framework allows a convenient treatment of inequality constraints and additional equality constraints and on the other hand, the rSQP method can reduce the complexity of the problem by exploiting the null space structure of some equality constraints. We now consider the following constrained optimization problem:

$$\begin{aligned} &\min_{y,p} f(y,p) \\ \text{s.t.}\quad & c(y,p) = 0\,, \qquad \frac{\partial c}{\partial y} \text{ nonsingular} \\ & h(y,p) \geq 0\,. \end{aligned}$$

The derivatives of the additional constraint are denoted by

$$H^{in} := \begin{bmatrix} \frac{\partial h}{\partial y} & \frac{\partial h}{\partial p} \end{bmatrix}\,.$$

So, the concept of prSQP methods is formulated in algorithm 7.4.

Algorithm 7.4 prSQP algorithm

```
function PRSQP
    Choose an initial iterate $y_0, p_0$
    $k \leftarrow 0$
    repeat
        Compute the reduced gradient $\gamma_k^{red} = T_k^\top \begin{pmatrix} \nabla_y f_k \\ \nabla_p f_k \end{pmatrix}$
        and some approximation $B_k$ of $T_k^\top H_k T_k$
        Solve the quadratic problem
            $\min_{\Delta p_k} \Delta p_k^\top B_k \Delta p_k + (\gamma_k^{red})^\top \Delta p_k$
            $\text{s.t. } H_k^{in} T_k \Delta p_k + h_k - H_k^{in} R_k c_k \geq 0$
        in order to get $\Delta p_k$ and the adjoint variable $\mu_{k+1}$
        $\Delta y_k \leftarrow -C_y^{-1} C_p \Delta p_k - C_y^{-1} c_k$
        $y_{k+1} \leftarrow y_k + \Delta y_k$
        $p_{k+1} \leftarrow p_k + \Delta p_k$
        $\lambda_{k+1} \leftarrow -R_k^\top [(H_k^{in})^\top \mu_{k+1} + \begin{pmatrix} \nabla_y f_k \\ \nabla_p f_k \end{pmatrix}]$
        $k \leftarrow k + 1$
    until convergence
end function
```

A detailed discussion of prSQP methods can be found in [163].

One-shot method for aerodynamic shape optimization

We will now again consider the aerodynamic shape optimization problem (2.29 - 2.31)

$$\begin{aligned} &\min_{y,p} f(y,p) \\ \text{s.t.}\quad &c(y,p) = 0 \\ &h(y,p) \geq 0\,. \end{aligned}$$

and introduce a one-shot method based on the above-mentioned methods following closely [58]. The scalar-valued constraint $h(y,p) \geq 0$ ensures that the optimized profile produces a given lift C_L^0 in order to obtain aerodynamically meaningful solutions. Due to the fact that loosing lift results in a better objective function value (drag), this constraint will be active. Thus, the constraint is a priori formulated as an equality constraint:

$$\min_{y,p} f(y,p) \tag{7.33}$$

$$\text{s.t.}\quad c(y,p) = 0 \tag{7.34}$$

$$h(y,p) = 0\,. \tag{7.35}$$

The Lagrangian of problem (7.33 - 7.35) is given as

$$\mathcal{L}(y,p,\lambda,\mu) = f(y,p) + \lambda^\top c(y,p) + \mu^\top h(y,p)\,.$$

Further, we assume that $\frac{\partial c}{\partial y} = C_y$ is nonsingular and the LICQ condition holds, i.e.

$$\{\nabla_{(y,p)} c_1, \dots, \nabla_{(y,p)} c_{n_{eq}}, \nabla_{(y,p)} h\}$$

is linear independent. The necessary optimality conditions (7.8 - 7.12) are given by

$$\begin{aligned} \nabla_y \mathcal{L} &= 0 \\ \nabla_p \mathcal{L} &= 0 \\ c(y,p) = \nabla_\lambda \mathcal{L} &= 0 \\ h(y,p) = \nabla_\mu \mathcal{L} &= 0\,. \end{aligned}$$

The SQP approach (cf. (7.21)) leads to the following KKT system

$$\begin{bmatrix} H_{yy} & H_{yp} & C_y^\top & (H_y^{lift})^\top \\ H_{py} & H_{pp} & C_p^\top & (H_p^{lift})^\top \\ C_y & C_p & 0 & 0 \\ H_y^{lift} & H_p^{lift} & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta\lambda \\ \Delta\mu \end{pmatrix} = \begin{pmatrix} -\nabla_y \mathcal{L} \\ -\nabla_p \mathcal{L} \\ -c \\ -h \end{pmatrix}, \tag{7.36}$$

where the matrix $\begin{bmatrix} H_{yy} & H_{yp} \\ H_{py} & H_{pp} \end{bmatrix}$ denotes the Hessian of the Lagrangian $\mathcal{L}$. Applying the partially reduced SQP approach (cf. algorithm 7.4), the system (7.36) reduces to

$$\begin{bmatrix} B & \gamma_h^{red} \\ (\gamma_h^{red})^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu_{k+1} \end{pmatrix} = \begin{pmatrix} -\gamma_f^{red} \\ -h + H_y^{lift} C_y^{-1} c \end{pmatrix}, \tag{7.37}$$

where the consistent reduced gradient of the drag coefficient is given as

$$\gamma_f^{red} = \nabla_p f + C_p^\top \sigma_f$$

and the consistent reduced gradient of the lift coefficient as

$$\gamma_h^{red} = \nabla_p h + C_p^\top \sigma_h \,.$$

The two adjoint variables σ_f and σ_h are computed by the adjoint flow solver, where the right hand side is the gradient of the drag coefficient

$$C_y^\top \sigma_f = -\nabla_y f$$

and the lift coefficient

$$C_y^\top \sigma_h = -\nabla_y h \,.$$

Finally, a block elimination leads to the following iteration:

$$\mu_{k+1} = \frac{h + \sigma_h c - ((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_f^{red})^{k+1}}{((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_h^{red})^{k+1}} \tag{7.38}$$

and

$$\Delta p = -B_k^{-1} (\gamma_f^{red})^{k+1} - B_k^{-1} (\gamma_h^{red})^{k+1} \mu_{k+1} \,. \tag{7.39}$$

If the state equation is solved sufficiently, i.e. $c \approx 0$, the term $\sigma_h c$, which corresponds to an additional Newton step to solve $y = \Theta(p)$, can be neglected. In [58], the two systems (7.36) and (7.37) are shown to be asymptotically equivalent and to converge to the same solution. So, the one-shot method in the context of aerodynamic shape optimization is summarized in algorithm 7.5.

Algorithm 7.5 One-shot algorithm

```
function ONE-SHOT
    Choose an initial iterate $y^0, p^0, \sigma_f^0, \sigma_h^0$
    $k \leftarrow 0$
    repeat
        Starting in $\sigma_f^k$, perform $n_a$ steps in the adjoint solver ($\nabla_y f$ as r.h.s) $\longrightarrow \sigma_f^{k+1}$
        Starting in $\sigma_h^k$, perform $n_a$ steps in the adjoint solver ($\nabla_y h$ as r.h.s) $\longrightarrow \sigma_h^{k+1}$
        Compute the reduced gradients
            $(\gamma_f^{red})^{k+1} = \nabla_p f(y^k, p^k) + C_p^\top (y^k, p^k) \sigma_f^{k+1}$
            $(\gamma_h^{red})^{k+1} = \nabla_p h(y^k, p^k) + C_p^\top (y^k, p^k) \sigma_h^{k+1}$
        Approximate the consistent reduced Hessian of the Lagrangian $\longrightarrow B_k$
        $\mu_{k+1} \longleftarrow \frac{h(y^k,p^k) - ((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_f^{red})^{k+1}}{((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_h^{red})^{k+1}}$
        $p^{k+1} \longleftarrow p^k - \tau_k B_k^{-1} (\gamma_f^{red})^{k+1} - B_k^{-1} (\gamma_h^{red})^{k+1} \mu_{k+1}$, $\tau_k \in (0, 1]$ damping factor
        Compute the corresponding geometry and adjust the computational mesh
        Starting in $y^k$, perform $n_f$ steps in the forward solver $\longrightarrow y^{k+1}$
        $k \leftarrow k + 1$
    until convergence
end function
```

Generalized one-shot method for multiple-setpoint aerodynamic shape optimization

In the robust optimization framework, the special structure of multiple-setpoint problems often arises as a result of the discretization of the probability space. A generalization of the one-shot approach described in algorithm 7.5 to the multiple-setpoint problems exploiting the structure of the underlying problem will be introduced in this section. Numerical approaches to this problem class have already been proposed in [20, 21]. For the sake of simplicity, we restrict the discussion to a problem of the form

$$\min_{y_1,y_2,p} \omega_1 \cdot f(y_1,p,s_1) + \omega_2 \cdot f(y_2,p,s_2) \tag{7.40}$$

$$\text{s.t.} \qquad c(y_i,p,s_i) = 0\,, \qquad i = 0,1,2 \tag{7.41}$$

$$h(y_0,p,s_0) \geq 0\,, \tag{7.42}$$

where s_i denotes the i-th setpoint. We can think of s_i as a realization of a random parameter, i.e. of the Mach number. The two setpoints in the objective function (7.40) are coupled via a weighted sum and the design variables p, which are the same for all set-points. In addition, the lift constraint is required at a third setpoint s_0. The generalization to more setpoints is then obvious.

The corresponding Lagrangian of problem (7.40 - 7.42) is given by

$$\begin{aligned}\mathcal{L}(y_0,y_1,y_2,p,\lambda_0,\lambda_1,\lambda_2,\mu) =&\omega_1 \cdot f(y_1,p,s_1) + \omega_2 \cdot f(y_2,p,s_2)\\ &+ \lambda_0^\top c(y_0,p,s_0) + \lambda_1^\top c(y_1,p,s_1) + \lambda_2^\top c(y_2,p,s_2)\\ &+ \mu^\top h(y_0,p,s_0)\,.\end{aligned}$$

Resulting from the necessary optimality conditions (7.8-7.12)

$$\begin{array}{rcl} \nabla_{y_0}\mathcal{L} &=& 0\\ \nabla_{y_1}\mathcal{L} &=& 0\\ \nabla_{y_2}\mathcal{L} &=& 0\\ \nabla_{p}\mathcal{L} &=& 0\\ c(y_0,p) = \nabla_{\lambda_0}\mathcal{L} &=& 0\\ c(y_1,p) = \nabla_{\lambda_1}\mathcal{L} &=& 0\\ c(y_2,p) = \nabla_{\lambda_2}\mathcal{L} &=& 0\\ h(y_0,p) = \nabla_{\mu}\mathcal{L} &=& 0\,, \end{array}$$

we obtain the following KKT system

$$\begin{bmatrix} H_{y_0y_0} & H_{y_0y_1} & H_{y_0y_2} & H_{y_0p} & (C^0_{y_0})^\top & 0 & 0 & (H^{lift}_{y_0})^\top\\ H_{y_1y_0} & H_{y_1y_1} & H_{y_1y_2} & H_{y_1p} & 0 & (C^1_{y_1})^\top & 0 & 0\\ H_{y_2y_0} & H_{y_2y_1} & H_{y_2y_2} & H_{y_2p} & 0 & 0 & (C^2_{y_2})^\top & 0\\ H_{py_0} & H_{py_1} & H_{py_2} & H_{pp} & (C^0_p)^\top & (C^1_p)^\top & (C^2_p)^\top & (H^{lift}_p)^\top\\ C^0_{y_0} & 0 & 0 & C^0_p & 0 & 0 & 0 & 0\\ 0 & C^1_{y_1} & 0 & C^1_p & 0 & 0 & 0 & 0\\ 0 & 0 & C^2_{y_2} & C^2_p & 0 & 0 & 0 & 0\\ H^{lift}_{y_0} & 0 & 0 & H^{lift}_p & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y_0\\ \Delta y_1\\ \Delta y_2\\ \Delta p\\ \Delta\lambda_0\\ \Delta\lambda_1\\ \Delta\lambda_2\\ \Delta\mu \end{pmatrix} = \begin{pmatrix} -\nabla_{y_0}\mathcal{L}\\ -\nabla_{y_1}\mathcal{L}\\ -\nabla_{y_2}\mathcal{L}\\ -\nabla_{p}\mathcal{L}\\ -c_0\\ -c_1\\ -c_2\\ -h \end{pmatrix}, \tag{7.43}$$

where C^i denotes the Jacobian of the flow equation of the i-th setpoint.
Approximating the Hessian of the Lagrangian in (7.43) by

$$\begin{bmatrix} H_{y_0y_0} & H_{y_0y_1} & H_{y_0y_2} & H_{y_0p} \\ H_{y_1y_0} & H_{y_1y_1} & H_{y_1y_2} & H_{y_1p} \\ H_{y_2y_0} & H_{y_2y_1} & H_{y_2y_2} & H_{y_2p} \\ H_{py_0} & H_{py_1} & H_{py_2} & H_{pp} \end{bmatrix} \approx \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B \end{bmatrix},$$

(cf. partially reduced SQP approach in section 7.2.1), the following linear system has to be solved

$$\begin{bmatrix} 0 & 0 & 0 & 0 & (C^0_{y_0})^\top & 0 & 0 & (H^{lift}_{y_0})^\top \\ 0 & 0 & 0 & 0 & 0 & (C^1_{y_1})^\top & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & (C^2_{y_2})^\top & 0 \\ 0 & 0 & 0 & B & (C^0_p)^\top & (C^1_p)^\top & (C^2_p)^\top & (H^{lift}_p)^\top \\ C^0_{y_0} & 0 & 0 & C^0_p & 0 & 0 & 0 & 0 \\ 0 & C^1_{y_1} & 0 & C^1_p & 0 & 0 & 0 & 0 \\ 0 & 0 & C^2_{y_2} & C^2_p & 0 & 0 & 0 & 0 \\ H^{lift}_{y_0} & 0 & 0 & H^{lift}_p & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y_0 \\ \Delta y_1 \\ \Delta y_2 \\ \Delta p \\ \Delta\lambda_0 \\ \Delta\lambda_1 \\ \Delta\lambda_2 \\ \Delta\mu \end{pmatrix} = \begin{pmatrix} -\nabla_{y_0}\mathcal{L} \\ -\nabla_{y_1}\mathcal{L} \\ -\nabla_{y_2}\mathcal{L} \\ -\nabla_p\mathcal{L} \\ -c_0 \\ -c_1 \\ -c_2 \\ -h \end{pmatrix}. \qquad (7.44)$$

Applying a block elimination to (7.44), we obtain

$$\begin{bmatrix} B & (H^{lift}_p)^\top - (C^0_p)^\top (C^0_{y_0})^{-\top} (H^{lift}_{y_0})^\top \\ ((H^{lift}_p)^\top - (C^0_p)^\top (C^0_{y_0})^{-\top} (H^{lift}_{y_0})^\top)^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \Delta\mu \end{pmatrix}$$

$$= \begin{pmatrix} (C^0_p)^\top (C^0_{y_0})^{-\top} \nabla_{y_0}\mathcal{L} + (C^1_p)^\top (C^1_{y_1})^{-\top} \nabla_{y_1}\mathcal{L} + (C^2_p)^\top (C^2_{y_2})^{-\top} \nabla_{y_2}\mathcal{L} - \nabla_p\mathcal{L} \\ -h - H^{lift}_{y_0} (C^0_{y_0})^{-\top} c_0 \end{pmatrix}.$$

Replacing $\Delta\mu$ by $\mu_{k+1} = \mu_k + \Delta\mu$ leads to

$$\begin{bmatrix} B & \gamma^{red}_h \\ (\gamma^{red}_h)^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu_{k+1} \end{pmatrix} = \begin{pmatrix} -\gamma^{red}_f \\ -h - H^{lift}_{y_0} (C^0_{y_0})^{-\top} c_0 \end{pmatrix},$$

with

$$\begin{aligned} \gamma^{red}_h &= (H^{lift}_p)^\top - (C^0_p)^\top (C^0_{y_0})^{-\top} (H^{lift}_{y_0})^\top \\ \gamma^{red}_f &= \omega_1 \cdot \gamma^{red}_{f1} + \omega_2 \cdot \gamma^{red}_{f2} \\ &= \omega_1 \cdot (\nabla_p f^1 - (C^1_p)^\top (C^1_{y_1})^{-\top} \nabla_{y_1} f^1) + \omega_2 \cdot (\nabla_p f^2 - (C^2_p)^\top (C^2_{y_2})^{-\top} \nabla_{y_2} f^2). \end{aligned}$$

We notice that the adjoint solution of

$$C^\top_{y_i} \sigma_{f^i} = -\nabla_{y_i} f^i, \qquad i = 1, 2$$

and the lift coefficient

$$C^\top_{y_0} \sigma_h = -\nabla_{y_0} h$$

are to be solved independently, and then collected in the reduced gradients.

Therefore, the iterations in the adjoint solvers can be done in parallel for each setpoint s_i. Then, the update in the design variables is computed and distributed to all forward problems, which can again be performed in parallel. The generalization to more setpoints, which is the case in robust optimization problems, is then obvious.

Figure 7.1 illustrates the optimization strategy for N setpoints.

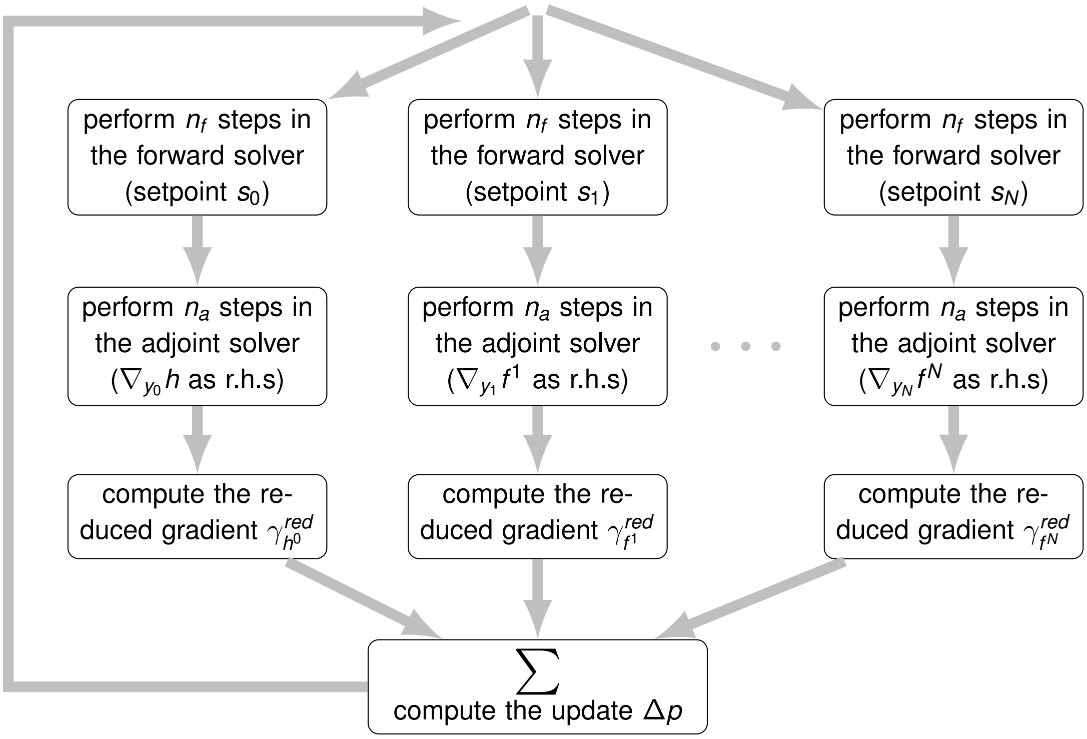

perform n_f steps in the forward solver (setpoint s_0)
perform n_f steps in the forward solver (setpoint s_1)
perform n_f steps in the forward solver (setpoint s_N)
perform n_a steps in the adjoint solver ($\nabla_{y_0} h$ as r.h.s)
perform n_a steps in the adjoint solver ($\nabla_{y_1} f^1$ as r.h.s)
perform n_a steps in the adjoint solver ($\nabla_{y_N} f^N$ as r.h.s)
compute the reduced gradient $\gamma_{h^0}^{red}$
compute the reduced gradient $\gamma_{f^1}^{red}$
compute the reduced gradient $\gamma_{f^N}^{red}$
$\sum$
compute the update Δp

Figure 7.1: Optimization strategy of the generalized one-shot approach.

The amount of additional computational effort considering a multiple-setpoint optimization problem can be efficiently overcome by a parallelization of function evaluations and gradient computations of the different setpoints. We will see later on in the numerical results (see chapter 8) that discretized semi-infinite formulations of the aerodynamic shape optimization problem belong to the class of multiple-setpoint problems and can therefore be efficiently solved by the generalized one-shot algorithm.

The generalized version of the one-shot approach is stated in algorithm 7.6.

Algorithm 7.6 Generalized one-shot algorithm

```
function GENERALIZED ONE-SHOT
    Choose an initial iterate $y_0^0, \dots, y_N^0, p^0, \sigma_{f^1}^0, \dots, \sigma_{f^N}^0, \sigma_h^0$
    $k \leftarrow 0$
    repeat
        compute in parallel
            Starting in ${\sigma_f^i}^k$, perform $n_a$ steps in the adjoint solver ($\nabla_{y_i} f^i$ as r.h.s)
                $\longrightarrow \sigma_{f^i}^{k+1}$, $\forall i = 1, \dots, N$
            Starting in $\sigma_h^k$, perform $n_a$ steps in the adjoint solver ($\nabla_{y_0} h$ as r.h.s)
                $\longrightarrow \sigma_h^{k+1}$
            Compute the reduced gradients

                $$\begin{aligned} (\gamma_{f^i}^{red})^{k+1} &= \nabla_p f^i(y_i^k, p^k) + (C_p^i)^\top (y_i^k, p^k) \sigma_{f^i}^{k+1}, \quad \forall i = 1, \dots, N \\ (\gamma_h^{red})^{k+1} &= \nabla_p h(y_0^k, p^k) + (C_p^0)^\top (y_0^k, p^k) \sigma_h^{k+1} \end{aligned}$$

        end parallel
        Compute the weighted sum of the reduced gradients

            $$(\gamma_f^{red})^{k+1} = \sum_{i=1}^{N} \omega_i \cdot (\gamma_{f^i}^{red})^{k+1}$$

        Approximate the consistent reduced Hessian of the Lagrangian $\longrightarrow B_k$
        $\mu_{k+1} \longleftarrow \frac{h(y_0^k, p^k) - ((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_f^{red})^{k+1}}{((\gamma_h^{red})^{k+1})^\top B_k^{-1} (\gamma_h^{red})^{k+1}}$
        $p^{k+1} \longleftarrow p^k - \tau_k B_k^{-1} (\gamma_f^{red})^{k+1} - B_k^{-1} (\gamma_h^{red})^{k+1} \mu_{k+1}$, $\tau_k \in (0, 1]$ damping factor
        Compute the corresponding geometry and adjust the computational mesh
        compute in parallel
            Starting in $y_i^k$, perform $n_f$ steps in the forward solver $\longrightarrow y_i^{k+1}$, $\forall i = 0, \dots, N$
        end parallel
        $k \longleftarrow k + 1$
    until convergence
end function
```

7.2.2 Nelder-Mead algorithm

The Nelder-Mead (Simplex) algorithm proposed by John Nelder and Roger Mead [129] is a gradient-free optimization method for unconstrained nonlinear problems, which is very popular among engineers for aerodynamic shape optimization. The method belongs to the direct search class of methods, which do not require derivatives and are therefore often claimed to be robust for problems with discontinuities or problems where the function values are affected by noise. Convergence results exist only for a restricted class of problems in one or two dimensions (see e.g. [99]). Several variants of the Nelder-Mead algorithm partly with provable convergence can be found in literature (cf. e.g. [141]).

The basic method solving the nonlinear problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

with $f : \mathbb{R}^n \to \mathbb{R}$ can be stated as follows:

Starting with a simplex in the domain of the function to be minimized, where a simplex S in $\mathbb{R}^n$ is defined as the convex hull

$$S = \{\sum_{i=0}^{n} \iota_i x_i : \iota_i \geq 0, i = 0, \dots, n, \sum_{i=0}^{n} \iota_i = 1\}$$

of $n+1$ affinely independent points $x_0, \dots, x_n \in \mathbb{R}^n$. The algorithm transforms the working simplex in each iteration by attempting to replace the vertex with the highest function value with a better one. If this attempt fails, the simplex is shrinked towards the best vertex. So, one iteration consists of the following steps:

1. Determine the worst $f_w = \max_i f(x_i)$, second worst $f_{sw} = \max_{i \neq w} f(x_i)$ and best $f_b = \min_i f(x_i)$ vertex
2. Compute the centroid $centroid = \frac{1}{n} \sum_{i \neq w} x_i$ with respect to x_w.
3. Transform the current simplex into the new working simplex by using the operations described below.

There are four basic operations in order to transform the simplex:

Reflection Compute the reflect point

$$x_r = centroid + \alpha_r(centroid - x_w)$$

scaled by a positive constant $0 < \alpha_r \leq 1$ called the reflection coefficient.

Expansion By extending the search in direction ($x_r - centroid$), compute the expand point

$$x_e = centroid + \gamma_e(x_r - centroid)\,,$$

where $\gamma_e > 1$ denotes the expansion coefficient.

Contraction

outside The contract-outside point is given by

$$x_c = centroid + \beta_c(x_r - centroid)$$

inside The contract-inside point is given by

$$x_c = centroid + \beta_c(x_w - centroid)$$

with $0 < \beta_c < 1$.

Algorithm 7.7 Nelder-Mead algorithm

```
function NELDER-MEAD
    Choose an initial simplex $S^0$
    $k \leftarrow 0$
    Evaluate the objective function at each of the $n + 1$ vertices.
    repeat
        Determine $f_w^k = \max_i f(x_i^k)$, $f_{sw}^k = \max_{i \neq w} f(x_i^k)$, $f_b^k = \min_i f(x_i^k)$.
        Calculate the reflect point $x_r^k$
        if $f(x_b^k) \leq f(x_r^k) < f(x_{sw}^k)$ then
            $x_w^k \leftarrow x_r^k$
        else
            if $f(x_r^k) < f(x_b^k)$ then
                Calculate the expansion point $x_e^k$
                if $f(x_e^k) < f(x_r^k)$ then
                    $x_w^k \leftarrow x_e^k$
                else
                    $x_w^k \leftarrow x_r^k$
                end if
            else
                if $f(x_{sw}^k) \leq f(x_r^k) < f(x_w^k)$ then
                    Compute the contract outside point $x_{co}^k$.
                    if $f(x_{co}^k) \leq f(x_r^k)$ then
                        $x_w^k \leftarrow x_{co}^k$
                    else
                        Shrink the simplex
                    end if
                else
                    Compute the contract inside point $x_{ci}^k$.
                    if $f(x_{ci}^k) \leq f(x_w^k)$ then
                        $x_w^k \leftarrow x_{ci}^k$
                    else
                        Shrink the simplex
                    end if
                end if
            end if
        end if
        $k \leftarrow k + 1$
    until convergence
end function
```

Shrinking Compute n new vertices

$$\hat{x}_i = x_b + \frac{1}{2}(x_i - x_b)\,, \qquad i = 0, \dots, n,\ i \not= b.$$

The basic Nelder-Mead algorithm is summarized in algorithm 7.7.

Further details on termination criteria and typical choices of the parameters $\alpha_r, \beta_c, \gamma_e$ can be found in [83]. If the variables are well scaled, the method will show a stable behavior in practice even for mildly discontinuous problems. The main disadvantage of the Nelder-Mead algorithm is the high number of function evaluations needed during the optimization. In order to use the Nelder-Mead algorithm for constrained optimization problems also, penalty function approaches can be applied, see e.g. [132] for more details.

7.3 Semi-infinite programming

The robust formulation (4.12 - 4.14) of the aerodynamic shape optimization problem leads to optimization problems of semi-infinite type. Therefore, we will briefly discuss methods for semi-infinite problems of the general form

$$\min_x f(x) \tag{7.45}$$

$$\text{s.t.} \quad h(x, u) \geq 0\,, \quad \forall u \in U, \tag{7.46}$$

where $U \subset \mathbb{R}^m$ is a compact set. The functions f and h are assumed to be real-valued and twice continuously differentiable functions in their respective domains. Problem (7.45 - 7.46) is called a semi-infinite programming problem, if $|U| = \infty$ and x denotes finitely many variables. A survey of theory, applications and methods for more general semi-infinite problems with weaker assumptions on the set U, the so called generalized semi-infinite optimization problems, can be found in [174]. In the application of interest, the set U corresponds to the realizations of the uncertain input parameter.

Obviously, problem (7.45 - 7.46) is equivalent to

$$\min_x f(x) \tag{7.47}$$

$$\text{s.t.} \quad \phi(x) \geq 0 \tag{7.48}$$

with $\phi(x) := \min_{u \in U} h(x, u)$. The function ϕ is the optimal value function of the so called lower level problem

$$\min_{u \in U} h(x, u)\,. \tag{7.49}$$

In the lower level problem, u is the decision variable and x plays the role of an n-dimensional parameter, whereas the upper level problem consists of minimizing f over the feasible set $\mathcal{F} = \{x \in \mathbb{R}^n \mid \phi(x) \geq 0\}$. The main challenge in semi-infinite programming is to solve the lower level problem (7.49) to global optimality [174]. The reformulation (7.47 - 7.48) of the semi-infinite problem (7.45 - 7.46) suggests the use of nondifferentiable optimization approaches, especially descent methods [72], which are robust, but expensive with respect to the computing time. In order to improve the convergence rate, the numerical methods for semi-infinite programming replace the standard problem (7.45 - 7.46) by a sequence of finite problems, which means problems with only a finite number of constraints [72]. Generating the finite problems, we will examine the structure of the feasible set $\mathcal{F} = \{x \in \mathbb{R}^n \mid \phi(x) \geq 0\}$ and especially the active set U_0 at a point x

$$U_0(x) = \{u \in U \mid h(x, u) = 0\}\,. \tag{7.50}$$

For special analysis of first and second order optimality conditions of semi-infinite programming problems, we refer to literature, see e.g. [167, 174], and concentrate in the following on the basics, which are used to construct numerical methods to solve problems of the form (7.45 - 7.46).

