

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.) genehmigte Dissertation

Robustness Measures and Optimization Strategies for Multi-Objective Robust Design

Lisa Kusch

Gutachter: Prof. Dr. Nicolas R. Gauger,
AG Scientific Computing,
Technische Universität Kaiserslautern

Prof. Dr. Andrea Walther,
Institut für Mathematik,
Humboldt-Universität zu Berlin

Datum der Disputation: 8. September 2020



Abstract

A significant step to engineering design is to take into account uncertainties and to develop optimal designs that are robust with respect to perturbations. Furthermore, it is often of interest to optimize for different conflicting objective functions describing the quality of a design, leading to a multi-objective optimization problem. In this context, generating methods for solving multi-objective optimization problems seek to find a representative set of solutions fulfilling the concept of Pareto optimality. When multiple uncertain objective functions are involved, it is essential to define suitable measures for robustness that account for a combined effect of uncertainties in objective space. Many tasks in engineering design include the solution of an underlying partial differential equation that can be computationally expensive. Thus, it is of interest to use efficient strategies for finding optimal designs. This research aims to present suitable measures for robustness in a multi-objective context, as well as optimization strategies for multi-objective robust design.

This work introduces new ideas for robustness measures in the context of multi-objective robust design. Losses and expected losses based on distances in objective space are used to describe robustness. A direct formulation and a two-phase formulation based on expected losses are proposed for finding a set of robust optimal solutions.

Furthermore, suitable optimization strategies for solving the resulting multi-objective robust design problem are formulated and analyzed. The multi-objective optimization problem is solved with a constraint-based approach that is based on solving several constrained single-objective optimization problems with a hybrid optimization strategy. The hybrid method combines a global search method on a surrogate model with adjoint-based optimization methods. In the context of optimization with an underlying partial differential equation, a one-shot approach is extended to handle additional constraints.

The developed concepts for multi-objective robust design and the proposed optimization strategies are applied to an aerodynamic shape optimization problem. The drag coefficient and the lift coefficient are optimized under the consideration of uncertainties in the operational conditions and geometrical uncertainties. The uncertainties are propagated with the help of a non-intrusive polynomial chaos approach. For increasing the efficiency when considering a higher-dimensional random space, it is made use of a Karhunen-Loève expansion and a dimension-adaptive sparse grid quadrature.

Zusammenfassung

Ein wesentlicher Schritt in der Entwicklung und Konstruktion von Bauteilen besteht darin, Unsicherheiten zu berücksichtigen und optimale Entwürfe zu finden, welche robust gegenüber gewissen Störungen sind. Ferner ist es häufig von Interesse, verschiedene widersprüchliche Zielfunktionen, welche die Güte der Entwürfe beschreiben, zu optimieren. Daraus resultiert ein Mehrzieloptimierungsproblem. Spezielle Methoden zum Lösen solcher Probleme dienen dazu, eine repräsentative Menge von Lösungen zu finden, welche die Anforderungen der Pareto-Optimalität erfüllen. Wenn mehrere mit Unsicherheiten behaftete Zielfunktionen betrachtet werden, ist es wichtig, geeignete Maße für Robustheit zu definieren, die den kombinierten Effekt der Unsicherheiten im Zielfunktionsraum berücksichtigen. Viele Aufgaben im Bereich der Entwicklung und Konstruktion von Bauteilen umfassen die rechnerische aufwändige Lösung einer zugrundeliegenden partiellen Differentialgleichung. Daher ist es von Interesse, möglichst effiziente Strategien zum Auffinden optimaler Lösungen zu verwenden. Ziel dieser Forschung ist es, geeignete Robustheitsmaße im Bereich der Mehrzieloptimierung und Optimierungsstrategien für den robusten Entwurf mit mehreren Zielfunktionen vorzustellen.

In der vorgelegten Arbeit werden erweiterte Ideen für Robustheitsmaße für den robusten Entwurf mit mehreren Zielfunktionen eingeführt. Verluste und erwartete Verluste, basierend auf Entfernungen im Zielfunktionsraum, beschreiben die Robustheit. Um eine Menge robuster optimaler Lösungen zu finden, werden eine direkte und eine zweiphasige Problemstellung, basierend auf erwarteten Verlusten, vorgeschlagen.

Darüber hinaus werden geeignete Optimierungsstrategien zum Lösen der resultierenden Mehrzieloptimierungsprobleme für den robusten Entwurf formuliert und analysiert. Das Mehrzieloptimierungsproblem wird mit einem Ansatz gelöst, welcher darauf basiert, das Problem in mehrere restringierte Einzieloptimierungsprobleme zu verwandeln und diese mit einer hybriden Optimierungsstrategie zu lösen. Das hybride Verfahren kombiniert ein globales Suchverfahren auf einem Ersatzmodell mit einem Optimierungsverfahren basierend auf der Adjungiertenmethode. Bei der Optimierung mit zugrundeliegenden partiellen Differentialgleichungen wird eine One-Shot-Methode auf die Behandlung zusätzlicher Nebenbedingungen erweitert.

Die entwickelten Ansätze für den robusten Entwurf mit mehreren Zielfunktionen und die vorgeschlagenen Optimierungsstrategien werden für eine aerodynamische Formoptimierung verwendet. Dabei werden der Widerstandsbeiwert und der Auftriebsbeiwert unter der Berücksichtigung von Unsicherheiten im Betriebszustand sowie geometrischen Unsicherheiten optimiert. Die Verbreitung von Unsicherheiten wird mit Hilfe einer nicht-intrusiven Polynomial-Chaos-Methode realisiert. Um die Effizienz bei höherdimensionalen Wahrscheinlichkeitsräumen zu erhöhen, werden eine Karhunen-Loève-Entwicklung und eine Sparse-Grid-Quadraturformel, die adaptiv in der Dimension ist, verwendet.

Acknowledgements

I would like to thank all people who supported me in working on and writing this thesis.

Foremost, I would like to thank my advisor Prof. Dr. Nicolas R. Gauger. He has supported my work, gave me the freedom in research, and has enabled me to make valuable contacts with other researchers. The work with him, starting already in 2011, has inspired my enthusiasm for research in the field of numerical optimization, and it initiated my work on multi-objective optimization. Furthermore, I would like to thank Prof. Dr. Andrea Walther, who initiated the work on extending the one-shot approach to additional equality constraints. We had fruitful discussions, and the cooperation with her was highly beneficial for the results presented in this thesis. I also wish to thank Dr. Lionel Mathelin for his valuable suggestions for my research on multi-objective robust design.

The open-source community of the multiphysics analysis and design optimization software SU2 deserves special thanks for offering the framework enabling the numerical analysis of the present thesis. Here, I want to especially mention my colleague Tim Albring, who introduced me to SU2 and was always ready to help in case of questions. Furthermore, I would like to thank my colleague Dr. Emre Özkaya for his work on the optimization tool RoDeO. This has been very helpful as a basis for my ideas for hybrid optimization methods based on surrogate models. The source code for dimension-adaptive sparse grid quadrature used in the thesis was kindly provided by Prof. Dr. Claudia Schillings, whom I would like to thank for her support. Furthermore, I would like to thank Dr. Katrin Stöbener for sharing her results on visualization in multi-objective optimization.

I owe thanks to my colleagues at the Chair for Scientific Computing at TU Kaiserslautern. The atmosphere in the group always motivated me to commute between my home in Aachen and my work in Kaiserslautern. Furthermore, I wish to thank my colleagues Tobias Kattmann and Dr. Max Sagebaum for proofreading parts of the thesis.

Finally, I would like to thank my family, who believes in me and always supports and encourages me. I want to thank my brother Jonas for proofreading parts of the thesis. My special thanks go to Matthias for his valuable support during the final stages of writing this work.

Contents

Abstract	i
Zusammenfassung	iii
Acknowledgements	v
Contents	vii
Lists	xi
1. Introduction	1
1.1. Motivation	1
1.2. Research Goals and Contributions	9
1.3. Composition of Thesis	13
2. Multi-Objective Optimization	15
2.1. Fundamentals and Definitions	16
2.1.1. Single-Objective Optimization	17
2.1.2. Pareto Optimality	19
2.2. Multi-Objective Optimization Algorithms	24
2.2.1. Direct Pareto Approaches	25
2.2.2. Weighted Sum Method	27
2.2.3. Equality Constraint Method	27
2.2.4. Epsilon-Constraint Method	34
2.2.5. Normal Constraint Method	35
2.3. Hybrid Optimization Strategy	37
2.3.1. Gradient-Based Algorithms for Unconstrained Optimization	38
2.3.2. Gradient-Based Algorithms for Constrained Optimization	41
2.3.3. Hybridization	44
2.4. Application and Results	51
2.4.1. Aerodynamic Shape Optimization Problem	52
2.4.2. Implementation Details	59
2.4.3. Results of Multi-Objective Aerodynamic Shape Optimization	64
3. One-Shot Method for PDE-constrained Optimization	71
3.1. Fundamentals of PDE-constrained optimization	72
3.1.1. Properties and Assumptions	72
3.1.2. Necessary and Sufficient Conditions for Optimality	74

3.1.3.	Sensitivity Computation	75
3.2.	The One-Shot Approach	80
3.2.1.	Fixed-Point Formulation	83
3.2.2.	Piggy-Back Iteration	84
3.2.3.	Single-Step One-Shot Approach	85
3.2.4.	The One-Shot Approach with Additional Equality Constraints	91
3.2.5.	Bound Constraints in the One-Shot Approach	103
3.2.6.	Algorithmic Differentiation	107
3.3.	Application and results	115
3.3.1.	Implementation in SU2	115
3.3.2.	Results of One-Shot Aerodynamic Shape Optimization	122
4.	Robustness in Multi-Objective Optimization	131
4.1.	Probability Theory	131
4.2.	Uncertainty Quantification	133
4.2.1.	Representation of Random Inputs	134
4.2.2.	Methods for Uncertainty Propagation	138
4.2.3.	Quadrature	143
4.3.	Robust Design	147
4.3.1.	Robust Design in a Single-Objective Context	148
4.3.2.	Robust Design in a Multi-Objective Context	152
4.3.3.	Expected Losses in Objective Space	158
4.4.	Application and Results	163
4.4.1.	Uncertainty Quantification and Robust Design in Aerodynamics	163
4.4.2.	Implementation Details	166
4.4.3.	Results for Uncertainties in the Operational Conditions	170
4.4.4.	Results for Geometrical Uncertainties	178
5.	Conclusion and Outlook	183
5.1.	Conclusion	183
5.2.	Outlook	186
	Bibliography	193
A.	Appendix	219
A.1.	Appendix for Chapter 2	219
A.1.1.	Variant of the Equality Constraint Method	219
A.1.2.	Variants of the Epsilon-Constraint Method	219
A.1.3.	Multi-Objective Shape Optimization of 3D U-Bend	221
A.2.	Appendix for Chapter 3	223
A.2.1.	Adjoint Approach: Duality Viewpoint	223
A.2.2.	Inexact rSQP Approach for One-Shot Optimization	223
A.2.3.	One-Shot Approach for Topology Optimization of Nonlinear Structures	224

A.2.4. Comparison of Results for Multi-Objective Optimization	227
A.2.5. Limited-Memory BFGS method for Bound Constraints	228
A.3. Appendix for Chapter 4	229
A.3.1. Approximating the Signed Distance Function	229
A.3.2. Monte Carlo Methods versus Non-Intrusive Polynomial Chaos . . .	230
A.3.3. Additional Results for Operational Uncertainties	231
A.3.4. Multi-Objective Robust Design of 3D U-Bend	233
Curriculum Vitae	235
Lebenslauf	237
List of Publications	239

Lists

List of Figures

2.1. Concept of dominance.	19
2.2. Local and global Pareto optimality.	20
2.3. Steps of equality constraint method.	31
2.4. Scanning the Pareto optimal front.	32
2.5. Problem of locally optimal solutions.	33
2.6. Procedure of epsilon-constraint method.	35
2.7. Aerodynamic forces.	55
2.8. Hicks-Henne parameterization.	56
2.9. Flow solution (NACA0012).	58
2.10. Results of expected improvement method.	62
2.11. Results for test problem.	63
2.12. Flow solution (optimized design).	64
2.13. Results of epsilon-constraint method (aerodynamic shape optimization).	65
2.14. Pressure coefficients for optimal designs.	65
2.15. Visualization of results.	67
2.16. Comparison of results (NC method and epsilon-constraint method).	68
3.1. Optimization history of single-step one shot approach.	123
3.2. Results for MOO with one-shot approach (one additional constraint).	123
3.3. Convergence history for one-shot approach with one additional constraint.	124
3.4. Results for MOO with one-shot approach (three additional constraints).	125
3.5. Convergence history for one-shot approach with three additional constraints.	126
3.6. Results for preconditioner using second-order adjoints.	127
3.7. Convergence history of using preconditioner for multiplier update.	128
4.1. Tensor grid and sparse grid.	146
4.2. Robustness in single- and multi-objective optimization.	152
4.3. Definite gains and losses in objective space.	158
4.4. Robustness based on losses.	159
4.5. Level sets of signed distance function (two-phase approach).	161
4.6. Level sets for estimated losses (direct approach).	163
4.7. Gauss-Hermite quadrature points.	168
4.8. Sparse grid Gauss-Hermite quadrature.	169
4.9. Results of expectation-based approach.	173
4.10. Robust optimal design.	173

4.11. Robustness with respect to uncertain Mach number.	174
4.12. Results of expectation-based approach (one-shot method).	175
4.13. Results of two-phase approach.	176
4.14. Robustness in two-phase approach.	177
4.15. Results of direct approach.	177
4.16. Robustness in direct approach.	178
4.17. Karhunen-Loève expansion.	179
4.18. Examples of dimension-adaptive sparse grids.	180
4.19. Results of expectation-based approach (geometrical uncertainties).	181
A.1. State and adjoint convergence (reverse accumulation).	222
A.2. Results of multi-objective optimization for U-shaped bend.	222
A.3. Topology optimization with one-shot method.	226
A.4. Optimal design: One-shot method.	227
A.5. Comparison of Monte Carlo approach and non-intrusive PC.	230
A.6. Non-intrusive polynomial chaos for variance.	231
A.7. Results for variance-based measure.	232
A.8. Results for two-dimensional uncertainties in operational conditions.	232
A.9. Results for multi-objective robust design of U-shaped bend.	233

List of Tables

3.1. Runtime factors for second-order adjoints.	127
---	-----

List of Algorithms

1. Lexicographic optimization for estimating the Nadir point.	22
2. Determination of step for equidistant scanning.	28
3. Equality constraint method.	30
4. Quasi-Newton method.	40
5. Backtracking Algorithm for finding step length.	41
6. Classical optimization approach.	80
7. BFGS method with bound constraints.	106
8. One-shot optimization in SU2 (pseudo-code).	121
9. Limited-memory BFGS with bound constraints.	228

List of Abbreviations

AD	Algorithmic Differentiation
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CFD	Computational Fluid Dynamics

CWEI	Constrained-Weighted Expected Improvement (method)
DoE	Design of Experiments
EC	Equality Constraint (method)
EI	Expected Improvement (method)
GPC	Generalized Polynomial Chaos
KKT	Karush-Kuhn-Tucker
KL	Karhunen-Loeve (expansion)
KTCQ	Kuhn-Tucker Constraint Qualification
GA	Genetic Algorithm
LICQ	Linear Independence Constraint Qualification
MDS	Multi-Dimensional Scaling
MOO	Multi-Objective Optimization
MPI	Message Passing Interface
NACA	National Advisory Committee for Aeronautics
NC	Normal Constraint (method)
NSGA	Non-dominated Sorting Genetic Algorithm
PC	Polynomial Chaos
PDE	Partial Differential Equation
SAC	Single Assignment Code
SPEA	Strength Pareto Evolutionary Algorithm
SQP	Sequential Quadratic Programming

List of Symbols

This list of symbols introduces all relevant symbols used in the thesis. Due to the different topics covered in the three main chapters some symbols are reused. If this is the case for relevant symbols, the description indicates the respective chapter. Otherwise, the meaning of the symbol is explicitly stated in the thesis. In general, the style of this work to use bold symbols for vectors of variables and for functions for which it is relevant that they map into a vector space.

A	surface area (airfoil)
\mathcal{A}	active set of inequality constraints
α	angle of attack (Chapter 2 and 4), penalty parameter for primal residual (Chapter 3)
B	design space preconditioner (one-shot)
\tilde{B}	preconditioner for multiplier update (one-shot)
β	penalty parameter for adjoint residual
$c(\cdot)$	partial differential equation (discretized form)
c_d	drag coefficient
c_l	lift coefficient
c_p	pressure coefficient
c_m	moment coefficient
$C(\cdot)$	covariance function

\mathbf{C}	covariance matrix
cov	covariance
γ	line search parameter (Chapter 2 and 3), orthogonality constant (Chapter 4)
Γ_0	representation of Pareto optimal front
d	maximum thickness (airfoil)
\mathbf{d}	search direction
\mathcal{D}	design space
Df	total derivative of f
Δ	difference quadrature rule
δ	finite difference step size
$\delta(\cdot)$	distance function
E	energy
$E(\cdot)$	expectation (function)
$f(\cdot)$	function of interest (general objective function)
\hat{f}	prediction of f
\mathbf{f}^*	ideal point
\mathbf{f}^N	nadir point
$\mathbf{f}^{(j)}$	vector of constraint values for constraint methods
\mathbf{F}	vector of objective functions
F_Z	cummulative distribution function
f_Z	probability density function
\mathcal{F}	power set of Ω
$g(\cdot)$	function of inequality constraints
$G(\cdot)$	fixed point iterator
$\boldsymbol{\eta}$	multiplier for inequality constraints
$h(\cdot)$	function of equality constraints
$H(\cdot)$	Hessian (Chapter 2 and 3), Hermite polynomial (Chapter 4)
$I(\cdot)$	improvement (expected improvement method)
\mathcal{I}	inactive set of inequality constraints (Chapter 2 and 3), general index set (Chapter 4)
k	number of objective functions
K	order of truncation
$L(\cdot)$	Lagrange function (Chapter 2 and 3), Lagrange polynomial (Chapter 4)
$L^a(\cdot)$	augmented Lagrange function
λ	eigenvalue
M	number of sampling points
Ma	Mach number
μ	expected value
$\boldsymbol{\mu}$	multiplier for equality constraint
n	dimension of design space

n_u	scanning resolution for constraint methods
\mathbf{n}	normal direction
N	dimension of random space
$N(\mu, \sigma^2)$	normal distribution
$N(\cdot)$	shifted Lagrange function
p	pressure (Chapter 2 and 3), polynomial order (Chapter 4)
\mathcal{P}	Pareto optimal set
P	probability measure
q	quadrature node
Q	quadrature rule
\mathcal{Q}	set of quadrature nodes
r	number of equality constraints
\mathbf{r}	vector of quasi-Newton method
R	number of quadrature nodes
$R(Z_1, Z_2)$	correlation function
\mathbf{R}	correlation matrix
ρ	density (Chapter 2), contraction rate (Chapter 3)
s	number of inequality constraints
$s(\cdot)$	one shot increment vector (function)
\mathbf{s}	vector of quasi-Newton method
$\hat{\sigma}_f$	standard deviation of prediction of f
S	sparse quadrature rule
σ	standard deviation (Chapter 2 and 4), factor for preconditioner (Chapter 3)
t	temporal variable (Chapter 2), temporal or spatial variable of random field (Chapter 4)
\mathbf{u}	design variable (shape optimization)
v	velocity (Chapter 2) eigenfunction/eigenvector (Chapter 4)
$var(\cdot)$	variance (function)
ϕ	GPC basis function (one-dimensional input)
Φ	GPC basis function (multi-dimensional input)
w	quadrature weight
\mathbf{x}	design variable (general)
ξ	spatial variable
\mathbf{y}	state variable (general)
$\bar{\mathbf{y}}$	adjoint state variable
$\dot{\mathbf{y}}$	tangent-linear state variable
ψ	random field
Z	random variable
$Z(\mathbf{y}, \mathbf{u})$	projection matrix (reduced Hessian)
ω	possible outcome
Ω	set of possible outcomes

1. Introduction

Multi-objective optimization and robust design are two well-established fields of research. Especially in engineering applications, it is important to optimize for different conflicting criteria like, for example, cost and quality aspects. This is realized using multi-objective optimization (MOO) strategies to find trade-off designs for different criteria. Another significant step towards realistic design is to take into account uncertainties for finding robust optimal solutions. Robust design aims at finding solutions that are optimal and robust with respect to perturbations.

However, multi-objective optimization and robust design are often treated as separate tasks when it comes to engineering design. Often, multi-objective aspects are considered in a conceptual design phase, and after that, the performance and robustness of a design are analyzed in a detailed design phase. If robust design is performed, it usually only takes into account a single criterion. The reason for this is that performing multi-objective optimization or robust design strategies separately already increases the computational costs, especially if underlying partial differential equations (PDEs) need to be solved in each design step. Furthermore, the definition of suitable measures for robust optimality is not straightforward in a multi-objective context.

Therefore, this thesis aims at presenting measures and corresponding problem formulations for robust design in a multi-objective context and optimization strategies that can be used to solve the robust design problem efficiently. The need for new and efficient strategies for robust design in a multi-objective context will be further motivated in Section 1.1. The section also gives a short literature overview. This is done more thoroughly for each topic in the main part of the thesis. In Section 1.2, research goals are formulated, the general methodologies are presented, and contributions of the thesis are summarized. Finally, an outline of the main part of the thesis is given in Section 1.3.

1.1. Motivation

The aim of the presented research is the definition of robustness measures and optimization strategies in the context of robust multi-objective design with an underlying PDE constraint. This comprises different areas of research, namely

- strategies for multi-objective optimization,
- strategies for PDE-constrained optimization, and
- strategies for robust design.

1. Introduction

This section presents an overview of the state-of-the-art of research for the different topics and possible shortcomings. It serves to demonstrate the need for new strategies in the respective areas of research.

Multi-Objective Optimization Conflicting criteria are present in everyday life. In engineering design, one has to find trade-offs between product quality and manufacturing cost. Nowadays, especially when it comes to rethinking in environmental sustainability and the associated demand for more sophisticated solutions, the complexity of problems in engineering design increases. Solutions need to be found in highly-nonlinear environments, often involving competing objectives. The aim is to find a design that fulfills various requirements. The requirements can be either integrated as multiple objective functions or, if the designer can formulate appropriate targets, as constraints of an optimization problem. In aerodynamic shape optimization, for example, the aim is to increase the performance by maximizing the lift of an airfoil and minimizing the drag, which implicitly decreases fuel consumption. Usually, manufacturing constraints, like the maximum space allowed for a component or the maximum amount of material, are prescribed by given bounds. Especially in the presence of multiple objectives, trial-and-error design based on intuition or experience in a specific field becomes impracticable. Thus, the use of numerical optimization algorithms is inevitable.

If the criteria are conflicting, there does not exist a single optimal solution to the multi-objective optimization problem. This means that a design that is optimal for one criterion is not necessarily optimal for the other criteria. Instead, the aim is to find a single trade-off solution or a set of trade-off solutions, which are offered to a decision-maker. The concept that is used for relating vectors of objective functions is the concept of Pareto optimality. Given a k -dimensional vector of objective functions, a feasible design \mathbf{x} is Pareto optimal if it is non-dominated, i.e., there does not exist any feasible design $\tilde{\mathbf{x}}$ such that $f_i(\tilde{\mathbf{x}}) \leq f_i(\mathbf{x})$ for every objective function f_i with $i \in \{1, \dots, k\}$ and there exists at least one strict inequality. The image of the set of Pareto optimal solutions in objective space is referred to as the Pareto optimal front.

Methods for multi-objective optimization use different strategies to find a set of Pareto optimal solutions. The aim is to find a set of representative solutions that are uniformly distributed on the Pareto optimal front. One may distinguish between direct Pareto approaches and scalarization approaches for multi-objective optimization.

Direct Pareto approaches try to find several Pareto optimal points at once. This can, for example, be realized using evolutionary approaches. Research in the field of multi-objective evolutionary methods is an ongoing topic. It started with Schaffer's Vector Evaluated Genetic Algorithm (VEGA, [234]) and the ideas of Goldberg [83]. The most popular algorithm used is the Non-Dominated Sorting Genetic Algorithm (NSGA-II, [49]). Direct Pareto approaches based on evolutionary strategies are popular due to their easy implementation and integration in existing frameworks. A drawback of evolutionary approaches is that there is no proof of convergence and, accordingly, there are no objective convergence criteria like a vanishing gradient as a necessary condition for optimality. Thus, in general, it is hard to define a suitable stopping criterion for

evolutionary algorithms. Another disadvantage is the high number of function evaluations reached to find an agreeable set of solutions, which is a significant problem if the computational costs for a single evaluation are already high.

In scalarization approaches, the problem is transformed into several single-objective optimization problems, which can be solved using efficient strategies for single-objective optimization. The most intuitive approach in this context is to optimize a weighted sum of objective functions with alternating weights. This method is referred to as the weighted sum method. A significant disadvantage is that the method can only find Pareto optimal points on convex parts of the Pareto optimal front. Additionally, the points on the Pareto optimal front are not necessarily uniformly distributed when choosing uniformly distributed weights [45]. This can be circumvented when using other scalarization approaches, for example, constraint methods like the epsilon-constraint method [177]. The idea of the epsilon-constraint method is to optimize one objective function while imposing inequality constraints on the remaining competing objective functions. The constraints, as well as the objective function, can be varied to find different Pareto optimal solutions. The advantage of using constraint methods over the weighted sum technique is the ability to find more evenly distributed Pareto points and analyze selected regions of the Pareto front by selectively choosing appropriate constraints.

A major advantage of scalarization approaches is the opportunity to make use of known strategies for single-objective optimization. Specific challenges for single-objective optimization algorithms in the context of multi-objective optimization can be formulated. These challenges are:

- Only the unique global solution to the single-objective optimization problem is a Pareto optimal solution. Therefore, it is important to use a method with a high chance of finding a global optimum.
- Several single-objective optimization problems have to be solved for finding solutions to the multi-objective optimization problem. Since the computational costs for finding a representative set of solutions increase exponentially with the number of objective functions, efficient optimization strategies are required.
- The optimization method shall work reliable and provide solutions for each of the single-objective optimization problems. One may use the notion of robustness in this context. However, the robustness of the methodology itself shall not be mixed up with the robustness of a solution.

These challenges often prevent the use of scalarization approaches. Unfortunately, possible remedies for the challenges mentioned above are contradicting each other:

- Global optimization is still an open research problem. There is no deterministic global optimization method with guaranteed convergence. The only reliable strategy for finding a global optimum is an exhaustive search of the design space. The high computational costs render this brute-force strategy prohibitive. The same

1. Introduction

applies to heuristic strategies for single-objective optimization like genetic algorithms [119]. They usually have a higher chance of finding a global optimum, but they share the same problems as heuristic methods in the context of multi-objective optimization.

- Opposed to heuristic optimization methods, gradient-based strategies are computationally efficient. Additionally, reasonable convergence criteria exist. Newton's method, which makes use of the Hessian, shows local quadratic convergence. Also, methods only based on first-order derivatives can achieve fast convergence, especially when making use of Hessian approximations. One-shot optimization strategies have the potential to decrease the computational effort by one further order of magnitude, and will be introduced in the next paragraph. However, gradient-based strategies can only guarantee to find local optimal solutions. Additionally, depending on the starting point of the algorithm, they can get easily stuck in local optima. Since they converge locally, the chance of finding the global optimum can only be increased by choosing the starting point in the vicinity of the global optimum, which is normally not known.
- Methods that are not making use of derivative information are, in general, more reliable and easier to implement in existing frameworks. When making use of inexact derivatives, obtained, for example, from a finite difference approximation, a gradient-based strategy may not show the expected convergence behavior or, in the worst case, fail to converge. Thus, gradient-based optimization strategies require the provision of accurate derivative information, which can be obtained from algorithmic differentiation.

The fact that the challenges cannot be met using either global search strategies or gradient-based methods motivates the use of hybrid optimization approaches that combine both strategies to increase the chance of finding a global optimum. The idea is to profit from the advantages of both strategies.

To summarize, the goal for multi-objective optimization is to revive constraint methods by using hybrid optimization strategies. Especially, the opportunity to use efficient methods using derivatives obtained with the help of algorithmic differentiation, e.g., the one-shot method, can significantly decrease the computational costs for the single-objective optimization problems arising from constraint methods for MOO. This makes a Pareto front exploration feasible when the determination of the objective function involves the computationally expensive solution of a system of partial differential equations. In the following, strategies for optimization with PDE constraints are presented, focusing on adjoint methods and one-shot methods.

PDE-Constrained Optimization Many tasks in the field of engineering involve optimization problems or optimal control problems with underlying partial differential equations. In fluid dynamics, this can, for example, be the Euler or the Reynolds-averaged Navier-Stokes equations as state equations. They serve as constraints to the given problem. Usually, these PDE constraints cannot be solved analytically but have to be solved

numerically. Calculating the solution to the PDEs is often, as in the case of fluid dynamics, expensive considering computing time and memory.

Because of the high computational effort when solving PDEs, it is costly to use classical optimization methods. An additional cause for high computational costs is the determination of the gradient of the objective functions and the gradient of the constraint functions with respect to the design variables when using gradient-based optimization methods. Typically the number of design variables is high in relation to the number of objective and constraint functions. This is, for example, the case for shape optimization problems.

Finite differences are a common and straightforward approach for obtaining the needed derivatives. However, the disadvantages are that these formulas only provide an approximation of the derivatives, and a time-consuming study of appropriate step sizes is needed. The step size needs to be re-evaluated for each new problem and each new optimization parameter, which is a constant overhead required by finite difference formulas. When using finite differences, the computational complexity increases linearly with the number of design variables. To solve this problem, one often makes use of adjoint approaches, which guarantee an evaluation of derivatives that is independent of the number of design variables. The main idea of the adjoint approach is to solve the so-called adjoint equation that can be derived by setting up the optimality conditions for the constrained optimization problem. It was first derived and applied by Pironneau [221] in 1974 in the context of control theory in fluid dynamics. Jameson [135] was the first one to apply the adjoint approach to a shape optimization problem of an airfoil subject to a potential equation. Later, he also applied it to problems subject to the Euler equations and the Reynolds-averaged Navier-Stokes equations. There exist two classes of adjoint approaches. In the continuous adjoint approach, the adjoint equation is derived from the flow equation and is then discretized. In contrast to that, in the discrete approach, the problem is discretized first, and the variations are taken afterwards. Jameson [135] made use of the continuous adjoint approach, whereas, e.g., Giles [81] employed the discrete adjoint method.

The discrete adjoint approach can be implemented in a robust and semi-automatic manner with the help of the reverse mode of algorithmic differentiation (AD, [93]). Algorithmic differentiation is a technique to differentiate a function that is implemented as an algorithm in a computer program, like a numerical flow solver and can be decomposed into elementary operations like $+$, \sin , etc. The application of AD to an algorithm augments it by its derivative by linking the derivatives of the elementary operations using the chain rule of differentiation. This can be either done using source code transformation or using operator overloading. Source code transformation interprets the original source code and produces a new source code augmented by the derivatives for each operation. For modern programming languages like C++ that enable operator overloading, each operation can be overloaded, such that a specific internal representation of the operation that serves to calculate the respective derivative is generated. The advantage is that all code changes can be automatically differentiated with minimal maintenance overhead. There exist two different modes of AD. The forward mode provides derivatives for each design parameter separately. Thus, as it is the case for finite difference approximations,

1. Introduction

the computational costs increase linearly with the number of design parameters. In contrast, the highly efficient reverse mode provides the full gradient of an objective function in a single run.

Besides the advantage of computational cost, the use of derivatives obtained from algorithmic differentiation enhances the reliability of gradient-based optimization since derivatives obtained from AD are machine-accurate. The success of gradient-based optimization methods depends on the availability of accurate derivatives. Quasi-Newton methods need accurate gradients to achieve a high-quality Hessian approximation and, as a result, locally super-linear convergence. Furthermore, methods for constrained optimization problems like projection methods rely on the accuracy of the derivatives of the constraint functions.

The reverse mode of algorithmic differentiation is connected to the adjoint equation in the discrete adjoint method. The differentiation of the fixed-point iterator of a scheme for solving the state solution of a PDE constraint results in a solution scheme for the adjoint solution. The discrete adjoint solver based on this strategy inherits the contraction properties of the solver for the state equation. Advanced techniques in the development of algorithmic differentiation facilitate the extension of the fixed-point solver when it is modified, as well as the incorporation of different objective functions and constraints for optimization. Furthermore, the fixed-point iterator can be treated in a black-box manner, such that methods involving the adjoint can be developed independently of its implementation. This enables the semi-automatic transition from simulation tools to optimization (see [70]). Especially, the availability of a robust discrete adjoint solver enables the use of advanced optimization techniques like the one-shot approach.

In classical hierarchical optimization approaches, the feasibility of the primal equation and of the adjoint equation are recovered in each optimization step. If the primal equation has a rather slow linear convergence, as it is the case for calculations of a flow solution, one-shot strategies can be advantageous. The one-shot approach needs only the computational costs of a small multiple of flow solutions for the optimization which is referred to as bounded retardation. The primal flow equations, the adjoint equations, and the design equation are iterated simultaneously in one optimization procedure in a coupled fixed-point iteration. One-shot strategies can be distinguished by the order of setting up the optimization problem and differentiation. The one-shot optimization problem can be set up to be discretized afterwards [113, 116, 114], thus using the ideas of the continuous adjoint approach. The solution procedure for this method can be interpreted as an inexact reduced sequential quadratic programming approach. On the other hand, the one-shot strategy can be formulated for the discretized primal equation [89, 72, 102, 103, 71]. This approach is referred to as the single-step one-shot approach as, opposite to the other approach, each update is based on the information of a single point at the old iterate. For the single-step one-shot approach convergence to a stationary point of the optimization problem can be guaranteed for a carefully chosen preconditioner based on a doubly augmented Lagrangian, which is used for the iteration in design space. The prerequisite for this is that the fixed-point iteration of the underlying primal equation is contractive. A review of one-shot approaches can be found in [24].

The coordination of primal, adjoint, and design iterations is well established when only the state equation is considered as an equality constraint. If additional constraints are considered, the usual strategy is to extend the objective function by a penalty term with a fixed penalty. However, the convergence properties of the single-step one-shot method cannot be directly adopted. In [239], an additional equality constraint is introduced in the context of an inexact reduced sequential quadratic programming approach, and in [23], the author extends the one-shot framework for additional constraints that allow the introduction of corresponding additional state variables. Constraint methods for MOO rely on a reliable optimization strategy for optimization with additional constraints. This motivates to extend the single-step one-shot approach and its theoretical results to the inclusion of additional constraints.

Having the opportunity of an efficient optimization strategy for MOO problems, the idea of introducing an additional dimension, namely random variables due to uncertainties in the underlying problem, is straightforward and will be discussed in the next paragraph.

Uncertainty Quantification and Robust Design Sources of uncertainty can be found in a lot of real-world scenarios as most processes are not deterministic. Instead, environmental factors may, for example, cause variations in the parameters or the boundary conditions of a model describing the problem. Opposed to epistemic uncertainties, which result from modeling, these types of uncertainties are inherent in the problem and cannot be reduced by further improving the model and are referred to as aleatoric uncertainties. In aerodynamic shape optimization, e.g., one has to consider aleatoric uncertainties in the operational conditions for flight. They may, for example, arise due to turbulence or uncertainties in the geometry caused by manufacturing tolerances or temporary factors like icing. Thus, realistic engineering design often involves the consideration of uncertainties affecting the system. If the effect of uncertainties is neglected in the design process, the performance of an optimal design that was optimized under fixed conditions may be significantly decreased for a small variation in these conditions. In the worst case of events, this may even lead to a failure with significant effects.

When introducing uncertainties in the design process, the first task is the modeling of uncertainties and the propagation of uncertainties through the system under consideration. Research in this area has been conducted in the last decades and is summarized under the term uncertainty quantification. Uncertainties caused, for example, by environmental factors, as described above, can be represented by introducing random variables with known distributions in the system of PDEs. Some uncertainties have to be modeled as random processes leading to an infinite-dimensional probability space. A Karhunen-Loève (KL, [141, 170]) expansion can be used to approximate the infinite-dimensional space with a finite number of random variables.

There exist different strategies for the propagation of uncertainties. The choice of strategies mainly depends on the number of uncertain variables. Monte Carlo approaches make use of random sampling for approximating quantities of interest like the expected value or the variance. While the number of needed samples is very high, these methods

1. Introduction

do not depend on the number of uncertain parameters.

For moderate dimensions of the random space (usually up to a maximum of ten uncertain parameters), stochastic spectral methods, which are also referred to as polynomial chaos (PC) approaches, are significantly more efficient in comparison to Monte Carlo approaches. The idea of stochastic spectral methods is to expand input and output random variables in terms of orthogonal polynomials depending on the probability density function of the random input variables. In the stochastic Galerkin method (see e.g. [292]), the expansion coefficients are found by projecting the residual of the system of PDEs onto the space of orthogonal polynomials. As a result, the approach is intrusive since a new system of equations needs to be solved that requires a different analysis and numerical treatment.