If U is a finite set, it is well known that the following inclusion $U_0(x) \subset U_0(\hat{x})$ holds for all x in a neighborhood of a given point $\hat{x}$ due to continuity of h. Whereas in the semi-infinite case, one can show that the active set $U_0(x)$ changes from point to point along the boundary of the feasible set $\mathcal{F}$, see [79] for more details. So, the control of the active set $U_0(x)$ is one of the main features in semi-infinite optimization. Let $\hat{x} \in \mathcal{F}$ and $U_0(\hat{x}) \neq \emptyset$. It is obvious that $\hat{u} \in U_0(\hat{x})$, iff $\hat{u}$ is a global minimum of the lower level problem (7.49). Further, suppose that every $\hat{u} \in U_0(\hat{x})$ is a nondegenerate local minimum of $h(\hat{x}, \cdot)|_U$, which means LICQ, strict complementarity and the second order sufficiency condition are satisfied. Then, it follows that each $\hat{u} \in U_0(\hat{x})$ is an isolated point [79]. Hence, the set $U_0(\hat{x})$ is a closed subset of U without accumulation points. Consequently, $U_0(\hat{x})$ only contains finitely many points, $U_0(\hat{x}) = \{\hat{u}_1, \cdots, \hat{u}_{n_U}\}$. This special structure of the active set U_0 can be exploited in order to replace the semi-infinite problem (7.45 - 7.46) by a sequence of finite ones. According to the way finite problems are generated, three types of methods can be distinguished: discretization methods, methods based on local reduction and semi-continuous methods. Since semi-continuous methods work continuously with respect to the variable u and therefore require the knowledge of all minimizers $u \in U$ of the lower level problem (7.49) for a given x, these methods are hardly applicable in the nonlinear framework, so that we will not treat this approach here and refer to [147] for more details.

7.3.1 Discretization

The most obvious approach solving semi-infinite problems of the form (7.45 - 7.46) by a sequence of finite subproblems is the discretization method. The set U is replaced by a finite subset $U^k \subset U$ and the resulting finite problem is solved using standard optimization algorithms. This procedure is possibly repeated for an enlarged or improved set U^{k+1} in order to achieve a higher approximation quality of the underlying semi-infinite problem. The choice of the discretization of U and the sequence U^k are a crucial point of this method in order to achieve convergence of the solutions of the finite subproblems to the solution of (7.45 - 7.46). The grids U^k can be described either a priori or adaptively during the optimization, which means information obtained by the k-th discretization level is utilized to define U^{k+1}. The general discretization algorithm is stated in algorithm 7.8.

In order to show convergence, it has to be ensured that each accumulation point of the sequence of solutions of the finite subproblems is a solution of the underlying semi-infinite problem (7.45 - 7.46). Considering a sequence of finite subsets U^k fulfilling the following two assumptions, two stability results can be shown [147].

- U^k is a sequence of compact subsets $U^k \subset U^{k+1} \subset U$ and $\lim_{k\to\infty} dist(U^k, U) = 0$.
- There exists $x^\star \in \{x \in \mathbb{R}^n \mid h(x, u) \geq 0, u \in U^0\}$ such that the level set $\Lambda(x^\star, U^0) = \{x \in \mathbb{R}^n \mid f(x) \geq f(x^\star)\} \cap \{x \in \mathbb{R}^n \mid h(x, u) \geq 0, u \in U^0\}$ is bounded.

Algorithm 7.8 Conceptual discretization algorithm

```
function CONCEPTUAL DISCRETIZATION ALGORITHM
    Choose an initial grid $U^0$
    $k \leftarrow 0$
    repeat
        Select a (finer) discretization $U^{k+1} \subset U$
        $k \leftarrow k + 1$
        Compute a solution $x^k$ of the finite subproblem
            $\min_x f(x)$
            s.t. $h(x, u) \geq 0 , \quad \forall u \in U^k$
    until $x^k$ is feasible within a given accuracy $\epsilon > 0$, i.e. $h(x^k, u) \geq -\epsilon,\ \forall u \in U$
end function
```

The first one states convergence, if global minimizers of the finite subproblems can be computed, which is usually only possible for linear or convex problems. The proof can be found e.g. in [146]. In [140], convergence results are presented, where x^k does not need to be a global solution of the finite subproblem. Many modifications and improvements of discretization methods for nonlinear semi-infinite programming problems are suggested in literature, see e.g. [147] providing a comprehensive survey.

7.3.2 Local reduction

The principle of local reduction proposed by Hettich and Jongen [73] locally reduces the semi-infinite problem (7.45 - 7.46) to a finite program keeping track of the elements of $U_0(x)$ for varying x. As discussed before, the feasible set of a semi-infinite problem cannot be locally represented only by the active constraints (in contrast to finite optimization problems). However, under additional assumptions, there exists a finite number of constraints such that the feasible set defined by these constraints coincides with the feasible set of the semi-infinite problem in a neighborhood. Then, a local reduction of the semi-infinite to a finite parametric optimization problem can be performed. The reduction ansatz is said to hold at $\hat{x} \in \mathcal{F}$ if all $u \in U_0(\hat{x})$ are nondegenerate minima of $h(\hat{x}, \cdot)|_U$. It can be shown that the set $U_0(\hat{x})$ is a finite set $U_0(\hat{x}) = \{\hat{u}_1, \cdots, \hat{u}_{n_U}\},\ \hat{x} \in \mathcal{F}$ under this assumption as discussed above. As each point $u_k \in U_0(\hat{x})$ is an isolated point, the local variation with x can be described by the implicit function theorem [174]. Then, there exist open neighborhoods $\mathcal{U}(\hat{x})$ of $\hat{x}$ and $\mathcal{U}(u_k)$ of $u_k,\ k = 1, ..., n_U$ and implicit functions $u_1(x), ..., u_{n_U}(x)$ such that

1. $u_k : \mathcal{U}(\hat{x}) \rightarrow \mathcal{U}(u_k) \cap U,\ k = 1, ..., n_U$
2. $u_k(\hat{x}) = u_k,\ k = 1, ..., n_U$
3. for all $x \in \mathcal{U}(\hat{x})$, $u_k(x)$ is a nondegenerate and isolated local minimizer of the lower level problem (7.49).

Therefore, it follows that

$$\{x \in \mathcal{U}(\hat{x}) \mid h(x, u) \geq 0,\ \forall u \in U\} \Leftrightarrow \{x \in \mathcal{U}(\hat{x}) \mid h_k(x) := h(x, u_k(x)) \geq 0,\ k = 1, ..., n_U\} .$$

So, it is possible to replace the infinite constraints of the semi-infinite problem (7.45 - 7.46) by finite constraints which are locally sufficient to define the feasible region. The finite reduced problem is then defined by

$$\min_{x \in \mathcal{U}(\hat{x})} f(x) \tag{7.51}$$

$$\text{s.t.} \quad h_k(x) \geq 0\,, \quad k = 1, \ldots, n_U. \tag{7.52}$$

For all $x \in \mathcal{U}(\hat{x})$, h_k is a twice continuously differentiable function with respect to x and $x \in \mathcal{U}(\hat{x})$ is a feasible point of the semi-infinite problem (7.45 - 7.46) if and only if $h_k(x) \geq 0,\ k = 1, \ldots, n_U$ [79]. It can be further proven that $x^\star \in \mathcal{U}(\hat{x})$ is a strict, isolated local minimizer of (7.45 - 7.46) iff $x^\star$ is a strict, isolated local minimizer of the reduced problem (7.51 - 7.52), (cf. [147]). The drawbacks of reduction based algorithms lie in the fact that the neighborhood $\mathcal{U}(\hat{x})$ and the function h_k are not known explicitly. However the function values $h_k(x)$ can be computed by $h(x, u_k(x))$, where the $u_k(x)$ are the local minimizers of $h(x, \cdot)$. For the numerical treatment of semi-infinite problems (7.45 - 7.46) by methods based on the reduction ansatz, the assumption on the global minimizers $u_k \in U_0(\hat{x})$ are not sufficient, in fact, this assumption has to be made also on the local minimizer of the lower level problem (7.49). Further, it has to be assumed that the cardinality of the set of local minimizers of the lower level problem (7.49) has to be finite. In literature, there can be found regularity conditions which imply the additional assumptions on the local minimizers, see e.g. [79]. In the following, we will give a conceptual description of methods based on the reduction ansatz and refer to literature for more details on the reduction based methods, e.g. [72, 147, 174]. In order to solve the inner reduced problem, SQP methods are often used providing an efficient way. Methods based on the reduction ansatz were successively applied to even highly nonlinear semi-infinite programming problems, cf. e.g. [67].

Algorithm 7.9 Conceptual reduction algorithm

```
function CONCEPTUAL REDUCTION ALGORITHM
    Choose a starting point $x_0$
    $k \leftarrow 0$
    repeat
        Determine all local minimizers $u_j(x_k),\ j = 1, \ldots, n_{U_l}$ of the lower level problem
            $\min_{u \in U} h(x, u)\,.$
        Apply $n_k$ steps of a finite programming algorithm to the reduced problem
            $\min_{x \in \mathcal{U}(x_k)} f(x)$
            s.t. $h_k(x) \geq 0\,, \quad k = 1, \ldots, n_{U_l}$
        with $h_k(x) = h(x, u_k(x))$. Let $x_k^i,\ i = 1, \ldots, n_k$ be the iterates.
        $x^{k+1} \leftarrow x_k^{n_k}$
        $k \leftarrow k + 1$
    until convergence
end function
```

Remark 7.3.1. Semi-infinite Optimization in the Robust Aerodynamic Framework

In the numerical results in section 8, we will see that the lift constraints show an almost linear behavior with respect to the uncertainties, even in the case of geometrical uncertainties. So, the reduction of the robust formulation (4.12 - 4.14) of the aerodynamic shape optimization problem to a finite optimization problem is straightforward. More precisely, the determination of all local minimizers of the lower level problem can be efficiently performed. In the case of scalar-valued uncertainties, a reduction based ansatz is chosen ensuring the feasibility of the optimized shape over the whole range of variations of the uncertain parameter. Further, the discretization method is applied in the high dimensional case of geometrical uncertainties, since discretization methods provide an efficient and straightforward way to obtain at least a sufficiently good approximate solution of the problem. Due to the monotone behavior of the lift constraints with respect to the geometrical uncertainties, the feasible set of the discretized problem coincides with the feasible set of the semi-infinite problem even for coarse discretization.

Chapter 8

Aerodynamic Applications and Numerical Results

The first part of this chapter is devoted to the numerical comparison of the semi-infinite formulation (4.12 - 4.14) and the chance-constrained formulation (4.18 - 4.20). Uncertainty quantification methods, introduced in chapter 5, are applied in order to transform the robust formulations into numerically tractable optimization problems. The comparison between the two robust formulations is performed considering the optimization of a 2D transonic RAE2822 profile in Euler flow under scalar-valued uncertainties. Further, the influence of adding higher order terms to the mean value in the objective function is investigated considering the robust optimization of the 2D RAE2822 profile under uncertainties in the angle of attack. Afterwards, numerical results of robust optimization considering the combination of two scalar-valued uncertainties in the 2D RAE2822 test case will be presented.

Furthermore, the optimal aerodynamic shape under geometrical uncertainties is computed in a 2D Euler and Navier-Stokes test case. The uncertainties of the shape are modeled as a Gaussian random field of perturbations, which is approximated by a Karhunen-Loève expansion in order to obtain a finite dimensional representation. Additionally, we will investigate the influence of the geometrical uncertainties on the shape to reduce the problem size. The discretization of the probability space then leads to a finite, approximating optimization problem, which will be solved by a parallel version of the one-shot method (cf. chapter 7). Since the same profile in the two test cases is considered, the numerical results allow conclusions regarding the influence of the perturbations on the physics. We will see that even small deviations from the planned shape may cause significant changes in the flow statistics. So, this test case illustrates the need of robust optimization in order to compute a shape, which shows still a good performance under (unavoidable) variations of the assumed setpoints.

In the next section, we will investigate the influence of geometrical uncertainties in a 3D industrial test case by uncertainty quantification methods. Quantities of interests will be expanded in a series of orthogonal polynomials by a non-intrusive polynomial chaos method. The surrogate function is then used to determine statistics of the solution like the mean or variance. We will see that the introduced methods provide an efficient way of quantifying the impact of input uncertainties on the output, even in highly sophisticated 3D test cases. This additional information can then be used e.g. during the preprocessing of a design process to decide whether an uncertainty has to be taken into account in the robust optimization or whether it can be disregarded.

One of the main tasks within the project MUNA was the applicability of the developed methods for industrial purposes. The introduced methods exploit the structure of the underlying problem, which arises from the PDE constrained optimization problem and also from the stochasticity involved. The numerical results will show that the proposed approach leads to numerically tractable optimization tasks and so makes robust design available for the aerodynamic framework.

8.1 Numerical comparison of the introduced robust formulations (test case RAE2822)

We investigate in the following the robust shape optimization of a RAE2822 profile in transonic Euler flow by the use of the structured flow solver FLOWer. In our example, the space is discretized by a 193×33 grid, see figure 8.1. For parametrization, the airfoil is decomposed into thickness and camber distribution. Then, only the camber of the airfoil is parametrized by 21 Hicks-Henne functions and the thickness is not changed during the optimization process.

The deterministic optimization problem is defined by

$$\min_{y,p} f(y,p) \tag{8.1}$$

$$\text{s.t.} \quad c(y,p) = 0 \tag{8.2}$$

$$h(y,p) \geq 0 \tag{8.3}$$

with $p \in \mathbb{R}^{21}$, $y \in \mathbb{R}^{31845}$. The objective function f denotes the drag coefficient C_D, c is the steady state Euler equation with appropriate boundary conditions and the inequality constraint is defined by

$$h(y,p) = C_L(y,p) - C_L^0 \tag{8.4}$$

with $C_L^0 = 0.8263$. The shape of the RAE2822 airfoil is to be optimized for transonic flight conditions, i.e.

$$M = 0.73\,, \qquad \alpha = 2^\circ\,. \tag{8.5}$$

The initial values of the drag and lift coefficients are

$$C_D = 86.88 \cdot 10^{-4}\,, \qquad C_L = 0.8263\,.$$

A single-setpoint optimization using the one-shot method introduced in algorithm 7.5 leads to the coefficients

$$C_D = 37.35 \cdot 10^{-4}\,, \qquad C_L = 0.8261\,,$$

which show a significant reduction of the drag coefficient. In order to compare the introduced robust formulations of the deterministic problem (8.1 - 8.3), we consider in the following the Mach number as an uncertain parameter. To include the uncertainty in the optimization problem, the Mach number is described by a real-valued random variable $s : \mathcal{O} \to \mathbb{R}$, with $(\mathcal{O}, Y, \mathcal{P})$ given probability space, with the assumptions

- The realizations of s are in the range of $[0.7, 0.76]$, i.e. $s(\zeta) \in [0.7, 0.76], \forall \zeta \in \mathcal{O}$.
- The expected value corresponds with the setpoint of the deterministic problem, $\mathbb{E}(s) = 0.73$.

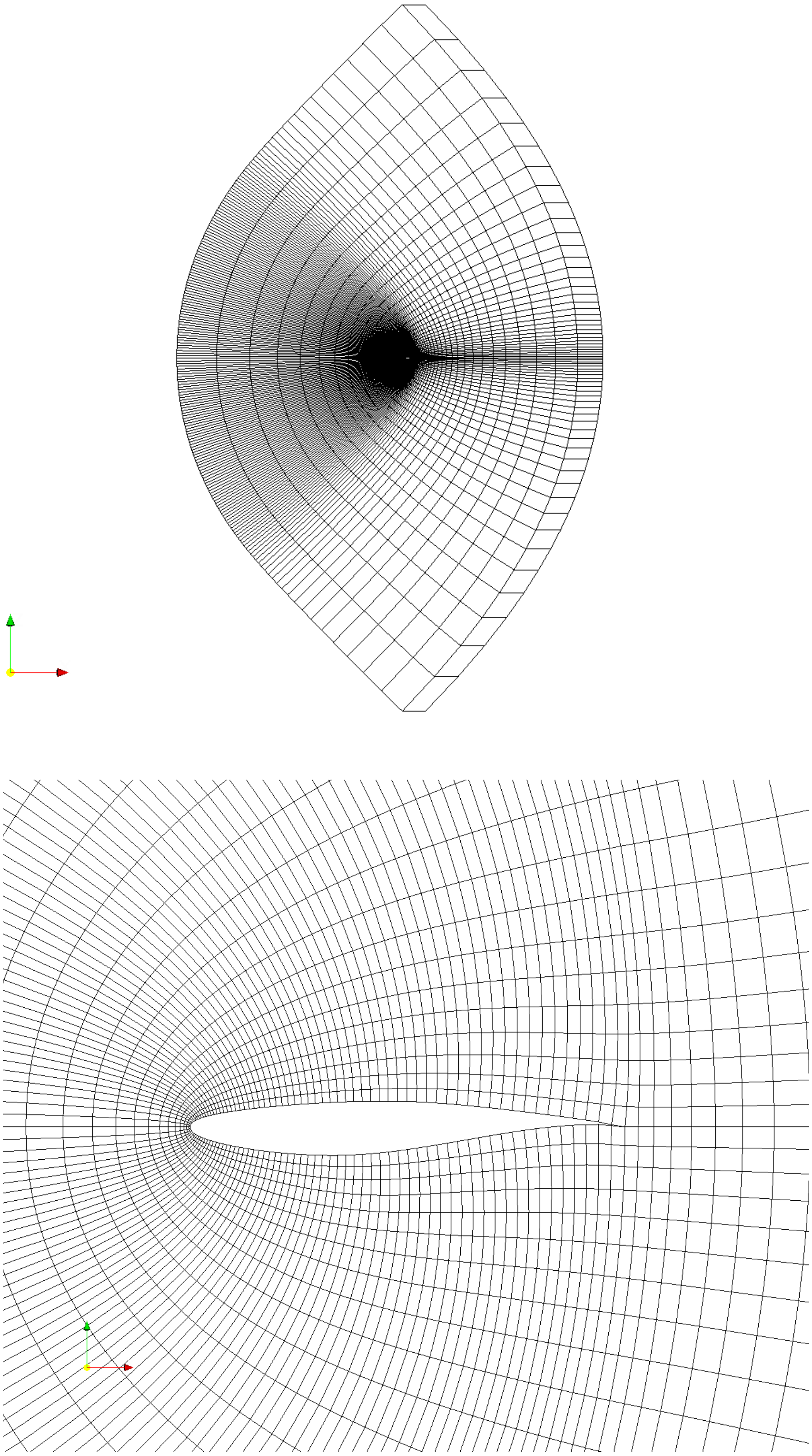


Figure 8.1: C-type grid for the RAE2822 airfoil (Euler flow): the total geometrical plane (above) and zoom around the airfoil (below).

- The random variable s is truncated normally distributed,

$$s \sim \frac{1}{const} \mathcal{N}(0.73, 0.001) \cdot \mathbb{1}_{[0.7,0.76]} \tag{8.6}$$

with $const = \int\limits_{0.7}^{0.76} \frac{1}{\sqrt{2\pi 0.001}} \exp\left(-\frac{(x-0.73)^2}{2 \cdot 0.001}\right) \mathrm{d}x$.

The resulting density function $\varphi_{truncated}$ was already depicted in chapter 3, cf. figure 3.1.

Motivating the robust optimization considering the Mach number as an uncertain parameter in the test case, we first investigate the influence of the Mach number on the target functional f and the inequality constraint h.

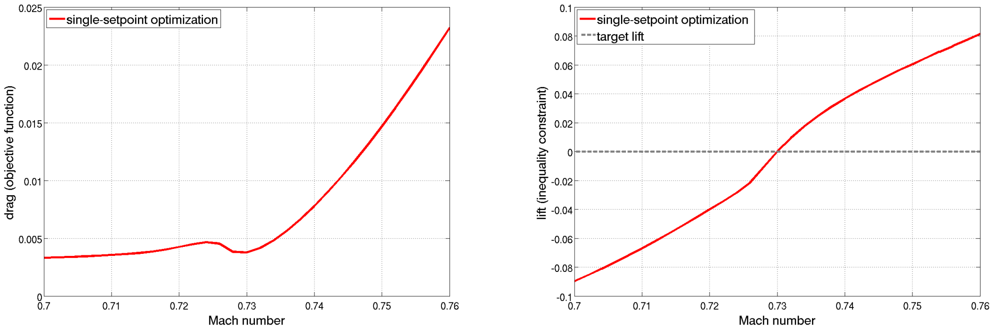


Figure 8.2: Drag and lift performance of the single-setpoint optimization with respect to variations of the Mach number.

As figure 8.2 shows, the drag is very sensitive to changes in the Mach number, especially in the region above the setpoint $M = 0.73$. Even small perturbations of the deterministic setpoint result in a large amount of the drag coefficient. Furthermore, the required target lift is not reached for realizations of the random variable, which are below the nominal point. Summarizing, the sensitivity analysis shows that the optimized shape is not robust against small variations of the Mach number and hence will not reach the expected performance in real world conditions. The task of robust optimization is now to compute a shape, which can improve the performance over the whole range of perturbations.

Another important observation of this study is the fact that the inequality constraint h shows a monotone and almost linear behavior with respect to the Mach number. This property can be used to efficiently approximate the lift coefficient with respect to the input random variable s by uncertainty quantification methods.

In order to numerically compare the robust formulations introduced in chapter 4, we first derive computationally tractable optimization problems using the uncertainty quantification techniques and discretization methods discussed in chapter 5 and 6.

8.1.1 Semi-infinite formulation

The semi-infinite formulation of problem (8.1 - 8.3) is of the form

$$
\begin{aligned}
&\min_{y(s(\zeta)),p} \ \mathbb{E}(f(y(s(\zeta)),p,s(\zeta))) && (8.7)\\
\text{s.t.}\quad & c(y(s(\zeta)),p,s(\zeta)) = 0\,, \qquad \forall\zeta\in\mathcal{O} && (8.8)\\
& h(y(s(\zeta)),p,s(\zeta)) \geq 0\,, \qquad \forall\zeta\in\mathcal{O}, && (8.9)
\end{aligned}
$$

where s denotes the random variable modeling the variations of the Mach number. The computation of the mean value in the objective function (8.7) requires the evaluation of the integral

$$
\begin{aligned}
\mathbb{E}(f(y(s(\zeta)),p,s(\zeta))) &= \int_{\mathcal{O}} f(y(s(\zeta)),p,s(\zeta))\,\mathrm{d}\mathcal{P}(\zeta) && (8.10)\\
&= \int_{\mathbb{R}} f(y(x),p,x)\cdot\varphi_{truncated}\,\mathrm{d}x && (8.11)\\
&= \frac{1}{const\sqrt{2\pi 0.001}}\int_{0.70}^{0.76} f(y(x),p,x)\exp\left(-\frac{(x-0.73)^2}{2\cdot 0.001}\right)\mathrm{d}x. && (8.12)
\end{aligned}
$$

Since there exists no closed form solution of (8.12), the integral has to be numerically approximated. As we consider a one-dimensional random variable, the numerical evaluation of (8.12) can be efficiently done by a Gauß-Hermite quadrature, where the quadrature points $\{s_i\}$ are the roots of the Hermite polynomials, cf. (6.11). Due to the exponential growth of the effort with increasing dimension, the full tensor product Gaussian quadrature rule should be replaced in the higher dimensional case by Smolyak type algorithms, as discussed in chapter 6. Figure 8.3 illustrates the density function $\varphi_{truncated}$ and the corresponding Gaussian points used to approximate the mean value (8.10).

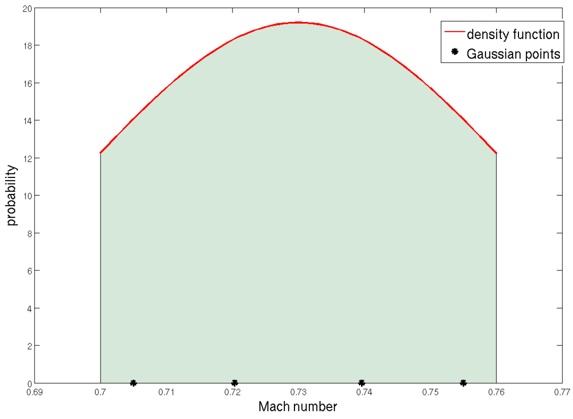


Figure 8.3: Density function of the random variables $M \sim \frac{1}{const}\mathcal{N}(0.73, 0.001)\cdot\mathbb{1}_{[0.7,0.76]}$ and the corresponding Gaussian points.

Therefore, we can reformulate the objective function (8.7) and the appropriate flow equation (8.8) in an approximative fashion in the form of a multiple-setpoint problem for the setpoints $\{s_i\}_{i=1}^N$:

$$\begin{aligned} \min_{y_i,p} \sum_{i=1}^{N} f(y_i, p, s_i)\omega_i & \\ \text{s.t.} \qquad c(y_i, p, s_i) = 0\,, & \qquad \forall i \in \{1, \dots, N\}, \end{aligned}$$

where ω_i denotes the quadrature weights determined by (6.12). Another numerically difficult task is to ensure the feasibility over the whole range of variations, which means to consider infinitely many inequality constraints given by (8.9). A local reduction ansatz as introduced in section 7.3.2 is applied. We assume that the initial shape given by p_0 is a feasible geometry of problem (8.7 - 8.9), i.e.

$$h(y(s(\zeta)), p_0, s(\zeta)) \geq 0\,,\ \forall \zeta \in \mathcal{O}.$$

Using the reduction approach, all local minimizers of the lower level problem

$$\min_{s \in [0.7, 0.76]} h(y(s), p_0, s) \tag{8.13}$$

have to be determined first, cf. algorithm 7.9. In general, the computation of all local minimizers is a difficult task , but due to the monotonicity and almost linearity of the lift constraint with respect to the Mach number, the set of local minimizers contains only the global minimum, which can be easily determined:

$$s_0 = 0.70 = \arg\min_{s \in [0.7, 0.76]} h(y(s), p_0, s)\,. \tag{8.14}$$

Since the changes in the geometry are very small during the optimization, the linearity of the constraint can be conserved and the set of local minimizers of the lower level problem (8.13) remains constant over the whole optimization procedure, i.e.

$$s_0 = 0.70 = \arg\min_{s \in [0.7, 0.76]} h(y(s), p, s) \tag{8.15}$$

is the only local (global) minimum. Therefore, the semi-infinite optimization problem (8.7 - 8.9) can be simplified to

$$\min_{y_i,p} \sum_{i=1}^{N} f(y_i, p, s_i)\omega_i \tag{8.16}$$

$$\text{s.t.} \qquad c(y_i, p, s_i) = 0\,, \qquad \forall i \in \{0, \dots, N\} \tag{8.17}$$

$$h(y_0, p, s_0) \geq 0\,. \tag{8.18}$$

The reformulation (8.16 - 8.18) leads to a finite multiple-setpoint problem, which can be efficiently solved by the generalized one-shot algorithm described in algorithm 7.6. One advantage of problem formulation (8.16 - 8.18) lies in the reusability of the methods developed for the deterministic single-setpoint aerodynamic optimization. The gradients of the objective function (8.16) and the constraint (8.18) can be efficiently computed by the use of the available adjoint solvers. Furthermore, the generalized one-shot method supports a parallel implementation very well, so that the additional flow simulations for each realization of the random variable s_i, $i = 0, \dots, N$ can be performed in parallel.

8.1.2 Chance-constrained formulation

The chance-constrained formulation of problem (8.1 - 8.3) is given by

$$\min_{y(s(\zeta)),p} \ \mathbb{E}(f(y(s(\zeta)),p,s(\zeta))) \tag{8.19}$$

$$\text{s.t.} \qquad c(y(s(\zeta)),p,s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.20}$$

$$\mathcal{P}(\{\zeta \mid h(y(s(\zeta)),p,s(\zeta)) \geq 0\}) \geq \mathcal{P}_0 \tag{8.21}$$

with given $\mathcal{P}_0$ indicating the probability for which the lift constraint h is fullfilled. We reformulate problem (8.19 - 8.21) to a computationally tractable problem following the ideas of [143]. The simplification is performed by applying a Taylor series expansion about the nominal setpoint $s_{no} = 0.73 = \mathbb{E}(s)$, which corresponds to the moment methods discussed in section 5.1. Suppressing further arguments (y,p) for the moment, the second order Taylor approximation of f in (8.19) gives

$$f(s) \doteq f(s_{no}) + \frac{\partial f(s_{no})}{\partial s}(s - s_{no}) + \frac{1}{2}(s - s_{no})^\top \frac{\partial^2 f(s_{no})}{\partial s^2}(s - s_{no})\,.$$

Integrating this, we obtain

$$\mathbb{E}(f) \doteq f(s_{no}) + \frac{1}{2}\frac{\partial^2 f(s_{no})}{\partial s^2}\mathbb{V}ar(s)\,,$$

which corresponds to the SOFM method introduced in (5.8). Obviously, a first order Taylor series approximation estimates the mean of the output using the first moment of the input, which corresponds to the deterministic solution in our application. The FOFM method will only lead to good approximations of the mean value, if the underlying function shows an almost linear behavior, which is not the case in the considered application. In order to deal with the probabilistic chance constraint (8.21), we also have to approximate the probability distribution of the output random variable h. Since the random variable s is assumed to be truncated Gaussian, a first order Taylor approximation of the inequality constraint h is applied, so that the approximation of the lift constraint h is again truncated Gaussian distributed (unlike the second order approximation) (cf. [66]).

$$h(s) \doteq h(s_{no}) + \frac{\partial h(s_{no})}{\partial s}(s - s_{no}) \quad \sim \quad \frac{1}{const}\mathcal{N}\left(h(s_{no}), \sigma^2_{approx}\right) \cdot \mathbb{1}_{R_{approx}}$$

with

$$\sigma_{approx} = \sqrt{\left(\frac{\partial h(s_{no})}{\partial s}\right)^2 \sigma^2}\,,$$

where σ denotes the standard deviation of s, i.e $\sigma = \sqrt{0.001}$, The normalizing constant *const* of the output density is equal to the constant of the input density due to the linearity of the transformation. The support of the resulting density function is given by

$$R_{approx} = \left[\frac{\partial h(s_{no})}{\partial s} \cdot (0.70 - s_{no}) + h(s_{no}), \frac{\partial h(s_{no})}{\partial s} \cdot (0.76 - s_{no}) + h(s_{no})\right]\,.$$

Furthermore, we use the following equivalent representation of the chance constraint (8.21)

$$\mathcal{P}(\{\zeta \mid h(y(s(\zeta)),p,s(\zeta)) \geq 0\}) \geq \mathcal{P}_0 \iff \mathcal{P}(\{\zeta \mid h(y(s(\zeta)),p,s(\zeta)) \leq 0\}) \leq 1 - \mathcal{P}_0$$

in order to reformulate problem (8.19 - 8.21) applying the discussed simplifications:

$$\min_{p} f(y(p,s_{no}),p,s_{no}) + \frac{1}{2}\frac{\partial^2 f(y(p,s_{no}),p,s_{no})}{\partial s^2}\mathbb{V}ar(s) \tag{8.22}$$

$$\text{s.t.} \quad \frac{1}{const\sqrt{2\pi\cdot\sigma^2_{approx}}}\int_{\mathbb{R}} \exp\left(-\frac{(x-h(y(p,s_{no}),p,s_{no}))^2}{2\sigma^2_{approx}}\right)\cdot \mathbb{1}_{\tilde{R}_{approx}}(x)\,\mathrm{d}x \leq 1-\mathcal{P}_0\,, \tag{8.23}$$

where

$$\tilde{R}_{approx} = \begin{cases} R_{approx}, & \text{if } 0 \;>\; \frac{\partial h(s_{no})}{\partial s}\cdot(0.76-s_{no})+h(s_{no}) \\ \left[\frac{\partial h(s_{no})}{\partial s}\cdot(0.70-s_{no})+h(s_{no}),0\right], & \text{else} \end{cases} .$$

Using the implicit function theorem, the flow vector y can be written as a function of the design variables p and the uncertain parameter$s$. Since the flow model (8.20) depends on the uncertainties s, we should be aware that the derivatives with respect to s are total derivatives. As in the semi-infinite formulation, the integral (8.23) is approximated by a numerical quadrature formula. Due to the variable bounds of the intergal, a Newton-Cotes formula of order 6 is applied.

In contrast to the semi-infinite formulation, gradients of the objective function (8.22) and the chance constraint (8.23) with respect to the design variables p cannot be evaluated by the use of the available adjoint solvers. Due to this fact, the Nelder-Mead method, which is a gradient-free algorithm, is chosen, cf. section 7.2.2. The derivatives with respect to the uncertain parameter s are computed by finite differences.

8.1.3 Numerical results

In the following, the results of a numerical comparison between the semi-infinite (8.16 - 8.18) and the chance-constrained formulation (8.22 - 8.23) considering the shape optimization of the introduced RAE2822 test case will be presented. In particular, we compare four formulations:

1. Non-robust single-setpoint optimization at the nominal point $M = 0.73$ (one-shot algorithm)
2. Semi-infinite formulation (8.16 - 8.18) (generalized one-shot algorithm)
3. Chance-constrained formulation (8.22 - 8.23) (Nelder-Mead algorithm)
4. Non-robust single-setpoint optimization at the nominal point $M = 0.73$ but maintaining feasibility over the whole range of perturbations (one-shot algorithm).

The following figures show evaluations of the objective (drag), figure 8.4, as well as the constraint (lift), figure 8.5, comparing the four formulations introduced above.

Comparing the drag performance of the two robust formulations, we can observe that the semi-infinite optimization leads in the region above the nominal point to a better drag coefficient than the chance-constrained optimization. Table 8.1 indicates the mean value of the drag, which is the chosen measure of robustness of the objective function for the four considered cases. The mean value of the semi-infinite formulation is a little bit higher than the mean value of the single-setpoint optimization, as expected due to the lift requirements. But the semi-infinite formulation as well as the chance-constrained formulation show a better drag performance in the mean than the single-setpoint profile, which fulfills the lift requirements over the whole range of variations of the Mach

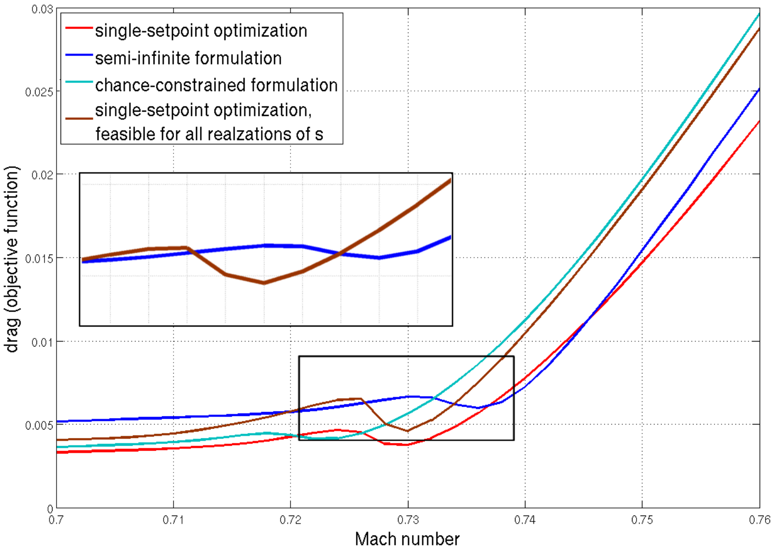


Figure 8.4: Drag performance of optimized airfoils.

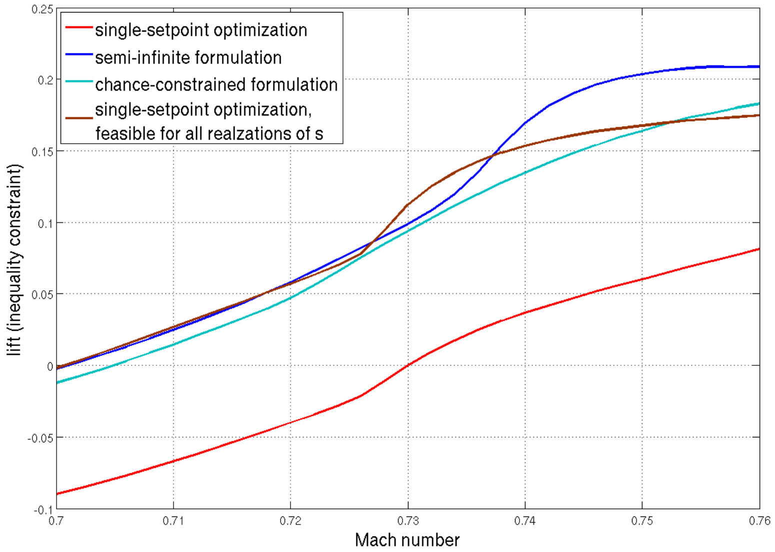


Figure 8.5: Lift performance of optimized airfoils.

	single-setpoint	semi-infinite	chance-constrained	single-setpoint, feasible over the whole range of variations
$\mathbb{E}(f)$	$7.496 \cdot 10^{-3}$	$8.774 \cdot 10^{-3}$	$9.887 \cdot 10^{-3}$	$9.739 \cdot 10^{-3}$

Table 8.1: Comparison of the mean value resulting from the introduced robust formulations.

number. So, the robust formulation of the objective function leads to profiles showing a little bit higher drag at the nominal point than the single-setpoint optimization, but which are more robust to variations of the uncertain parameter and are therefore superior to the single-setpoint optimized shapes. In figure 8.6, the distributions of the pressure coefficient over the airfoil at three different Mach numbers considering the four different optimization strategies are depicted. We can observe

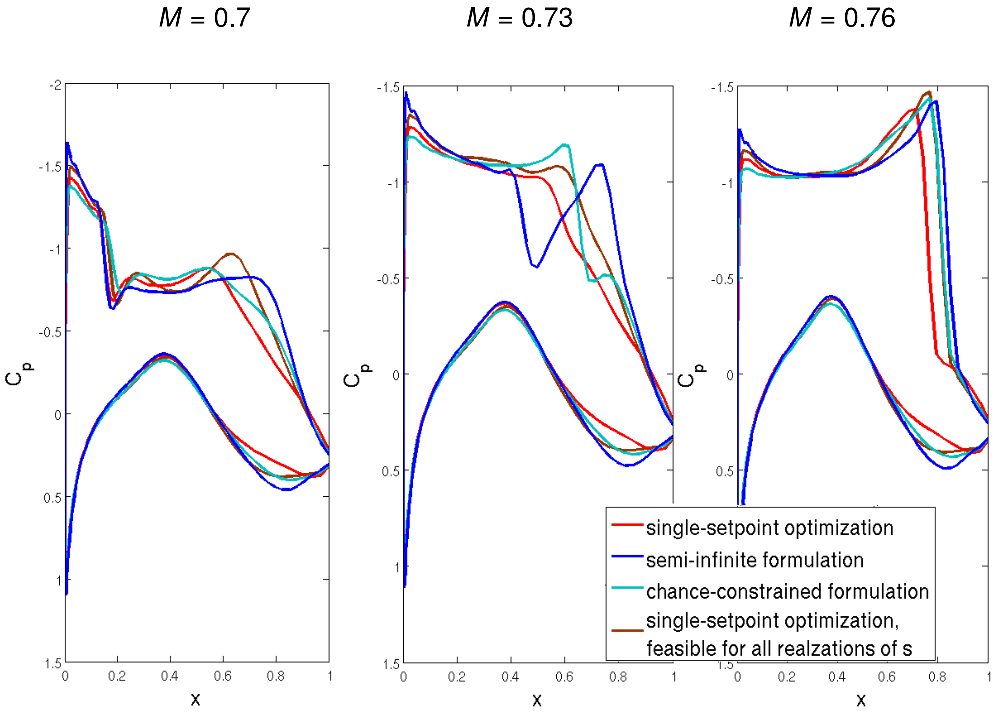


Figure 8.6: Comparison of the distribution of the pressure coefficient over the airfoil.

that the semi-infinite formulation still shows a little shock wave in the nominal point $M = 0.73$ producing a higher drag than the other solutions, whereas the single-setpoint optimization results in a shock free profile. The following four figures show the distribution of the pressure coefficient C_P in space around the airfoil for $M = 0.70$, $M = 0.73$ and $M = 0.76$ of the single-setpoint optimization (cf. figure 8.7), the semi-infinite formulation (cf. figure 8.8), the chance-constrained formulation (cf. figure 8.9) and the single-setpoint optimization maintaining feasibility over the whole range of variations (cf. figure 8.10).

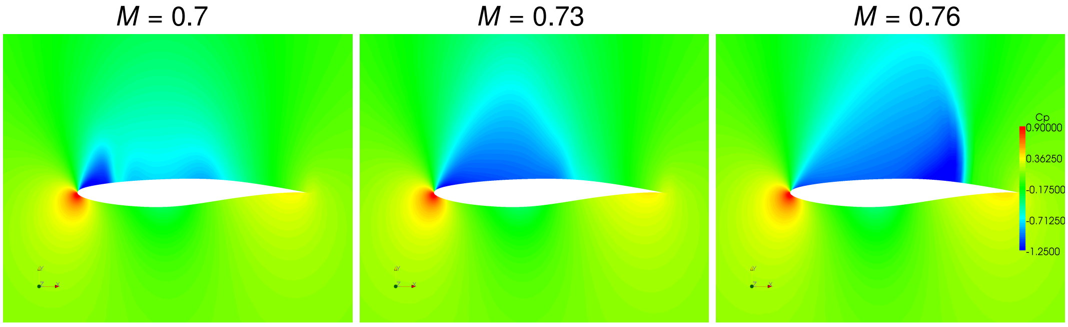

M = 0.7 *M* = 0.73 *M* = 0.76

Figure 8.7: Pressure distribution around the airfoil, *M* = 0.70, *M* = 0.73 and *M* = 0.76 (single-setpoint optimization).

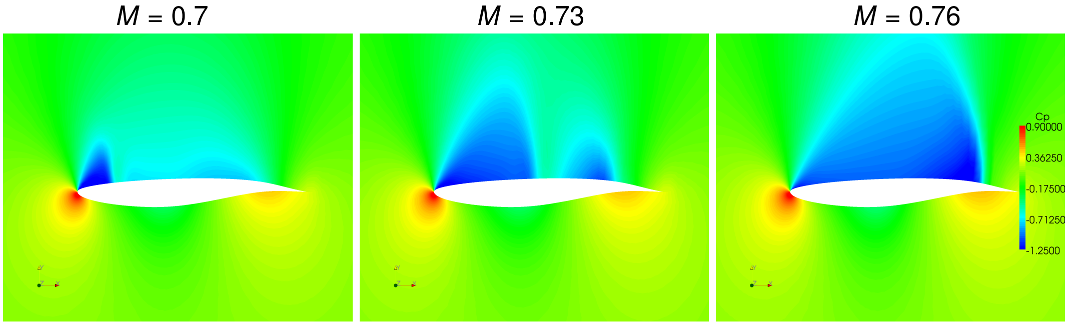

M = 0.7 *M* = 0.73 *M* = 0.76

Figure 8.8: Pressure distribution around the airfoil, *M* = 0.70, *M* = 0.73 and *M* = 0.76 (semi-infinite formulation).

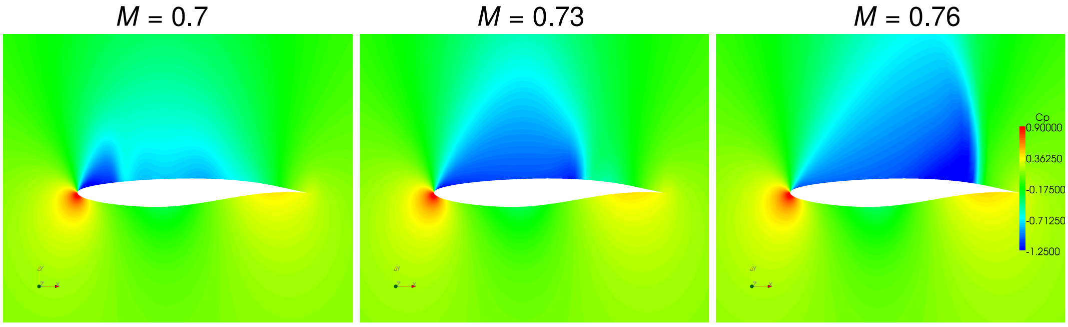

M = 0.7 *M* = 0.73 *M* = 0.76

Figure 8.9: Pressure distribution around the airfoil, *M* = 0.70, *M* = 0.73 and *M* = 0.76 (chance-constrained formulation).

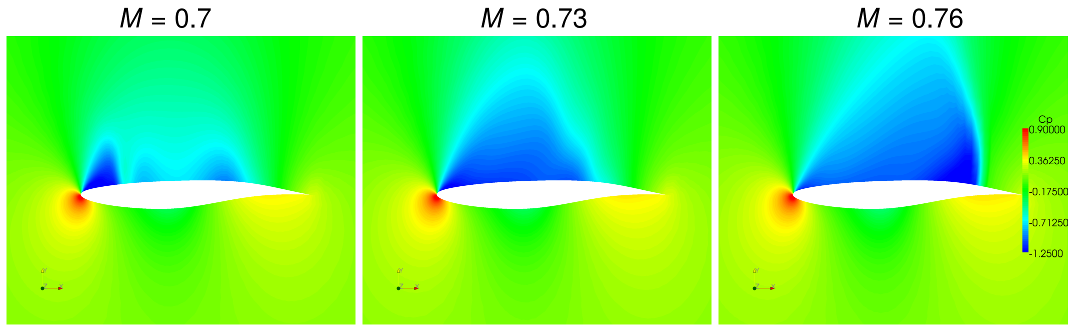

M = 0.7 M = 0.73 M = 0.76

Figure 8.10: Pressure distribution around the airfoil, M = 0.70, M = 0.73 and M = 0.76 (single-setpoint optimization, feasible $\forall s(\zeta)$, $\zeta \in \mathcal{O}$).

In the contour plots 8.7 and 8.8, the two shock waves of the semi-infinite formulation and the smoothed out pressure distribution of the single-setpoint optimization at the nominal point M = 0.73 are clearly visible.

Last, the resulting shapes are compared in figure 8.11.

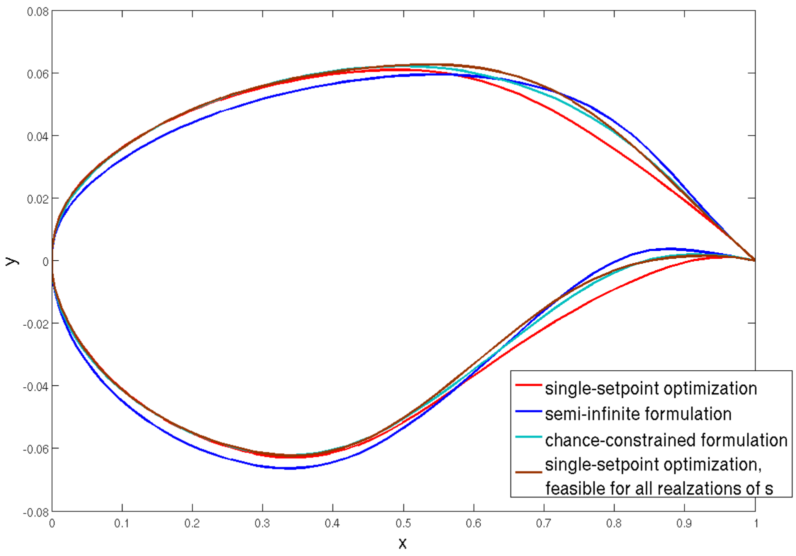


Figure 8.11: Comparison of the resulting shapes.

Comparing the chance-constrained formulation and the semi-infinite formulation, we can state the following observations: The chance-constrained is feasible with a probability $\mathcal{P}_0$ of 90%, whereas the semi-infinite optimization fulfills the lift requirements over the whole range of variations, as required. Further, table 8.1 indicates the better drag performance of the semi-infinite formulation compared to the chance-constrained formulation.

In summary, it can be said that the semi-infinite formulation has a better lift to drag ratio than the chance-constrained formulation, in particular in the region above the setpoint 0.73, due to the fact that the semi-infinite formulation shows a higher lift over the whole range of variations (cf. figure 8.5) and for the greater part a better drag performance than the chance-constrained (cf. figure 8.4). Further, the semi-infinite formulation allows to apply the generalized one-shot method and the adjoint approach to calculate gradients, which is much more computationally efficient than the gradient-free optimization method used to solve the chance-constrained problem. For this reason, the semi-infinite approach is more suitable for high dimensional design tasks.

Due to the better performance over the whole range of perturbations and the more efficient optimization strategy, the semi-infinite formulation seems to be favorable in our application. The following numerical results are therefore based on this formulation.

8.2 Variance reduction under Mach number uncertainties (test case RAE2822)

In this section, we will investigate the influence of adding higher order terms to the mean value in the objective function. This approach is often discussed in the context of robust optimization in order to compute more conservative designs. The different measures of robustness are compared considering the optimization of the RAE2822 profile in transonic Euler flow. In the underlying deterministic shape optimization problem defined by (8.1 - 8.3), the angle of attack is assumed to be uncertain and modeled as a real-valued random variable $s : \mathcal{O} \to \mathbb{R}$, with $(\mathcal{O}, Y, \mathcal{P})$ given probability space. The following assumptions are made:

- $s(\zeta) \in [1.8, 2.2]$, $\forall \zeta \in \mathcal{O}$.
- $\mathbb{E}(s) = 2.0$.
- $s \sim \frac{1}{const}\mathcal{N}(2.0, 0.1) \cdot \mathbb{1}_{[1.8,2.2]}$, $\quad const = \int\limits_{1.8}^{2.2} \frac{1}{\sqrt{2\pi 0.1}} \exp\left(-\frac{(x-2.0)^2}{2\cdot 0.1}\right) \mathrm{d}x.$

Using the combination of mean value and variance as a measure of robustness, the semi-infinite formulation of the deterministic problem (8.1 - 8.3) is then given by

$$\min_{y(s(\zeta)),p} \ \mathbb{E}(f(y(s(\zeta)), p, s(\zeta))) + \theta \cdot \mathbb{V}ar(f(y(s(\zeta)), p, s(\zeta))) \tag{8.24}$$

$$\text{s.t.} \qquad c(y(s(\zeta)), p, s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.25}$$

$$h(y(s(\zeta)), p, s(\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}, \tag{8.26}$$

where the coefficient θ indicates the risk aversion, cf. section 4.3.

Since the lift constraint also shows a monotone behavior with respect to the angle of attack in the considered range of variations, the infinitely many inequality constraints can be replaced by only one constraint using the reduction ansatz.

Due to the low dimension of the probability space, the integrals in the objective function (8.24) are approximated by a Gaussian quadrature formula. The truncated normal density function of s and the corresponding quadrature points are shown in figure 8.12.

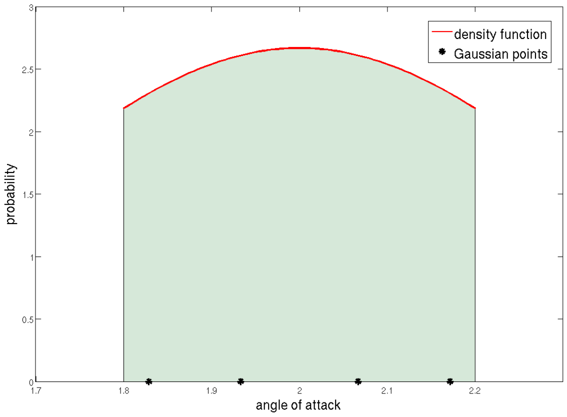


Figure 8.12: Density function of the random variables $\alpha \sim \frac{1}{const}\mathcal{N}(2.0, 0.1) \cdot \mathbb{1}_{[1.8,2.2]}$ and the corresponding Gaussian points.

The discretization of (8.24 - 8.26) leads to

$$\min_{y_i,p} \quad \sum_{i=1}^{N} f(y_i,p,s_i)\,\omega_i + \theta \cdot \left(\sum_{i=1}^{N} f(y_i,p,s_i)^2\,\omega_i - (\sum_{i=1}^{N} f(y_i,p,s_i)\,\omega_i)^2 \right) \tag{8.27}$$

$$\text{s.t.} \qquad c(y_i,p,s_i) = 0\,, \qquad \forall i \in \{0,\dots,N\} \tag{8.28}$$

$$h(y_0,p,s_0) \geq 0\,, \tag{8.29}$$

which is solved by the generalized one-shot method described in algorithm 7.6. During the optimization, the required iterations in the flow and adjoint solver are performed in parallel for each discretization point.

The numerical results presented below compare the robust solutions of problem (8.27 - 8.29) for four different values of the parameter θ, which controls the risk aversion of the designs. The highest considered value of the risk aversion parameter is chosen such that the mean and the variance are of the same magnitude.

First, we can observe that the lift constraint is not influenced by the different risk measures in the objective function. As figure 8.14 indicates, the lift performance of the four optimized airfoils is almost the same and the required lift is reached over the whole range of variations. However, the drag performances of the optimized shapes shown in figure 8.13 clearly demonstrate the influence of the parameter θ on the conservatism of the resulting airfoils. A high value of the risk aversion parameter θ strongly penalizes a high value of the variance. Hence, the drag functions of the formulations with higher values of θ get closer to the variance optimal case, that means a variance equals zero resulting in a constant drag over the whole range of variations.

Further, we can state that the variance reduction leads to a higher drag than the optimization of the mean value. The tradeoff between mean value and variance minimization is illustrated in table 8.2.

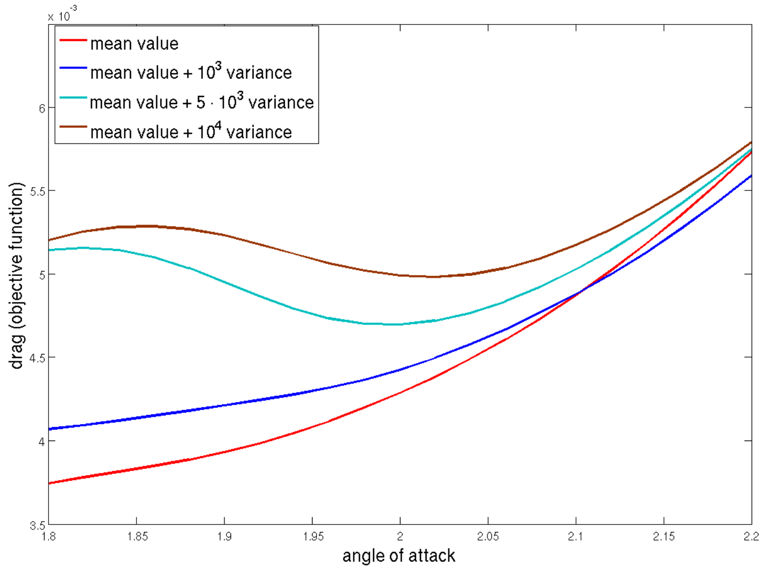


Figure 8.13: Comparison of drag performance for different risk aversion values.

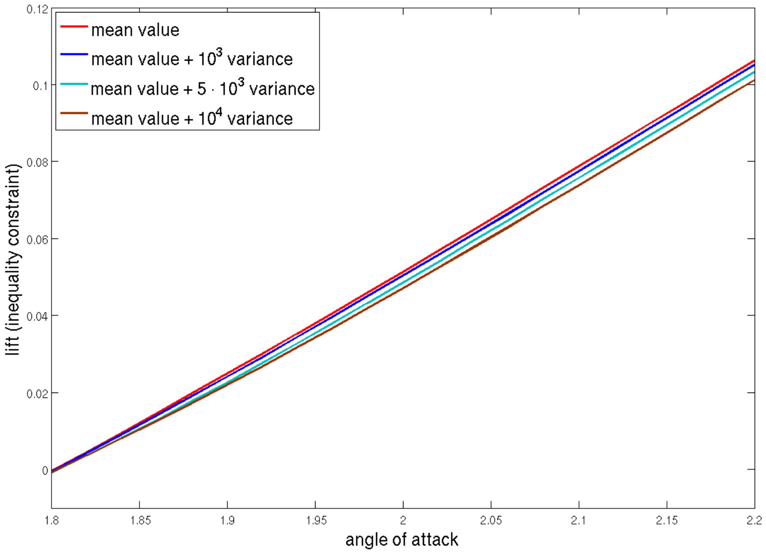
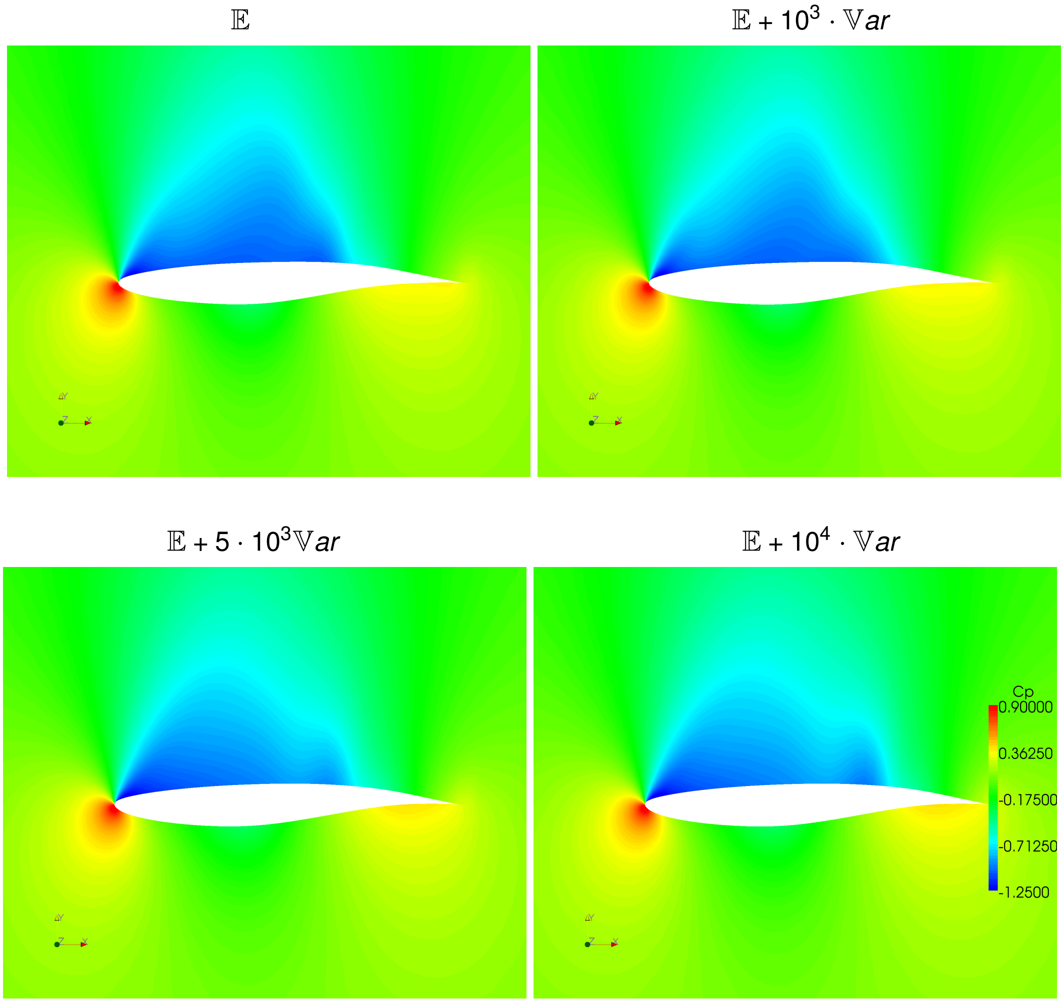


Figure 8.14: Comparison of lift performance for different risk aversion values.

objective function	$\mathbb{E}$	$\mathbb{E} + 10^3 \cdot \mathbb{V}ar$	$\mathbb{E} + 5 \cdot 10^3\ \mathbb{V}ar$	$\mathbb{E} + 10^4 \cdot \mathbb{V}ar$
$\mathbb{E}(f)$	$4.430 \cdot 10^{-3}$	$4.564 \cdot 10^{-3}$	$5.000 \cdot 10^{-3}$	$5.204 \cdot 10^{-3}$
$\mathbb{V}ar(f)$	$3.106 \cdot 10^{-7}$	$1.740 \cdot 10^{-7}$	$6.420 \cdot 10^{-8}$	$3.268 \cdot 10^{-8}$

Table 8.2: Comparison of the mean value and variance resulting from the different risk measures.