This can be avoided when using non-intrusive PC approaches. In collocation methods, the expansion coefficients are found based on a given number of samples for which the output variables are evaluated. Stochastic collocation [179], for example, makes use of interpolation for finding the expansion coefficients. In contrast, in the non-intrusive polynomial chaos approach [289] the expansion coefficients are found by projecting the residual of the expansion of the output onto the space of orthogonal polynomials. The resulting integral for the coefficients is then solved, for example, by using a quadrature method. The computational effort required for non-intrusive strategies may increase exponentially with the dimension of the random space due to the convergence properties of interpolation or quadrature methods. This effect is referred to as the curse of dimensionality. For intrusive methods, the same effect can be observed in a weaker form. The use of sparse grids, e.g., dimension-adaptive sparse grids for quadrature [77], can mitigate the curse of dimensionality.

The effects of uncertainties may be included in the optimization, yielding robust design strategies. The aim of robust design is to find solutions that are robust and optimal, also referred to as robust optimal, under the considerations of uncertainties. This requires the definition of robustness as a modeling aspect and its integration in the formulation of the optimization problem to find robust optimal designs. First ideas for robust design were formulated in the middle of the last century by Taguchi [265], who proposed to describe robustness with a quality loss function. The average quality loss inspired the definition of robustness based on expectation and variance as quantities of interest, which is still popular today. The quantities of interest can be obtained with the help of the strategies for uncertainty quantification described above.

In general, including uncertainties in the optimization approach will naturally increase the computational effort. However, the use of efficient optimization techniques based on adjoint methods makes robust design a feasible option. Schillings et al. [235] successfully demonstrate this for robust aerodynamic design. The key for efficiency is the utilization of a one-shot optimization method for solving a semi-infinite robust design formulation based on the expected value of the drag coefficient. A non-intrusive polynomial chaos approach with the help of dimension-adaptive sparse grid quadrature is used for uncertainty propagation. This motivates an extension of these strategies to robust design in a multi-objective context.

There can be seen a growing interest in finding robust optimal solutions when different

conflicting objective functions are considered in an uncertain environment. Variations in parameters and boundary conditions have a combined effect on all objective functions. The effect expresses itself in a probability distribution in objective space. Similarly to the case of MOO without considering uncertainties, there does not necessarily exist a single robust optimal solution. Multi-objective robust design strategies were first formulated in an evolutionary context. Most of the research in this field is restricted to direct Pareto approaches. The application in the context of evolutionary multi-objective optimization enables the definition of a probability of dominance [269, 124], which can be used for ranking members of a population, or a dominance relation based on worst case analyses [10]. A common strategy is to replace the original fitness value of the evolutionary strategy by an expected fitness value. Deb and Gupta [47] introduced different multi-objective robust design formulations using a mean effective objective function as a fitness function. Inspired by measures for robustness in single-objective robust design, the formulations can be extended to the use of the expected value and the variance [197]. These formulations can also be used together with constraint methods for MOO.

The definition of robustness in MOO using the expectation and variance of the individual objective function does not necessarily reflect the combined effect of uncertainties in the form of a distribution in objective space. The measures used for dominance relations under uncertainties, in general, assume that objective functions are independently distributed. Furthermore, these measures cannot be directly used in scalarization approaches for MOO. This motivates the definition of a measure for robust optimality that accounts for the combined effects of uncertainty on all objective functions.

1.2. Research Goals and Contributions

This section defines the research goals for the different areas of research that are of interest for enabling multi-objective robust design and presents scientific contributions connected to this thesis.

One may distinguish between the different research goals that were motivated in the previous section:

- Constraint methods are an interesting option when it comes to multi-objective optimization with expensive PDE constraints. However, they require the use of a strategy with a high chance to find a globally optimal solution to the constrained single-objective optimization problems. Therefore, one goal of this work is to demonstrate the success of constraint methods when using hybrid optimization strategies.
- The one-shot approach is an efficient strategy for PDE-constrained optimization. A further goal of this work is to integrate the one-shot strategy into a hybrid strategy for multi-objective optimization. This requires the successful handling of additional constraint functions. Consequently, the aim is to extend the framework of the one-shot strategy to additional constraints and present corresponding theoretical and numerical results.

1. Introduction

- The incorporation of uncertainties in a design process with multiple objectives is not straightforward. Especially since existing strategies do not account for the effect of uncertainty in multi-objective space, another aim of this work is to introduce a suitable robustness measure and corresponding strategies for robust design with multiple objectives. The strategies of the previous key points shall help to enable the application of the robust design formulations to PDE-constrained problems.

Next to the analysis and implementation of the proposed strategies, the aim is to show the application to shape optimization in fluid flow. A test case based on two-dimensional aerodynamic shape optimization shall serve to demonstrate the research aspects. In this context, the multi-objective task is to minimize the drag and maximize the lift. The open-source multi-physics simulation and design framework SU2 [216] is employed for the implementation. SU2 is designed for supporting shape optimization problems in fluid flow by providing the needed solvers and all steps in the design chain. It implements the solvers for the state equations describing the fluid flow, as well as discrete adjoint capabilities [4] supported by algorithmic differentiation with the help of the AD tool CoDiPack [231].

The following paragraphs summarize the scientific contributions of the present work. Furthermore, published results are brought in context.

Constraint Methods for Multi-Objective Optimization using Hybrid Strategies Contributions of the present work in the field of multi-objective optimization involve the rediscovery of constraint methods for multi-objective optimization using highly efficient single-objective optimization algorithms, for example, the one-shot approach. Furthermore, hybrid optimization algorithms based on surrogate models are used to support the multi-objective optimization process by increasing the chance to find global optima and, as a result, Pareto optimal solutions. The general idea is to use a global search method on a surrogate model to avoid the high computational costs needed by global search methods to explore the design space. An efficient gradient-based optimization method is used afterwards for a local search on the original expensive PDE-constrained optimization problem.

In the context of own previous work [158], different multi-objective optimization algorithms were explored. Constraint methods for multi-objective optimization, namely the equality constraint method and the epsilon-constraint method, were identified as valuable methods for generating a representative set of Pareto optimal points. A hybrid algorithm was proposed for solving the constrained single-objective optimization problems to increase the success of the overall optimization process. The hybrid method couples a derivative-free algorithm with a high probability for global solutions and a local line search method to combine the advantages of both approaches. In a first step, a genetic algorithm was applied for a search on a Kriging metamodel of the expensive objective functions. Afterwards, the found solution was used as a starting point for a quasi-Newton method. The overall methodology was applied in an industrial context for improving exhaust after-treatment based on selective catalytic reduction. The two

objective functions under consideration were the pressure drop and the flow uniformity index. The proposed method, as well as the results, were published in [158].

In the present work, the idea described above is extended by further developments of the hybrid optimization strategy such that it is improved for single-objective optimization problems with additional constraints. The global search conducted on the Kriging surrogate model (see [213]) is tailored for constrained optimization problems by using the constraint-weighted expected improvement method. The emphasis for implementation was put on interfaces for objective and constraint functions to allow the integration of the Kriging approach in the multi-objective optimization strategies. The proposed multi-objective optimization methodology was integrated into the framework of SU2 and successfully applied to shape optimization problems in fluid mechanics. In first applications, an interior point method was used as a strategy for gradient-based optimization. To further increase the overall efficiency, suitable one-shot optimization strategies were developed. They will be the focus of the following paragraph.

Theoretical and Numerical Analysis of the One-Shot Method with Additional Constraints Scientific contributions in the field of efficient strategies for PDE-constrained optimization are based on the extension of the single-step one-shot method to include additional equality constraints. This involves the theoretical and numerical analysis of the approach and its application to multi-objective shape optimization problems.

Research in this field started with the work of Walther et al. [278], who extended the single-step one-shot strategy [72, 102, 103] to additional equality constraints. The basic idea of the extension is the introduction of a modified Lagrange function. The inclusion of additional equality constraints complicates the analysis, but many properties of the approach by Hamdi and Griewank are inherited. The work initiated a collaboration in refining the results on convergence properties and developing practical solutions for implementation. A contribution of this thesis is the implementation of the single-step one-shot strategy and its extension to an additional equality constraint in SU2 (see also [155]).

The modification of the Lagrange function to include additional equality constraints implies the introduction of an additional vector of constraint multipliers. The choice of a suitable preconditioner for the design update as well as for the constraint multiplier update is essential to achieve bounded retardation. Refined results for the theoretical analysis are published in [280], providing simpler conditions for both preconditioners. Furthermore, a suitable preconditioner for the constraint multiplier update is proposed, thus simplifying the overall implementation. The extension to inequality constraints with the help of strategies for bound projection is discussed to enable the use of constraint-based MOO strategies, leading to constrained optimization problems with inequality constraints.

Finally, the one-shot strategy with additional equality constraints is firstly applied in a multi-objective context [154]. The multi-objective optimization problem is formulated for aerodynamic shape optimization and treated using the equality constraint method. The constrained single-objective optimization problems arising from the equality con-

1. Introduction

straint method can thus be solved efficiently. Furthermore, the use of second-order derivatives in the one-shot approach is investigated. The results show the successful transition from simulation to optimization in a semi-automatic fashion.

Robustness Measures for Robust Design with Multiple Objectives Contributions in the field of robust design are the formulation and analysis of strategies for robust design in a multi-objective context. The general idea is to measure robust optimality based on losses in multi-objective space. The losses are defined using the general concept of Pareto optimality. Expected losses are integrated in the multi-objective optimization problem with the help of two different strategies to find robust optimal designs. The uncertainty is propagated with the help of the non-intrusive polynomial chaos approach.

Multi-objective robust design using the expectation of the respective objective functions for measuring robustness is performed in [156]. The expectation-based robustness measure is a common approach to robust design with multiple objectives. It is inspired by robust design strategies in a single-objective context. The epsilon-constraint method supported by the hybrid optimization strategy is employed for solving the multi-objective robust design problem. The robust design task is applied in the context of aerodynamic shape optimization and implemented in SU2. Uncertainties in the operational conditions and the geometry are considered separately and propagated with the help of a non-intrusive polynomial chaos approach. The random process that describes geometrical perturbations is approximated using a Karhunen-Loève expansion. The integration in random space is performed using dimension-adaptive sparse grids to increase the overall efficiency. The work extends the ideas of Schillings et al. [235] to multi-objective optimization using the epsilon-constraint method.

In [157], expected losses in objective space are introduced as a new measure for finding robust optimal designs. Two different formulations are presented. They use different definitions of losses in objective space. The first formulation is a two-phase approach in which the losses are defined with regards to an approximation of the Pareto optimal front without considering uncertainties. The Pareto optimal points for approximation are found in the first phase. The expected losses are described with the help of a signed distance function in objective space. Afterwards, a new optimization problem that includes a constraint for the expected losses is solved. In the direct approach, on the other hand, the Pareto optimal front in a deterministic setting is unknown. Therefore, the losses are described based on the distance to the deterministic outcome of the design under consideration. Both approaches are applied for robust airfoil design with multiple objectives. When compared to the results of standard measures, the desired effect can be observed. As a result, perturbations for the found designs are expected to stay close to the Pareto optimal front.

1.3. Composition of Thesis

In line with the composition of the introductory sections, the thesis is split into three thematic chapters. Each of the chapters serves to introduce needed notation and to give details on existing strategies for the respective topic. Then new strategies are explained in detail. Results for the application of the proposed methods to aerodynamic shape optimization are shown at the end of each chapter. To summarize these chapters:

- In Chapter 2, the hybrid strategy for multi-objective optimization is presented. Fundamentals of optimization in a single-objective and a multi-objective setting are introduced. Several methods for multi-objective optimization are reviewed, focusing on constraint methods. Finally, a hybrid optimization strategy is presented, followed by results for shape optimization in fluid flow.
- Chapter 3 focuses on the development and application of the one-shot method with additional equality constraints. In the beginning, general concepts for PDE-constrained optimization are described with an emphasis on adjoint methods for sensitivity computation. The one-shot approach and its extension to additional equality constraints are presented along with important results on convergence properties. Finally, the one-shot method with additional equality constraints is applied to multi-objective aerodynamic shape optimization problems.
- Strategies for robust design in a multi-objective context are presented in Chapter 4. After an introduction of notations in probability theory, methods for uncertainty quantification and propagation are presented. Robust design formulations are discussed in a single-objective and a multi-objective context. A new robustness measure is introduced for robust design with multiple objectives and applied in the context of aerodynamic shape optimization under uncertainty.

Finally, a conclusion and an outlook are given in Chapter 5. The appendix comprises additional information and results for some research aspects.

2. Multi-Objective Optimization

Conflicting objectives are a natural phenomenon, which can be found in various fields of application. We have to make decisions based on costs versus quality in everyday life. Similarly, a lot of research in engineering involves the simultaneous optimization of different competing objectives. In fluid dynamics, for example, fluid mechanical performances to be optimized may be represented by different objectives. In aerodynamic design, the aim is to maximize the lift of an airfoil while minimizing the drag for fuel-efficiency. When designing a system for internal flow, the improvement of specific flow properties often leads to a pressure loss. One refers to *multi-objective optimization* (MOO) for optimization including different competing objectives.¹

Typically, there does not exist a unique solution to the multi-objective optimization problem. Instead, there exists a set of compromise solutions to the problem. The reason for that is that the objectives are partly conflicting with each other. A well-known concept for comparing solutions is the notion of Pareto optimality. Multi-objective optimization algorithms aim to find a representative subset of solutions fulfilling the conditions for Pareto optimality. Such a set of solutions can be offered to a *decision-maker* to choose a single optimal design. A decision-maker is a person with enough insight into the underlying application (regardless of expertise in mathematical modeling or optimization) to decide for a given set of solutions according to preferences and possible constraints for realization. When provided a representative set of the Pareto optimal solutions, the decision-maker may also gather more information on the problem and possible trade-offs. The present work does not focus on the decision-making process, but solely on methodologies for finding a representative set of solutions.

Note that conflicting objectives can also arise from disciplines as a special situation of multi-disciplinary optimization. Multi-disciplinary optimization problems comprise any kind of optimization problem where different disciplines are involved, e.g., fluid mechanics and structural mechanics in engineering design. If multiple objectives are considered, standard multi-objective optimization algorithms may be utilized. However, in this work, one focuses on problems involving a single discipline.

In the field of multi-objective optimization, the contribution of the present work involves the rediscovery of constraint methods for multi-objective optimization using highly efficient single-objective optimization algorithms (for example one-shot approaches). Furthermore, hybrid optimization algorithms based on surrogate models are used to support the multi-objective optimization process.

In this chapter, the concept of Pareto optimality is introduced together with conditions for Pareto optimality (Section 2.1). Different methods for multi-objective optimization

¹Other notions used in this context are *multi-criteria optimization* or *vector optimization*.

2. Multi-Objective Optimization

are presented and compared in Section 2.2 with a strong focus on constraint methods for multi-objective optimization. The present work uses the equality constraint method (Section 2.2.3), the epsilon-constraint method (Section 2.2.4) and the normal constraint method (Section 2.2.5) for solving MOO problems. The underlying algorithm for solving the constrained single-objective optimization problem is a hybrid algorithm, coupling a heuristic search method with a high probability for finding global solutions and a computationally efficient gradient-based search method to combine the advantages of both algorithms. The hybrid optimization strategy is presented in Section 2.3. Finally, results for the application to aerodynamic shape optimization are presented in Section 2.4.

2.1. Fundamentals and Definitions

The general constrained minimization problem for a vector of objective functions $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^\top$ can be posed as

$$\min_{\mathbf{x} \in \mathcal{D}} \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (2.1)$$

with the vector of design variables $\mathbf{x} \in \mathbb{R}^n$ and the design space

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = \mathbf{0}, g(\mathbf{x}) \leq \mathbf{0}\},$$

which is a subset of the finite dimensional Euclidean space.²

The function $\mathbf{F}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is vector-valued for $k \geq 2$ comprising k multiple objectives. One may then refer to problem (2.1) as a multi-objective optimization (MOO) problem and the minimization is to be understood component-wise for each objective function. In the following, also the shorter notation $\min_{\mathbf{x} \in \mathcal{D}} \mathbf{F}(\mathbf{x})$ is used. The functions $h : \mathbb{R}^n \rightarrow \mathbb{R}^r$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^s$ describe the equality and inequality constraints that define the design space. The design space of shape optimization problems in fluid dynamics is, for example, limited by the designs that provide reasonable flow solutions fulfilling the equations of fluid dynamics. It is also constrained by the dimensions of the installation space.

The design space \mathcal{D} is often referred to as a *feasible region* with a *feasible design* $\mathbf{x} \in \mathcal{D}$. In the following, one assumes that the objective functions and constraint functions are at least twice continuously differentiable and the set $\mathcal{L} := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{F}(\mathbf{x}) \leq \mathbf{l}\} \cap \mathcal{D}$ is closed, bounded and non-empty for a real vector $\mathbf{l} \in \mathbb{R}^n$ to guarantee the existence of a minimum. Furthermore, in the following definitions the *active set* of inequality constraints is defined as

$$\mathcal{A}(\mathbf{x}) := \{j \in \{1, \dots, s\} \mid g_j(\mathbf{x}) = 0\}.$$

The *inactive set* is defined as

$$\mathcal{I}(\mathbf{x}) := \{j \in \{1, \dots, s\} \mid g_j(\mathbf{x}) < 0\}.$$

²It is referred only to minimization problems. Every maximization problem $\max(f(\mathbf{x}))$ can be transformed into a minimization problem $-\min(-f(\mathbf{x}))$.

This section serves to define the notion of Pareto optimality. Since single-objective optimization strategies are used in constraint methods for MOO, definitions in this context can be found in Section 2.1.1 before turning to the concept of Pareto optimality in Section 2.1.2.

2.1.1. Single-Objective Optimization

Before introducing concepts of multi-objective optimization the focus will be on single objective optimization problems, assuming $k = 1$ in this section. The scalar version of the minimization problem (2.1) to be considered is

$$\min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}). \quad (2.2)$$

The reason for introducing concepts of single objective optimization is the fact that scalarization methods for solving multi-objective optimization problems make use of single objective optimization.

When dealing with solutions to minimization problems, it is important to introduce the concept of global and local minima, as well as necessary and sufficient conditions for optimality.

Definition 1 (global minimum). *A point $\mathbf{x}^* \in \mathbb{R}^n$ is a strict global minimum of f if $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x} \neq \mathbf{x}^*$. If $\mathbf{x}^* \in \mathcal{D}$, then it is denoted as a global solution to (2.2).*

Definition 2 (local minimum). *Let $\mathbf{x}^* \in \mathbb{R}^n$ and $U(\mathbf{x}^*) \subset \mathbb{R}^n$ be an open neighborhood of \mathbf{x}^* , then \mathbf{x}^* is a strict local minimum of f if $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x} \in U(\mathbf{x}^*)$, $\mathbf{x} \neq \mathbf{x}^*$. If $\mathbf{x}^* \in \mathcal{D}$ and $\mathbf{x} \in U(\mathbf{x}^*) \cap \mathcal{D}$, then \mathbf{x}^* is the local solution to (2.2).*

A weak minimum is defined in the same way without using strict inequalities.