Figure 8.15 compares the four different C_P distributions around the airfoil at the nominal point $\alpha = 2.0°$. The pressure distributions of the first two optimized airfoils with $\theta = 0$ and $\theta = 10^3$ are almost the same. If we take a look at the results in table 8.2, both lead to almost the same mean


$\mathbb{E}$

$\mathbb{E} + 10^3 \cdot \mathbb{V}ar$

$\mathbb{E} + 5 \cdot 10^3 \mathbb{V}ar$

$\mathbb{E} + 10^4 \cdot \mathbb{V}ar$

Figure 8.15: Pressure distribution around the airfoil resulting from the different risk measures at the nominal point $\alpha = 2°$.

and variance of the drag. Thus, the risk aversion parameter θ seems to be too small to effect the results. On the other hand, the other two formulations with $\theta = 5 \cdot 10^3$ and $\theta = 10^4$ show a reduction of the variance, an amount in the mean value and pressure distributions, which differs quite strongly from the mean value optimization. Comparing the resulting optimized shapes (cf. figure 8.16), the same effect can be observed.

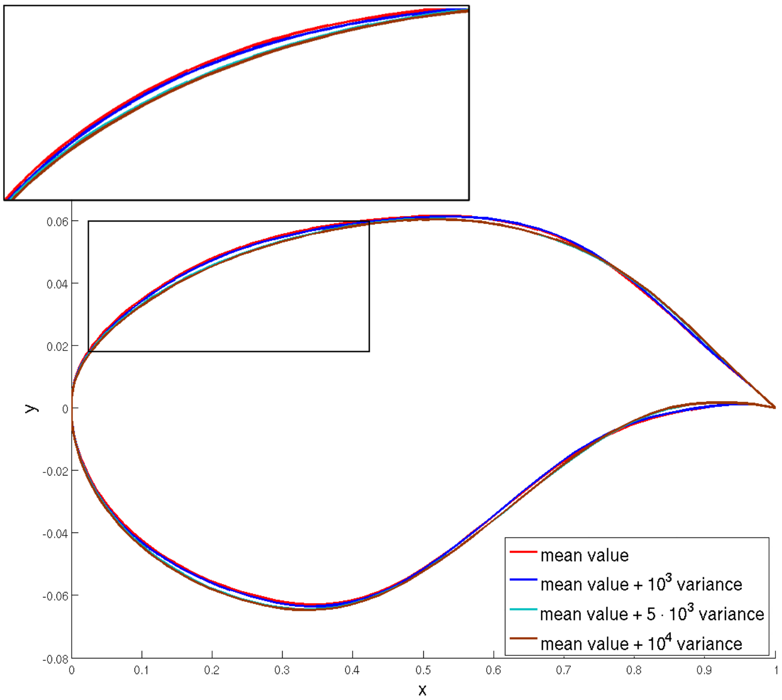


Figure 8.16: Comparison of the resulting shapes optimized for different risk aversion values.

This example points out how important the choice of the risk aversion parameter θ is. A too small value of the parameter θ will not affect the results and a too large value may lead to overly conservative designs. In order to properly determine the value of the risk aversion parameter, additional knowledge on the magnitude of the variance and the mean as well as on the desired grade of robustness needs to be at hand. Another possibility to formulate a variance reducing optimization problem is to define an additional constraint leading to the following problem

$$\min_{y(s(\zeta)),p} \; \mathbb{E}(f(y(s(\zeta)),p,s(\zeta))) \tag{8.30}$$

$$\text{s.t.} \quad c(y(s(\zeta)),p,s(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.31}$$

$$h(y(s(\zeta)),p,s(\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.32}$$

$$\mathbb{V}ar(f(y(s(\zeta)),p,s(\zeta))) \leq b_{\mathbb{V}ar}\,. \tag{8.33}$$

The variance of the objective function is required to be less or equal than a given bound $b_{\mathbb{V}ar}$. Then, the risk aversion parameter is related to the Lagrangian multiplier resulting from the corresponding Lagrangian function of problem (8.30 - 8.33). Considering the variance as an additional constraint, the bound $b_{\mathbb{V}ar}$ has to be given. This bound can be determined i.e. by policies, which are required, or by design requirements. Therefore, formulation (8.30 - 8.33) may offer a practical alternative to problem (8.24 - 8.26) from a modeling point of view. The application of the discussed methods to the optimization problem (8.30 - 8.33) is straightforward.

8.3 Numerical results of optimization under uncertain Mach number and angle of attack (test case RAE2822)

Beside the Mach number, the angle of attack is considered as an additional uncertainty. Again, the robust optimization of the RAE2822 profile in transonic Euler as in the previous sections is investigated. The same space discretization and parametrization of the shape is chosen, so that the deterministic optimization problem coincides with (8.1 - 8.3). The Mach number and the angle of attack are modeled as real-valued random variables $s_M,\ s_\alpha$ and the following assumptions on the uncertain parameter are made:

- $s_M,\ s_\alpha$ are independent.
- $s_M \sim \frac{1}{const_M}\mathcal{N}(0.73, 0.001) \cdot \mathbb{1}_{[0.7,0.76]}$ with $const_M = \int\limits_{0.7}^{0.76} \frac{1}{\sqrt{2\pi 0.001}} \exp\left(-\frac{(x-0.73)^2}{2\cdot 0.001}\right) \mathrm{d}x.$
- $s_\alpha \sim \frac{1}{const_\alpha}\mathcal{N}(2.0, 0.1) \cdot \mathbb{1}_{[1.8,2.2]}$ with $const_\alpha = \int\limits_{1.8}^{2.2} \frac{1}{\sqrt{2\pi 0.1}} \exp\left(-\frac{(x-2.0)^2}{2\cdot 0.1}\right) \mathrm{d}x.$

Considering the random vector $\boldsymbol{s} = \begin{pmatrix} s_M \\ s_\alpha \end{pmatrix}$, the joint distribution of $\boldsymbol{s}$ is given by

$$\frac{1}{const}\mathcal{N}\left(\begin{pmatrix} 0.73 \\ 2.0 \end{pmatrix}, \begin{pmatrix} 0.001 & 0 \\ 0 & 0.1 \end{pmatrix}\right) \cdot \mathbb{1}_{[0.7,0.76]\times[1.8,2.2]}$$

with $const = const_M \cdot const_\alpha$.
The joint probability density function is depicted in figure 8.17. Since the semi-infinite formulation outperforms the chance-constrained formulation in the numerical comparison, the semi-infinite formulation is used to include the uncertainties in the deterministic shape optimization problem (8.1 - 8.3):

$$\min_{y(\boldsymbol{s}(\zeta)),p} \ \mathbb{E}(f(y(\boldsymbol{s}(\zeta)), p, \boldsymbol{s}(\zeta))) \tag{8.34}$$

$$\text{s.t.} \quad c(y(\boldsymbol{s}(\zeta)), p, \boldsymbol{s}(\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.35}$$

$$h(y(\boldsymbol{s}(\zeta)), p, \boldsymbol{s}(\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}. \tag{8.36}$$

The evaluation of the mean value (8.34) is performed by a 2D tensor grid based on 1D Gaussian quadrature formulas, which is in the low dimensional case an efficient way to approximate the integral. In the higher dimensional case, we refer to sparse grids to circumvent the curse of dimensionality. The resulting quadrature points are illustrated in figure 8.18.

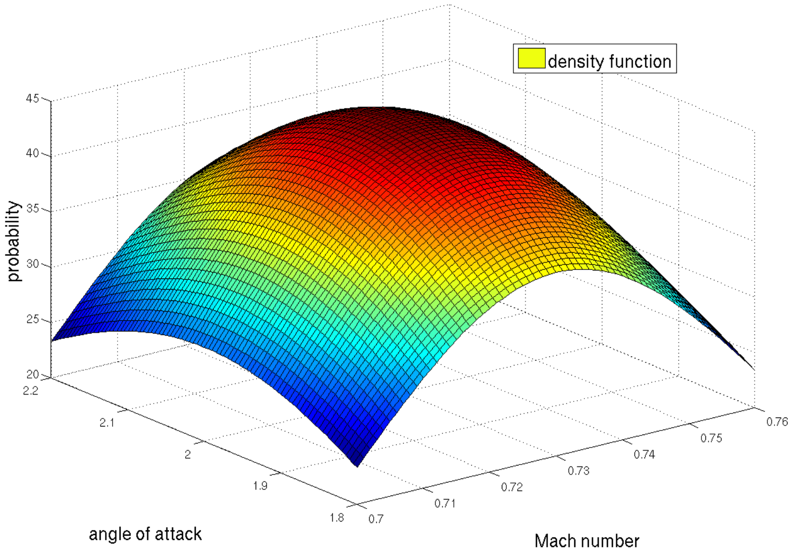


Figure 8.17: Joint probability density function of Mach number and angle of attack.

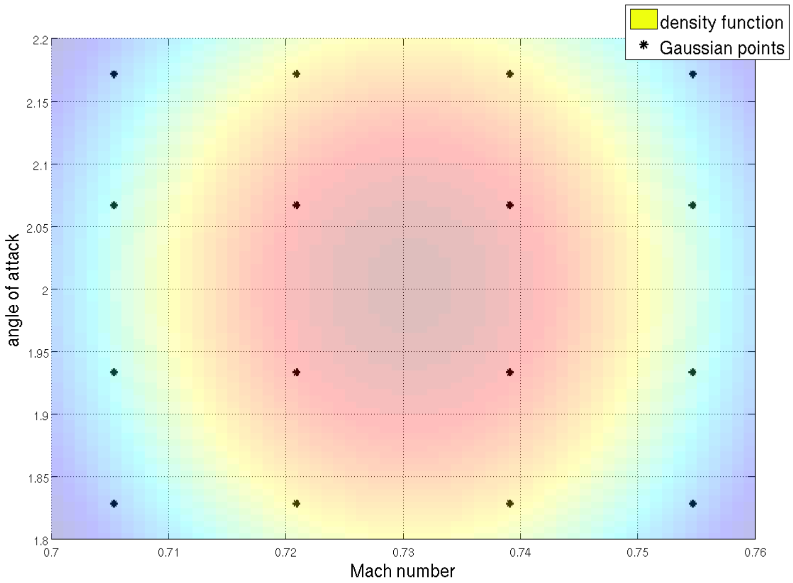


Figure 8.18: Gaussian quadrature points in order to approximate statistics.

Due to the monotonicity and almost linearity of the lift constraint with respect to the Mach number and angle of attack, the reduction ansatz (cf. section 7.3.2) to reduce the infinitely many inequalities to a finite number of constraints can be applied. Hence, (8.34 - 8.36) can be simplified to

$$\min_{y_i, p} \sum_{i=1}^{N} f(y_i, p, \boldsymbol{s}_i)\omega_i \tag{8.37}$$

$$\text{s.t.} \qquad c(y_i, p, \boldsymbol{s}_i) = 0\,, \qquad \forall i \in \{0, \dots, N\} \tag{8.38}$$

$$h(y_0, p, \boldsymbol{s}_0) \geq 0\,. \tag{8.39}$$

with $\boldsymbol{s}_0 = \begin{pmatrix} 0.70 \\ 1.8 \end{pmatrix}$. Altogether, iterations in the flow solver and the adjoint solvers have to be performed for 17 realizations of the Mach number and angle of attack in each iteration of the optimization algorithm. Using the generalized one-shot method (cf. algorithm 7.6), the required iterations can be done in parallel, so that the additional computational time resulting from the robust optimization can be significantly reduced. Figure 8.19 and figure 8.20 illustrate the results of the robust optimization of (8.37 - 8.39) compared to the single-setpoint optimization.

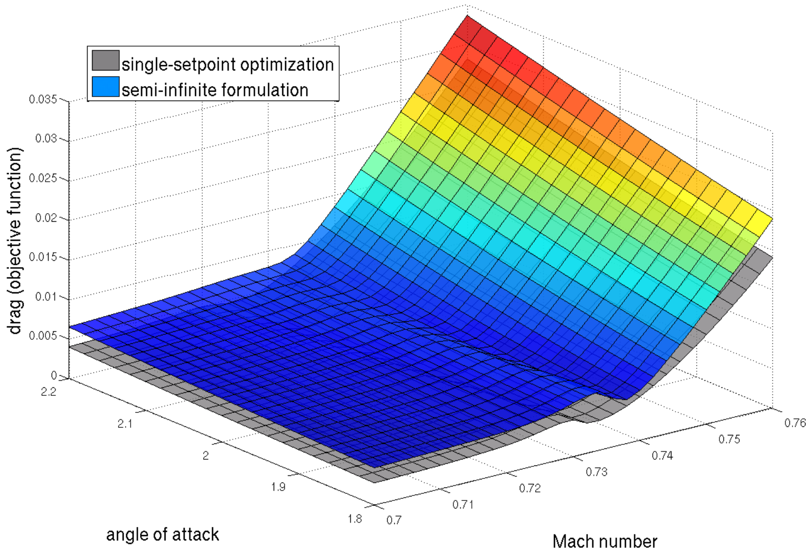


Figure 8.19: Drag performance under Mach number and angle of attack uncertainties.

The semi-infinite formulation gives a higher drag over the whole range of variations than the usual single-setpoint case. The mean value of the single-setpoint optimization is equal to

$$\mathbb{E}_{single-setpoint} = 7.565 \cdot 10^{-3},$$

whereas the robust optimization gives

$$\mathbb{E}_{semi-infinite} = 9.806 \cdot 10^{-3}.$$

But the solution of the semi-infinite formulation is always feasible, as required, whereas the single-setpoint optimization achieves the given lift only in a small region of the variations (cf. figure 8.20).

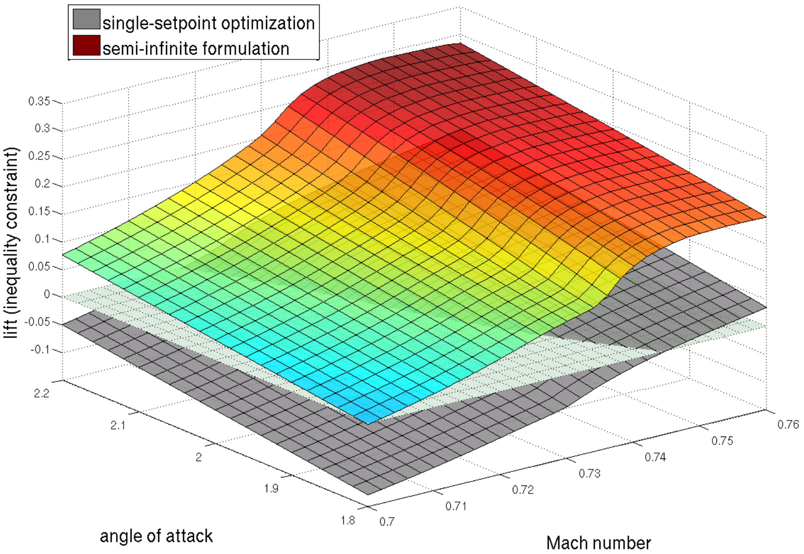


Figure 8.20: Lift performance under Mach number and angle of attack uncertainties.

In summary, the semi-infinite formulation leads once again to a better lift to drag ratio just as in the one-dimensional stochastic case. In figure 8.21, the comparison between the pressure distribution around the airfoil of the semi-infinite and single-setpoint optimization at the nominal point $M = 0.73$ is depicted. We can observe that the semi-infinite formulation leads to two small shock waves on the upper surface, inducing the higher drag at the nominal point than the single-setpoint profile, which is a shock free profile at this setpoint.

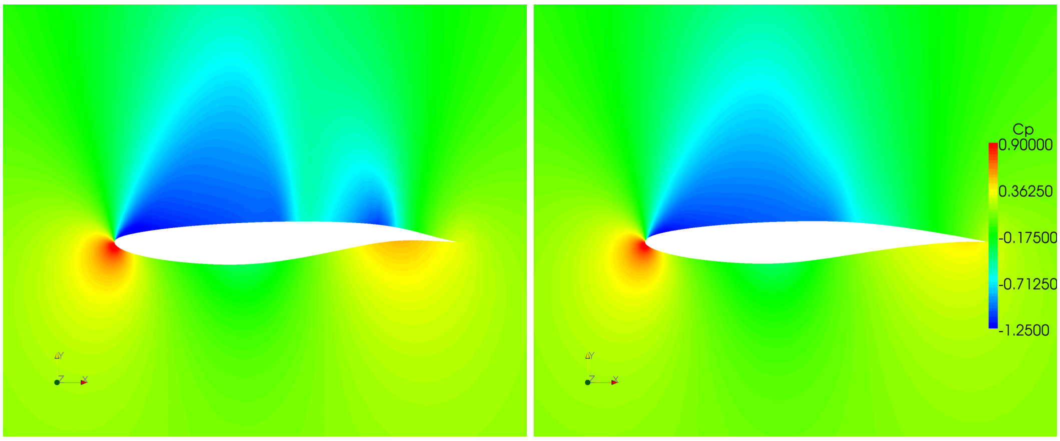


Figure 8.21: Pressure distribution around the airfoil at $M = 0.73$ of the semi-infinite (left) and the single-setpoint (right) optimization.

Last, the resulting shape of the semi-infinite formulation compared to the single-setpoint optimization is depicted in figure 8.22.

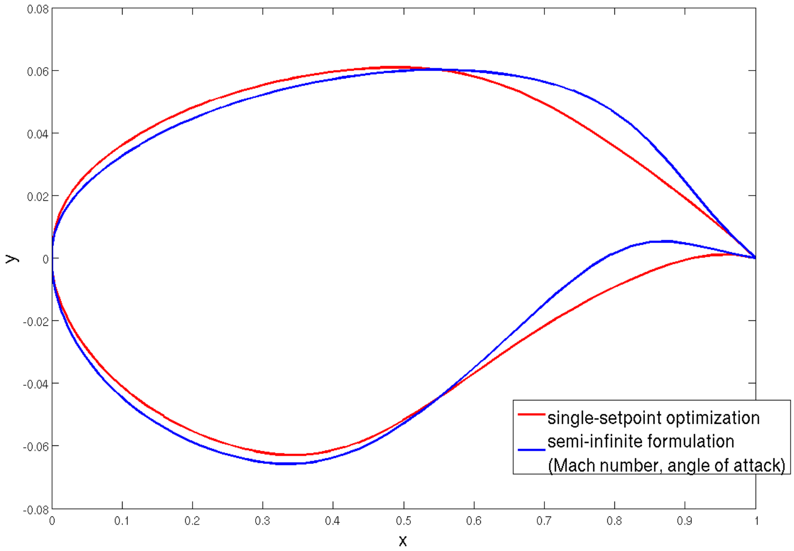


Figure 8.22: Comparison of the resulting shapes under Mach number and angle of attack uncertainties.

Considering the Mach number and the angle of attack as uncertain parameters, the resulting robust shape differs quite strongly from the single-setpoint optimized shape, which can be partly attributed to the higher lift requirements resulting from the semi-infinite formulation.

8.4 Numerical results considering geometrical uncertainties (test case RAE2822)

In this section, we present the numerical results of robust optimization under shape uncertainties of a RAE2822 profile in Euler and Navier-Stokes flow. Since the robust optimization problem is solved within a one-shot framework, we use the flow solver TAU provided by DLR, which allows the computation of gradients by the adjoint approach in the Euler as well as in the Navier-Stokes test case. The grid used in the Euler case is the structured FLOWer grid depicted in figure 8.1 converted to the TAU format, which describes the profile by 129 surface grid points. In the Navier-Stokes case, an unstructured grid with 192 surface grid points is considered (cf. figure 8.23). Again, the airfoil is parametrized by 21 Hicks-Henne functions in both test cases.

The geometrical uncertainties are characterized by a Gaussian random field defined on the shape Γ and on a given probability space $(\mathcal{O}, Y, \mathcal{P})$

$$\psi : \Gamma, \mathcal{O} \to \mathbb{R}\,. \tag{8.40}$$

In each point $x \in \Gamma$, a normally distributed random variable $\psi(x, \cdot)$ models the uncertainty of the

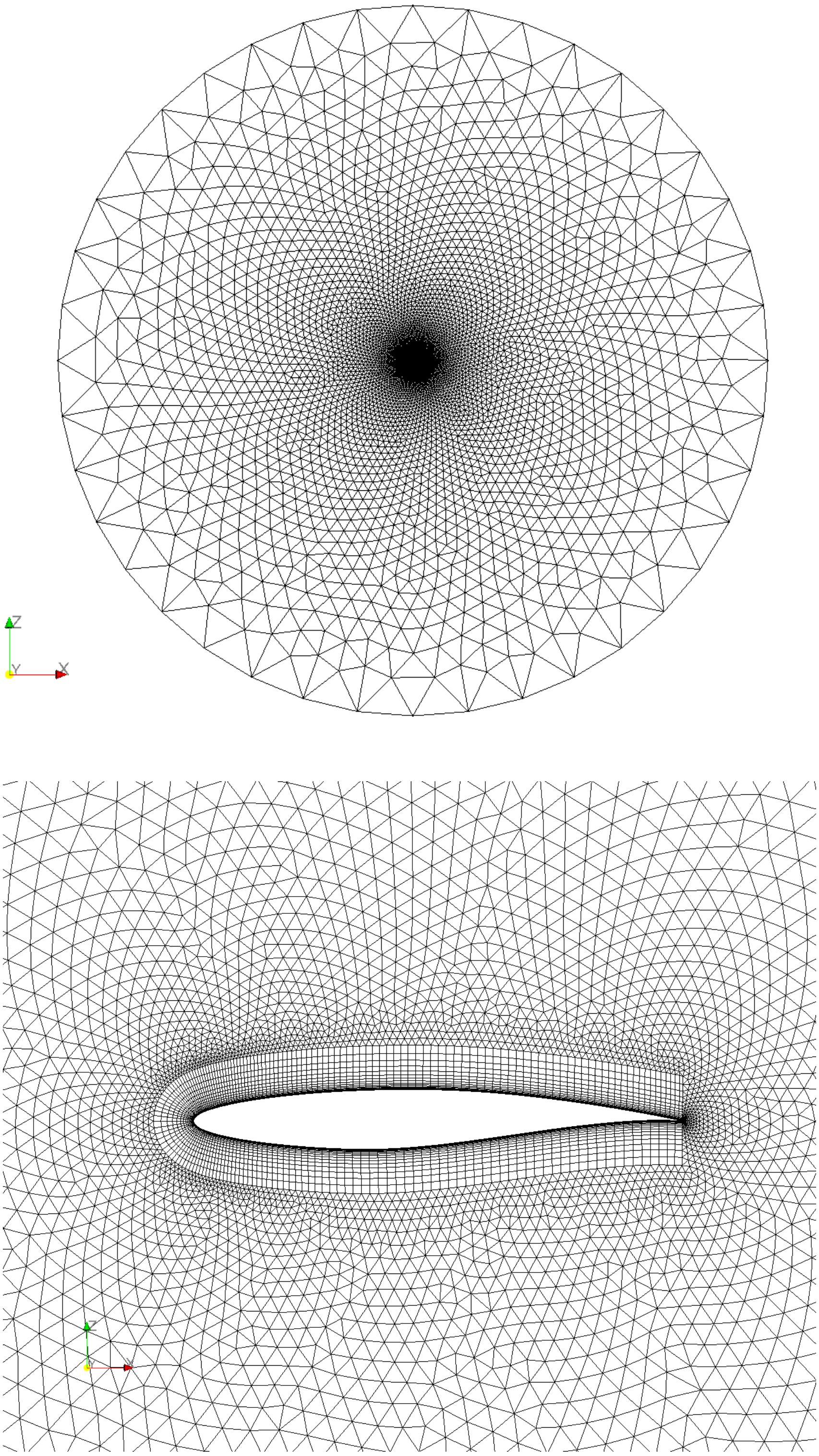


Figure 8.23: Unstructured grid for the RAE2822 airfoil (Navier-Stokes flow): the total geometrical plane (above) and zoom around the airfoil (below).

geometry. To properly describe the random field ψ of perturbations, the following second order statistics are assumed:

- Since no perturbations are expected, the mean value of the random field ψ is equal to 0 for all $x \in \Gamma$, i.e.
$$\mathbb{E}\left(\psi\left(x,\zeta\right)\right) = 0\,, \qquad \forall x \in \Gamma. \tag{8.41}$$
- The interaction of the normally distributed random variables on the shape is described by a squared exponential covariance function
$$Cov\left(x,y\right) = (0.005)^2 \cdot \exp\left(-\frac{\|x-y\|^2}{(0.1)^2}\right)\,, \qquad \forall x,y \in \Gamma. \tag{8.42}$$

Then, a perturbed geometry is given by

$$v\left(x,\zeta\right) = x + \psi\left(x,\zeta\right)\cdot n\left(x\right)\,, \qquad \forall x \in \Gamma,\ \zeta \in \mathcal{O}\,, \tag{8.43}$$

where n is the unit vector in x normal to the profile Γ. In order to compute a flow simulation and adjoint solution considering the perturbed geometry, the TAU deformation tool is used to adjust the mesh accordingly to the realization of the random field ψ of perturbations. Since the convergence behavior of the solver is very sensitive to changes of the trailing edge, this part of the profile cannot be deformed. Figure 8.24 shows the part of the wing, where perturbations of the shape may occur. Hence, the Gaussian random field ψ is defined on a smaller domain $\Gamma_{red} = \{(x,y) \in \Gamma |\ y \leq 0.8\}$

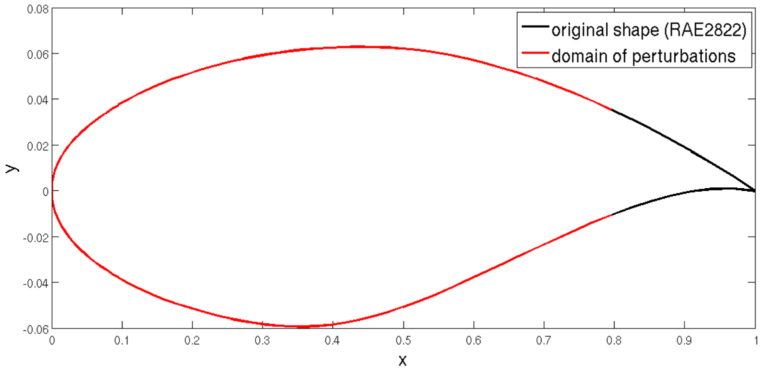


Figure 8.24: Domain of perturbations (test case RAE2822).

$$\psi : \Gamma_{red}, \mathcal{O} \rightarrow \mathbb{R}\,.$$

ensuring the applicability of the deformation tool. A realization of the random field on the reduced domain is depicted in figure 8.25. The resulting perturbed shape is shown in figure 8.26. Using the semi-infinite formulation to include the geometrical uncertainties in the deterministic shape optimization problem, we obtain the following robust optimization problem

$$\min_{y(\psi(\cdot,\zeta)),p} \ \mathbb{E}(f(y(\psi(\cdot,\zeta)),p,\psi(\cdot,\zeta))) \tag{8.44}$$
$$\text{s.t.} \qquad c(y(\psi(\cdot,\zeta)),p,\psi(\cdot,\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.45}$$
$$h(y(\psi(\cdot,\zeta)),p,\psi(\cdot,\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}. \tag{8.46}$$

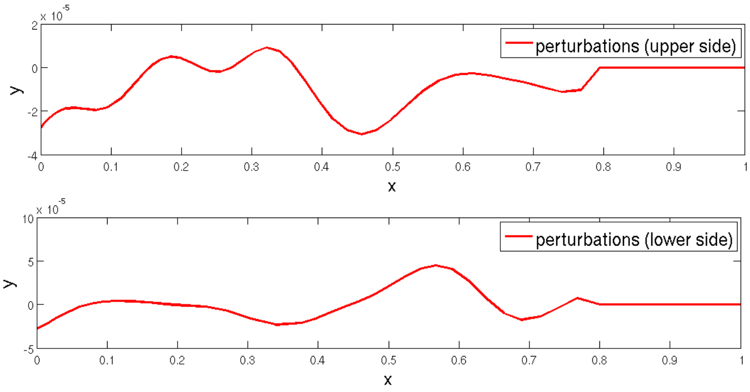


Figure 8.25: One realization of the random field ψ on the reduced domain Γ_{red}: perturbations on the upper side of the profile (above) and on the lower side (below).

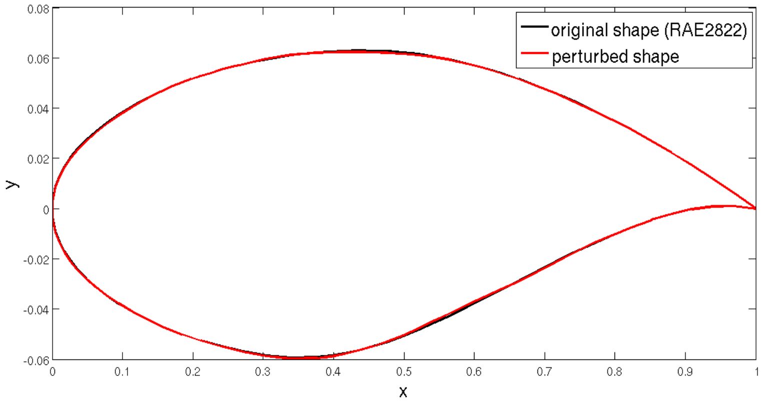


Figure 8.26: Resulting perturbed geometry (in red) on the reduced domain Γ_{red} compared with the original shape (in black).

Representing the random field for the numerical treatment ψ in a finite number of independent random variables using the Karhunen-Loève expansion (cf. section 5.2), one has to solve the eigenvalue problem

$$\int_{\Gamma_{red}} cov\,(x,y)\, z_i^{KL}\,(y) = \varsigma_i^{KL} z_i^{KL}\,(x)\ .$$

In the underlying two-dimensional test case, the discretization of the profile leads to a matrix of size (129×129) and (192×192) in the Navier-Stokes case, so the eigenvalues and eigenvectors can be computed by common methods. The distribution of the eigenvalues of the given random field ψ is shown in the next figure 8.27. As stated before, the eigenvalues exponentially converge towards zero. For the numerical results presented below, we have considered only the first four eigenvalues and eigenvectors to represent the random field ψ of perturbations. The corresponding eigenvectors are shown in figure 8.28.

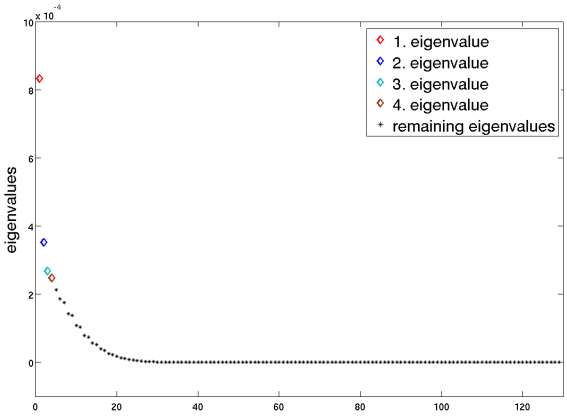


Figure 8.27: Distribution of the eigenvalues of the given random field ψ of perturbations.

Using the truncated Karhunen-Loève representation, the Gaussian random field is approximated by

$$\psi_4(x,\zeta) = \sum_{i=1}^{4} \sqrt{\varsigma_i^{KL}} z_i^{KL} X_i(\zeta)\,.$$

The random variables X_i are uncorrelated normally distributed, i.e. $X_i \sim \mathcal{N}(0,1)$, and therefore independent.

Applying the Karhunen-Loève representation, problem (8.44 - 8.46) can be approximated by

$$\min_{y(\psi_4(\cdot,\zeta)),p} \quad \mathbb{E}(f(y(\psi_4(\cdot,\zeta)),p,\psi_4(\cdot,\zeta))) \tag{8.47}$$

$$\text{s.t.} \quad c(y(\psi_4(\cdot,\zeta)),p,\psi_4(\cdot,\zeta)) = 0\,, \qquad \forall \zeta \in \mathcal{O} \tag{8.48}$$

$$h(y(\psi_4(\cdot,\zeta)),p,\psi_4(\cdot,\zeta)) \geq 0\,, \qquad \forall \zeta \in \mathcal{O}. \tag{8.49}$$

In each iteration of the optimization procedure, the evaluation of the four-dimensional integral (8.47) is required. In order to reduce the computational effort, we investigate the influence of the

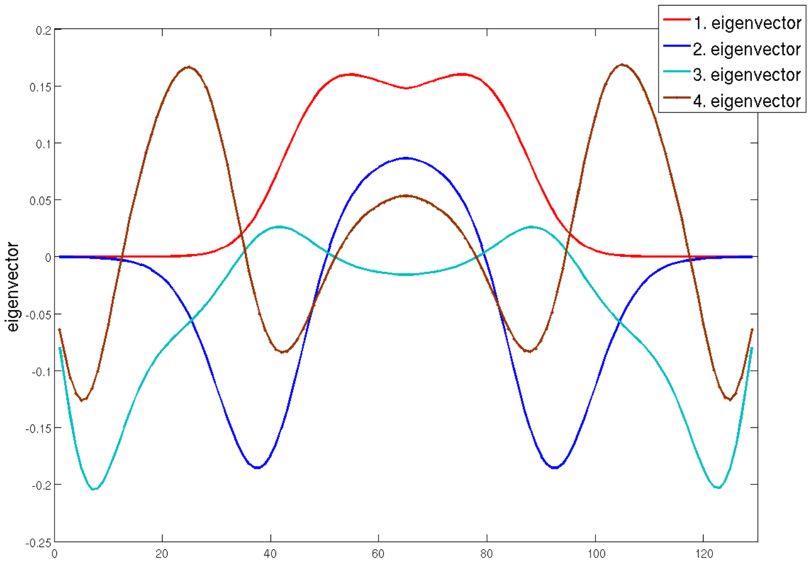


Figure 8.28: First four eigenvectors of the given random field ψ.

individual eigenvectors to reject those eigenvectors from the reduced basis, which have no impact on the target functional. Since the following results are problem dependent, we will now distinguish between the Euler and Navier-Stokes case.

8.4.1 Euler flow

Following the ideas of a goal-oriented choice of the Karhunen-Loève basis discussed in section 5.3, the influence of the first four eigenvectors of the Karhunen-Loève basis on the target functional *f* is investigated. As figure 8.29 shows, the third eigenvector has no impact on the objective function, hence it can be rejected from the Karhunen-Loève basis. If we take a look at the sensitivity of the drag performance with respect to changes of the shape depicted in figure 8.30, we can observe that changes of the front part of the profile, which corresponds to the middle part of the figure due to the numbering of the surface points, have a large impact on the drag. Further, perturbations of the back part except for the trailing edge hardly effect the drag performance. Since the third eigenvector of the Karhunen-Loève basis shown in figure 8.28 induces large perturbations of the back part and only very small perturbations of the leading edge, the sensitivity analysis approves the discussed observations. The introduced indicator, which measures the influence of the individual eigenvectors on the target functional, also captures this behavior, cf. table 8.3.

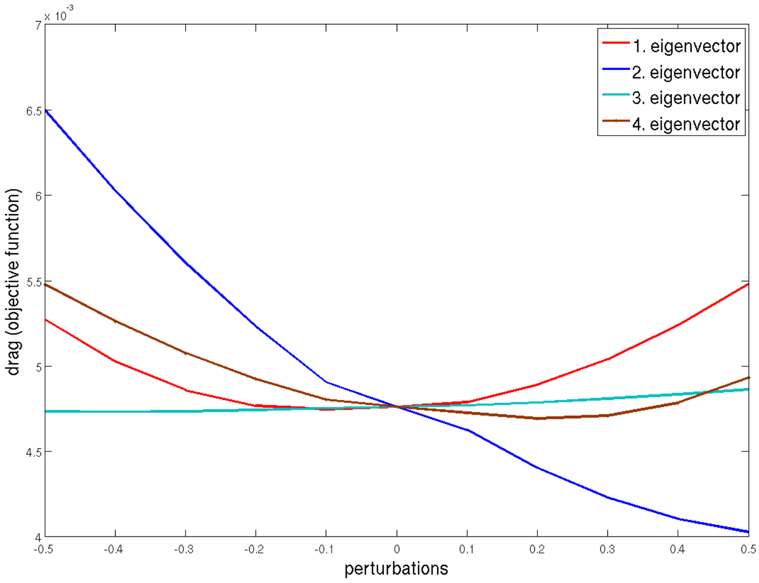


Figure 8.29: Influence of the first four eigenvectors on the target functional (Euler flow).

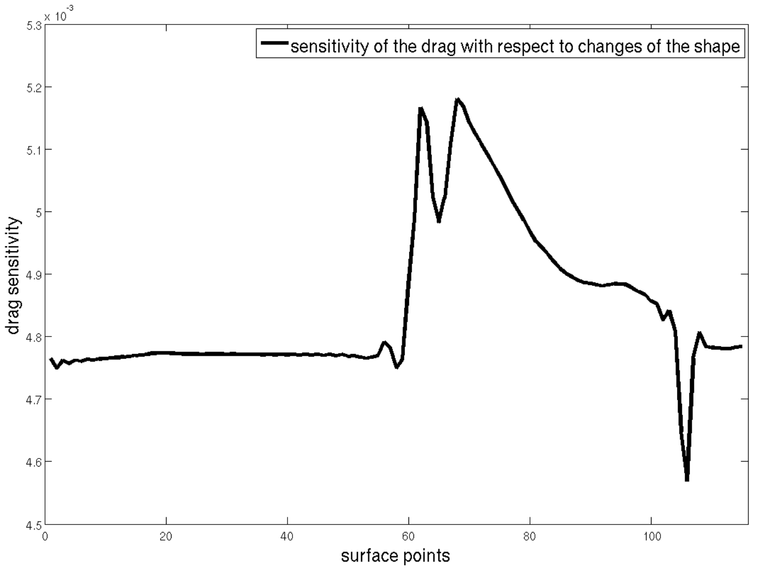


Figure 8.30: Sensitivity of the drag with respect to the shape (Euler flow).

	1. eigenvector	2. eigenvector	3. eigenvector	4. eigenvector
indicator	$2.256 \cdot 10^{-4}$	$-1.421 \cdot 10^{-3}$	$7.722 \cdot 10^{-6}$	$-4.745 \cdot 10^{-4}$

Table 8.3: Indicator of the first four eigenvectors measuring the influence on the target functional (Euler flow).

Consequently, the mean value is given by

$$\mathbb{E}\left(f(p, \psi_4^{reduced}(\cdot,\zeta))\right) = \int_{\mathcal{O}}\int_{\mathcal{O}}\int_{\mathcal{O}} (f(p, \sum_{\substack{i=1\\ i\neq 3}}^{4} \sqrt{\varsigma_i^{KL}} z_i^{KL} X_i(\zeta))) \, \mathrm{d}\mathcal{P}(\zeta)\, \mathrm{d}\mathcal{P}(\zeta)\, \mathrm{d}\mathcal{P}(\zeta) \ . \qquad (8.50)$$

If one approximates the expected value (8.50) using a full tensor grid interpolation (6.1), 729 grid points will be needed to reach the error tolerance of $3 \cdot 10^{-4}$, which is approximately the accuracy of the drag coefficient given by the flow solver. The resulting full grid is shown in figure 8.31. Since we want to compare the efficiency of the different introduced methods, we have chosen multilinear hierarchical basis functions as ansatz functions for the sparse and full tensor grid.

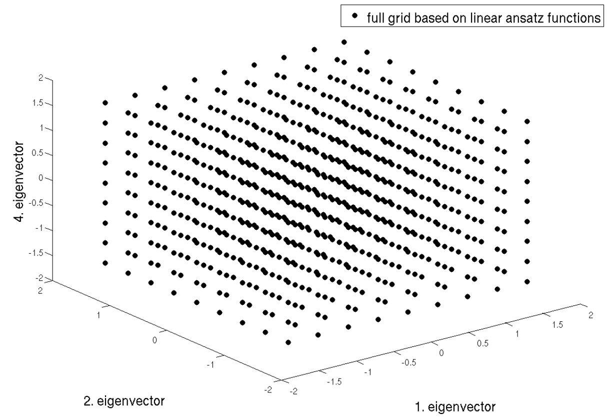


Figure 8.31: Full tensor grid based on linear ansatz functions with 729 grid points (Euler flow).

The sparse grid method can reduce the computational effort by a factor of 10 maintaining the same approximation quality. The corresponding grid is depicted in figure 8.32.

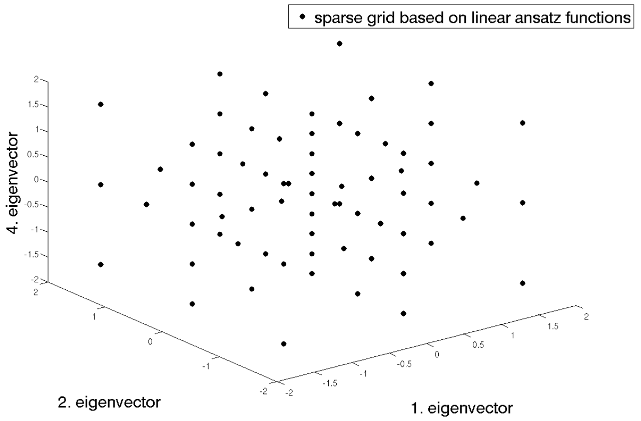


Figure 8.32: Sparse grid based on linear ansatz functions with 69 grid points (Euler flow).

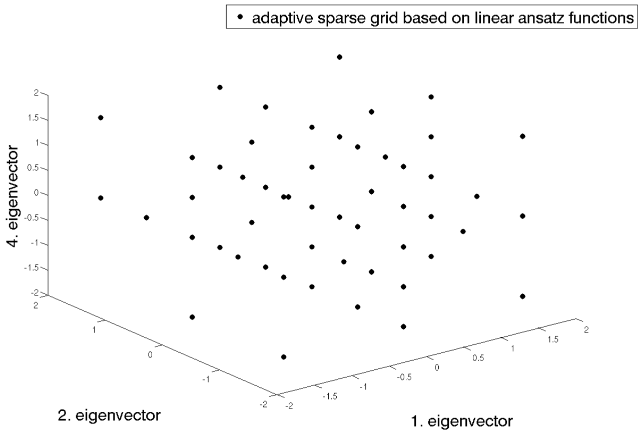


Figure 8.33: Adaptive sparse grid based on linear ansatz functions with 52 grid points (Euler flow).

As figure 8.33 shows, the number of grid points can be further reduced from 69 grid points to 52 grid points, i.e. a reduction of 17 flow simulations in each iteration is reached using a local refinement strategy as described in algorithm 6.3. Since the optimization requires the evaluation of the mean value in each iteration, this reduction by factor 15 compared to the full grid takes place in each step of the optimization algorithm and hence significantly speeds up the whole algorithm. The construction of the adaptive sparse grid although needs some additional function evaluations in order to compute the adaptivity indicator, but this amount of computational effort occurs outside the optimization loop, i.e. these costs are negligible.

Beside the local refinement strategy, a dimension adaptive sparse grid was introduced in section 6.4.1. The main advantage of this approach lies in the possibility to choose the underlying quadrature formulas problem dependent. As discussed before, Gauß-Hermite formulas are used, since the weighting function of the polynomials corresponds to the density function of the random variables X_1, X_2, X_4. Due to the higher accuracy of Gauß-Hermite quadrature formulas, an error tolerance of 10^{-5} is required in the following. The computed full tensor grid depicted in figure 8.34 consists of 343 grid points.

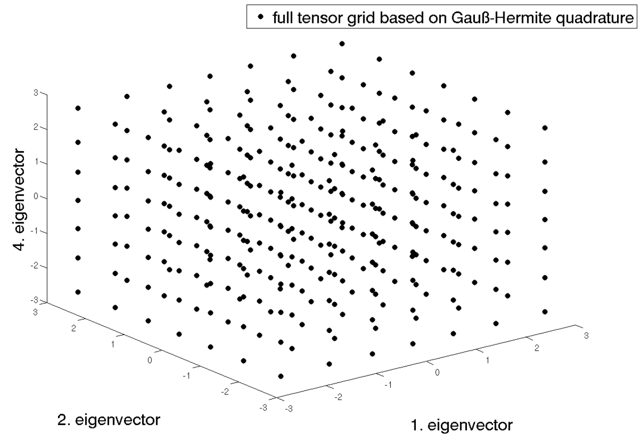


Figure 8.34: Full tensor grid based on Gauß-Hermite quadrature with 343 grid points (Euler flow).

A usual sparse grid based on equation (6.5) can reduce the number of grid points from 343 to 37, again almost a factor of 10 is achieved, cf. figure 8.35.

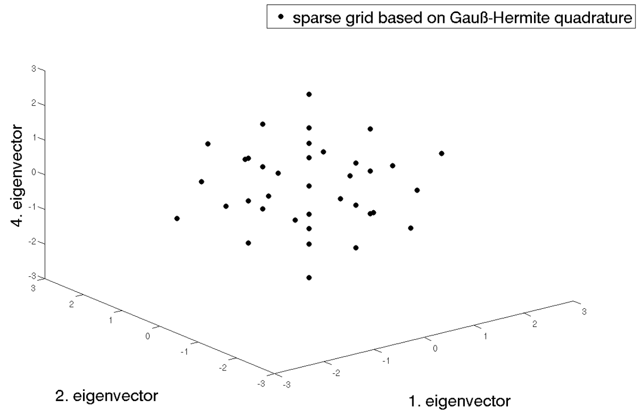


Figure 8.35: Sparse grid based on Gauß-Hermite quadrature with 37 grid points (Euler flow).

The dimension adaptive strategy results in a grid consisting of 21 points shown in figure 8.36.

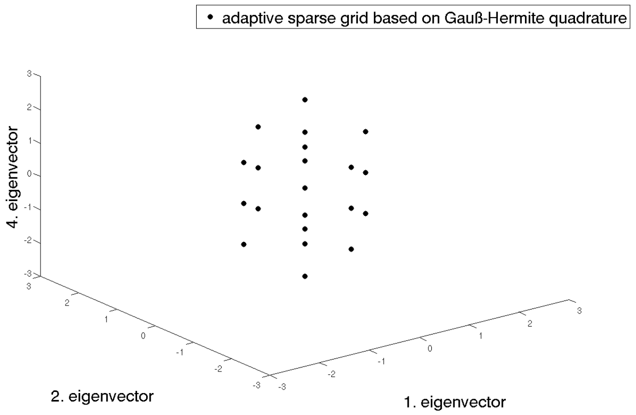


Figure 8.36: Adaptive sparse grid based on Gauß-Hermite quadrature with 21 grid points (Euler flow).

The comparison of the two refinement strategies shows that the use of problem dependent quadrature formulas can significantly reduce the size of the grid and increases the approximation quality at the same time. Hence, the objective function in the semi-infinite formulation (8.47 - 8.49) is approximated by the dimension adaptive grid consisting of 21 grid points shown in figure 8.36.

Remark 8.4.1. The construction of full tensor grids and sparse grids is problem independent, i.e. the discretization is based on the approximation of the input parameters. In our application, the domain of the input random variables and their distribution determine the resulting grids. Since these input quantities remain constant during the optimization, the grids used for the approximation of the mean value need not be modified in each iteration. On the other hand, adaptive strategies try to use additional information of the underlying function to reduce the number of discretization points. Hence, the adaptive grids are problem dependent and may change during the optimization, since the underlying function differs from the original function due to changes of the optimization variables. In general, a trust region approach has to be considered to update the grids during the optimization. In the aerodynamic shape optimization framework, the changes of the geometry are very small, so that the main characteristics with respect to the input uncertainties are not affected by the optimization. Numerical tests have shown that the initial sparse grids constructed by an adaptive strategy need not be updated and the approximation quality can be conserved over the whole optimization procedure. But one should be aware of the dependency of adaptive approaches on the underlying function and check the approximation quality e.g. by trust region methods.

The discretization of the probability space to approximate the mean value is further applied to the semi-infinite inequality constraint (8.49) leading to

$$\min_{y_i,p} \textstyle\sum_{i=1}^{21} f(y_i, p, \psi_i^{disc}) \tag{8.51}$$

$$\text{s.t.} \quad c(y_i, p, \psi_i^{disc}) = 0\,, \qquad \forall i = 1, \dots, 21 \tag{8.52}$$

$$h(y_i, p, \psi_i^{disc}) \geq 0\,, \qquad \forall i = 1, \dots, 21, \tag{8.53}$$

where ψ_i^{disc} denotes the i-th realization of the random field $\psi_4^{reduced}$ defined by the adaptive sparse grid based on Gauß-Hermite quadrature. Since the treatment of the inequality constraint (8.49) is much more complicated than in the scalar-valued test cases, a discretization approach (cf. section 7.3.1) is chosen to solve (8.49) by a sequence of finite subproblems. The infinite support of the random variables describing the random field of perturbations is approximated by a finite one resulting from the sparse grid, i.e. using the adaptive sparse grid depicted in figure 8.36, realizations of the normally distributed random variables X_1, X_2, X_4 lying in $[-1.225, 1.225] \times [-1.225, 1.225] \times [-2.652, 2.652]$ will be considered. The probability of this event is equal to 80%, approximately. This relatively small region of perturbations is assumed, as the deformation tool of the flow solver does not allow arbitrary large modifications of the grid. Further, large variations of the shape due to uncertainties in the manufacturing process or due to fatigue of material are not reasonable from the physical point of view. However, this assumption may be replaced by measurements, if at hand. Applying the discretization approach to semi-infinite problems, the convergence of the finite subproblems to the original semi-infinite problem is a crucial task, cf. section 7.3.1. As a first step, the influence of the individual eigenvectors on the lift constraints is investigated, shown in figure 8.37.

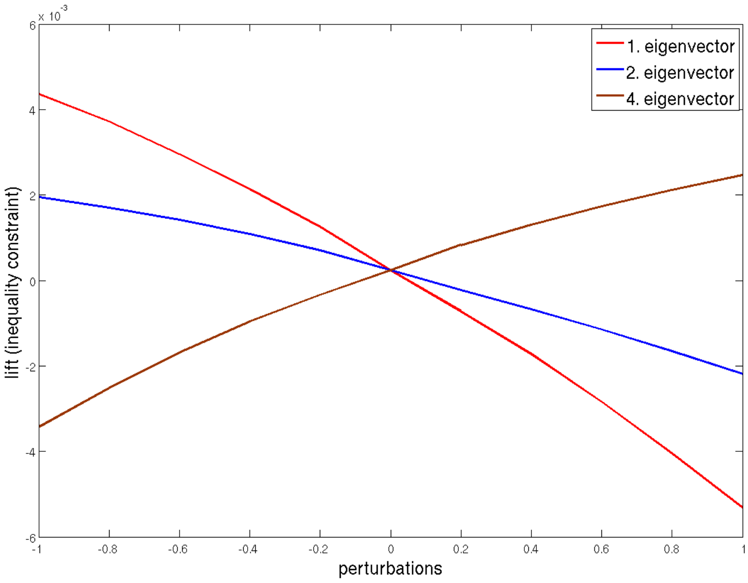


Figure 8.37: Influence of the eigenvectors of the reduced KL-basis on the lift constraint (Euler flow).

We can observe that the lift coefficient is strictly monotone and almost linear with respect to the perturbations. This property of the lift coefficient greatly simplifies the proper treatment of the semi-infinite constraint. The solution of the lower level problem

$$\min_{(X_1,X_2,X_4)\in\mathcal{D}} h(y(X_1,X_2,X_4),p,\sum_{\substack{i=1\\ i\neq 3}}^{4}\sqrt{\varsigma_i^{KL}}z_i^{KL}X_i)$$

with $\mathcal{D} = [-1.225, 1.225] \times [-1.225, 1.225] \times [-2.652, 2.652]$ can be therefore determined by checking the function values of the vertices. Hence, a discretization including the vertices of the domain $[-1.225, 1.225] \times [-1.225, 1.225] \times [-2.652, 2.652]$ leads to a finite subproblem, which is equivalent to the semi-infinite formulation. In more general cases considering arbitrary nonlinear constraints, the approximation of the semi-infinite constraint may require much more effort. Using the introduced sparse grid techniques, an adaption criterion specifically tailored to the approximation of the objective function and the lift constraint needs to be developed. A further possibility to achieve convergence of the finite subproblems to the semi-infinite problem is to additionally apply a multigrid method, which successively refines the discretization during the optimization.

In the following, the results of the robust optimization is compared to the single-setpoint optimization. The robust optimization problem (8.51 - 8.53) is solved by the generalized version of the one-shot method, cf. algorithm 7.6 and the adaptive sparse grid based on Gauß-Hermite polynomials shown in figure 8.36 is used for the discretization of the probability space. The drag and lift performance of the 21 perturbed geometries is shown in figure 8.38 and 8.39. The dashed line in figure 8.38 indicates the mean value of the drag.

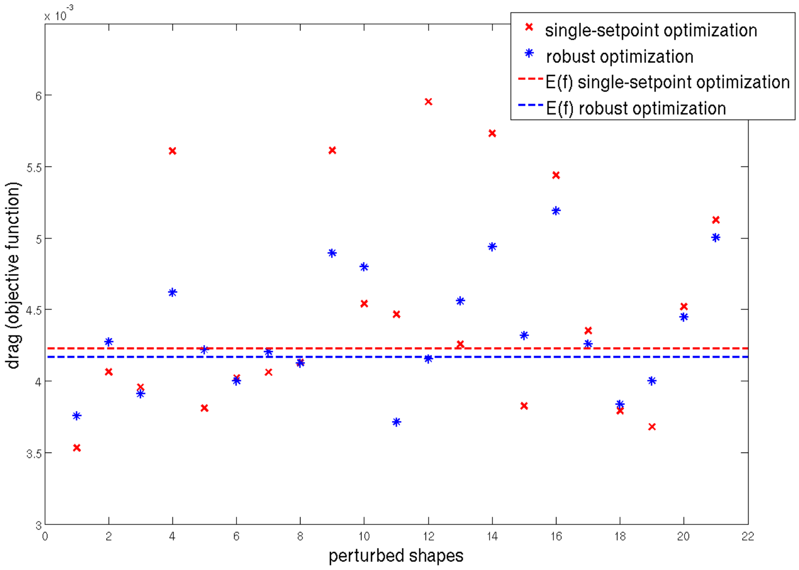


Figure 8.38: Drag performance of the 21 perturbed shapes (Euler flow).

The robust optimization gives a little bit higher drag at the nominal point, which is the first discretization point, than the single-setpoint optimization. But over the whole range of variations, the robust optimization shows a similar drag performance and improves the mean value of the target functional by 0.5 drag counts compared to the single-setpoint optimization. At the same time, the robust shape leads to a better lift performance over the whole range of perturbations, whereas the single-setpoint optimization is infeasible in more than half of the considered grid points. Summarizing, it can be said that the robust optimization leads to a better lift to drag ratio than the single-setpoint optimization and the resulting profile is more robust against small perturbations of the shape itself.

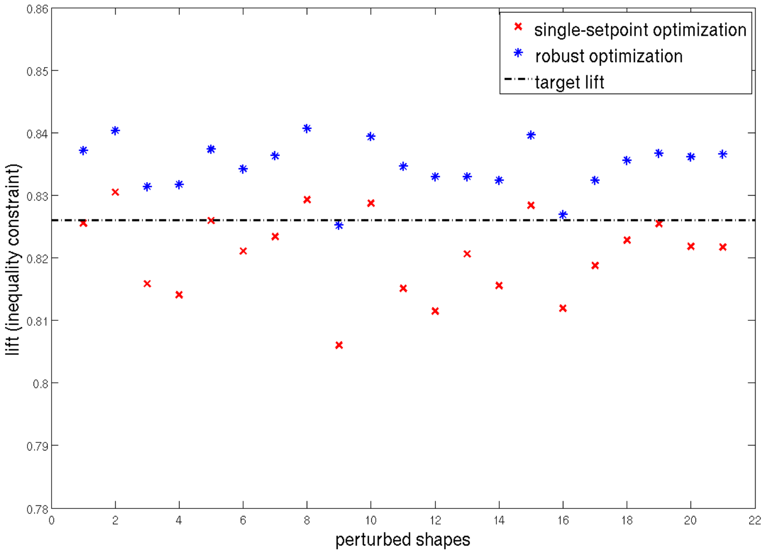


Figure 8.39: Lift performance of the 21 perturbed shapes (Euler flow).

In figure 8.40, the pressure distribution of the robust and single-setpoint optimized shape at the nominal point (without considering any perturbations of the shape) is compared.

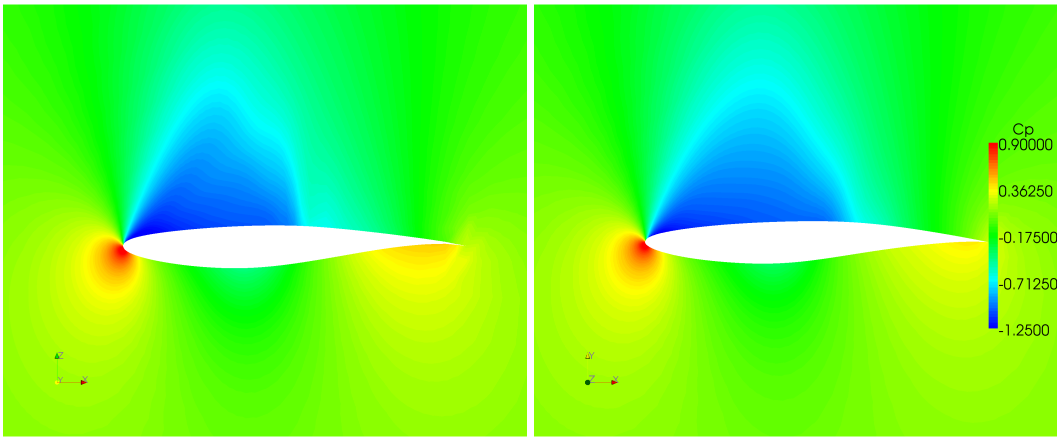


Figure 8.40: Pressure distribution around the airfoil at the nominal point of the semi-infinite (left) and the single-setpoint (right) optimization (Euler flow).

The shock of the single-setpoint optimized shape at the nominal point is smoothed out resulting in a drag coefficient $C_D = 3.531 \cdot 10^{-3}$. The stronger shock of the robust shape on the upper surface (cf. 8.40) leads to a little bit higher drag coefficient $C_D = 3.754 \cdot 10^{-3}$ than the single-setpoint optimization. The next figure 8.41 compares the pressure distribution around the airfoil at discretization point 9. Here, the single-setpoint optimized shape cannot reach the target lift, cf. figure 8.39, whereas the semi-infinite formulation is feasible. Further, the single-setpoint optimization

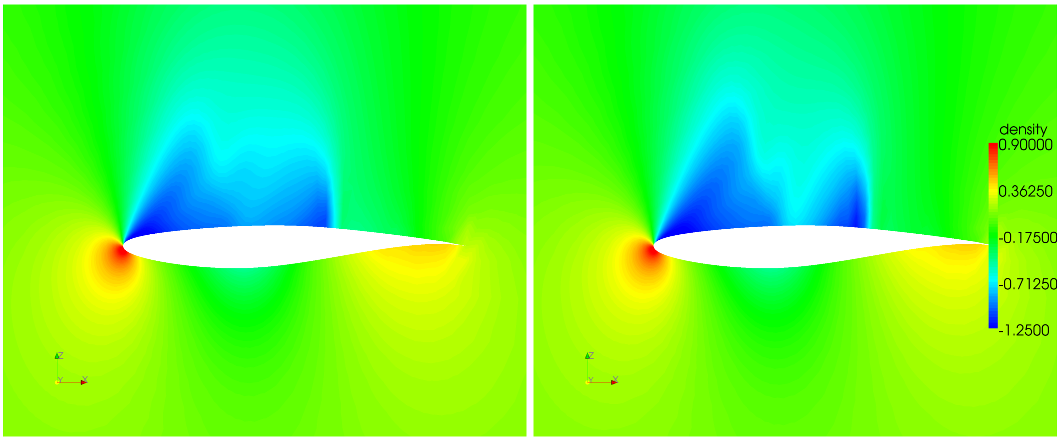


Figure 8.41: Pressure distribution around the airfoil at discretization point 9 of the semi-infinite (left) and the single-setpoint (right) optimization (Euler flow).

leads to a drag coefficient $C_D = 5.612 \cdot 10^{-3}$, which can be reduced by the robust optimization to $C_D = 4.896 \cdot 10^{-3}$.