The necessary and sufficient condition for optimality can be expressed in terms of a *Lagrange function* or *Lagrangian*.

Definition 3 (Lagrange function). *The function*

$$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\eta}) = f(\mathbf{x}) + \boldsymbol{\mu}^\top h(\mathbf{x}) + \boldsymbol{\eta}^\top g(\mathbf{x}) \quad (2.3)$$

is the Lagrange function associated to (2.2) with Lagrange multipliers $\boldsymbol{\mu} \in \mathbb{R}^r$ and $\boldsymbol{\eta} \in \mathbb{R}^s$.

There exist several regularity conditions, which are referred to as *constraint qualifications*. The following regularity definition by Kuhn and Tucker [153] is used in the context of necessary and sufficient conditions for optimality and can be found similarly in [190, p.38] and [31, p.51].

Definition 4 (Kuhn-Tucker constraint qualification (KTCQ)). *Let h, g be continuously differentiable at \mathbf{x}^* . Then the KTCQ is satisfied at \mathbf{x}^* if for any $\mathbf{d} \in \mathbb{R}^n$ that fulfills $\nabla h_j(\mathbf{x}^*)^\top \mathbf{d} = 0$ for all $j \in \{1, \dots, r\}$ and $\nabla g_l(\mathbf{x}^*)^\top \mathbf{d} \leq 0$ for all $l \in \mathcal{A}(\mathbf{x}^*)$, there exists a function $a : [0, 1] \rightarrow \mathbb{R}^n$ that is continuously differentiable at 0 and $\alpha > 0 \in \mathbb{R}$ such that*

$$a(0) = \mathbf{x}^*, \quad g(a(t)) = \mathbf{0} \text{ and } h(a(t)) \leq \mathbf{0} \quad \forall 0 \leq t \leq 1, \text{ and } \quad a'(0) = \alpha \mathbf{d}.$$

2. Multi-Objective Optimization

The KTCQ is implied by the condition that the constraints should be satisfied at \mathbf{x}^* and that the gradients of the constraints $\nabla h_j(\mathbf{x}^*)$ with $j \in \{1, \dots, r\}$ and $\nabla g_l(\mathbf{x}^*)$ with $l \in \mathcal{A}(\mathbf{x}^*)$ should be linearly independent [31, p.51,56]. This condition is also called the LICQ (Linear Independence Constraint Qualification) condition. It is more practical to use and is required for second-order necessary conditions.

Theorem 1 (Karush-Kuhn-Tucker first-order necessary condition for optimality). *Let $\mathbf{x}^* \in \mathcal{D}$ be a local minimum. Let $f(\mathbf{x}), h(\mathbf{x}), g(\mathbf{x})$ be continuously differentiable at \mathbf{x}^* and let the KTCQ be fulfilled at \mathbf{x}^* . Then there exist unique vectors $\boldsymbol{\mu} \in \mathbb{R}^r$ and $\boldsymbol{\eta} \in \mathbb{R}^s$ with $\eta_i \geq 0 \forall i \in \{1, \dots, s\}$ such that*

$$\nabla L(\mathbf{x}^*, \boldsymbol{\mu}, \boldsymbol{\eta}) = \nabla f(\mathbf{x}^*) + \sum_{j=1}^r \mu_j \nabla h_j(\mathbf{x}^*) + \sum_{l=1}^s \eta_l \nabla g_l(\mathbf{x}^*) = 0, \quad (2.4)$$

and

$$\eta_j g_j(\mathbf{x}^*) = 0 \forall j \in \{1, \dots, s\}. \quad (2.5)$$

Proof. The theorem is taken from [206, p.321] and can be found similarly in [31, p.56]. A detailed proof can be found in [206] (p.323-329) or in the original paper by Kuhn and Tucker [153]. \square

The conditions in Theorem 1 are also referred to as the KKT (Karush-Kuhn-Tucker) conditions.

Theorem 2 (second-order necessary condition for optimality). *Let $\mathbf{x}^* \in \mathcal{D}$ be a local minimum. Let $f(\mathbf{x}), h(\mathbf{x}), g(\mathbf{x})$ be twice continuously differentiable at \mathbf{x}^* and let the LICQ be fulfilled at \mathbf{x}^* . Then $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ of Theorem 1 fulfill*

$$\mathbf{w}^\top \nabla^2 L(\mathbf{x}^*, \boldsymbol{\mu}, \boldsymbol{\eta}) \mathbf{w} \geq 0 \forall \mathbf{w} \in W(\mathbf{x}^*), \quad (2.6)$$

with $W(\mathbf{x}^*) = \{\mathbf{w} \in \mathbb{R}^n \mid \nabla h_i(\mathbf{x}^*)^\top \mathbf{w} = 0 \forall i \in \{1, \dots, r\}, \nabla g_i(\mathbf{x}^*)^\top \mathbf{w} = 0 \forall i \in \mathcal{A}(\mathbf{x}^*) \text{ with } \eta_i > 0, \nabla g_i(\mathbf{x}^*)^\top \mathbf{w} \leq 0 \forall i \in \mathcal{A}(\mathbf{x}^*) \text{ with } \eta_i = 0\}$.

Proof. The proof is given in [206, p.332f]. \square

To find a local minimum of a problem, the sufficient condition for optimality has to be fulfilled.

Theorem 3 (second-order sufficient condition for optimality). *Let $\mathbf{x}^* \in \mathcal{D}$ and let f, g, h be twice continuously differentiable at \mathbf{x}^* . If $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ of Theorem 1 fulfill*

$$\mathbf{w}^\top \nabla^2 L(\mathbf{x}^*, \boldsymbol{\mu}, \boldsymbol{\eta}) \mathbf{w} > 0 \forall \mathbf{w} \in W(\mathbf{x}^*) \text{ with } \mathbf{w} \neq \mathbf{0}, \quad (2.7)$$

then \mathbf{x}^* is a strict local solution to (2.2).

Proof. The proof is given in [206, p.333-335]. \square

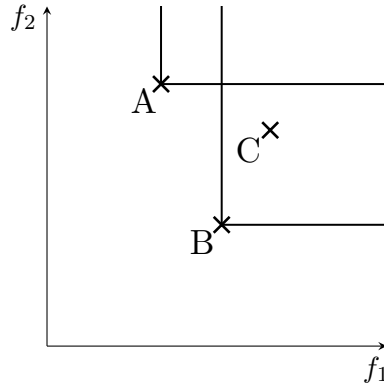


Figure 2.1.: Concept of dominance: Solution C is dominated by solution B, which is indicated by the horizontal and vertical lines. When restricting the feasible space to the set $\{A,B,C\}$, solution A and B are non-dominated.

2.1.2. Pareto Optimality

As a basis for multi-objective optimization, it is important to define a theory of relations for objective vectors. Pareto used a concept of dominance for multiple criteria optimization in the context of economics in 1896 [217, 218]. As a result, this concept of optimality is called *Pareto optimality*.³ The two following definitions are based on [254].

Definition 5 (Pareto dominance). *Let \mathbf{z}^1 and \mathbf{z}^2 be two vectors in \mathbb{R}^k . A vector \mathbf{z}^1 dominates \mathbf{z}^2 iff $z_i^1 \leq z_i^2 \forall i \in \{1, 2, \dots, k\}$ and $z_j^1 < z_j^2$ for at least one $j \in \{1, 2, \dots, k\}$.*

Definition 6 (Non-dominance). *Let $\mathbf{z}, \bar{\mathbf{z}} \in \mathbb{R}^k$. $\bar{\mathbf{z}}$ is non-dominated iff there does not exist any \mathbf{z} that dominates $\bar{\mathbf{z}}$.*

Figure 2.1 shows the concept of dominance for three arbitrary solutions in a two-dimensional objective space. Solution C is dominated by solution B. When restricting the feasible space to the set $\{A, B, C\}$, solutions A and B are non-dominated.

Having defined a concept of domination, one can now introduce an optimal solution in the sense of Pareto. The definition can also be found in [190, p.11].

Definition 7 (global Pareto optimality). *A design vector $\mathbf{x}^* \in \mathcal{D}$ is globally Pareto optimal iff there does not exist any design \mathbf{x} such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i \in \{1, \dots, k\}$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \{1, \dots, k\}$.*

Thus, the set of globally Pareto optimal designs corresponds to the set of feasible solutions whose objective vectors are non-dominated. It is called a *Pareto optimal set* and is denoted by \mathcal{P} . The image of the Pareto optimal set in objective space is a *Pareto optimal front*. To objective vectors evaluated at a Pareto optimal design one may also refer as Pareto optimal (solutions). If one uses strict inequalities $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for all $i \in \{1, \dots, k\}$ in Definition 7, the solution is referred to as *weakly Pareto optimal*.

³The terms *noninferiority* and *efficiency* are sometimes used in the same context.

2. Multi-Objective Optimization

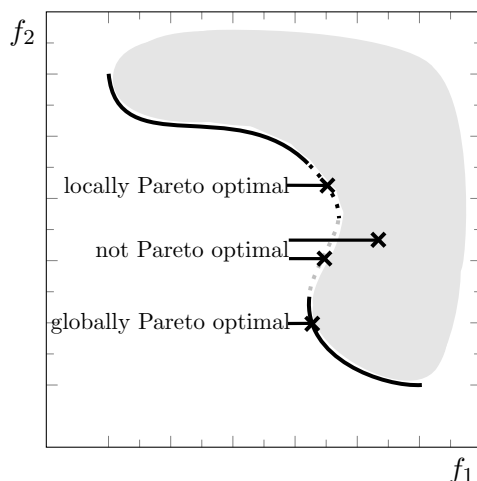


Figure 2.2.: Local (dashed bold line) and global (bold line) Pareto optimality.

Some of the computational algorithms for multi-objective optimization may find locally Pareto optimal solutions. Therefore, it is necessary to introduce local Pareto optimality.

Definition 8 (local Pareto optimality). *A point \mathbf{x}^* is locally Pareto optimal iff there exists an open neighborhood $U(\mathbf{x}^*) \subset \mathbb{R}^n$ of \mathbf{x}^* that does not contain a point that dominates \mathbf{x}^* .*

Figure 2.2 illustrates the concept of Pareto optimal solutions. The grey surface represents the image of the feasible region in objective space. The continuous line represents the Pareto optimal front. Points on the black, dashed line belong to the set of only locally Pareto optimal solutions. The curve section at the boundary of the feasible region that is indicated by the grey-colored dashed line is the part of the curve where one cannot find any locally or globally Pareto optimal point. In higher dimensions, the curve sections shown in Figure 2.2 will be higher-dimensional surfaces. The example shows that the Pareto optimal front can be non-convex and disconnected. The union of the above-described surfaces, which contains globally Pareto optimal points, only locally Pareto optimal points and points that are neither globally nor locally Pareto optimal, will be denoted as the *containing surface*.

In the following, it is essential to distinguish between local and global solutions to single-objective optimization problems and locally and globally Pareto optimal solutions. Furthermore, one has the concept of optimal solutions and strict optimal solutions in single-objective optimization problems and weakly Pareto optimal and Pareto optimal solutions in multi-objective optimization problems.

For a convex multi-objective optimization problem, it can be proven that all locally Pareto optimal solutions are also globally Pareto optimal (see [190, p.12, Theorem 2.2.3]). Convexity of the multi-objective optimization problem is defined in the following.

Definition 9 (convex MOO problem). *The multi-objective optimization problem (2.1) is convex if all objective functions and constraint functions are convex and the feasible set \mathcal{D} is convex. A general set $\mathcal{X} \subset \mathbb{R}^n$ is convex if*

$$(1-t)\mathbf{u} + t\mathbf{v} \in \mathcal{X} \quad \forall \mathbf{u}, \mathbf{v} \in \mathcal{X} \text{ and } \forall t \in [0, 1].$$

A function f is convex if it is defined on a convex set \mathcal{X} and

$$f((1-t)\mathbf{u} + t\mathbf{v}) \leq (1-t)f(\mathbf{u}) + tf(\mathbf{v})$$

for all $\mathbf{u}, \mathbf{v} \in \mathcal{X}$ and for all $t \in [0, 1]$.

Later, it will be made use of vectors defining the range of the Pareto optimal front, namely the *ideal point* and the *nadir point*.⁴

Definition 10 (ideal point). *The ideal point $\mathbf{f}^* = (f_1^*, \dots, f_k^*)^\top$ is the vector consisting of the global minima of the individual objective functions, i.e., f_i^* is the solution to the single-objective optimization problem*

$$\min_{\mathbf{x} \in \mathcal{D}} f_i(\mathbf{x}) \tag{2.8}$$

for $i = 1, \dots, k$.

Note that the ideal point is only a solution to (2.1) if all objective functions are non-conflicting and can be minimized independently, which is usually not the case.

Definition 11 (nadir point). *The nadir point $\mathbf{f}^N = (f_1^N, \dots, f_k^N)^\top$ comprises the individually worst objective functions values in the Pareto optimal set \mathcal{P} , i.e., f_i^N solves*

$$\max_{\mathbf{x} \in \mathcal{P}} f_i(\mathbf{x}) \tag{2.9}$$

for $i = 1, \dots, k$.

Maximizing over the Pareto-optimal set to find the Nadir point is, in general, not feasible since the complete Pareto optimal set is not known. However, the Nadir point can be estimated by using a payoff table based on the solutions to problem (2.8). When denoting the set of solutions to problem (2.8) as \mathcal{S} , the components of the Nadir point are defined as solutions to the problem

$$\max_{\mathbf{x} \in \mathcal{S}} f_i(\mathbf{x}) \text{ for } i = 1, \dots, k. \tag{2.10}$$

When using this approach, the Nadir point may be over- or underestimated, which is discussed in literature (see e.g. [190, 56]). Only for unique solutions to problem (2.8) in the case $k = 2$ it can be guaranteed to find the exact Nadir point. Uniqueness of the solution to problem (2.8) for $k > 2$ will avoid an overestimation of the Nadir point. If solutions are not unique, one may obtain weakly efficient solutions and, as a result, may overestimate the Nadir point. This problem can be overcome for $k = 2$ and relaxed

2. Multi-Objective Optimization

Algorithm 1: Lexicographic optimization for estimating the Nadir point.

```

 $\tilde{\mathcal{S}} = \emptyset$ 
for  $i = 1, \dots, k$  do
   $\mathbf{p}^0 = \arg \min_{\mathbf{x} \in \mathcal{D}} f_i(\mathbf{x})$ 
  for  $j = 1, \dots, k$  do
    if  $j \neq i$  then
       $\mathbf{p}^j = \arg \min_{\mathbf{x} \in \mathcal{D}} f_j(\mathbf{x})$ 
      s.t.  $f_l(\mathbf{x}) = f_l(\mathbf{p}^l) \forall l = 1, \dots, j - 1$  with  $l \neq i$ ,
       $f_i(\mathbf{x}) = f_i(\mathbf{p}^0)$ .
    end if
  end for
   $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup \{\mathbf{p}^l\}$  with  $l = \max\{j \mid j \in \{1, \dots, k\}, j \neq i\}$ 
end for

```

for $k > 2$ by performing a lexicographic optimization based on the solutions to (2.8) as expressed in Algorithm 1.

The Nadir points can then be estimated with the maximization problem (2.10) over the set $\tilde{\mathcal{S}}$ obtained from the lexicographic optimization. The estimate can be different for different orders of indices in the inner loop of the algorithm.

The ideal and nadir point can be used to define the outlines of the Pareto optimal front. They are often used to normalize objective functions in a given problem (cp. [188]). The normalized objective function \hat{f}_i is then given by

$$\hat{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^*}{f_i^N - f_i^*}. \quad (2.11)$$

Necessary and sufficient conditions can also be defined for Pareto optimality following Kuhn and Tucker [153]. The presentation in the following is based on the formulation in [190].

Theorem 4 (Kuhn-Tucker necessary condition for Pareto optimality). *Let $\mathbf{x}^* \in \mathcal{D}$. Let $\mathbf{F}(\mathbf{x})$, $h(\mathbf{x})$ and $g(\mathbf{x})$ be continuously differentiable at \mathbf{x}^* and let $h(\mathbf{x})$ and $g(\mathbf{x})$ satisfy the KTCQ at \mathbf{x}^* . The necessary condition for \mathbf{x}^* to be Pareto optimal is that there exist multipliers $\boldsymbol{\lambda} \in \mathbb{R}^k$ with $\lambda_i \geq 0 \forall i \in \{1, \dots, k\}$ and $\boldsymbol{\lambda} \neq \mathbf{0}$, $\boldsymbol{\mu} \in \mathbb{R}^r$ and $\boldsymbol{\eta} \in \mathbb{R}^s$ with $\eta_i \geq 0 \forall i \in \{1, \dots, s\}$ such that*

$$\sum_{i=1}^k \lambda_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^r \mu_j \nabla h_j(\mathbf{x}^*) + \sum_{l=1}^s \eta_l \nabla g_l(\mathbf{x}^*) = \mathbf{0} \text{ and} \quad (2.12)$$

$$\mu_j g_j(\mathbf{x}^*) = 0 \forall j \in \{1, \dots, s\}. \quad (2.13)$$

⁴Sometimes the ideal point is also referred to as utopian point.

Proof. The proof can be found in [190, p.39f] for inequality constraints. The extension to equality constraints is straightforward. \square

For the second-order necessary condition, the LICQ needs to be fulfilled.

Theorem 5 (second-order necessary condition for Pareto optimality). *Let $\mathbf{x}^* \in \mathcal{D}$ be Pareto optimal. Let $\mathbf{F}(\mathbf{x})$, $g(\mathbf{x})$ and $h(\mathbf{x})$ be twice continuously differentiable at \mathbf{x}^* and let $g(\mathbf{x})$ and $h(\mathbf{x})$ satisfy the LICQ at \mathbf{x}^* . Then $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ of Theorem 4 fulfill*

$$\mathbf{d}^\top \left(\sum_{i=1}^k \lambda_i \nabla^2 f_i(\mathbf{x}^*) + \sum_{j=1}^p \mu_j \nabla^2 h_j(\mathbf{x}^*) + \sum_{l=1}^q \eta_l \nabla^2 g_l(\mathbf{x}^*) \right) \mathbf{d} \geq 0, \quad (2.14)$$

with $\mathbf{d} \in \{\mathbf{0} \neq \mathbf{d} \in \mathbb{R}^n \mid \nabla f_i(\mathbf{x}^*)^\top \mathbf{d} \leq 0 \forall i \in \{1, \dots, k\}, \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} = 0 \forall j \in \{1, \dots, r\}, \nabla g_l(\mathbf{x}^*)^\top \mathbf{d} = 0 \forall l \in \mathcal{A}(\mathbf{x}^*)\}$.