Last, we will compare the different resulting shapes in figure 8.42.

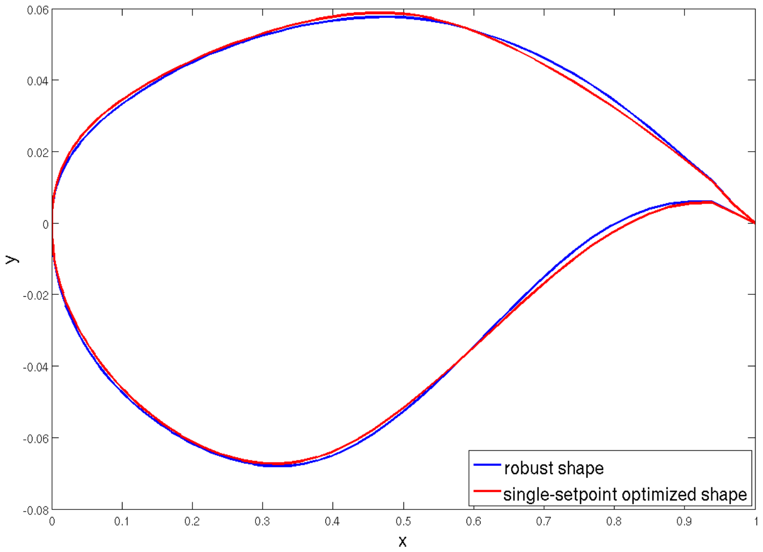


Figure 8.42: Comparison of the single-setpoint optimized shape and the robust shape with respect to geometrical uncertainties (Euler flow).

Although we have assumed only small perturbations of the shape itself (cf. figure 8.26), the difference between the robust shape and the single-setpoint is well recognizable.

The test case demonstrates how important robust design in the aerodynamic framework is. The numerical results show that even small deviations from the planned geometry have a significant effect on the drag and lift coefficient. Since variations of the planned shape due to manufacturing tolerances or fatigue of material cannot be avoided, geometrical uncertainties have to be taken into account in the aerodynamic design optimization problem to ensure a robust solution. The results of the semi-infinite optimization show a better lift to drag ratio over the whole range of considered perturbations than the single-setpoint optimization.

8.4.2 Navier-Stokes flow

In order to determine the goal-oriented Karhunen-Loève basis, we first investigate the influence of the individual eigenvector on the target functional shown in figure 8.43. The sensitivity analysis depicted in figure 8.44 shows a similar result to the Euler case. Perturbations of the leading edge have a large impact on the drag, whereas changes of the back part of the profile do not effect the drag performance. Due to this reason, the third eigenvector of the Karhunen-Loève basis can be rejected from the reduced basis. As table 8.4 shows, the introduced indicator properly reflects the influence of the individual eigenvectors on the target functional, so that the reduced basis can be chosen automatically.

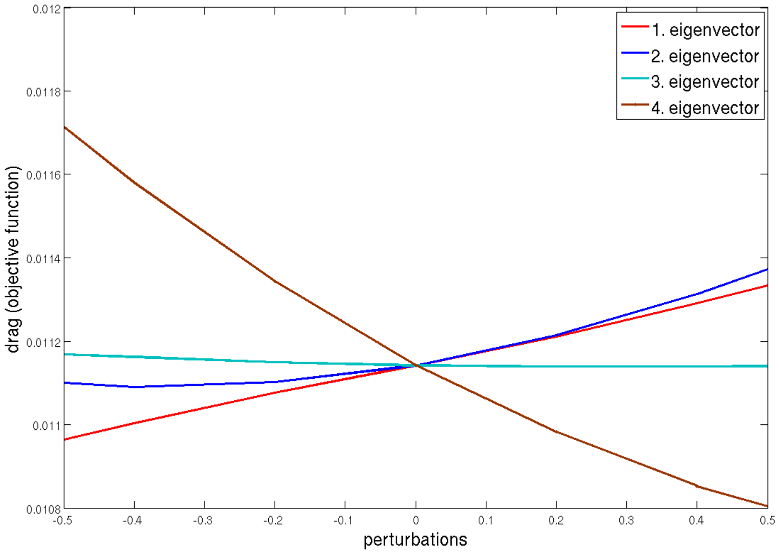


Figure 8.43: Influence of the first four eigenvectors on the target functional (Navier-Stokes flow).

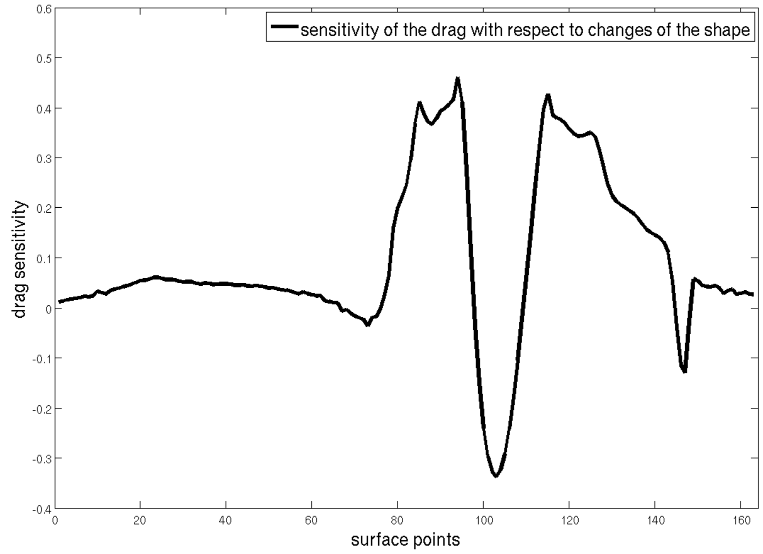


Figure 8.44: Sensitivity of the drag with respect to the shape (Navier-Stokes flow).

	1. eigenvector	2. eigenvector	3. eigenvector	4. eigenvector
indicator	$1.518 \cdot 10^{-4}$	$1.176 \cdot 10^{-4}$	$-1.313 \cdot 10^{-5}$	$-4.009 \cdot 10^{-4}$

Table 8.4: Indicator of the first four eigenvectors measuring the influence on the target functional (Navier-Stokes flow).

Hence, the mean value is approximated using the first, second and fourth basis vector

$$\mathbb{E}\left(f(p, \psi_4^{reduced}(\cdot,\zeta))\right) = \int_O \int_O \int_O (f(p, \sum_{\substack{i=1\\ i\neq 3}}^{4} \sqrt{\varsigma_i^{KL}} z_i^{KL} X_i(\zeta))) \, d\mathcal{P}(\zeta) \, d\mathcal{P}(\zeta) \, d\mathcal{P}(\zeta) \ .$$

If the same error tolerance $3 \cdot 10^{-4}$ as in the previous test case is required, a full grid of 4913 grid points based on multilinear ansatz functions has to be used in order to compute the objective function, cf. figure 8.45. A reduction factor of 28 can be achieved by a sparse grid approach, which is further improved by a local refinement strategy. The resulting adaptive grid consists of 99 grid points, which leads to a reduction factor of 50. The computed grids are illustrated in figure 8.46.

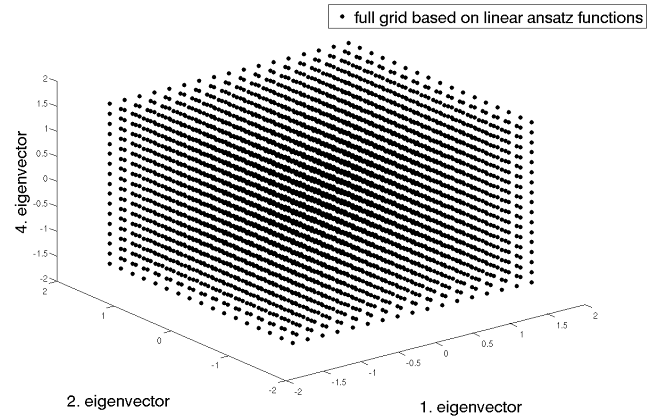


Figure 8.45: Full tensor grid based on linear ansatz functions with 4913 grid points (Navier-Stokes flow).

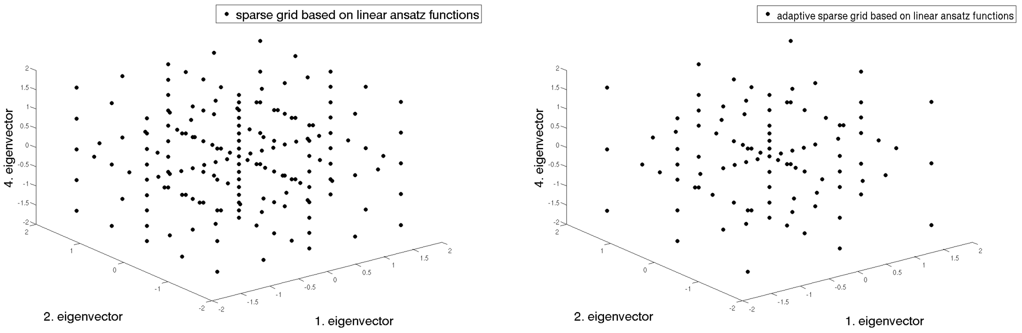


Figure 8.46: Sparse grid based on linear ansatz functions with 177 grid points and locally refined sparse grid with 99 grid points (Navier-Stokes flow).

Since the computational effort in the Navier-Stokes case is much higher than in the Euler test case, the number of grid points need to be further reduced in order to make a robust optimization possible. The use of Gauß-Hermite quadrature formulas results in the same full grid and sparse grid as in the Euler test case fulfilling the error tolerance $3 \cdot 10^{-4}$, cf. figure 8.34 and figure 8.35. Applying the dimension adaptive approach, an adaptive sparse grid consisting of 15 grid points shown in figure 8.48 is obtained.

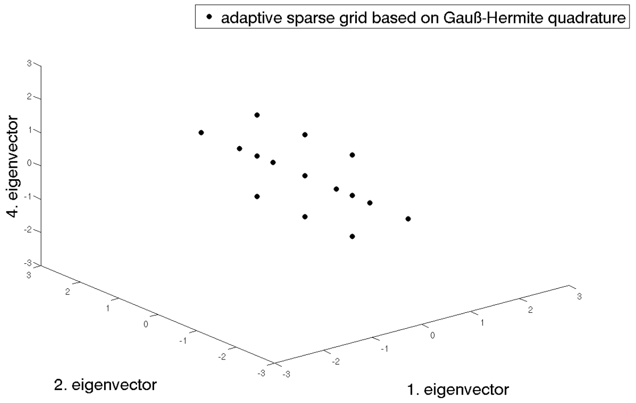


Figure 8.47: Dimension adaptive sparse grid with 15 grid points (Navier-Stokes flow).

The linearity of the drag depending on the first eigenvector (cf. figure 8.43) is recognized by the dimension adaptive algorithm, so that this dimension is not refined further. Since the linear behavior might change during the optimization, we add two points ensuring that the first eigenvector is taken into account during the optimization, see figure 8.48. Using the adaptive sparse grid consisting of

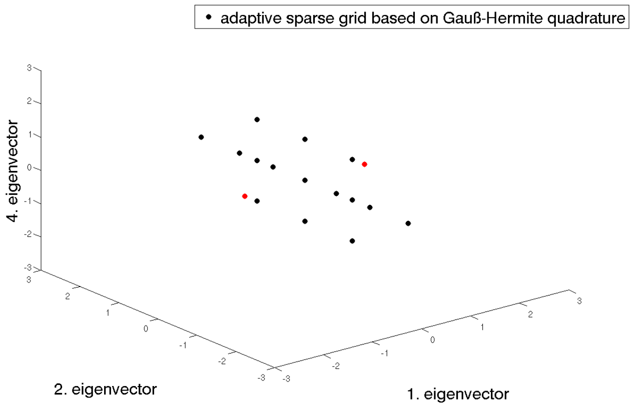


Figure 8.48: Dimension adaptive sparse grid with 17 grid points (Navier-Stokes flow).

17 grid points, the robust optimization problem is given by

$$\min_{y_i,p} \sum_{i=1}^{17} f(y_i, p, \psi_i^{disc}) \tag{8.54}$$

$$\text{s.t.} \quad c(y_i, p, \psi_i^{disc}) = 0\,, \qquad \forall i = 1, \dots, 17 \tag{8.55}$$

$$h(y_i, p, \psi_i^{disc}) \geq 0\,, \qquad \forall i = 1, \dots, 17. \tag{8.56}$$

The semi-infinite constraint is again solved by a discretization method, which exactly solves the original problem due to the monotonicity of the lift constraint with respect to the reduced basis.

Further, problem (8.54 - 8.56) is solved by the generalized one-shot algorithm 7.6. The results of the robust optimization compared to the single-setpoint solution are illustrated in figure 8.49 and figure 8.50.

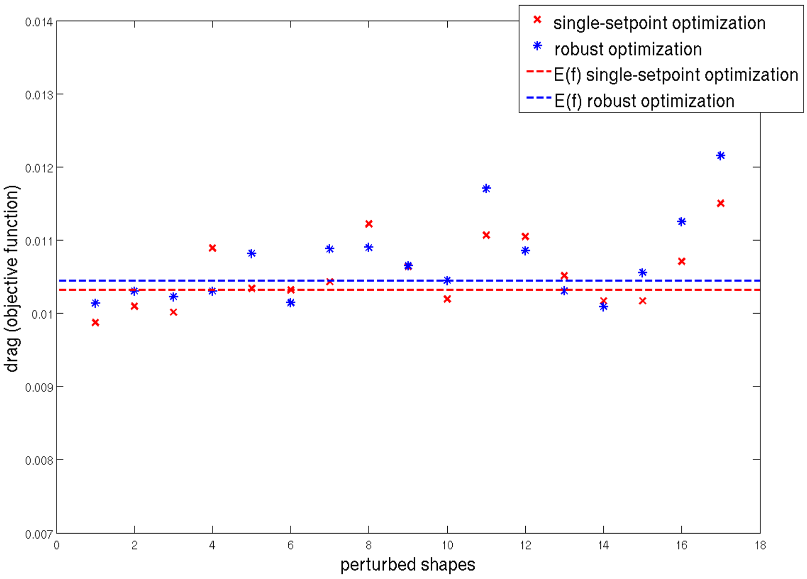


Figure 8.49: Drag performance of the 17 perturbed geometries (Navier-Stokes flow).

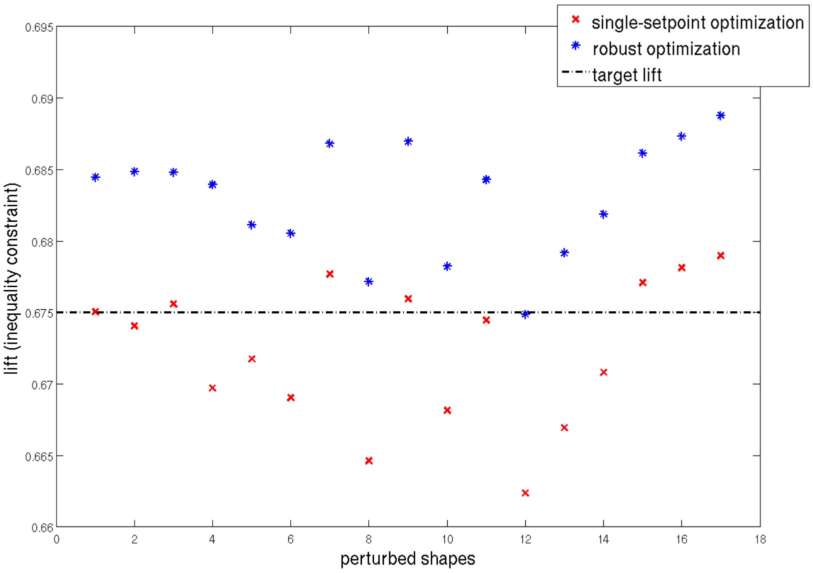


Figure 8.50: Lift performance of the 17 perturbed geometries (Navier-Stokes flow).

The mean value of the drag in the robust case is a little bit higher than the mean value of the single-setpoint optimization (1.3 drag counts), which can be attributed to the lift constraints. The robust optimization is feasible over the whole range of perturbations as required, whereas the single-setpoint optimization cannot reach the target lift in more than 60% of the realizations. Hence, we can state that the semi-infinite optimization again leads to a better lift to drag ratio as in the Euler case.

Figure 8.51 compares the pressure distribution around the airfoil of the semi-infinite and single-setpoint solution at the nominal point. The semi-infinite formulation leads to two small shocks on the upper surface of the airfoil inducing the higher drag than the single-setpoint optimized shape. Moreover, we will compare the pressure distribution at grid point 12, where the single-setpoint solution cannot reach the target lift and at the same time shows a higher drag than the robust optimization. The position of the first shock wave of the semi-infinite optimized profile differs from the single-

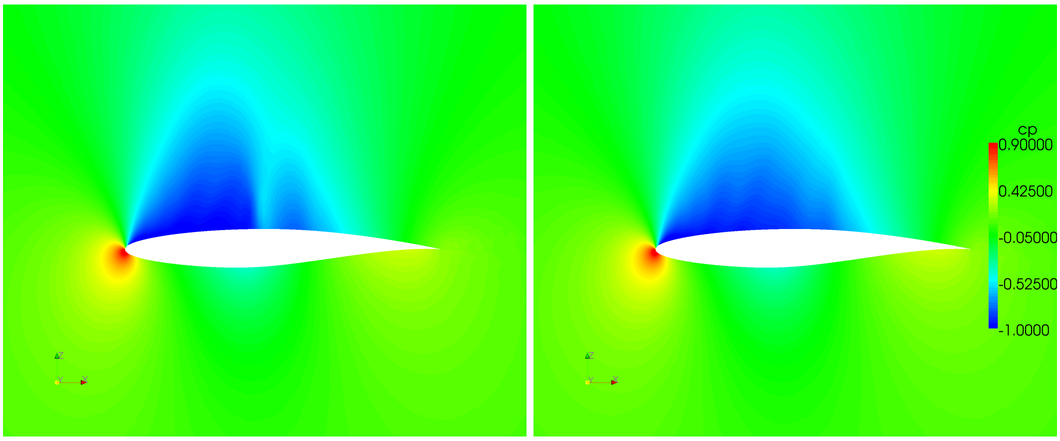


Figure 8.51: Pressure distribution around the airfoil at the nominal point of the semi-infinite (left) and the single-setpoint (right) optimization (Navier-Stokes flow).

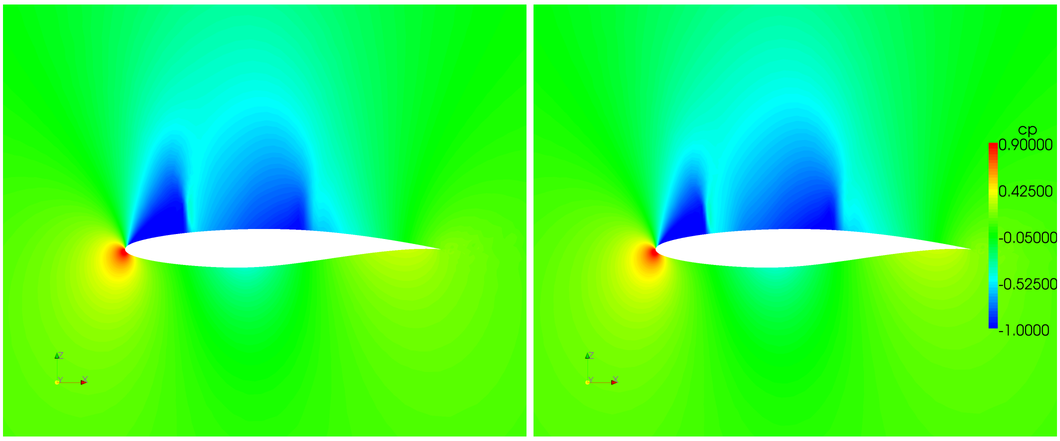


Figure 8.52: Pressure distribution around the airfoil at discretization point 12 of the semi-infinite (left) and the single-setpoint (right) optimization (Navier-Stokes flow).

setpoint optimization leading to a reduction of the drag coefficient, cf. figure 8.52.

The resulting shapes are depicted in figure 8.53.

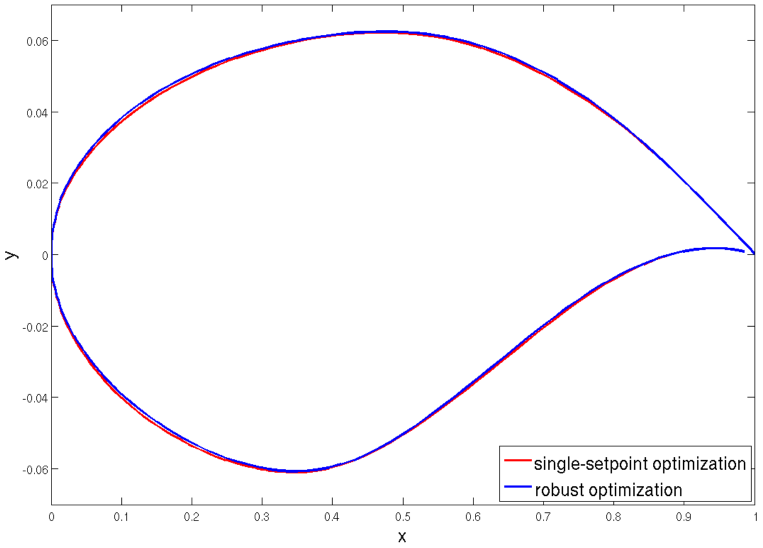


Figure 8.53: Comparison of the single-setpoint optimized shape and the robust shape with respect to geometrical uncertainties (Navier-Stokes flow)

The difference between the robust shape and the single-setpoint optimized shape is smaller than in the Euler case, indicating that the profile is more sensitive to changes of the shape.

8.5 Numerical study of the influence of geometrical uncertainties (test case SFB-401)

A non-intrusive polynomial chaos method is applied to a 3D test case in order to determine the influence of geometrical uncertainties on flow parameters. The unstructured grid of the 3D wing and a close-up are shown in figure 8.54. The space is discretized by 2506637 grid points, where the surface is described by 80903 points. The grid generated by Centaur consists of 1636589 tetraeders, 4363281 prisms and 170706 surface triangles, 5427 surface quadrilaterals.

Here, the transonic flow at Mach number $M = 0.8$ is modeled using the Reynolds-averaged Navier-Stokes equations. The geometrical uncertainties are assumed to occur only on the upper part of the wing, the perturbed region is depicted in figure 8.55. The perturbations are modeled as a Gaussian random field defined by the following second order statistics

$$\mathbb{E}\left(\psi\left(x,\zeta\right)\right) = 0\,, \qquad \forall x \in \Gamma, \tag{8.57}$$

$$cov\left(x,y\right) = (0.0016)^2 \cdot \exp\left(-\frac{\|x-y\|^2}{(0.06)^2}\right)\,, \qquad \forall x,y \in \Gamma. \tag{8.58}$$

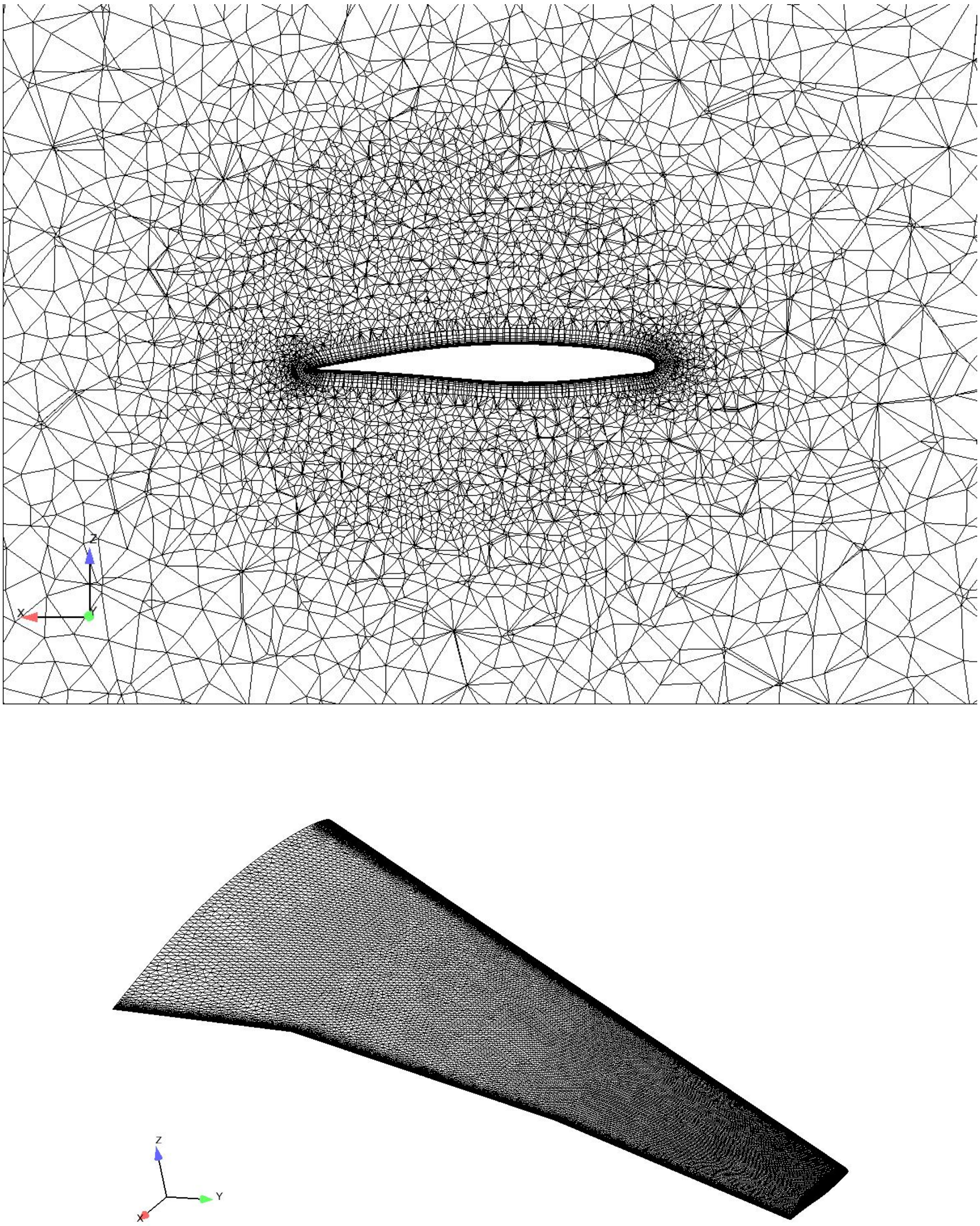


Figure 8.54: Grid for the SFB-401 test case: cut of the grid at y=0.5 (above) and surface of the wing (below).

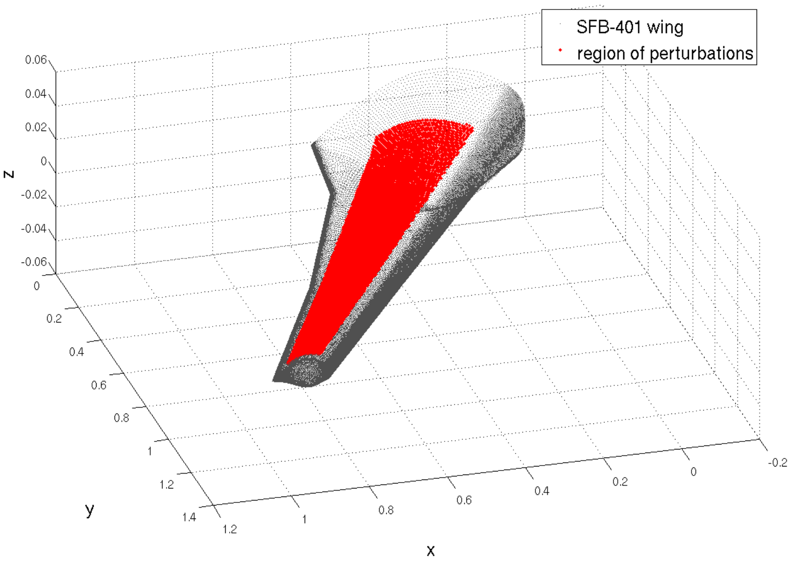


Figure 8.55: Perturbed region of the SFB-401 wing.

In order to take the curvature of the wing into account computing the norm in (8.58), the selected area is transformed into 2D approximating the distance by a polygon path on the surface. The projection is depicted in figure 8.56. Due to the problem size, the block version of the iterative eigensolver LOPEX (cf. algorithm 5.1) is used in order to solve the eigenvalue problem arising from the Karhunen-Loève expansion. The resulting eigenvalue distribution of the first 50 eigenvalues is shown in figure 8.57.

We consider the first 15 eigenvalues and eigenvectors to represent the random field, as an example the first, 8-th and 15-th eigenvectors and resulting perturbed shapes are depicted in figure 8.58. In order to approximate statistics of the flow solution depending on the considered perturbations, the drag, the lift and the pressure coefficient C_P are expanded into the first 16 multi-dimensional Hermite polynomials using a non-intrusive polynomial chaos method, cf. section 5.4. The scalar-valued coefficients C_D and C_L are approximated by

$$C_D(p, X_1, \dots, X_{15}) = \sum_{k=0}^{15} (\tilde{C_D})_k(p) \cdot \mathbf{H}_k \, (X_1, \dots, X_{15}) \tag{8.59}$$

and

$$C_L(p, X_1, \dots, X_{15}) = \sum_{k=0}^{15} (\tilde{C_L})_k(p) \cdot \mathbf{H}_k \, (X_1, \dots, X_{15}) \ . \tag{8.60}$$

Due to the complexity of the 3D test case, one flow simulation takes about 30 days performed on our computing server (8 cores AMD Opteron with 8 GB memory). The computational effort can be reduced to 15 days by a restart strategy, but the computational costs restrict the number of flow simulations to determine the polynomial chaos representation. Therefore, a sparse grid consisting

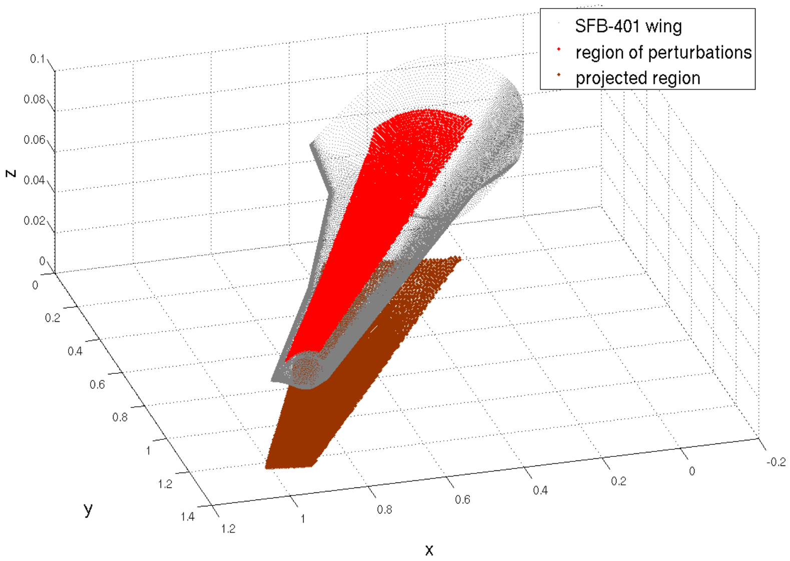


Figure 8.56: Projection in 2D of the region of perturbations (SFB-401 wing).

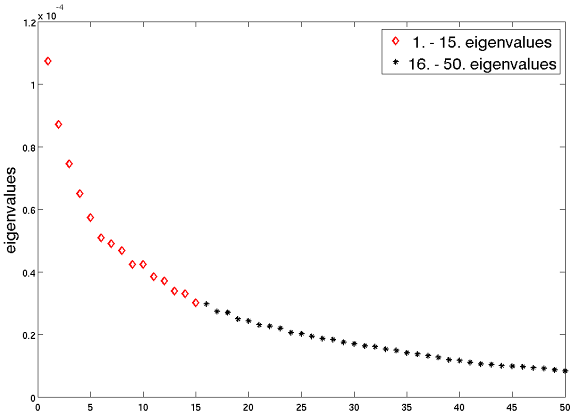


Figure 8.57: Distribution of the first 50 eigenvalues of the given random field ψ.

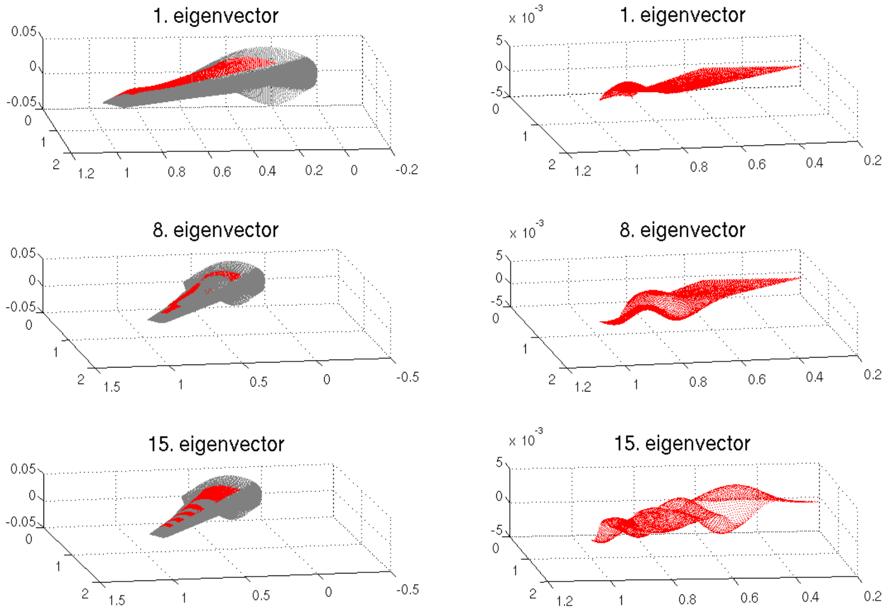


Figure 8.58: First, 8-th and 15-th eigenvectors and resulting perturbed shapes.

of 31 grid points is chosen in order to approximate the integrals

$$
\begin{aligned}
(\tilde{C}_D)_k(p) &= \frac{\langle C_D(p,\cdot), \mathbf{H}_k\rangle}{\langle \mathbf{H}_k^2\rangle} \\
&= \frac{1}{\langle \mathbf{H}_k^2\rangle} \int_{\mathbb{R}^{15}} C_D(p,\mathbf{y})\mathbf{H}_k(\mathbf{y})\varphi(\mathbf{y})\,\mathrm{d}\mathbf{y}\,, \quad k = 1,\dots,M
\end{aligned}
$$

and

$$
\begin{aligned}
(\tilde{C}_L)_k(p) &= \frac{\langle C_L(p,\cdot), \mathbf{H}_k\rangle}{\langle \mathbf{H}_k^2\rangle} \\
&= \frac{1}{\langle \mathbf{H}_k^2\rangle} \int_{\mathbb{R}^{15}} C_L(p,\mathbf{y})\mathbf{H}_k(\mathbf{y})\varphi(\mathbf{y})\,\mathrm{d}\mathbf{y}\,, \quad k = 1,\dots,M,
\end{aligned}
$$

respectively. In the next two figures 8.59 and 8.60, the drag and the lift of each perturbed shape and the corresponding mean values are illustrated.

As figure 8.59 indicates, the geometrical uncertainties have a large impact on the target functional. The standard deviation from the mean value is equal to 1.65 drag counts, and the mean value is 6.55 drag counts higher than the nominal, unperturbed geometry, which corresponds to the first grid point in figure 8.59. The mean value of the lift $\mathbb{E}(C_L) = (\tilde{C}_L)_0 = 0.2625$ is about 7 lift counts smaller than the lift at the nominal point. The standard deviation from the mean value is equal to 3.5 lift counts. The uncertainty quantification demonstrates that even small perturbations of the shape result in large variations of the lift to drag ratio.

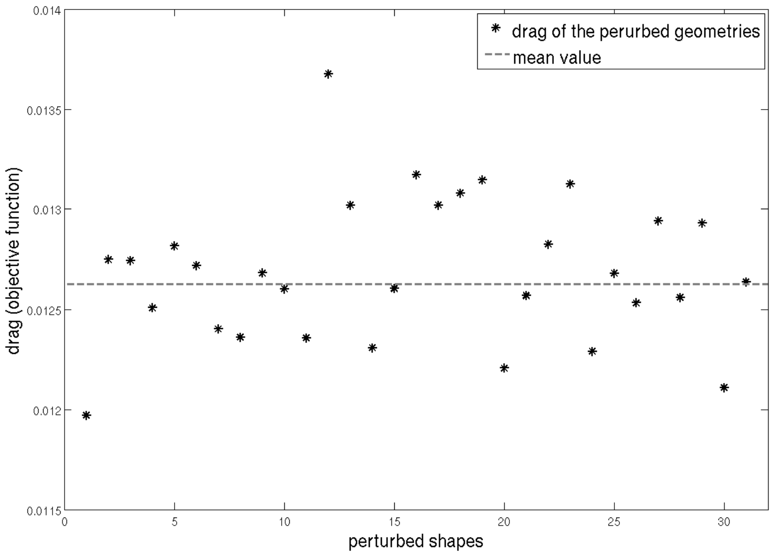


Figure 8.59: Drag performance and mean value of the perturbed 3D shapes (SFB-401 Navier-Stokes flow).

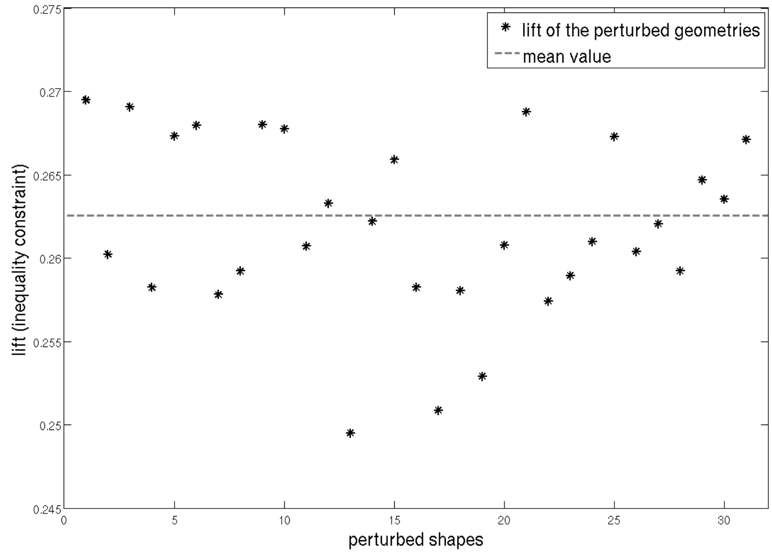


Figure 8.60: Lift performance and mean value of the perturbed 3D shapes (SFB-401 Navier-Stokes flow).

Beside the scalar-valued flow coefficients, the influence of the geometrical uncertainties on the pressure distribution C_P is quantized by a non-intrusive polynomial chaos method. The deterministic coefficients of the polynomial chaos approximation $\tilde{C_P}$ also depend on the surface points

$$\begin{aligned}(\tilde{C_P})_k(p,x) &= \frac{\langle C_P(p,x,\cdot),\mathbf{H}_k\rangle}{\langle \mathbf{H}_k^2\rangle} \\ &= \frac{1}{\langle \mathbf{H}_k^2\rangle}\int_{\mathbb{R}^{15}} C_P(p,x,\mathbf{y})\mathbf{H}_k(\mathbf{y})\varphi(\mathbf{y})\,\mathrm{d}\mathbf{y}\,, \quad k=1,\dots,M,\ x\in\Gamma.\end{aligned}$$

The comparison between the C_P distribution of the unperturbed geometry (cf. figure 8.61) and the C_P distribution of the mean value (cf. figure 8.62) shows that an additional shock on the upper side of the shape occurs due to the uncertainties of the geometry.

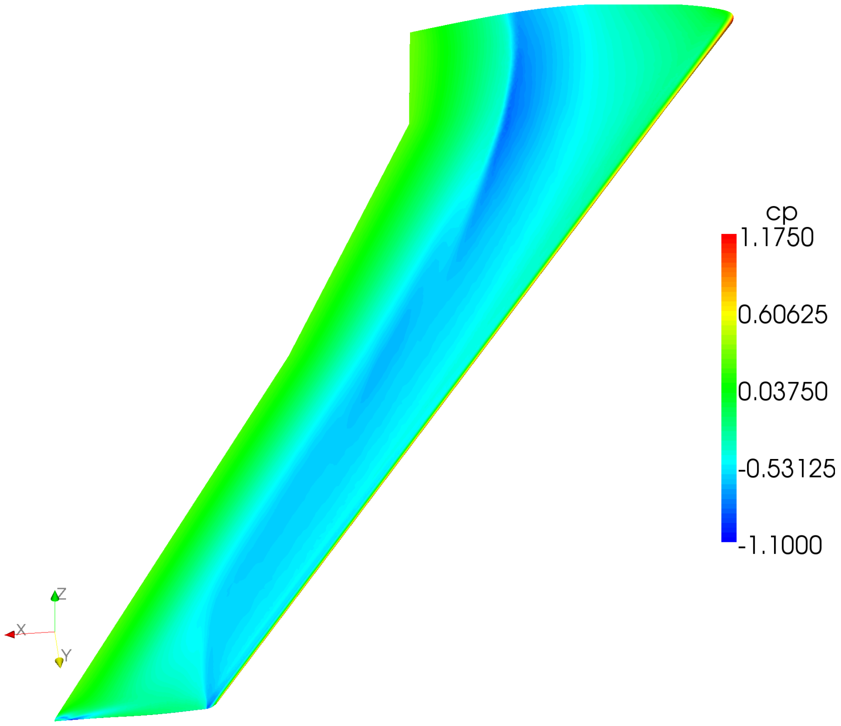


Figure 8.61: C_P distribution of the unperturbed geometry (SFB-401 Navier-Stokes flow).

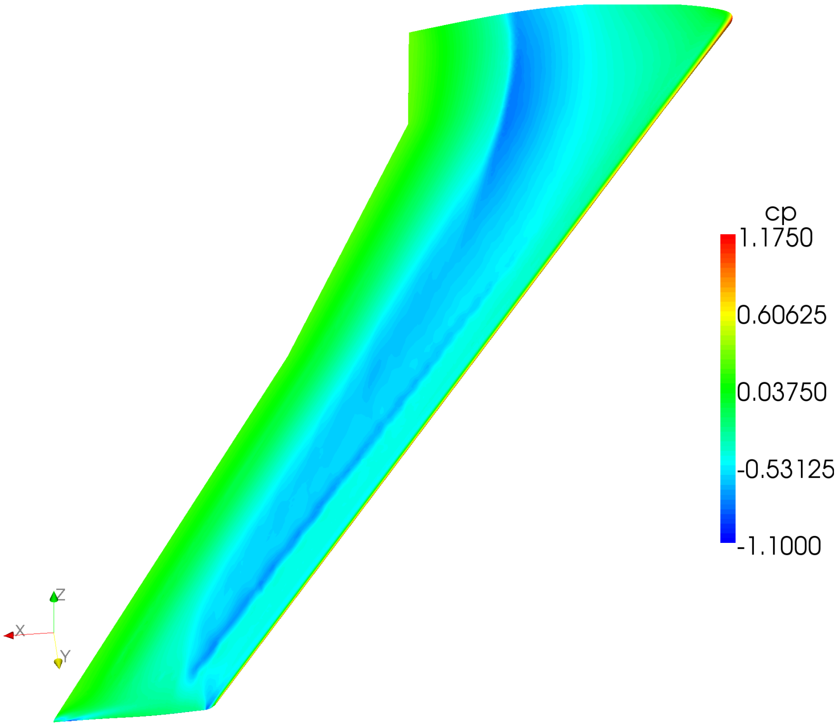


Figure 8.62: C_P distribution of the mean value (SFB-401 Navier-Stokes flow).

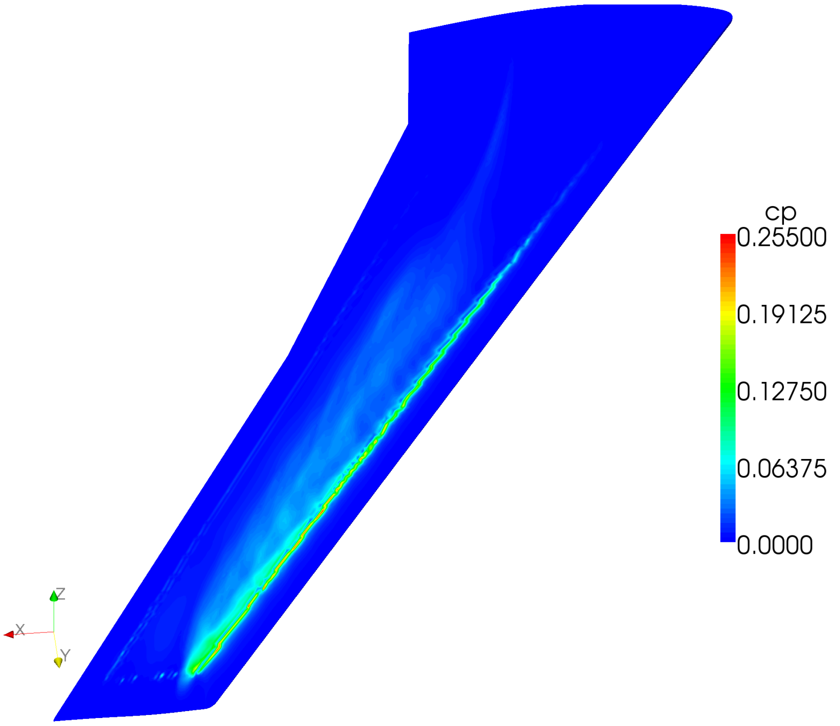


Figure 8.63: Variance of the C_P distribution (SFB-401 Navier-Stokes flow).

Figure 8.63 emphasizes the influence of the perturbations showing the variance of the C_P distribution. The variance and higher order statistics as well as probability distribution functions of the quantities of interest with respect to geometrical uncertainties can be efficiently approximated using the polynomial chaos model. After determination of the deterministic coefficients, the moments of the flow parameters can be directly computed from the polynomial chaos representation and no further flow simulations are required. Summarizing the results of the study, it can be stated that polynomial chaos methods have the potential to efficiently quantize uncertainties even in highly sophisticated test cases. As already observed in the 2D studies, geometrical uncertainties have a significant effect on the performance of the profile. In order to ensure a robust performance in real life conditions, the consideration of uncertainties in the aerodynamic design process chain is absolutely essential.

Chapter 9

Conclusions and Outlook

The aim of this work was to provide a general framework for robust optimization in aerodynamics with the focus on industrial applicability. Based on the existing simulation and optimization tools, efficient methods were developed in order to identify, quantize and include uncertainties in the overall optimization procedure, thus making robust design in this sense possible.

9.1 Conclusions

First of all, a classification of uncertainties arising in aerodynamic optimization problems was discussed. We concentrated on aleatory uncertainties caused by natural, unpredictable variations of parameter values, initial and boundary conditions and of the geometry itself. Due to the stochastic nature of the considered uncertainties, a stochastic approach modeling the behavior of the uncertain quantities was chosen. The proposed model allows to include arbitrary input uncertainties avoiding a parametrization of the input uncertainties and hence a reduction of the space of realizations. Furthermore, this approach gives the flexibility to adapt the robust optimization to new information of the input uncertainties, e.g. if new measurements are available. The additional information of the uncertain parameters given by the stochastic model can be used to reduce the computational effort arising by the computation of statistics of the quantities of interest. The numerical results demonstrate the potential of exploiting the stochastic information of the input uncertainties. Considering geometrical uncertainties, the use of problem dependent discretization techniques with respect to the density function of the random variables is essential in order to obtain a numerically tractable robust optimization problem.

In the next step, we discussed several formulations of the deterministic aerodynamic shape optimization problem to include the identified uncertainties in the optimization procedure. We concentrated on the chance-constrained and the semi-infinite formulation, both belonging to the class of stochastic optimization methods. The semi-infinite as well as the chance-constrained formulation minimize the expected value of the objective function as a measure of robustness. The difference between these two robust formulations lies in the treatment of the additional lift constraint. The semi-infinite approach aims at maintaining the lift constraint for all realizations of the uncertain parameter, whereas the chance-constrained formulation requires to fulfill the lift constraint only with a certain probability.

Both robust counterparts of the deterministic aerodynamic shape optimization problem involve the evaluation of statistics of quantities of interest depending on the uncertain parameter. In order to efficiently quantize the influence of uncertainties in the input data on quantities in the output of a simulation, several uncertainty quantification methods were proposed. The moment methods, which are based on a Taylor series approximation, were applied to the chance-constrained formulation to approximate the objective as well as the lift constraint with respect to the uncertain input data. However, the numerical results indicate that even a second order approach cannot capture the nonlinear behavior of the objective function. As an alternative to the moment methods, we considered the non-intrusive polynomial chaos approach expanding the quantities of interest in a series of orthogonal polynomials with respect to the distribution of the random input data. The polynomial chaos representation can then be used to efficiently compute statistics of the solution, which was demonstrated in a highly sophisticated 3D test case. For the special case of input random fields modeling geometrical uncertainties, another well known uncertainty quantification technique, the Karhunen-Loève expansion, which is based on the spectral decomposition of the covariance kernel, was introduced. This approach is not the appropriate choice in order to compute statistics of the solution, as the covariance function of the random output of a simulation is not known. However, if the covariance function is at hand, the Karhunen-Loève expansion will be the preferred choice, since the approximation is optimal in the mean-square sense. An enhancement of the approach using a goal-oriented choice of the Karhunen-Loève basis was proposed, which has the potential to significantly reduce the computational effort as demonstrated in the numerical results.

Evaluation of statistics and uncertainty quantification representations usually requires the computation of high dimensional integrals, which cannot be solved analytically. To circumvent the curse of dimensionality, sparse grid methods in combination with dimension adaptive and local refinement strategies were discussed. The reduction of the number of discretization points is an important issue in the context of robust design, since the computational effort of the numerical quadrature arises in every iteration of the optimization algorithm. Both adaptive refinement strategies have the potential to achieve an enormous improvement compared to the tensor grid and usual sparse grid quadrature methods, as shown in the numerical results. Due to the possibility to use problem dependent quadrature formulas, the dimension adaptive refinement method shows a better performance in the considered test cases than the local refinement strategy.

In order to optimize the discretized semi-infinite formulation, we generalized the one-shot method to the multiple-setpoint case. For each realization of the uncertain input data, the inner iterations in the flow and adjoint solvers can be done in parallel, so that the computational time evaluating the flow equation and gradients in the multiple-setpoint case is comparable to the deterministic optimization. The semi-infinite lift constraint can be efficiently handled by a reduction ansatz considering scalar-valued uncertainties and by a discretization approach in the case of geometrical uncertainties. In general, the treatment of semi-infinite problems is a difficult task. However, in the application of interest, we could observe a monotone behavior of the lift constraint with respect to the uncertain input data, which simplifies the optimization of the lower level problem. In contrast to the semi-infinite formulation, gradients of the chance-constrained formulation with respect to the design variables cannot be evaluated by the use of the available adjoint solvers. Due to this fact, the Nelder-Mead method was chosen to solve the optimization problem. As this optimization method is designed for unconstrained problems, the chance-constraint was included by a penalty approach.

The first application presented in this thesis consists of a numerical comparison between the chance-constrained and the semi-infinite formulation considering a 2D transonic Euler test case under uncertain Mach number. It can be observed that the semi-infinite optimization leads to the better drag to lift ratio over the whole range of variations than the chance-constrained formulation. The moment method, which was applied to the chance-constrained formulation, is not able to capture the nonlinear behavior of the objective function. Further, the proposed method to solve the semi-infinite optimization problem based on the generalized one-shot method and the adjoint approach to calculate gradients is much more computationally efficient than the gradient-free optimization method used to solve the chance-constrained problem. Therefore, the semi-infinite approach is the preferable method, especially in high dimensional robust design tasks, due to the better performance over the whole range of perturbations and the more efficient optimization strategy.

In the second application, the influence of adding higher order terms to the mean value of the objective function was investigated in a 2D transonic Euler optimization under uncertain angle of attack. The results of the robust optimization with different weights of the variance point out how important the proper choice of the risk aversion parameter is. Additional knowledge in order to reach the desired grade of conservatism with respect to the input uncertainty needs to be at hand. To circumvent this difficulty, we proposed to add an additional inequality constraint, so that the risk aversion parameter is related to the Lagrangian multiplier resulting from the corresponding Lagrangian function of the robust optimization problem.

Further, the robust optimization of the 2D transonic profile under uncertain Mach number and angle of attack was investigated. In the two-dimensional case, a tensor grid based on Gaussian quadrature formulas adapted to the truncated normal distribution of the uncertain parameters was used in order to discretize the semi-infinite formulation. The resulting multiple-setpoint problem was efficiently solved by the generalized one-shot method and by a reduction ansatz. The optimized robust shape differs quite strongly from the single-setpoint optimized shape, which can be partly attributed to the higher lift requirements of the semi-infinite formulation. The numerical results show that the drag to lift ratio could be essentially improved over the whole range of variations by the robust approach. So, the robust optimization enables the computation of designs, which still show a good performance under variations of the uncertain input data.

Summarizing the results of the robust optimization under scalar-valued uncertainties, we can state that even small perturbations of the nominal points of the uncertain input data may cause a serious deterioration of performance of the deterministic optimized profile. As variations of the Mach number and angle of attack are inherently present in real life conditions due to atmospheric turbulences, they have to be considered in the design process chain. The amount of computational effort resulting from the robust optimization can be efficiently reduced by exploiting the structure of the underlying problem. The use of a parallel version of the generalized one-shot method, problem dependent choice of the quadrature points in order to compute the statistics of the solution and the reduction ansatz to solve the semi-infinite constraint can significantly reduce the costs of the robust optimization, so that robust design becomes numerically tractable in the aerodynamic framework.

In the special case of geometrical uncertainties, the computational complexity increases additionally due to the stochastic model of the random geometry. In this thesis, we proposed methods to handle this high dimensional optimization tasks and presented results of optimization under shape uncertainties in a 2D transonic Euler and Navier-Stokes test case. The random field describing the geometrical perturbations could be efficiently approximated by a finite number of random variables using a Karhunen-Loève expansion. In the Euler as well as in the Navier-Stokes case, it was shown that a goal-oriented choice of the Karhunen-Loève basis, which was automatically determined by the introduced indicator measuring the influence of the random variables on the target functional, further decreased the dimension of the probability space. In order to compute statistics of quantities of interest, sparse grids combined with two different adaptive refinement strategies were considered. The numerical comparison between the local and dimension adaptive refinement indicates that the dimension adaptive approach seems to be of advantage in our application due to the possibility to choose the underlying quadrature formulas with respect to the distributions of the input uncertainties. The number of discretization points could be reduced by a factor of 16 in the Euler case and by a factor of 20 in the Navier-Stokes test case using the dimension adaptive sparse grids. Since the evaluation of statistics is required in each iteration of the optimization procedure, this reduction of costs is also achieved in each iteration. Furthermore, the solution of the semi-infinite constraint was approximated by a discretization method. The studies of the influence of the geometrical uncertainties on the lift indicated a monotone behavior guaranteeing the convergence of the finite subproblems to the original semi-infinite problem. The numerical results of the robust optimization show that even small deviations from the planned geometry have a significant effect on the drag and lift coefficient, so that geometrical uncertainties have to be taken into account in the aerodynamic design optimization problem to ensure a robust solution. The amount of computational effort compared to optimization under scalar-valued uncertainties can be overcome by adaptive uncertainty quantification and sparse grid techniques.

The last application presented in this work is on uncertainty quantification in a 3D test case in order to determine the influence of geometrical uncertainties on flow parameters. A non-intrusive polynomial chaos method was applied to compute statistics of the scalar-valued coefficients, drag and lift, and of the pressure distribution on the surface. Due to the complexity of the test case, a linear approximation with respect to the shape uncertainties was chosen. The results demonstrate that non-intrusive polynomial chaos methods in combination with a Karhunen-Loève approximation of the input random field and sparse grid methods are able to efficiently quantize the influence of geometrical uncertainties on quantities of interest. As already observed in the 2D test cases, shape uncertainties have a significant impact on the performance of the airfoil and therefore have to be included in the design process chain. A general framework for robust aerodynamic design attacking the additional computational complexity of the treatment of uncertainties was proposed in this work, so that the identified uncertainties can be taken into account.

9.2 Future Work

The methods developed in this thesis provide a great contribution to the treatment of uncertainties in the context of aerodynamic design and proved their efficiency in several industrial test cases. Being able to consider uncertainties in aerodynamic optimization tasks, new challenges and open questions arise. The main task is to further reduce the additional computational complexity of the robust approach in order to enable the application of the methods to highly sophisticated 3D test cases, especially to complete aircraft models. The evaluation of high dimensional integrals resulting from the direct approximation of statistics as well as from uncertainty quantification approximations is a crucial part of robust optimization with regard to the computational costs. Adaptive sparse grid methods, which have the potential to significantly reduce the flow simulations required to approximate the integrals, were proposed in this work. It can be expected that gradient information can further enhance the approximation quality and should therefore be included in the discretization method. Additionally, due to the active research in the area of uncertainty quantification methods in the context of computational fluid dynamics, there are now several enhancements and further developments of non-intrusive polynomial chaos and stochastic collocation methods improving the approximation behavior available. So, the new uncertainty quantification methods should be taken into account within the robust optimization approach.

Another open question is the proper treatment of additional constraints in the optimization problem. Two robust formulations of constraints with uncertain input data were discussed, the chance-constrained and semi-infinite formulation. In the case of lift requirements, the considered uncertainties showed a monotone behavior simplifying the approach to solve the robust constraints. To handle arbitrary input uncertainties, additional effort is needed to properly model the stochastic output of the constraint with respect to uncertain input data, e.g. by the use of uncertainty quantification techniques in combination with adaptive discretization methods.

Furthermore, the coupling of robust optimization with novel developments in highly efficient deterministic numerical optimization methods based on shape calculus are under current research, cf. [158]. Parametric design optimization suffers from unavoidable construction of mesh sensitivities, which becomes a major bottleneck, when the level of detail of the geometrical resolution is increased beyond some handfuls of geometrical parameters. The use of the shape calculus approach is able to overcome this drawback. It is to be expected that a combination of shape calculus and robust optimization methodologies leads to novel efficient, robust and accurate methods for aerodynamic design. Moreover, the coupling of robust aerodynamic and aerostructure methods leading to multidisciplinary aerodynamic shape and structural design optimization of airplane wings can be envisioned.

Bibliography

[1] E. Acar and R. T. Haftka, *Reliability based aircraft structural design optimization with uncertainty about probability distributions*, in 6th World Congress of Structural and Multidisciplinary Optimization, 2005.

[2] S. Acharjee and N. Zabaras, *A non-intrusive stochastic Galerkin approach for modeling uncertainty propagation in deformation processes*, Computers and Structures, 85 (2007), pp. 244–254.

[3] R. J. Adler and J. Taylor, *Random Fields and Geometry*, Springer, 2007.

[4] S. Ahmed and A. Shapiro, *Solving chance-constrained stochastic programs via sampling and integer programming*, in TutORials in Operations Research, Z. L. Chen and S. Raghavan, eds., INFORMS, 2008, pp. 261–269.