Proof. The theorem is given in [190, p.42] with a reference to the proof in [282]. \square

A sufficient condition for Pareto optimality can be stated as a second-order condition.

Theorem 6 (sufficient condition for Pareto optimality). *Let $\mathbf{x}^* \in \mathcal{D}$. Let $\mathbf{F}(\mathbf{x})$, $g(\mathbf{x})$ and $h(\mathbf{x})$ be twice continuously differentiable at \mathbf{x}^* and let $g(\mathbf{x})$ and $h(\mathbf{x})$ satisfy the Kuhn-Tucker Constraint Qualification at \mathbf{x}^* . The sufficient condition for \mathbf{x}^* to be locally Pareto optimal is that*

$$\exists \boldsymbol{\lambda} \in \mathbb{R}^k \text{ with } \lambda_i \geq 0 \forall i \in \{1, \dots, k\}, \boldsymbol{\mu} \in \mathbb{R}^r \text{ and } \boldsymbol{\eta} \in \mathbb{R}^s \text{ with } \eta_i \geq 0 \forall i \in \{1, \dots, s\}$$

such that (2.12) and (2.13) are fulfilled, and

$$\mathbf{d}^\top \left(\sum_{i=1}^k \lambda_i \nabla^2 f_i(\mathbf{x}^*) + \sum_{j=1}^r \mu_j \nabla^2 g_j(\mathbf{x}^*) + \sum_{l=1}^s \eta_l \nabla^2 h_l(\mathbf{x}^*) \right) \mathbf{d} > 0 \quad (2.15)$$

with $\mathbf{d} \in \{\mathbf{0} \neq \mathbf{d} \in \mathbb{R}^n \mid \nabla g_j(\mathbf{x}^*)^\top \mathbf{d} = 0 \forall j \in \{1, \dots, r\}, \nabla h_l(\mathbf{x}^*)^\top \mathbf{d} = 0 \forall l \in \mathcal{A}(\mathbf{x}^*) \text{ with } \mu_l > 0, \nabla h_m(\mathbf{x}^*)^\top \mathbf{d} \leq 0 \forall m \in \mathcal{A}(\mathbf{x}^*) \text{ with } \mu_m = 0\}$
 or $\mathbf{d} \in \{\mathbf{0} \neq \mathbf{d} \in \mathbb{R}^n \mid \nabla f_i(\mathbf{x}^*)^\top \mathbf{d} \leq 0 \forall i \in \{1, \dots, k\}, \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} = 0 \forall j \in \{1, \dots, r\}, \nabla g_l(\mathbf{x}^*)^\top \mathbf{d} \leq 0 \forall l \in \mathcal{A}(\mathbf{x}^*)\}$.

Proof. The theorem is given in [190, p.43] with a reference to the proof in [282]. \square

In the above conditions one assumes that the solution is non-trivial, i.e., $\boldsymbol{\lambda} \neq \mathbf{0}$. Note that the above necessary conditions are also fulfilled for weakly Pareto optimal points and local (weakly) Pareto optimal points. The sufficient condition then gives a locally Pareto optimal point in the stricter sense.

2.2. Multi-Objective Optimization Algorithms

There exist different approaches to solve multi-objective optimization problems. They can be categorized according to the time of decision-making. One approach is to make a decision before defining the optimization problem. This is often referred to as an *a priori method*. It is common to define a so-called utility function based on preferences. The utility function is minimized, such that a single optimal solution to the problem is found. This procedure enables the reduction of complexity and is applicable to an arbitrary number of objective functions. However, it will significantly restrict the outcome and raises problems concerning the articulation of preferences since the utility function has to be carefully chosen. Examples of methods with a priori decision-making are the weighted sum method without weight scanning, goal programming [33], compromise programming [296, 299], and lexicographic approaches based on Algorithm 1 for finding a Pareto optimal point. They will all find a single Pareto optimal solution.

There also exist so-called *interactive methods* in which preferences are set during the optimization like in the method of Steuer [255, 254], the STEM method [15] or the NIMBUS method [191].

In methods for Pareto front exploration or *generating methods*, the challenge is to find an evenly distributed set of objective vectors that approximate the Pareto optimal front. This is a challenge because the Pareto optimal front can be convex, non-convex, continuous, or discontinuous. Methods for Pareto front exploration are also referred to as *a posteriori methods* since the decision-maker selects a trade-off solution after the generation of Pareto optimal points. The present work is restricted to these types of methods. Generating methods may provide a better understanding of the overall problem and give freedom to the decision-maker. The decision-making process can be further supported by adding solutions in interesting regions of the objective space. A drawback of this methodology is that the decision-making process is not very intuitive in situations where more than three objectives are present. A suitable visualization strategy can be of interest for helping in the decision-making process.

Among algorithms for Pareto front exploration one can distinguish between *scalarization methods* and *direct Pareto methods*. Direct Pareto methods find a set of representative solutions on the Pareto optimal front directly in one optimization procedure. They are a popular choice for MOO. Therefore, Section 2.2.1 gives a condensed overview of direct Pareto approaches, focusing on evolutionary strategies. Although these methods manage to generate several proposed solutions simultaneously, they are usually computationally expensive and may be slow in terms of convergence. Opposed to gradient-based methods, they lack mathematically well-defined convergence criteria.

Scalarization methods reduce the multi-objective optimization problem to one or several single-objective optimization problems, such that only one optimal solution can be found in each optimization run.⁵ The single-objective optimization problems can then, for example, be solved using an efficient gradient-based strategy. Examples for

⁵Sometimes scalarization only refers to a certain kind of aggregation of objective functions. Here, it is used as a more general description.

scalarization methods are the weighted sum method, the equality constraint method, the epsilon-constraint method and the normal constraint method, which are presented in Sections 2.2.2, 2.2.3, 2.2.4 and 2.2.5.

Some studies in literature exist that compare scalarization approaches and evolutionary approaches (see e.g. [192, 250]). Gambier [69], for example, compares a scalarization method (normal boundary intersection) with a multi-objective evolutionary algorithm (NSGA-II). The author concludes that for the specific application, i.e., the design of a control system, the scalarization approach is computationally more efficient, and the resulting set of Pareto optimal solutions is better distributed in objective space. However, in [11], a multi-objective evolutionary algorithm (MOGA) outperforms a scalarization approach (a combination of weighted sum method and epsilon-constraint method) with respect to computational effort for a structural design problem with two objective functions. It is important to point out that neither Pareto approaches nor scalarization approaches are superior. On the contrary, the choice of a methodology depends on the underlying optimization problems. This includes the computational costs for function evaluations, the availability of gradients, additional constraints, the complexity of the design space as well as the dimensionality of the MOO problem. In general, the multi-objective nature of the problem introduces an additional level of complexity. Both types of strategies are suitable for parallelization strategies, e.g., by simultaneously solving the single-objective optimization problems or by simultaneously evaluating the quality of a set of design points.

Nevertheless, a continuous optimization problem with a high-dimensional design space is predestined for gradient-based optimization methods. A common mistake is the assumption that problems with discontinuous Pareto fronts cannot be solved with such type of methods. As indicated in the previous section, the Pareto optimal front of problems with continuously differentiable objective functions is not necessarily continuous. As a result, in the following, the focus is on scalarization approaches for multi-objective optimization methods with continuous objective functions. Here, the methods of choice are constraint methods.

2.2.1. Direct Pareto Approaches

Direct Pareto approaches generate several solutions simultaneously. Therefore, they are often easy to parallelize. Examples for direct Pareto approaches are *multi-objective evolutionary algorithms*. They are based on the generation of a population of solutions. Other examples of direct Pareto approaches are multi-objective particle swarm optimization [198] or multi-objective simulated annealing [272].

Evolutionary algorithms exist for single-objective optimization as well as for multi-objective optimization. Prominent evolutionary algorithms are *genetic algorithms*. They were developed by Holland [119] and use the evolutionary concepts of Darwinism. In genetic algorithms, a *population* of individuals is created in each iteration. A *fitness value* is assigned to every individual of a population. According to their fitness value, they are selected for reproduction. Crossover and mutation are applied to generate a new population, which forms the new generation of solutions. The algorithms stop after a

2. Multi-Objective Optimization

certain number of generations, typically if no further improvement is observed. Genetic algorithms differ in their methods for selection, mutation, and crossover.

Evolutionary methods are, in general, easy to implement and are reliable in providing solutions. A negative aspect of these strategies is that they are computationally expensive and may be slow in terms of convergence. In general, the convergence properties of evolutionary algorithms are hard to analyze and can only be formulated in a probabilistic setting (see e.g. [274] for multi-objective evolutionary algorithms). Therefore, when evolutionary algorithms are applied in the context of multi-objective optimization, it is not guaranteed that they approximate globally Pareto optimal points. It is only possible to judge Pareto optimality within the population of individuals. As a result, Pareto optimality cannot generally be guaranteed.

Several multi-objective evolutionary algorithms were developed in the last decades. The Vector-Evaluated Genetic Algorithm (VEGA) was introduced by Schaffer in 1985 [234] and was the first multi-objective genetic algorithm. It is not a Pareto approach as it is not based on Pareto optimality. The selection phase is different from a conventional genetic algorithm, as the population is divided into a fixed number of sub-populations. In these populations, the best individual for a single objective is selected. The major drawback of this method is the risk for tending to the optimization of only one objective so that, for example, non-convex regions of Pareto optimal fronts are not covered.

Most of the other multi-objective evolutionary approaches use Pareto ranking for the selection process, i.e., a ranking based on the concept of Pareto optimality. So-called niching algorithms are often applied either to the objective domain or the decision domain to ensure diversity among the Pareto optimal front. Goldberg introduced the first algorithm that was based on Pareto ranking in 1989 [83]. It was, for example, implemented in the Non-dominated Sorting Genetic Algorithm (NSGA) by Srinivas and Deb in [253], which was further extended in [49] (NSGA-II). Pareto ranking is performed by giving all non-dominated members of the population the same fitness value and repeatedly removing them from the population to rank the non-dominated members in the remaining population. Other prominent algorithms are the Multi-Objective Genetic Algorithm (MOGA) by Fonseca and Fleming [67] in which the fitness value of a member of the population depends on the number of individuals that are dominated by the member, the Niche Pareto Genetic Algorithm (NPGA) by Horn, Nafpliotis and Goldberg [120, 121], which compares two candidates based on a selected set of the population, or the Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler and Thiele [300, 301]. It archives a second population of non-dominated solutions that is used for ranking the original population.

The advantage of multi-objective evolutionary approaches is that, similarly to the single-objective evolutionary approaches, they are reliable and easy to implement since the evaluation of the objective function is treated as a black-box. They are based on the concept of domination and find a set of solutions in one run. If suitable strategies for ranking, mutation, and crossover are chosen, they may generate a representative set of solutions in objective space in a single run. However, an even distribution of solutions in objective space cannot be guaranteed. The multi-objective evolutionary strategies suffer from the same drawbacks as the corresponding single-objective optimization strategies.

They need a high number of function evaluations and are slow in terms of convergence. This is especially the case for high-dimensional objective function spaces, as evolutionary strategies need a large population to find a representative set of solutions in the objective function space [64].

2.2.2. Weighted Sum Method

In the *weighted sum method*, presented, for example, in [297], the objectives are combined to a single objective in a convex combination that is to be minimized, i.e.,

$$\min_{x \in \mathcal{D}} \sum_{i=1}^k w_i f_i(\mathbf{x}) \quad (2.16)$$

with weights $w_i \geq 0$ for $i = 1, \dots, k$ and $\sum_{i=1}^k w_i = 1$. To obtain a set of Pareto optimal solutions the values for the weights w_i are systematically modified for every optimization. There are special concepts for modifying the weights like the *adaptive weighted sum method* [143] or simple weight scanning.

All minimizers of the single objective optimization problem for (2.16) with strictly positive weights, as well as all unique minimizers, are globally Pareto optimal solutions. Otherwise, the solution is weakly Pareto optimal. This is, for example, proven in [254, p.167] and [190, p.78].

A significant disadvantage of using the weighted sum method for a generation of a Pareto optimal front is the fact that the points on the Pareto optimal front are not necessarily uniformly distributed when choosing uniformly distributed weights w_i . Additionally, the algorithm cannot find Pareto optimal solutions in non-convex regions of the Pareto optimal front. These drawbacks are discussed in [45].

2.2.3. Equality Constraint Method

The *equality constraint method* (EC) is also known as the *proper equality constraint method* and was formulated by Lin [165, 167, 166]. The idea is to minimize one objective function f_s for a fixed $s \in \{1, \dots, k\}$ while imposing equality constraints on the remaining objective functions f_i with $i \in \{1, \dots, k\}$ and $i \neq s$.

The constraints, as well as the objective function to be minimized, are varied to find different Pareto optimal points that are distributed in objective space. One can think of this procedure as *scanning* with a specific resolution in all directions of the objective space. The task is then to solve a total number of $n_o = k \cdot ((n_u + 1)^{k-1} - 1)$ constrained single-objective optimization problems, where n_u denotes the user-defined resolution in objective space. This means that a number of $n_s = (n_u + 1)^{k-1} - 1$ constrained optimization problems are solved when minimizing one objective function $s \in \{1, \dots, k\}$. This number arises from all possible combinations of prescribed constraint values. The

2. Multi-Objective Optimization

j -th minimization problem for $j = 1, \dots, n_s$ with $s \in \{1, \dots, k\}$ is then given by

$$\begin{aligned} & \min_{\mathbf{x} \in \mathcal{D}} f_s(\mathbf{x}) & (2.17) \\ \text{s.t. } & f_i(\mathbf{x}) = f_i^{(j)} \quad \forall i \in \{1, \dots, k\} : i \neq s. \end{aligned}$$

The constraint values $f_i^{(j)}$ for the different steps can be distributed in equidistant fashion. Since the constraints are given in objective space this allows an equidistant scanning of the Pareto optimal front. When additionally altering the objective function to be minimized one can find a well-distributed set of Pareto optimal points in objective space. One can further improve this result with a normalization based on Equation (2.11), although one cannot expect to obtain a perfectly even distribution. When a decision-maker is interested in specific parts of the Pareto optimal front, the constraints can be chosen according to his or her preferences. This can, for example, be done interactively.

Algorithmic Details In the following, the details on finding a step size for scanning and defining suitable bounds for the equality constraints are presented.

The range of the Pareto optimal front can be obtained by finding the ideal point and the nadir point (see Definition 10 and 11). The constraint values for the i -th objective function $f_i^{(j)}$ can then be constructed in the interval $[f_i^*, f_i^N]$ with a constant step $\Delta f_i > 0$. The corresponding algorithm for determining the step for equidistant scanning is given in Algorithm 2 as the first step of the method.

Algorithm 2: Determination of step for equidistant scanning.

Input:

k: number of objective functions
 n_u : resolution in objective space

Function:

```

for  $s = 1$  to  $k$  do
   $\mathbf{x}_s^* = \arg \min_{\mathbf{x} \in \mathcal{D}} f_s(\mathbf{x})$ 
   $f_s^* = f_s(\mathbf{x}_s^*)$ 
end for
for  $s = 1$  to  $k$  do
   $f_s^N = \max_{i=1, \dots, k} f_s(\mathbf{x}_i^*)$ 
   $\Delta f_s = \frac{1}{n_u+1} (f_s^N - f_s^*)$ 
end for

```

One can make the following important remarks:

- **Nadir point:** An underestimation of the nadir point prevents the constraint method from finding Pareto optimal points on all parts of the Pareto optimal front. An overestimation, on the contrary, will produce single-objective optimization problems defined outside of the range of the Pareto optimal front. As a result,

the chosen resolution of the Pareto optimal front may be deteriorated. Therefore, a good estimation is crucial for the success of the algorithm (see Section 2.1.2 for details on the estimation of nadir points).

- **Scanning:** In literature some authors (see e.g. [183]) also suggest to use a scanning step size $\Delta f_{s,i}$ depending on the i -th objective function to be minimized. It is given by

$$\Delta f_{s,i} = \frac{1}{n_u + 1} (f_s(\mathbf{x}_i^*) - f_s^*), \quad (2.18)$$

i.e., choosing $f_s(\mathbf{x}_i^*)$ instead of the respective value of the nadir point. The starting value for determining the constraint values is then chosen as $f_{s,i}^0 = f_s(\mathbf{x}_i^*)$. This choice will guarantee that the resulting optimal points lie within the range of the Pareto optimal front, but it cannot guarantee to find all Pareto optimal points. The modified procedure using $\Delta f_{s,i}$ is introduced in Section A.1.1 of the appendix.

- **Weak Pareto optimality:** When risking to find weakly Pareto optimal points, it is necessary to find \mathbf{x}_s^* in the above algorithm with the help of a lexicographic optimization to minimize the risk that \mathbf{x}_s^* is only weakly Pareto optimal [183]. Otherwise, since the nadir point is expectedly based on a weakly Pareto optimal solution, a choice of the step size based on the nadir point may result in finding weakly Pareto optimal solution outside of the range of the Pareto optimal front. Therefore, in this case, it can also be of advantage to use the method described in the second item. However, this may result in an incomplete representation of the Pareto optimal front. The following notation is restricted to problems with unique optimal solutions. This restriction is a natural assumption in engineering problems with continuous objective functions and underlying partial differential equations. For shape optimization problems, for example, one usually assumes that a unique state is given for each design point. The uniqueness of the objective function value often holds for the respective objective functions.
- **Existence of solutions:** For $k > 2$, it can happen that a feasible solution to a resulting constrained optimization problem does not exist since the bounds are too restrictive. Since the algorithm is constructed by starting with the maximum upper bound, which is lowered during the steps of the algorithm, the bounds become more restrictive. Whenever there does not exist a feasible solution to the optimization problem, the minimum feasible bound is reached, and all subsequent reductions of this bound can be discarded.

The algorithm for the equality constraint method is shown in Algorithm 3, using a recursive strategy for setting all possible combinations of constraint values. The points \mathbf{x}_s^* for $s \in \{1, \dots, k\}$ found with Algorithm 2 already represent Pareto optimal points, which are often referred to as *anchor points*. After having found the step size Δf_i for $i \in \{1, \dots, k\}$, it can be used to alter the constraint values for the constrained single-objective optimization problems starting from $f_i^{(0)} = f_i^N$. The updating of the

2. Multi-Objective Optimization

Algorithm 3: Equality constraint method.

Global variables:

k : number of objective functions (input)

n_u : resolution in objective space (input)

$\Delta \mathbf{F}$: resolution step size

j : counter

\mathcal{P} : Pareto optimal set (output)

Function Main(k, n_u):

Obtain \mathbf{x}_s^* , Δf_s , f_s^N for $s = 1, \dots, k$ from Algorithm 2

$\mathcal{P} \leftarrow \{\mathbf{x}_1^*, \dots, \mathbf{x}_k^*\}$

for $s = 1$ to k **do**

$j \leftarrow 0$

 Call **CalculatePoint**($1, s, \mathbf{0}$)

end for

Function CalculatePoint(l, s, ε):

if $l == s$ **then**

$l \leftarrow l + 1$

end if

for $m = 0$ to n_u **do**

$\varepsilon_l \leftarrow f_l^N + m\Delta f_l$

if $l == \max\{j \mid i \in \{1, \dots, k\}, i \neq s\}$ **then**

$f_i^{(j)} = \varepsilon_i$ for $i \in \{1, \dots, k\}, i \neq s$

if $j \neq 0$ **then**

$\mathbf{x}^p \leftarrow \arg \min_{\mathbf{x} \in \mathcal{D}} f_s(\mathbf{x})$ s.t. $f_i(\mathbf{x}) = f_i^{(j)} \forall i \in \{1, \dots, k\}, i \neq s$

$\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{x}^p\}$

end if

$j \leftarrow j + 1$

else

 Call **CalculatePoint**($l + 1, s, \varepsilon$)

end if

end for

return

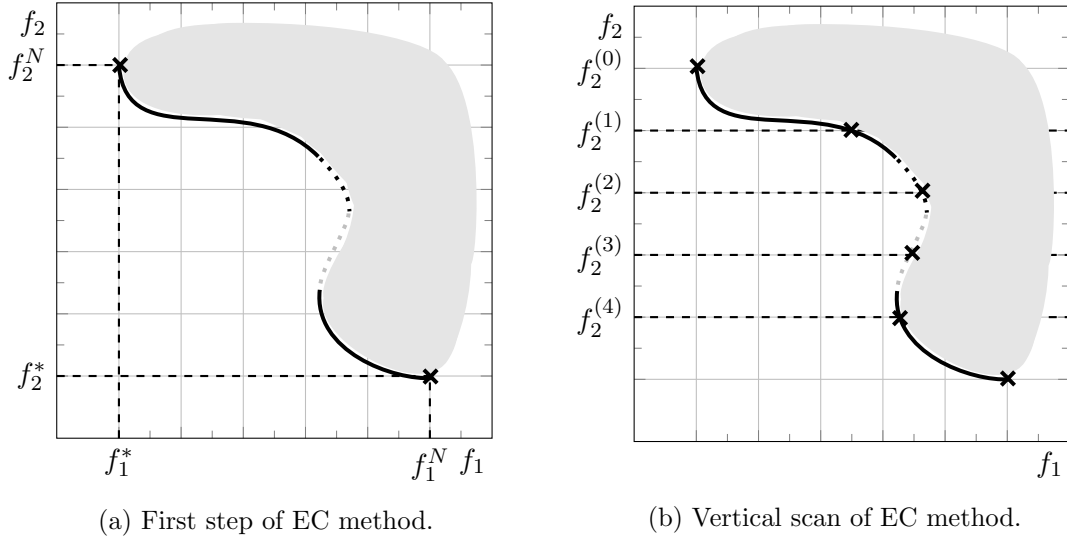


Figure 2.3.: First step of the equality constraint method, i.e., the determination of the ideal point f_i^* and the nadir point f_i^N for $i \in \{1, 2\}$ (a), and vertical scan of the Pareto optimal front (b). The objective function f_1 is minimized while constraining f_2 with the prescribed constraints $f_2^{(i)}$ for $i = 1, \dots, 4$.

constraints is performed in such a way that all possible combinations of constraint values are represented. Given an objective function $s \in \{1, \dots, k\}$, there are $(n_u + 1)^{k-1}$ possible combinations of constraint values when choosing from $n_u + 1$ values for each of the $k - 1$ objective functions. For each objective function to be constrained, the value of the nadir point and n_u equidistantly distributed values are set as constraints. Let

$$\mathcal{S}_i = \{f_i^{(j)} \mid f_i^{(j)} = f_i^{(0)} - m\Delta f_i \text{ for } m = 0, \dots, n_u\}$$

be the set of all possible constraint values $f_i^{(j)}$ for the i -th objective function for $i = 1, \dots, k$ with $i \neq s$. Then the set of all possible combinations of constraint functions can be expressed as the Cartesian product $\prod_{i \in \{1, \dots, k\}, i \neq s} \mathcal{S}_i$. The j -th combination of constraint values $f_i^{(j)}$ for $j = 0, \dots, (n_u + 1)^{k-1}$ is chosen from this set. As an optimization when setting the value of the nadir point for each constrained objective function will result in an anchor point, this case can be neglected.

The procedure of Algorithm 3 is shown in the following figures at the example of a two-dimensional objective space with artificial objective functions f_1 and f_2 . Figure 2.3a depicts the first step of the algorithm defining the extent of the Pareto optimal front with the help of the ideal point f_i^* and the nadir point f_i^N for $i \in \{1, 2\}$. Figure 2.3b shows the vertical scanning of the Pareto optimal front by minimizing f_1 and constraining f_2 with constraint values $f_2^{(j)}$ for $j = 1, \dots, n_u$ with $n_u = 4$. The constraint values are indicated by the dashed lines.

As can be seen in Figure 2.3b, the method can only guarantee to find points on the containing surface. The solution found for the constraint value $f_2^{(2)}$ is only locally Pareto

2. Multi-Objective Optimization

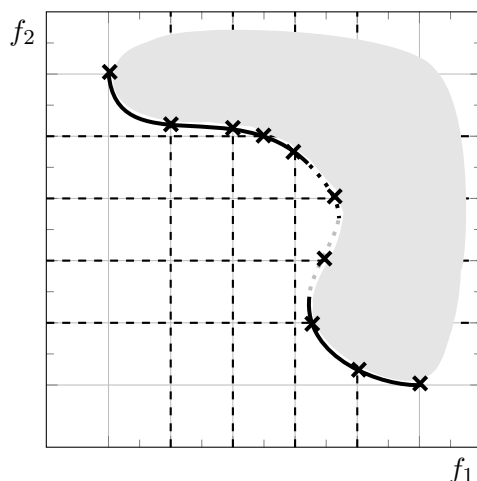


Figure 2.4.: Scanning the Pareto optimal front for both objectives. This ensures a good distribution of Pareto optimal points.

optimal, and the solution for the constraint value $f_2^{(3)}$ is not Pareto optimal. Another problem that can be seen in this context is that the choice of equidistant step sizes $\Delta f_i^{(j)}$ does not guarantee that the Pareto optimal front is covered uniformly. The large part of the Pareto optimal front between the points for $f_2^{(0)}$ and $f_2^{(1)}$ is not covered.

The first problem can only be solved by using inequality constraints instead of equality constraints (see Section 2.2.4). At the end of the present section, one can find a comment on strategies for judging if a solution is Pareto optimal. However, the second problem can be overcome by scanning the Pareto optimal front in all directions, i.e., by solving constrained single-objective optimization problems with varying objective functions.

The algorithm improves the approximation of the whole Pareto optimal front, as shown in Figure 2.4. Now, the region that is not covered in Figure 2.3b is represented by points found during the horizontal scanning of the front.

Properties of the method As it is the case for all scalarization approaches, finding a global solution to the single-objective optimization algorithms is crucial for the success of the multi-objective optimization procedure. Only a globally optimal solution can potentially give a globally Pareto optimal point. Figure 2.5 visualizes the situation when a single objective optimization algorithm only finds a local solution, which is the case for the two exemplary points that do not lie on the Pareto optimal front.

We have already seen in the example that the equality constraint method cannot guarantee to find only globally Pareto optimal solutions. It may find solutions on the containing surface that are only locally Pareto optimal or solutions that are not Pareto optimal at all. Only for convex multi-objective optimization problems, the equality constraint method will provide Pareto optimal solutions.

Lin [166] offers a theorem to show when solutions to (2.17) are globally Pareto optimal.

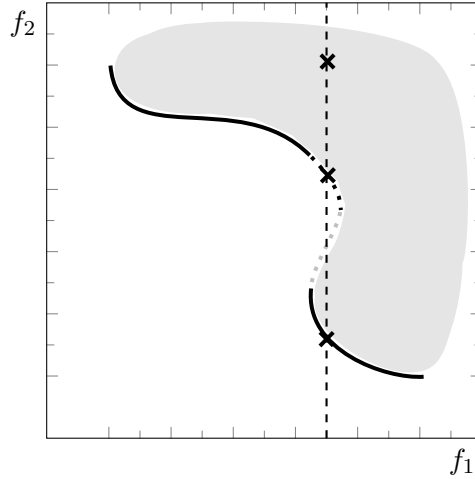


Figure 2.5.: Local solutions to the constrained single optimization problem.

Theorem 7. A feasible solution \mathbf{x}^* to (2.17) with a vector of chosen constraint values $\boldsymbol{\varepsilon}^* := (f_1(\mathbf{x}^*), \dots, f_{s-1}(\mathbf{x}^*), f_{s+1}(\mathbf{x}^*), \dots, f_k(\mathbf{x}^*))^\top$ and finite $f_s(\mathbf{x}^*)$ is globally Pareto optimal iff for $\Phi(\boldsymbol{\varepsilon}) := \inf(f_s(\mathbf{x}) \mid \mathbf{x} \in \mathcal{D}, f_i(\mathbf{x}) = \varepsilon_i, i \neq s)$:

- $\Phi(\boldsymbol{\varepsilon}) \geq \Phi(\boldsymbol{\varepsilon}^*)$ for all $\boldsymbol{\varepsilon}$ corresponding to a feasible solution to (2.17) satisfying $\varepsilon_i \leq \varepsilon_i^*, i \neq s$ and
- $\Phi(\boldsymbol{\varepsilon}) > \Phi(\boldsymbol{\varepsilon}^*)$ for all $\boldsymbol{\varepsilon}$ corresponding to a feasible solution \mathbf{x} to (2.17) with $\Phi(\boldsymbol{\varepsilon}) > -\infty$ and $\Phi(\boldsymbol{\varepsilon}) = f_s(\mathbf{x})$, and satisfying $\varepsilon_i \leq \varepsilon_i^*, i \neq s$ with the strict inequality holding for at least one i .

Proof. The theorem can be found in [166] for a maximization problem and in [31, p.147] for a minimization problem. It was proven by Lin in [166]. \square

This means that a solution is globally Pareto optimal if it is not dominated by any solutions with smaller constraint values. However, checking the requirements of Theorem 7 is not feasible since it requires the complete set of points that can be possibly found with the equality constraint method.

A simple, but not reliable, alternative is to discard solutions that are dominated by other solutions found by the algorithm. This procedure is also referred to as *Pareto filtering*. It is not guaranteed that all relevant solutions are discarded if the algorithm does not find the whole set of Pareto optimal solutions.

In [17] the authors derive a check for Pareto optimality of a solution \mathbf{x}^* . It involves solving an additional single-objective optimization. The approach is a more practicable reformulation of the ideas of Theorem 7. Given the vector $\boldsymbol{\delta} \in \mathbb{R}^k$, iff $\Phi = 0$ is the solution to

$$\Phi = \sup_{\mathbf{x}, \boldsymbol{\delta}} \sum_{i=1}^k \delta_i \quad \text{s.t.} \quad f_i(\mathbf{x}) + \delta_i = f_i(\mathbf{x}^*), \quad \delta_i \geq 0 \text{ for } i = 1, \dots, k, \quad (2.19)$$

2. Multi-Objective Optimization

then \mathbf{x}^* is Pareto optimal. Any solution $(\mathbf{x}^*, \boldsymbol{\delta}^*)$ to the above problem with $\Phi < \infty$ is a globally Pareto optimal solution.

Theorem 7 and its reformulation already motivate to use a formulation based on inequality constraints instead of equality constraints. The epsilon-constraint method, which is presented in the following section, is based on inequality constraints.

2.2.4. Epsilon-Constraint Method

The *epsilon-constraint method* (or *ε -constraint method*) was developed by Marglin in 1967 [177]. One individual objective function is minimized while imposing inequality constraints on the other objective functions. Thus, it is closely related to the equality constraint method introduced in Section 2.2.3, which was also analyzed by Lin [168]. The resulting modified optimization problem for minimizing f_s in the j -th step is given by

$$\min_{\mathbf{x} \in \mathcal{D}} f_s(\mathbf{x}) \text{ s.t. } f_i(\mathbf{x}) \leq f_i^{(j)} \quad \forall i \in \{1, \dots, k\}, i \neq s. \quad (2.20)$$

Again, the upper bound is varied for every iteration to generate a Pareto optimal set. Since the notation $\varepsilon_i^{(j)} := f_i^{(j)}$ is often used to describe the upper bound, the method is referred to as epsilon-constraint method.

The main advantage of the epsilon-constraint method in comparison to the equality constraint method is that a solution to problem (2.20) is weakly Pareto optimal. Actually, it can be shown that a solution \mathbf{x}^* to (2.20) is globally Pareto optimal if and only if it is an optimal solution to (2.20) for all $s = 1, \dots, k$ with upper bounds $\varepsilon_i = f_i(\mathbf{x}^*)$ for $i = 1, \dots, k$ with $i \neq s$ (see [190, p.85f]). Theorem 8 gives the stronger result.

Theorem 8. *All unique solutions to (2.20) are globally Pareto optimal for any upper bound $f_i^{(j)}$ for $i \in \{1, \dots, k\}$ with $i \neq s$.*

Proof. The proof is given in [190, p. 86]. □

The algorithm for the epsilon-constraint method can be constructed in the same way as the algorithms presented for the equality constraint method but with inequality constraints instead of equality constraints. Some details are distinctive for the epsilon-constraint method:

- **Nadir point:** The overestimation of the nadir point does not have as significant effects as in the case of the equality constraint method. The same applies to the choice of the nadir point as a starting value for prescribing upper bounds. In the case of an overestimation of upper bounds, the resulting solution to the given constrained optimization problem is (weakly) Pareto optimal with inactive constraints. Whenever the constraints of the optimal solution are inactive with values $\tilde{\varepsilon}_i$ the subsequent constrained optimization problems can be skipped until the prescribed upper bound is lower, i.e., $f_i^{(j)} \leq \tilde{\varepsilon}_i$. As a result, an overestimation may only affect the resolution of the Pareto optimal front.

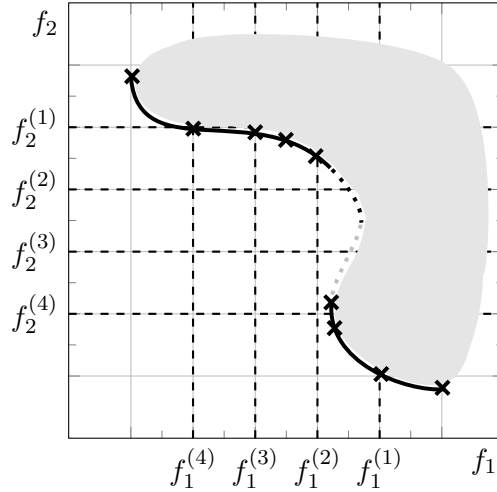


Figure 2.6.: Solutions for the epsilon-constraint method. The constraint values are indicated for both scanning directions.

- **Active and inactive constraints:** If an optimal solution with an active constraint was found in the j -th step of the algorithm and inactive constraints occur for the step $j + 1$, then the Pareto optimal front is disconnected [284] and the same skipping procedure as in the first item can be applied.

Figure 2.6 shows the points found with the help of the epsilon-constraint method for the two-dimensional artificial problem. It can be seen that the method indeed only finds globally Pareto optimal points if the solution is unique. When minimizing f_1 with the constraint $f_2 \leq f_2^{(2)}$, the method will find the starting point of the part of the disconnected front with lower f_2 -values. As a result, the single-objective optimization problem with the constraint value $f_2^{(3)}$ can be skipped since it would find the same point.

Some variants of the epsilon-constraint are presented in Section A.1.2 of the appendix. Since one assumes the absence of weakly Pareto optimal solutions, the equality constraint method and the epsilon-constraint method will be the method of choice in this thesis. However, other interesting scalarization methods exist. The normal constraint method and methods related to the normal constraint method will be presented in the following section.

2.2.5. Normal Constraint Method

The *normal constraint method* (NC, [133]) is directly related to the epsilon-constraint method. One objective function, e.g., f_1 is minimized, and the design space is gradually reduced by imposing additional inequality constraints for $k - 1$ planes in objective space. The main difference is that the feasible region in objective space is not constructed using planes parallel to the coordinate axes. Instead, the planes (or hyperplanes) that are used to define the feasible region are orthogonal to a *utopia plane* in objective space

2. Multi-Objective Optimization

(referring to the ideal point) in which all anchor points lie. Given the ideal point \mathbf{x}_i^* , the utopia plain in the original NC approach is built with the help of the normal directions $\mathbf{N}_i = \mathbf{F}(\mathbf{x}_i^*) - \mathbf{F}(\mathbf{x}_1^*)$. Then points on the utopia plane are described by

$$\mathbf{p}_j = \sum_{i=1}^k \alpha_{ij} \mathbf{F}(\mathbf{x}_i^*) \text{ with } \sum_{i=1}^k \alpha_{ij} = 1 \text{ and } 0 \leq \alpha_{ij} \leq 1.$$

One can generate an even distribution of points with the increment $\delta_i = (m_i - 1)^{-1}$, with the user-defined resolution m_i . The problem is transformed into a single-objective optimization problem with inequality constraints to restrict the feasible space. The inequality constraints are defined based on planes that are perpendicular to the utopia plane and intersect in \mathbf{p}_j , such that the task is to solve

$$\min_{\mathbf{x} \in \mathcal{D}} f_1 \quad \text{s.t. } \mathbf{N}_i^\top (\mathbf{F}(\mathbf{x}) - \mathbf{p}_j) \leq 0 \text{ for } i = 2, \dots, k \quad (2.21)$$

for all points \mathbf{p}_j generated in the utopia plane.

In general, for the NC method, one can expect a better distribution of Pareto optimal points. However, in the case of almost vertical or almost horizontal sections of the Pareto optimal front, the distribution will be uneven. This effect can be toned down by using a normalization in objective space based on the nadir point (see Equation (2.11)) as proposed in the *normalized normal constraint method* [132]. Since the method may produce non-Pareto optimal points or locally Pareto optimal points, the authors also propose to use a Pareto filtering approach.