[5] AIAA, *Guide for the verification and validation of computational fluid dynamics simulations*, American Institute of Aeronautics & Astronautics, 1998.

[6] J. D. Anderson, *Computational Fluid Dynamics: The Basics with Applications*, McGraw Hill, 1995.

[7] B. M. Ayyub and G. J. Klir, *Uncertainty Modeling and Analysis in Engineering and the Sciences*, CRC Press, 2006.

[8] I. Babuska, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[9] V. Barthelmann, E. Novak, and K. Ritter, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 12 (2000), pp. 273–288.

[10] R. Becker, M. Braack, D. Meidner, R. Rannacher, and B. Vexler, *Adaptive finite element methods for PDE-constrained optimal control problems*, in Reactive Flows, Diffusion and Transport, 2006, pp. 177 – 205.

[11] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton Series in Applied Mathematics, Princeton University Press, October 2009.

[12] A. Ben-Tal and A. Nemirovski, *Robust solutions of uncertain linear programs*, Operations Research Letters, 25 (1999), pp. 1–13.

[13] ——, *Selected topics in robust convex optimization*, Mathematical Programming, 112 (2008), pp. 125–158.

[14] F. E. BENTH AND J. GJERDE, *Convergence rates for finite element approximations of stochastic partial differential equations*, Stochastics and stochastics reports, 63 (1998), pp. 313 – 326.

[15] D. BERTSIMAS AND M. SIM, *Robust discrete optimization and network flows*, Mathematical Programming, 98 (2003), pp. 49–71.

[16] ——, *Tractable approximations to robust conic optimization problems*, Mathematical Programming, 107 (2006), pp. 5–36.

[17] H.-G. BEYER AND B. SENDHOFF, *Robust optimization - a comprehensive survey*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 3190–3218.

[18] J. R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer, 1997.

[19] J. BLAZEK, *Computational Fluid Dynamics: Principles and Applications*, Elsevier, 2004.

[20] H. BOCK, W. EGARTNER, W. KAPPIS, AND V. SCHULZ, *Practical shape optimization for turbine and compressor blades by the use of PRSQP methods*, Optimization and Engineering, 3 (2002), pp. 395–414.

[21] H. BOCK, E. KOSTINA, A. SCHÄFER, J. SCHLÖDER, AND V. SCHULZ, *Multiple set point partially reduced SQP method for optimal control of PDE*, in Reactive Flows, Diffusion and Transport. From Experiments via Mathematical Modeling to Numerical Simulation and Optimization Final Report of SFB (Collaborative Research Center) 359, R. R. W. Jäger and J. Warnatz, eds., 2007.

[22] A. BORZÌ AND G. VON WINCKEL, *Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2172–2192.

[23] J. BREZILLON AND R. DWIGHT, *Aerodynamic shape optimization using the discrete adjoint of the Navier-Stokes equations: Applications towards complex 3D configurations*, in Proceedings of the CEAS/KATnet II Conference on Key Aerodynamic Technologies, Paper No. 36-1, CEAS, 2009.

[24] H.-J. BUNGARTZ, *Finite Elements of Higher Order on Sparse Grids*, Habilitationsschrift, Fakultät für Informatik, Technische Universität München, 1998.

[25] H.-J. BUNGARTZ AND S. DIRNSTORFER, *Multivariate quadrature on adaptive sparse grids*, Computing, 71 (2003), pp. 89–114.

[26] G. CALAFIORE AND L. EL GHAOUI, *Ellipsoidal bounds for uncertain linear equations and dynamical systems*, Automatica, 40 (2004), pp. 773–787.

[27] R. H. CAMERON AND W. T. MARTIN, *The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals*, The Annals of Mathematics, 48 (1947), pp. 385–392.

[28] M. C. CAMPI, *Uncertain convex programs: Randomized solutions and confidence levels*, Mathematical Programming, 102 (2005), pp. 25–46.

[29] P. Castonguay and S. Nadarajah, *Effect of shape parameterization on aerodynamic shape optimization*, in 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.

[30] A. Celikyilmaz and I. B. Türksen, *Modeling Uncertainty with Fuzzy Logic - With Recent Theory and Applications*, Springer, 2009.

[31] A. Charnes, W. Cooper, and G. Symonds, *Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil*, Management Science, 4 (1958), pp. 183–195.

[32] H. Cheng and A. Sandu, *Numerical study of uncertainty quantification techniques for implicit stiff systems*, in ACM-SE 45: Proceedings of the 45th Annual Southeast Regional Conference, New York, 2007, ACM, pp. 367–372.

[33] S. Conti, H. Held, M. Pach, M. Rumpf, and R. Schultz, *Risk averse shape optimization*, Tech. Rep. SPP1253-074, DFG-SPP 1253, 2009.

[34] A. W. Date, *Introduction to Computational Fluid Dynamics*, Cambridge University Press, 2005.

[35] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Dover Publications, November 2007.

[36] A. Deane et al., *Parallel Computational Fluid Dynamics: Theory and Applications: Proceedings of the Parallel CFD 2005 Conference*, Elsevier Science & Technology Books, 2006.

[37] M. Diehl, H. G. Bock, and E. Kostina, *An approximation technique for robust nonlinear optimization*, Mathematical Programming, 107 (2006), pp. 213–230.

[38] U. Diwekar, *Optimization under uncertainty: An overview*, SIAG/Optimization Views-and-News, SIAM Journal of Scientific Computing, 13 (2002).

[39] R. A. Durrett, *Probability. Theory and Examples*, Duxbury Press, 3 ed., 2004.

[40] R. Duvigneau, *Aerodynamic shape optimization with uncertain flow conditions*, in 44eme Colloque d'Aérodynamique Appliquée AAAF, 2009.

[41] R. Dwight and J. Brezillon, *Effect of approximations of the discrete adjoint on gradient-based optimization*, AIAA Journal, 44 (2006), pp. 3022–3071.

[42] L. El Ghaoui, M. Oks, and F. Oustry, *Worst-case value-at-risk and robust portfolio optimization: A conic programming approach*, Operations Research, 51 (2003), pp. 543–556.

[43] A. H. El-Shaarawi and W. W. Piegorsch, eds., *Encyclopedia of Environmetrics*, John Wiley & Sons, 2001.

[44] M. Eldred and J. Burkardt, *Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification*, in Proceedings of the 47th AIAA Aerospace Sciences Meeting, 2009.

[45] M. Eldred, C. Webster, and P. Constantine, *Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos*, in Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (10th AIAA Non-Deterministic Approaches Conference), 2008.

[46] J. S. Faragher, *Probabilistic methods for the quantification of uncertainty and error in computational fluid dynamics simulations*, DSTO Platforms Sciences Laboratory, 2004.

[47] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics, Volume 1. Fundamental and General Techniques*, Springer, 2003.

[48] ——, *Computational Techniques for Fluid Dynamics, Volume 2. Specific Techniques for Different Flow Categories*, Springer, 2003.

[49] P. Frauenfelder, C. Schwab, and R. A. Todor, *Finite elements for elliptic problems with stochastic coefficients*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 205 – 228. Selected papers from the 11th Conference on The Mathematics of Finite Elements and Applications.

[50] ——, *Finite elements for elliptic problems with stochastic coefficients*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 205–228.

[51] J. Garcke, *A dimension adaptive sparse grid combination technique for machine learning*, in Proceedings of the 13th Biennial Computational Techniques and Applications Conference, CTAC-2006, W. Read, J. W. Larson, and A. J. Roberts, eds., vol. 48 of ANZIAM Journal, 2007, pp. 725 – 740.

[52] J. Garcke and M. Griebel, *Classification with sparse grids using simplicial basis functions*, Intelligent Data Analysis, 6 (2002), pp. 483–502.

[53] J. Garcke, M. Griebel, and M. Thess, *Data mining with sparse grids*, Computing, 67 (2001), pp. 225–253.

[54] N. Gauger, *Das Adjungiertenverfahren in der aerodynamischen Formoptimierung*, PhD thesis, Technische Universität Braunschweig, 2003.

[55] T. Gerstner and M. Griebel, *Dimension-adaptive tensor-product quadrature*, Computing, 71 (2003), pp. 65–87.

[56] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover Publications, 2003.

[57] L. E. Ghaoui, F. Oustry, and H. Lebret, *Robust solutions to uncertain semidefinite programs*, SIAM Journal on Optimization, 9 (1998), pp. 33–52.

[58] I. Gherman, *Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems*, PhD thesis, University of Trier, 2008.

[59] A. M. Giese, C. Kaebe, J. H. Maruhn, and E. W. Sachs, *Efficient calibration for problems in option pricing*, PAMM, 7 (2007), pp. 1062601 – 1062602.

[60] M. B. GILES AND N. A. PIERCE, *Adjoint equations in CFD: Duality, boundary conditions and solution behaviour*, AIAA, 97 (1997).

[61] ——, *An introduction to the adjoint approach to design*, Flow, Turbulence and Combustion, 65 (2000), pp. 393 – 415.

[62] P. GLASSERMAN, *Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability)*, Springer, August 2003.

[63] A. GRIEWANK, *Projected Hessians for preconditioning in one-step one-shot design optimization*, Nonconvex Optimization and its Application, 83 (2006), pp. 151–172.

[64] M. GRIGORIU, *Applied non-Gaussian processes: Examples, theory, simulation, linear random vibration, and MATLAB solutions*, PTR Prentice Hall, 1995.

[65] C. R. GUMBERT, P. A. NEWMAN, AND G. J.-W. HOU, *Effect of random geometric uncertainty on the computational design of a 3D flexible wing*, in 20th AIAA Applied Aerodynamics Conference, 2002.

[66] ——, *High-fidelity computational optimization for 3D flexible wings: Part II - effect of random geometric uncertainty on design*, Optimization and Engineering, 6 (2005), pp. 139 – 156.

[67] E. HAAREN-RETAGNE, *A semi-infinite programming algorithm for robot trajectory planning*, PhD thesis, University of Trier, 1992.

[68] J. HAMMERSLEY AND D. HANDSCOMB, *Monte Carlo Methods*, Methuen & Co, 1964.

[69] S. HAYASAKA, A. M. PEIFFER, C. E. HUGENSCHMIDT, AND P. J. LAURIENTI, *Power and sample size calculation for neuroimaging studies by non-central random field theory*, NeuroImage, 37 (2007), pp. 721 – 730.

[70] R. HEINRICH, R. DWIGHT, M. WIDHALM, AND A. RAICHLE, *Algorithmic developments in TAU*, in MEGAFLOW - Numerical Flow Simulation for Aircraft Design, N. Kroll and J. Fassbender, eds., Springer, 2005, pp. 93–108.

[71] R. HENRION, *Structural properties of linear probabilistic constraints*. Stochastic Programming E-Print Series, `http://www.speps.org`, 2005.

[72] R. HETTICH AND K. O. KORTANEK, *Semi-infinite programming: Theory, methods, and applications*, SIAM Review, 35 (1993), pp. 380–429.

[73] R. P. HETTICH AND H. JONGEN, *Semi-infinite programming: Conditions of optimality and applications*, in Optimization Techniques, vol. 7, Springer, 1978, pp. 1 – 11.

[74] R. M. HICKS AND P. A. HENNE, *Wing design by numerical optimization*, Journal of Aircraft, 15 (1978), p. 407–412.

[75] S. HOSDER, R. WALTERS, AND R. PEREZ, *A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-891, 2006.

[76] L. HUYSE, R. LEWIS, W. LI, AND S. PADULA, *Probabilistic approach to free-form airfoil shape optimization under uncertainty*, AIAA Journal, 40 (2002), pp. 1764–1772.

[77] A. JAMESON, *Aerodynamic design via control theory*, Journal of Scientific Computing, 3 (1988), pp. 233–260.

[78] ——, *Computational Aerodynamics for Aircraft Design*, Science, 245 (1989), pp. 361–371.

[79] H. JONGEN AND J.-J. RÜCKMANN, *On stability and deformation in semi-infinite optimization*, in Semi-Infinite Programming, R. Reemtsen and J.-J. Rückmann, eds., Kluwer, 1998, pp. 29–67.

[80] P. KALL AND S. WALLACE, *Stochastic Programming*, Wiley, Chichester, 1994.

[81] A. KEESE, *A review of recent developments in the numerical solution of stochastic PDEs (stochastic finite elements)*, Tech. Rep. 2003-6, Technische Universität Braunschweig, Brunswick, 2003.

[82] ——, *Numerical solution of systems with stochastic uncertainties: A general purpose framework for stochastic finite elements*, PhD thesis, Technische Universität Braunschweig, 2004.

[83] C. T. KELLEY, *Iterative methods for optimization*, Society for Industrial Mathematics, 1987.

[84] B. N. KHOROMSKIJ, A. LITVINENKO, AND H. G. MATTHIES, *Application of hierarchical matrices for computing the Karhunen-Loève expansion*, Computing, 84 (2009), pp. 49–67.

[85] D. KHOSHNEVISAN, *Multiparameter Processes (An Introduction to Random Fields)*, Springer, 2002.

[86] J. M. KILNER, S. J. KIEBEL, AND K. J. FRISTON, *Applications of random field theory to electrophysiology*, Neuroscience Letters, 374 (2005), pp. 174 – 178.

[87] A. KLIMKE, *Uncertainty modeling using fuzzy arithmetic and sparse grids*, PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.

[88] A. KLIMKE AND B. WOHLMUTH, *Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB*, ACM Transactions on Mathematical Software, 31 (2005), pp. 561–579.

[89] O. M. KNIO AND O. P. LE MAÎTRE, *Uncertainty propagation in CFD using polynomial chaos decomposition*, Fluid Dynamics Research, 38 (2006), pp. 616–640.

[90] A. V. KNYAZEV, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace*, in International Series Numerical Mathematics, Eigenwertaufgaben in Natur- und Ingenieurwissenschaften und ihre numerische Behandlung, Oberwolfach, Birkhauser, 1991, pp. 143–154.

[91] ——, *Preconditioned eigensolvers: Practical algorithms*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 352–368.

[92] ——, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM Journal of Scientific Computing, 23 (2001), pp. 517–541.

[93] A. V. Knyazev, M. E. Argentati, I. Lashuk, and E. E. Ovtchinnikov, *Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in Hypre and PETSc*, SIAM Journal of Scientific Computing, 29 (2007), pp. 2224–2239.

[94] R. Koekoek and R. F. Swarttouw, *The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue*, Tech. Rep. 94-05, Faculty of Technical Mathematics and Informatics, Technische Universiteit Delft, 1994.

[95] K. Kolanek and S. Jendo, *Random field models of geometrically imperfect structures with clamped boundary conditions*, Probabilistic Engineering Mechanics, 23 (2008), pp. 219 – 226. 5th International Conference on Computational Stochastic Mechanics.

[96] T. Kröger, I. Altrogge, O. Konrad, R. M. Kirby, and T. Preusser, *Estimation of probability density functions for parameter sensitivity analyses*, in SimVis, 2008, pp. 61–74.

[97] N. Kroll and J. K. Fassbender, eds., *MEGAFLOW - Numerical Flow Simulation for Aircraft Design*, Springer, 2005.

[98] N. Kroll, D. Schwamborn, K. Becker, H. Rieger, and F. Thiele, eds., *MEGADESIGN and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, Springer, 2010.

[99] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder-Mead Simplex algorithm in low dimensions*, SIAM Journal of Optimization, 9 (1996), pp. 112–147.

[100] B. Lastdrager and B. Koren, *Error analysis for function representation by the sparse-grid combination technique*, Tech. Rep. MAS-R9823, CWI, Amsterdam, 1998.

[101] O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics (Scientific Computation)*, 1 ed., 2010.

[102] O. P. Le Maître, O. M. Knio, H. N. Najm, and R. G. Ghanem, *A stochastic projection method for fluid flow - I: Basic formulation*, Journal of Computational Physics, 173 (2001), pp. 481–511(31).

[103] ——, *Multi-resolution analysis of Wiener-type uncertainty propagation schemes*, Journal of Computational Physics, 197 (2004), pp. 502 – 531.

[104] ——, *Uncertainty propagation using Wiener-Haar expansions*, Journal of Computational Physics, 197 (2004), pp. 28 – 57.

[105] O. P. Le Maître, H. N. Najm, P. P. Pèbay, R. G. Ghanem, and O. M. Knio, *Multi-resolution-analysis scheme for uncertainty quantification in chemical systems*, SIAM Journal of Scientific Computing, 29 (2007), pp. 864–889.

[106] O. P. Le Maître, M. T. Reagan, H. N. Najm, R. G. Ghanem, and O. M. Knio, *A stochastic projection method for fluid flow II: Random process*, Journal of Computational Physics, 181 (2002), pp. 9–44.

[107] C. Lemieux, *Monte Carlo and Quasi-Monte Carlo Sampling (Springer Series in Statistics)*, Springer, 1 ed., 2009.

[108] P. Li, H. Arellano-Garcia, and G. Wozny, *Chance constrained programming approach to process optimization under uncertainty*, Computers & Chemical Engineering, 32 (2008), pp. 25 – 45. Process Systems Engineering: Contributions on the State-of-the-Art - Selected extended Papers from ESCAPE '16/PSE 2006.

[109] W. Li, L. Huyse, S. Padula, and L. R. Center., *Robust airfoil optimization to achieve consistent drag reduction over a mach range*, National Aeronautics and Space Administration, Langley Research Center, 2001.

[110] W. Li and S. L. Padula, *Robust airfoil optimization in high resolution design space*, in ICASE NASA Langley Research Centre, 2003.

[111] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, October 2002.

[112] Z. Liu, Y. Lai, X. Zhang, and M. Zhang, *Random temperature fields of embankment in cold regions*, Cold Regions Science and Technology, 45 (2006), pp. 76 – 82.

[113] M. Loève, *Probability Theory 1*, Springer, 4 ed., 1977.

[114] ——, *Probability Theory 2*, Springer, 4 ed., 1978.

[115] G. Loeven, *Efficient uncertainty quantification in computational fluid dynamics*, PhD thesis, Technische Universiteit Delft, 2010.

[116] G. Loeven and H. Bijl, *Airfoil analysis with uncertain geometry using the probabilistic collocation method*, in 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA-2008-2070, 2008.

[117] X. Ma and N. Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations*, Journal of Computational Physics, 228 (2009), pp. 3084–3113.

[118] G. I. Marchuk, V. I. Agoshkov, and V. P. Shutyaev, *Adjoint Equations and Perturbation Algorithms*, CRC Press, 1996.

[119] H. Markowitz, *Portfolio selection*, The Journal of Finance, 7 (1952), pp. 77–91.

[120] J. H. Maruhn, *Robust Static Super-Replication of Barrier Options*, de Gruyter, 2009.

[121] L. Mathelin, M. Y. Hussaini, and T. A. Zang, *Stochastic approaches to uncertainty quantification in CFD simulations*, Numerical Algorithms, 38 (2005), pp. 209 – 236.

[122] H. G. MATTHIES, *Quantifying uncertainty: Modern computational representation of probability and applications*, in Extreme Man-Made and Natural Hazards in Dynamics of Structures, A. Ibrahimbegovic and I. Kozar, eds., vol. 21 of NATO Security through Science Series, Springer Netherlands, 2007, pp. 105–135.

[123] J. MERCER, *Functions of positive and negative type and their connection with the theory of integral equations*, Philosophical Transactions of the Royal Society, London, (1909).

[124] M. MEYER AND H. MATTHIES, *Efficient model reduction in nonlinear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods*, Computational Mechanics, 31 (2003).

[125] R. B. MORGAN, *Preconditioning eigenvalues and some comparison of solvers*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 101 – 115.

[126] A. MOUSAVI, P. CASTONGUAY, AND S. K. NADARAJAH, *Survey of shape parametrization techniques and its effect on three-dimensional aerodynamic shape optimization*, in 18th AIAA Computational Fluid Dynamics Conference, 2007.

[127] J. M. MULVEY, R. J. VANDERBEI, AND S. A. ZENIOS, *Robust optimization of large-scale systems*, Operations Research, 43 (1995), pp. 264–281.

[128] H. N. NAJM, *Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics*, Annual Review of Fluid Mechanics, 41 (2009).

[129] J. A. NELDER AND R. MEAD, *A simplex method for function minimization*, The Computer Journal, 7 (1965), pp. 308–313.

[130] A. NEMIROVSKI AND A. SHAPIRO, *Scenario approximations of chance constraints*, in Probabilistic and Randomized Methods for Design under Uncertainty, Springer, 2004, pp. 3–48.

[131] H. NIEDERREITER, *Random number generation and Quasi-Monte Carlo methods*, Society for Industrial and Applied Mathematics, Philadelphia, USA, 1992.

[132] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, August 2000.

[133] W. OBERKAMPF, J. HELTON, S. WOJTKIEWICZ, C. JOSLYN, AND S. FERSON, *Epistemic uncertainty workshop*, Reliable Computing, 8 (2002), pp. 503–505.

[134] W. L. OBERKAMPF, S. M. DELAND, B. M. RUTHERFORD, K. V. DIEGERT, AND K. F. ALVIN, *Error and uncertainty in modeling and simulation*, Reliability Engineering & System Safety, 75 (2002), pp. 333 – 357.

[135] H. OGURA, *Orthogonal functionals of the poisson process*, IEEE Transactions on Information Theory, 18 (1972), pp. 473–481.

[136] S. PADULA, C. GUMBERT, AND W. LI, *Aerospace applications of optimization under uncertainty*, Optimization and Engineering, 7 (2006), pp. 317–328.

[137] K. PETRAS, *Smolyak cubature of given polynomial degree with few nodes for increasing dimension*, Numerische Mathematik, 93 (2003), pp. 729–753.

[138] G. C. PFLUG, *Random planar shapes and their statistical recognition*, Annals of Mathematics and Artificial Intelligence, 13 (2005), pp. 267 – 279.

[139] O. PIRONNEAU, *On optimum design in fluid mechanics*, Journal of Fluid Mechanics Digital Archive, 64 (1974), pp. 97–110.

[140] E. POLAK AND L. HE, *Rate-preserving discretization strategies for semi-infinite programming and optimal control*, SIAM Journal on Control and Optimization, 30 (1992), pp. 548–572.

[141] C. J. PRICE, I. D. COOPE, AND D. BYATT, *A convergent variant of the Nelder-Mead algorithm*, Journal of Optimization Theory and Applications, 113 (2002), pp. 5–19.

[142] A. PRÉKOPA, *Stochastic Programming*, Kluwer Academic Publisher, 1995.

[143] M. PUTKO, P. NEWMAN, A. TAYLOR III, AND L. GREEN, *Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives*, in AIAA 15-th Computational Fluid Dynamics Conference, AIAA, 2001, pp. 2001–2528.

[144] M. RAKOWITZ, B. EISFELD, D. SCHWAMBORN, AND M. SUTCLIFFE, *Structured and unstructured computations on the DLR-F4 wing-body configuration*, Journal of Aircraft, 40 (2003), pp. 256 – 264.

[145] C. E. RASMUSSEN AND C. K. I. WILLIAMS, *Gaussian Processes for Machine Learning*, The MIT Press, 2002.

[146] R. REEMTSEN, *Discretization methods for the solution of semi-infinite programming problems*, Journal of Optimization Theory and Applications, 71 (1991), pp. 85 – 103.

[147] R. REEMTSEN AND J.-J. RÜCKMANN, eds., *Semi-Infinite Programming*, Kluwer, 1998.

[148] J. REUTHER AND A. JAMESON, *A comparison of design variables for control theory based airfoil optimization*, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1995.

[149] J. REUTHER, A. JAMESON, J. FARMER, L. MARTINELLI, AND D. SAUNDERS, *Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation*, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1996.

[150] C. P. ROBERT AND G. CASELLA, *Monte Carlo Statistical Methods*, Springer-Verlag, 1 ed., August 1999.

[151] A. RUSZCZYŃSKI, *Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra*, Mathematical Programming, 93 (2002), pp. 195–215.

[152] A. RUSZCZYŃSKI AND A. SHAPIRO, *Stochastic Programming. Handbook in Operations Research and Management Science*, vol. 10, Elsevier, 2003.

[153] ——, *Optimization of risk measures*, Risk and Insurance, EconWPA, 2004.

[154] J. A. SAMAREH, *A survey of shape parameterization techniques*, Tech. Rep. NASA/CP-1999-209136, Multidisciplinary Optimization Branch, NASA Langley Research Center, 1999.

[155] ——, *Aerodynamic shape optimization based on free-form deformation*, in Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, vol. 6, Multidisciplinary Optimization Branch, NASA Langley Research Center, 2004, pp. 3672–3683.

[156] S. SANKARAN, *Stochastic optimization using a sparse grid collocation scheme*, Probabilistic Engineering Mechanics, 24 (2009), pp. 382 – 396.

[157] C. A. SCHENK AND G. I. SCHUELLER, *Uncertainty Assessment of Large Finite Element Systems*, Springer, 2005.

[158] C. SCHILLINGS, S. SCHMIDT, AND V. SCHULZ, *Efficient shape optimization for certain and uncertain aerodynamic design*, Computers & Fluids, In Press, Accepted Manuscript (2010).

[159] S. SCHMIDT, C. ILIC, N. GAUGER, AND V. SCHULZ, *Shape gradients and their smoothness for practical aerodynamic design optimization*, Tech. Rep. SPP1253-10-03, DFG-SPP 1253, 2008. submitted (OPTE).

[160] S. SCHMIDT AND V. SCHULZ, *Impulse response approximations of discrete shape Hessians with application in CFD*, SIAM Journal on Control and Optimization, 48 (2009), pp. 2562–2580.

[161] W. SCHOUTENS, *Stochastic Processes and Orthogonal Polynomials*, vol. 146, Springer, 2000.

[162] G. I. SCHUELLER, *Computational stochastic mechanics - recent advances*, Computers & Structures, 79 (2001), pp. 2225 – 2234.

[163] 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, 1996.

[164] V. SCHULZ AND I. GHERMAN, *One-shot methods for aerodynamic shape optimization*, in MEGADESIGN and MegaOpt - Aerodynamic Simulation and Optimization in Aircraft Design, N. Kroll, D. Schwamborn, K. Becker, H. Rieger, and F. Thiele, eds., Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer, 2008.

[165] V. SCHULZ AND C. SCHILLINGS, *On the nature and treatment of uncertainties in aerodynamic design*, AIAA Journal, 47 (2009), pp. 646 – 654.

[166] D. SCHWAMBORN, T. GERHOLD, AND R. HEINRICH, *The DLR TAU-code: Recent applications in research and industry*, in European Conference on Computational Fluid Dynamics, ECCOMAS CFD, 2006.

[167] A. SHAPIRO, *First and second order optimality conditions and perturbation analysis of semi-infinite programming problems*, in Semi-Infinite Programming, R. Reemtsen and J.-J. Rückmann, eds., Kluwer, 1998, pp. 29–67.

[168] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Soviet Mathematics Doklady, 4 (1963), pp. 240–243.

[169] W. Song and A. J. Keane, *A study of shape parameterisation methods for airfoil optimisation*, in Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004.

[170] A. L. Soyster, *Convex programming with set-inclusive constraints and applications to inexact linear programming*, Operations Research, 21 (1973), pp. 1154–1157.

[171] C. Stachniss, C. Plagemann, and A. J. Lilienthal, *Learning gas distribution models using sparse Gaussian process mixtures*, Autonomous Robots, 26 (2009), pp. 187 – 202.

[172] S. Ta'asan, G. Kuruvila, and M. Salas, *Aerodynamic design and optimization in one shot*, in 30th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 92-0025, 1992.

[173] N. M. Temme and J. L. López, *The Askey scheme for hypergeometric orthogonal polynomials viewed from asymptotic analysis*, Journal of Computational and Applied Mathematics, 133 (2002), pp. 623–633.

[174] F. G. Vázques, J.-J. Rückmann, O. Stein, and G. Still, *Generalized semi-infinite programming: A tutorial*, Journal of Computational and Applied Mathematics, 217 (2008), pp. 394–419.

[175] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: The finite volume method*, Prentice Hall, 2 ed., 2007.

[176] X. Wan and G. E. Karniadakis, *Multi-element generalized polynomial chaos for arbitrary probability measures*, SIAM Journal of Scientific Computing, 28 (2006), pp. 901–928.

[177] G. W. Wasilkowski and H. Wozniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, Journal of Complexity, 11 (1994), pp. 1–56.

[178] N. Wiener, *The homogeneous chaos*, American Journal of Mathematics, 60 (1938), pp. 897–936.

[179] R. C. Wilcox and B. M. Ayyub, *Uncertainty modeling of data and uncertainty propagation for risk studies*, in Proceedings of the 4th International Symposium on Uncertainty Modelling and Analysis, ISUMA 2003, Washington, DC, USA, 2003, IEEE Computer Society, pp. 184 – 191.

[180] Z. Xiaoping, D. Jifeng, L. Weiji, and Z. Yong, *Robust airfoil optimization with multi-objective estimation of distribution algorithm*, Chinese Journal of Aeronautics, 21 (2008), pp. 289 – 295.

[181] D. Xiu, *Fast numerical methods for stochastic computations: A review*, Communications in Computational Physics, 5 (2009), pp. 242 – 272.

[182] D. Xiu and G. E. Karniadakis, *The Wiener-Askey polynomial chaos for stochastic differential equations*, SIAM Journal of Scientific Computing, 24 (2002), pp. 619–644.

[183] ——, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, Journal of Computational Physics, 187 (2003), pp. 137 – 167.

[184] J. YUAN, K. WANG, T. YU, AND M. FANG, *Reliable multi-objective optimization of high-speed WEDM process based on Gaussian process regression*, International Journal of Machine Tools and Manufacture, 48 (2008), pp. 47 – 60.

[185] M. S. YUCEMEN, *A random field model for the estimation of seismic hazard*, Structural Safety, 12 (1993), pp. 187 – 203.

[186] Y. ZHANG, *General robust-optimization formulation for nonlinear programming*, Journal of Optimization Theory and Applications, 132 (2007), pp. 111 – 124.

[187] D. W. ZINGG AND S. ELIAS, *On aerodynamic optimization under a range of operating conditions*, in 44th AIAA Aerospace Sciences Meeting, Reno, 2006.