In [188], the authors present a new feature that guarantees the coverage of the whole Pareto optimal front by enlarging the area for distributing points \mathbf{x}_j in the utopia plane. The hypervolume that contains all possible solutions that are not dominated by the anchor points is projected onto the utopian plane to find the respective area of interest. Similarly to the epsilon-constraint method, the hypervolume is constructed based on the nadir point.

There exists a scalarization method, the *normal boundary intersection method* (NBI, [44]), that uses a similar strategy based on equality constraints instead of inequality constraints. The origin of both methods is the goal attainment method proposed by Gembicki [75, 76], which is an a priori strategy for MOO. In [188], it is observed that the NC method is more stable and less likely to generate locally optimal solutions than the NBI method. Furthermore, the NBI method, like the original version of the NC method, may miss parts of the Pareto optimal front.

In [62] the *directed search domain* (DSD) method is presented as a combination of NBI and NC method. The idea is to build a utopian hyperplane as in the NC method but to shrink the search domain. Points on the hyperplane are used as reference points constraining the search space. Then the search space is further reduced to a search cone. For some occasions, the cone has to be rotated or inverted to find Pareto optimal points. Again, Pareto filtering has to be used to avoid non-Pareto optimal points.

2.3. Hybrid Optimization Strategy

The scalarization methods presented in Section 2.2 lead to single-objective optimization problems. Moreover, only the global solution to these problems can lead to Pareto optimal solutions. As proposed in [158], the idea is to employ a hybrid optimization method for efficient single-objective optimization with a high chance of finding a global solution. The general strategy is to combine the advantages of gradient-based methods and heuristic methods. In general, optimization methods can be divided into *gradient-based methods* and *derivative-free methods*.

In gradient-based methods, the derivative of the objective function with respect to the design variable is evaluated and used in the optimization algorithm. As the explicit relation between these two values is often unknown, e.g., for PDE-constrained optimization, there exist different approaches to determine derivatives for optimization (see Section 3.1.3). Gradient-based methods usually ensure a fast convergence. The drawback of gradient-based methods is that they can get stuck at local optima. Therefore, they belong to the class of *local search methods*.

The following sections give a thorough introduction to gradient-based optimization algorithms, starting with basic strategies for unconstrained optimization in Section 2.3.1. Since the optimization problems arising from the scalarization methods are based on introducing additional constraints, the focus is primarily on constraint handling in gradient-based optimization algorithms in Section 2.3.2.

Optimization algorithms that make use of gradient calculations are usually *deterministic* approaches, which means that they will find the same solution when using the same starting value.

Contrary to that, derivative-free methods are usually not deterministic approaches. Instead, they are *heuristic*. Examples of heuristic methods are genetic algorithms, which were already introduced in Section 2.2.1, but also particle swarm optimization and simulated annealing. The advantage of heuristic methods is that they often have a higher chance of finding a globally optimal solution if they are not based on a specific direction in search space but aim at exploring the whole design space. Such methods are referred to as *global search methods*. Unfortunately, they require a lot of evaluations of the objective function and, therefore, a high computational effort. The computationally most expensive strategy for exploring the design space is to use an exhaustive search method that performs a systematical brute-force search in the design space and tries to evaluate all possible candidates. If this is not feasible, the method can be replaced by an explorative random or targeted sampling in design space. Random sampling can be interpreted as the simplest type of heuristic strategy.

The general idea of *hybrid* optimization strategies is to combine two or more different optimization strategies, for example, a gradient-based strategy and a heuristic strategy, such that it can be taken advantage of the desired features of each of the strategies during the optimization.

Hybrid optimization strategies for multi-objective optimization can be implemented at the level of the multi-objective optimization algorithm or, when using a scalarization method, at the level of the single-objective optimization algorithm.

2. Multi-Objective Optimization

A common approach for hybrid optimization strategies for MOO is to support an operation in a multi-objective evolutionary algorithm by a local search strategy. This results in a hybridization at the level of multi-objective optimization. Vicini and Quagliarella [275], for example, use a gradient-assisted multi-objective evolutionary algorithm for aerodynamic shape optimization to increase the performance. Bosman and de Jong [22] make use of descent directions in multi-objective space for hybridizing a multi-objective evolutionary algorithm. In their studies, they conclude that, regarding efficiency, multi-objective evolutionary algorithms with local search strategies are not as promising as a hybridization of evolutionary algorithms for single-objective optimization. The main problem in this context is that descent in multi-objective space may disturb the exploration capability of the multi-objective evolutionary method, making a construction of a successful hybridization more difficult.

The work in this thesis is restricted to hybrid optimization strategies for the constrained single-objective problems resulting from scalarization. Although the multi-objective problem is transformed into a series of single-objective optimization problems, the design space for the optimization is the same for all optimization problems. This feature can be exploited in the single-objective optimization approaches. When only gradient-based strategies are employed, the found Pareto optimal points can be used as starting values for the new optimization with altered constraint. A heuristic strategy for single-objective optimization can reuse the information of each optimization run. For the hybrid strategy that is used in the present work, the idea is to reuse a Kriging meta-model. It is built and updated during optimization based on the expected improvement method to reduce the computational effort for the non-deterministic search method. As a result, the design space can be efficiently explored and provide suitable starting values for a gradient-based optimization strategy. An overview of hybrid approaches and details on the method suggested in the context of this thesis are given in Section 2.3.3.

2.3.1. Gradient-Based Algorithms for Unconstrained Optimization

A general strategy for gradient-based optimization is to iteratively choose a search direction \mathbf{d}_j in each step j of the optimization algorithm with $j = 0, \dots, N$ and move into this direction for each iteration step. The corresponding update formula for each iteration j is then given by

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \gamma_j \mathbf{d}_j. \quad (2.22)$$

The step length γ_j can be determined by different strategies (see Section 2.3.1.4). The determination of a suitable step length γ_j in each iteration is referred to as *line search*, which is why methods based on a step length are referred to as *line search methods*. Note that there also exist other types of gradient-based optimization methods, e.g., *trust-region methods*.

The maximum number of iteration steps j is reached for an abort criterion. One abort criterion for the iteration can be a minimal value for the norm of the update $\|\mathbf{d}_j\|$. Furthermore, the necessary condition for optimality gives a practical condition. The optimization method should stop when $\|\nabla f(\mathbf{x}_j)\|$ reaches a value close to zero. This

value should be of higher or the same order than the numerical error of the gradient itself, which depends on the approximation quality of the objective function f .

2.3.1.1. Method of Steepest Descent

A simple choice for \mathbf{d}_j is the direction of the steepest descent, which is the negative gradient $-\nabla f(\mathbf{x}_j)$ of the objective function. The advantage of the *method of steepest descent* is that only the first derivative of the objective function has to be calculated. A major drawback of the steepest descent method is the slow convergence compared to other choices of the search direction \mathbf{d} .

2.3.1.2. Newton's Method

The search direction for Newton's method is derived from a second-order Taylor series. When the second-order derivative $\nabla^2 f(\mathbf{x}_j)$ is nonsingular and symmetric positive definite, the direction given by

$$\mathbf{d}_j := -(\nabla^2 f(\mathbf{x}_j))^{-1} \nabla f(\mathbf{x}_j) \quad (2.23)$$

is a descent direction. A locally quadratic order of convergence can be obtained with the help of Newton's method, such that a fast convergence behavior is observed if the starting value is close to the optimal solution. A major drawback of Newton's method is that the calculation of the second derivative can be computationally expensive. Therefore, one can turn to quasi-Newton methods where the Hessian is approximated in each iteration step.

2.3.1.3. Quasi-Newton Methods (BFGS, DFP)

In *quasi-Newton methods* the true Hessian $\nabla^2 f(\mathbf{x}_j)$ is approximated by H_j in each step. The search direction can be expressed as

$$\mathbf{d}_j = -H_j^{-1} \nabla f(\mathbf{x}_j). \quad (2.24)$$

In the BFGS⁶ method the approximation of the Hessian matrix H_{j+1} is given by

$$H_{j+1} = H_j + \frac{\mathbf{r}_j \mathbf{r}_j^\top}{\mathbf{r}_j^\top \mathbf{s}_j} - \frac{H_j \mathbf{s}_j \mathbf{s}_j^\top H_j}{\mathbf{s}_j^\top H_j \mathbf{s}_j}, \quad (2.25)$$

with $\mathbf{s}_j := \mathbf{x}_{j+1} - \mathbf{x}_j$ and $\mathbf{r}_j := \nabla f(\mathbf{x}_{j+1}) - \nabla f(\mathbf{x}_j)$. The BFGS update H_{j+1} is symmetric positive definite if H_j is a symmetric positive definite matrix and $\mathbf{s}_j^\top \mathbf{r}_j > 0$ (cp. [206, p.24]). The last condition is automatically satisfied when the Wolfe conditions (see Section 2.3.1.4) hold. For a symmetric positive definite Hessian the search direction \mathbf{d}_j is a descent direction. Instead of inverting H_j , it is also possible to directly update the inverse by the formula

$$H_{j+1}^{-1} = H_j^{-1} + \frac{\mathbf{s}_j \mathbf{s}_j^\top}{\mathbf{r}_j^\top \mathbf{s}_j} - \frac{\mathbf{s}_j \mathbf{r}_j^\top H_j^{-1} + H_j^{-1} \mathbf{r}_j \mathbf{s}_j^\top}{\mathbf{r}_j^\top \mathbf{s}_j} + \frac{(\mathbf{r}_j^\top H_j^{-1} \mathbf{r}_j)(\mathbf{s}_j \mathbf{s}_j^\top)}{(\mathbf{r}_j^\top \mathbf{s}_j)^2}. \quad (2.26)$$

⁶Broyden, Fletcher, Goldfarb and Shanno

2. Multi-Objective Optimization

A similar method to the BFGS method is the DFP⁷ method. The direct DFP update for the inverse of the Hessian is given by

$$H_{j+1}^{-1} = H_j^{-1} + \frac{\mathbf{s}_j \mathbf{s}_j^\top}{\mathbf{r}_j^\top \mathbf{s}_j} - \frac{H_j^{-1} \mathbf{r}_j \mathbf{r}_j^\top H_j^{-1}}{\mathbf{r}_j^\top \mathbf{s}_j}. \quad (2.27)$$

Quasi-Newton methods are able to achieve locally superlinear convergence. General experience shows that the DFP method is not as efficient as the BFGS method. As an example, the DFP method is not as good as the BFGS method in self-correcting bad approximations of the Hessian (cp. [206, p.142]).

The optimization algorithm for quasi-Newton methods is given in Algorithm 4.

Algorithm 4: Quasi-Newton method.

```

j = 0, ε1 > 0, ε2 > 0, H0 = I
while ||∇f(xj)|| > ε1 and ||xk+1 - xk|| > ε2 do
  Find a suitable γj
  xj+1 = xj - γjHj-1∇f(xj)
  sj = xj+1 - xj
  rj = ∇f(xj+1) - ∇f(xj)
  calculate Hj+1-1 with the updating formula (2.26) or (2.27)
  j = j + 1
end while

```

2.3.1.4. Line Search

The step length γ_j can be found by solving the one-dimensional optimization problem

$$\min_{\gamma_j} f(\mathbf{x}_j + \gamma_j \mathbf{d}_j). \quad (2.28)$$

Usually, it is computationally expensive to solve a minimization problem in each step of the optimization algorithm, i.e to perform an exact line search. Alternatively, one may choose γ_j such that the Wolfe conditions are fulfilled. The Wolfe conditions are given in [206, p.33] for an inexact line search for γ_j as

$$f(\mathbf{x}_j + \gamma_j \mathbf{d}_j) \leq f(\mathbf{x}_j) + c_1 \gamma_j \nabla f(\mathbf{x}_j)^\top \mathbf{d}_j, \quad c_1 \in (0, 1) \quad (2.29)$$

and

$$\nabla f(\mathbf{x}_j + \gamma_j \mathbf{d}_j)^\top \mathbf{d}_j \geq c_2 \nabla f(\mathbf{x}_j)^\top \mathbf{d}_j, \quad c_2 \in (c_1, 1). \quad (2.30)$$

Equation (2.29) is the *sufficient decrease condition* or *Armijo condition* and (2.30) is referred to as the *curvature condition*. For Newton's method or quasi-Newton methods typical values for the constants are $c_1 = 0.0001$ and $c_2 = 0.9$.

⁷Davidon, Fletcher and Powell

For Newton's method or a method of steepest descent, it is sufficient to use a backtracking algorithm to fulfill the first Wolfe condition by reducing the step length γ_j until the condition is guaranteed. The simplest example of a backtracking algorithm is to use a bisection strategy for finding γ_j , presented in Algorithm 5.

Algorithm 5: Backtracking Algorithm for finding step length.

```

 $\gamma > 0, c = 10^{-4}$ 
while  $f(\mathbf{x}_j + \gamma \mathbf{d}_j) > f(\mathbf{x}_j) + c\gamma \nabla f(\mathbf{x}_j)^\top \mathbf{d}_j$  do
     $\gamma = 0.5\gamma$ 
end while
 $\gamma_j = \gamma$ 

```

Alternatively, one may find a suitable γ_j iteratively with a quadratic or higher-order approximation of the objective function $f(\gamma_j)$, which is to be optimized in (2.28). Quasi-Newton methods need a backtracking algorithm that ensures that both Wolfe conditions are satisfied. An example of a backtracking line search for quasi-Newton methods is given in [206, p.60ff].

2.3.2. Gradient-Based Algorithms for Constrained Optimization

As constraint methods for multi-objective optimization require the solution of several constrained optimization problems, it is important to discuss the issue of constraint handling in gradient-based optimization methods. Penalty methods and augmented Lagrangian methods will be introduced as classical strategies for formulating unconstrained optimization problems with additional penalty parameters. The idea of augmented Lagrangian methods will also become important in Chapter 3. The sequential quadratic programming method and the interior point method are introduced to present alternative strategies for constraint handling.

2.3.2.1. Penalty Methods

The idea of *penalty methods* is to transform the constrained optimization problem into an unconstrained problem with an additional penalty parameter. A classical strategy, which goes back to Courant [42], is to penalize constraint violation by adding a quadratic penalty term, such that the augmented objective function f^a is given by

$$f^a(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2}\rho \|h(\mathbf{x})\|_2^2, \quad (2.31)$$

with $\rho > 0$. Convergence to a stationary point in a bounded design space can be shown for sufficiently large ρ (see [63]). It can be achieved algorithmically by choosing a strictly increasing sequence of ρ_k . Alternatives to the quadratic penalty function are, for example, the l_1 penalty function. If inequality constraints $g(\mathbf{x}) \leq \mathbf{0}$ are involved, penalty terms are based on the l_1 -norm or l_2 -norm of $\max\{0, g(\mathbf{x})\}$ to penalize constraint violation. Note, however, that such penalty terms are not twice continuously differentiable. Line search methods or trust region method can be used to solve for optimality of f^a .

2. Multi-Objective Optimization

2.3.2.2. Augmented Lagrangian Methods

In *augmented Lagrangian methods* [117, 222] the original Lagrangian is augmented by a penalty term. An augmented Lagrangian L^a with a quadratic penalty term is, for example,

$$L^a(\mathbf{x}, \boldsymbol{\mu}, \rho) = f(\mathbf{x}) + \boldsymbol{\mu}h(\mathbf{x}) + \frac{1}{2}\rho\|h(\mathbf{x})\|_2^2. \quad (2.32)$$

An unconstrained subproblem

$$\min_x L^a(\mathbf{x}, \boldsymbol{\mu}_k, \rho_k) \quad (2.33)$$

is then solved in an iterative fashion for fixed ρ_k and $\boldsymbol{\mu}_k$. The values for ρ and $\boldsymbol{\mu}$ have to be chosen carefully. The value for ρ shall be neither too small nor too large to avoid an ill-conditioned Hessian matrix. The speed of convergence to the optimum solution strongly depends on the speed of convergence of $\boldsymbol{\mu} \rightarrow \boldsymbol{\mu}^*$. This means that in augmented Lagrangian methods, similarly to penalty methods, special care has to be taken for updating the additional parameters. Furthermore, the risk of running into an ill-conditioned system is given. A possible updating scheme is to update $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \rho h(\mathbf{x})$ if the equality constraint is small enough and increase ρ if this is not the case. Additional inequality constraints may be treated analogously to the equality constraints by introducing slack variables (see 2.3.2.4) or using the penalty term introduced in the penalty method.

One may refer to f^a and L^a as *penalty functions* or *merit functions*. The following definition introduces the notion of an *exact penalty function* (see e.g. [206, p.435]), which will be of importance in Chapter 3.

Definition 12 (Exact Penalty Function). *A penalty function $p^a(\mathbf{x}, \boldsymbol{\mu})$ is exact if there exist multipliers $\boldsymbol{\mu} > 0$ such that for $\boldsymbol{\mu}$ being component-wise large enough any local solution of the original optimization problem (2.2) is a local minimizer of p^a .*

It can be shown that the penalty method based on the l_1 -norm provides an exact penalty function for a given choice of $\boldsymbol{\mu}$ (cp. [206, Theorem 17.3]). In the augmented Lagrangian method, the Hessian of $L^a(\mathbf{x}^*, \boldsymbol{\mu}, \rho)$ is positive definite for sufficiently large ρ (cp. [51]) if the reduced Hessian of the original problem is positive definite and the stationary points of L^a and the original problem are the same. Thus, if the second-order sufficient condition for optimality is fulfilled, the strict locally optimal solution of the original problem is also a strict local minimizer of L^a . As a result, L^a is an exact penalty function.

2.3.2.3. Sequential Quadratic Programming

The *sequential quadratic programming method* (SQP) was firstly introduced by Wilson in 1963 and was made popular by Han (1977, [104]) and Powell (1978, [223]). Instead

of solving the constrained optimization problem (2.2), the corresponding quadratic programming problem

$$\begin{aligned} \min_{\mathbf{d}} \quad & \nabla f(\mathbf{x})^\top \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \nabla_{\mathbf{x}\mathbf{x}}^2 L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\eta}) \mathbf{d} \\ \text{s.t.} \quad & h(\mathbf{x}) + \nabla h(\mathbf{x})^\top \mathbf{d} = \mathbf{0}, \quad g(\mathbf{x}) + \nabla g(\mathbf{x})^\top \mathbf{d} \leq \mathbf{0} \end{aligned} \quad (2.34)$$

is solved. Equation (2.34) is solved in an iterative algorithm for \mathbf{d}_k at a point \mathbf{x}_k . One distinguishes between approaches for inequality and equality constrained optimization problems. For inequality constraints the problem has to be solved with an algorithm for quadratic programming for each iteration step. Popular methods used in this context are active set strategies. When using equality constraints, \mathbf{d}_k can be found by solving the linear system of equations

$$\begin{pmatrix} H_k(\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\eta}_k) & \nabla h(\mathbf{x}_k)^\top \\ \nabla h(\mathbf{x}_k) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d}_k \\ \boldsymbol{\lambda}_{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla f(\mathbf{x}_k) \\ -h(\mathbf{x}_k) \end{pmatrix} \quad (2.35)$$

under the condition that H_k is positive definite and $\nabla h(\mathbf{x})$ has full rank. The result \mathbf{d} serves as a direction of descent to update the design \mathbf{x} by $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma \mathbf{d}_k$. The step length γ is determined by an appropriate line search suitable for constrained optimization, for example, using penalty functions or augmented Lagrange functions. The matrix H_k is either an approximation of the Hessian of the Lagrange function of the problem (realized, for example, by a damped BFGS update [223]) or the exact Hessian in the case of Newton's algorithm. The system of equations (2.35) is equivalent to the system of equations that results when applying a Newton algorithm to the KKT condition where $\mathbf{d}_k = \Delta \mathbf{x}_k$, so that (2.34) can be seen as a linearization of the KKT condition (cp. [206, p.530ff]). Powell [223] states that locally a superlinear rate of convergence can be guaranteed under some regularity assumptions and for a positive definite approximation of the Hessian. An SQP strategy is, for example, implemented in SNOPT [82] or in SciPy's SLSQP method [149].

2.3.2.4. Interior Point Method

In the *interior point method* [63] or *barrier method* inequality constraints are transformed into equality constraints using slack variables. The constraint function $g(\mathbf{x}) \leq \mathbf{0}$ becomes $g(\mathbf{x}) + \mathbf{s} = \mathbf{0}$ with $\mathbf{s} \geq \mathbf{0}$. As a result, the design space is augmented by the slack variables $\mathbf{s} \in \mathbb{R}^q$.

Problems of the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h(\mathbf{x}) = \mathbf{0}, \quad g(\mathbf{x}) + \mathbf{s} = \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0} \end{aligned} \quad (2.36)$$

are then transformed into the barrier function problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{s}} \quad & \phi(\mathbf{x}, \mathbf{s}) := f(\mathbf{x}) - \beta \sum_{i=1}^q \ln(s_i) \\ \text{s.t.} \quad & h(\mathbf{x}) = \mathbf{0}, \quad g(\mathbf{x}) + \mathbf{s} = \mathbf{0}. \end{aligned} \quad (2.37)$$

2. Multi-Objective Optimization

The barrier problem may be solved using a damped Newton strategy to solve the KKT system of (2.37). As the KKT system is also often referred to as the primal-dual system, the strategy is often also called primal-dual interior point method. The barrier parameter β is sequentially reduced during optimization leading to new KKT systems. Convergence is observed for $\beta \rightarrow 0$ (see [63]). Opposed to the penalty methods, where ρ is increased during optimization, and the solution is typically approached from the exterior of the feasible set, the interior point method will remain in the interior of the feasible set of the inequality constraints. The use of the logarithmic barrier with $s \geq 0$ prevents the algorithm from finding points outside of the feasible set. Therefore, interior point methods are especially useful in handling inequality constraints.

An interior point approach is implemented in IPOPT (Interior Point OPTimizer, [277]). The authors use a line search filter strategy [276] for finding an admissible step size. The idea is to use a line search that either improves $\phi(\mathbf{x}, \mathbf{s})$ or the constraint violation $\|(h(\mathbf{x}), g(\mathbf{x}) + \mathbf{s})^\top\|$ in a suitable vector norm resulting in a bi-objective optimization problem, which is treated using the concept of Pareto dominance. A filter storing the non-dominated vector pairs of the objective function and constraint violation is updated during the optimization. Only points that are not dominated by any point in the filter are accepted in the course of the optimization algorithm. Furthermore, the line search strategy is chosen in such a way that a sufficient reduction in the objective function or the constraint violation, as well as feasibility ($\mathbf{s} \geq \mathbf{0}$), are guaranteed. The filter method can be applied to SQP methods or the equality constraints in interior-point methods, as shown in [276].

2.3.3. Hybridization

The advantages and disadvantages of gradient-based local search methods and heuristic global search methods were already discussed at the beginning of this section.⁸ They are summarized in the following:

- **Speed of convergence:** Gradient-based methods allow a fast convergence, especially when using Newton's method. Quasi-Newton methods can also achieve superlinear convergence. For PDE-constrained optimization problems, the efficiency can be further increased when using one-shot optimization strategies. Convergence of heuristic methods can only be proven in a probabilistic setting. The speed of convergence of such methods is rather slow.
- **Stopping criterion:** Stopping criteria for gradient-based optimization strategies can be clearly defined based on optimality conditions like a vanishing gradient. For heuristic approaches, there usually does not exist a clear stopping criterion.
- **Computational efficiency:** Gradient-based optimization methods need, next to a function evaluation, the evaluation of first-order derivatives and depending on

⁸It is important to emphasize that the distinction between gradient-based and derivative-free methods, and local and global search method is not unique. However, the present work is restricted to gradient-based strategies that are used for local search as well as derivative-free methods that are used for global search. The derivative-free strategy is based on a heuristic approach.

the method also second-order derivatives. Obtaining derivatives can potentially be computationally expensive, especially for PDE-constrained optimization problems with a high number of design variables. However, when using adjoint methods, the computational effort is comparable to a function evaluation. Heuristic approaches can be used for non-differentiable or discrete functions since they only need the function value. However, they require a high number of function evaluations to converge. In genetic algorithms, for example, each member of the population has to be evaluated in each step of the algorithm.

- **Implementation:** Gradient-based optimization strategies need the derivative of the objective and constraint functions with respect to the design variable. When using a black-box approach for obtaining derivatives, e.g., finite difference, the computational costs can be high. Adjoint approaches can overcome this problem but need to be implemented. Therefore, gradient-based strategies are often abandoned. Algorithmic differentiation, however, helps in setting up adjoint approaches in a semi-automatic fashion and, as a result, allows the semi-automatic transition of a simulation code to an optimization code. The implementation of heuristic methods is straightforward since the simulation tool can be treated as a black-box.
- **Global optimum:** Gradient-based optimization strategies can easily get stuck in local optima. Heuristic methods have a higher chance of finding the global optimum since they explore the design space.
- **Algorithmic robustness/reliability :** Since gradient-based optimization approaches are based on a single iterative search in design space, they have a higher chance of failing. Reasons for this can be that a design can fail to converge, derivatives are inaccurate, or, in the worst case, derivatives cannot be provided. Moreover, if the objective function is noisy, a gradient does not provide a good search direction. Heuristic approaches, on the contrary, are reliable to provide a good solution as they will not stop if a design does not converge.

One can establish a hybridization that uses two different methods for solving one optimization problem to combine the advantages of the different types of algorithms. A straightforward strategy, which will be pursued in this work, is to use these methods sequentially. This is also referred to as *pipelining*. Another idea is to apply each of the methods in situations where they are the most suitable, e.g., by dividing the design or solution space. In some cases, it might be interesting to allow cooperation between the algorithms by sharing information on the solution space or the search space. One option for hybridization can be to insert ideas from one method as an operator into the other method.

Several approaches can be used when hybridizing local and global search methods in this context. A pipelining strategy is used in [294]. It changes between a genetic algorithm and a gradient-based optimization strategy to avoid getting stuck in a local optimum while at the same time reducing the computational costs. In [295], a genetic algorithm is hybridized by introducing new genetic operators for reproduction using the

2. Multi-Objective Optimization

simplex method, which is a derivative-free local search method. A similar strategy is used in [227] with a quasi-Newton method applied for a local search to improve some members of the population.

2.3.3.1. General Strategy

The concept that is employed in the present thesis is a sequential hybridization that first uses a heuristic global search strategy followed by a gradient-based optimization strategy. The order of construction is straightforward and represents a popular pipelining strategy. Heuristic methods have a better chance to find a globally optimal solution, but they especially need a high number of function evaluations when refining the solution near an optimum. Instead of demanding a high level of approximation for the heuristic method, the idea is to switch to a gradient-based strategy that uses the result of the heuristic approach as a starting value for refining the solution locally near the optimum.

The main aspect of the hybridization in the present work is to perform the global search on a surrogate model. Since the function evaluations are cheap, especially when compared to solving a PDE for each objective function, this will reduce the computational costs of the global search strategy to the costs needed to construct the surrogate model and, if needed, the costs for refining it during the optimization. As a result, surrogate models are often used in combination with global search methods.

One may distinguish between three different types of *surrogate models* (also referred to as *metamodels*): regression or interpolation-based models, projection-based models (e.g. reduced-order models) and hierarchical models. Regression models treat the original model as a black-box.⁹ Examples of regression models are quadratic response surface models, Kriging models, and radial basis function regression. The method of choice for the hybrid optimization strategy in the present work is Kriging, which is presented in Section 2.3.1.1. Kriging is a popular choice and has proven to be successful in various applications. A minor drawback is that it is only feasible for a moderate number of design variables, i.e., a few dozens. The global search method on the surrogate model is the expected improvement method (see Section 2.3.3.3). The method is suitable for design space exploration and exploitation. In [146], the author uses a Kriging surrogate model for approximating the objective functions that arise from a scalarization approach for multi-objective optimization. The scalarization method is a variant of compromise programming with weight scanning. An expected improvement method is used for a global search supported by a genetic algorithm. As a result, the authors show that the technique outperforms NSGA-II.

As a gradient-based strategy, the method of choice in the present work is either the interior-point optimizer implemented in IPOPT or the one-shot strategy that is presented in Chapter 3. Both make use of derivatives obtained from a robust discrete adjoint approach. Note that the constraint handling has to be a special focus for all optimization strategies.

A significant task of the hybridization is to find the point when the local search algorithm should be applied. In general, it is not possible to judge whether a heuristic

⁹Regression models are sometimes also referred to as response surface models or approximation models.

method is close to a global optimal solution. An indicator for reaching a locally or globally optimal solution can be that the change in the function value has reached a lower bound. For the specific strategy employed in the present work, which is based on a surrogate model, a second criterion is the approximation quality of the function value itself. If the objective function does not change significantly with an acceptable approximation quality, the algorithm has found an optimal solution on the surrogate model. The Kriging surrogate model does not only predict a function value but also provides a level of confidence for the prediction. This value can be used to assess the approximation quality. When a satisfying quality of approximation or a maximum number of function evaluations is reached, the strategy of the presented approach is to stop the refinement of the surrogate model. A major advantage of using a surrogate model is that an intensive search, e.g., based on random sampling in design space, can be employed to find the global optimum during the iterations of the expected improvement methods. Especially, the intensive search can be performed on the fixed surrogate model when stopping the expected improvement method. Furthermore, since the design space is the same for all constrained single-objective optimization problems, the Kriging surrogate model can be reused for each optimization run. It will need less function values for later optimization runs, such that the maximum number of function evaluations can be distributed strategically.

Furthermore, the gradient-based optimization algorithm needs to produce iterates in design space that are close to the design proposed by the global search. This is possible by restricting the design space or the norm of the design update. The restriction is especially required at the beginning of the algorithm. The former is used in the present work since it does not intervene with the general procedure of the algorithm.

The resulting hybrid strategy for constrained single-objective optimization is summarized in the following:

1. **DoE:** In the preliminary step an initial design of experiments (DoE) of size n_1 is performed.
2. **Expected improvement method:** The Kriging model is constructed and refined with the help of the expected improvement method with constraint handling. The expected improvement method stops
 - when a fixed number of refinements n_2 is reached, or
 - for feasible designs $\mathbf{x} \in \mathcal{D}$ with different minimum objective function values f_{\min} the change for successive iterations is small, i.e., $|f_{\min}^{(j)} - f_{\min}^{(j+1)}| \leq \epsilon_1$, or
 - $|\hat{f}_{\min}^{(j)} - \hat{f}_{\min}^{(j+1)}| \leq \epsilon_1$ for a change in the prediction and $\hat{s}^2 \leq \epsilon_2$ for the mean squared error \hat{s} of the prediction.
3. **Explorative search:** A further explorative search on the Kriging model based on random sampling is run to find the optimal solution \mathbf{x}^K on the Kriging surrogate model.
4. **Gradient-based search:** The solution \mathbf{x}^K is chosen as a starting value for a gradient-based search with $\mathbf{x} \in \mathcal{D} \cap \{\mathbf{x} \mid x_i^K - \delta \leq x_i \leq x_i^K + \delta \text{ with } i = 1, \dots, n\}$.

2. Multi-Objective Optimization

Details on the Kriging model and the expected improvement method with constraint handling are presented in the following sections.

2.3.3.2. Kriging

Kriging [152, 181], also referred to as Gaussian process regression, can be used to approximate highly nonlinear functions. In contrast to least squares regression, the Kriging method will not smooth the approximation at data points, but it interpolates the data points. The general idea of Kriging is to interpolate function values at specific points by using weights for given nodes. The nodes that are closer to the point have higher weights. The weighting factors are determined by considering a statistical variation over the distance of the nodes. The main advantage of a Kriging model is that it provides estimates for function values as well as for the modeling error at unsampled points. To model such kind of behavior, the output of a function is described by assuming a regression model as an underlying global trend function with an additional error term represented by a random process. The initial data acquisition can be made, for example, by using Latin Hypercube sampling (LHS, [184]). Then the Kriging model is trained.

Given a design vector $\mathbf{x} \in \mathbb{R}^n$ and M samples in design space, for this particular section, let the output vector be defined as $\mathbf{y} \in \mathbb{R}^M$. It comprises all evaluations of a function f , for example an objective function or a constraint function, such that $y_i = f(\mathbf{x}^{(i)})$ for $i = 1, \dots, M$. The global trend function can be expressed as $\beta(\mathbf{x})$, such that the general modeling assumption is

$$Y(\mathbf{x}) = \beta(\mathbf{x}) + Z(\mathbf{x}), \quad (2.38)$$

where Y and Z describe random processes. The global trend function is a constant β_0 for *ordinary Kriging* or a linear function for *universal Kriging*. In the following, it is focused on ordinary Kriging. The error term $Z(\mathbf{x})$ is assumed to be a Gaussian random process with the expectation function $E(Z(\mathbf{x})) = 0$ and the covariance function $C(Z(\mathbf{x}), Z(\mathbf{x}')) = \sigma^2 R(\mathbf{x}, \mathbf{x}') \geq 0$ (see also Section 4.1). The correlation function R , which describes how the error at a given design point $\mathbf{x}^{(i)}$ is correlated with other design points $\mathbf{x}^{(j)}$, can, for example, be prescribed as

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \prod_{k=1}^M e^{-\theta_k |x_k^{(i)} - x_k^{(j)}|^{\gamma_k}}$$

with $\theta_k \geq 0$ and $0 < \gamma_k \leq 2$. Note that the values for θ_k and γ_k can be determined using maximum likelihood estimates to fit the model to the samples. The underlying optimization can, for example, be done using a genetic algorithm.

The optimal predictor that minimizes the mean squared prediction error is the conditional expectation (see [43]). Therefore, the best linear, unbiased predictor of the model, is

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^\top(\mathbf{x}) \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\beta}) \quad (2.39)$$

with $\mathbf{r}(\mathbf{x}) = (R(\mathbf{x}, \mathbf{x}^{(1)}), \dots, R(\mathbf{x}, \mathbf{x}^{(M)}))^\top$, the correlation matrix $\mathbf{R} \in \mathbb{R}^{M \times M}$, where $R_{i,j} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and the least squares estimate

$$\hat{\beta} = (\mathbf{1}^\top \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^\top \mathbf{R}^{-1} \mathbf{y}. \quad (2.40)$$

The maximum likelihood estimate of σ is

$$\sigma^2 = \frac{1}{M} (\mathbf{y} - \mathbf{1} \hat{\beta})^\top \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\beta}). \quad (2.41)$$

Equation (2.39) can be rewritten as a linear predictor

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \sum_{i=1}^M w^{(i)} R(\mathbf{x}, \mathbf{x}^{(i)})$$

with weights $(w^{(1)}, \dots, w^{(M)}) = \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\beta})$, which can be found by solving a linear system of equations. This reformulation mirrors the general idea of Kriging as an interpolation method. The correlation function $R(\mathbf{x}, \mathbf{x}^{(i)})$ can be interpreted as a Gaussian radial basis function centered at a sample point $\mathbf{x}^{(i)}$.

The mean squared error of the prediction can be expressed accordingly as

$$\hat{s}^2(\mathbf{x}) = \sigma^2 \left[1 - (\mathbf{1} \quad \mathbf{r}^\top(\mathbf{x})) \begin{pmatrix} 0 & \mathbf{1}^\top \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}(\mathbf{x}) \end{pmatrix} \right]. \quad (2.42)$$

Kriging is a highly popular method for function approximation, which has been further developed in recent years. In *direct gradient-enhanced Kriging* [159, 105], for example, the predictor additionally consists of the weighted sum of gradients of the output with respect to the input variables. The use of gradients improves the accuracy of the Kriging model. Issues concerning scalability of the correlation matrix and ill-conditioning for noise data can often be overcome, although this requires, for example, the introduction of regularization parameters, which need to be tuned.

2.3.3.3. Expected Improvement Method

The Kriging model can be used for the optimization of an objective function f . In the following, the *expected improvement method* (EI), which was originally proposed by Mockus et al. [194], is outlined. It can be used for adaptive sampling during optimization using the above described Kriging model and belongs to the class of Bayesian optimization methods. The idea of the expected improvement method is to exploit the given model for finding a minimum while at the same time exploring the search space. The improvement I for a sample \mathbf{x} with prediction \hat{f} and mean squared error $\hat{s}_f^2 \neq 0$ can be expressed as

$$I(\mathbf{x}) = \begin{cases} f_{\min} - \hat{f}, & \text{for } \hat{f} < f_{\min} \\ 0, & \text{otherwise.} \end{cases} \quad (2.43)$$

2. Multi-Objective Optimization

Here, f_{\min} is the minimum value of f found so far.¹⁰ Then, the expected improvement is derived as

$$E(I(\mathbf{x})) = \begin{cases} (f_{\min} - \hat{f}) \cdot F_{\tilde{Z}}\left(\frac{1}{\hat{s}_f}(f_{\min} - \hat{f})\right) + \hat{s}_f \cdot f_{\tilde{Z}}\left(\frac{1}{\hat{s}_f}(f_{\min} - \hat{f})\right), & \text{for } \hat{s}_f > 0 \\ 0, & \text{for } \hat{s}_f = 0, \end{cases} \quad (2.44)$$

where $F_{\tilde{Z}}$ and $f_{\tilde{Z}}$ are the cumulative distribution function and the probability density function of the standard normal distribution. The derivation is done by using the normalization $u = \hat{s}_f^{-1}(f_{\min} - \hat{f})$ and $v = \hat{s}_f^{-1}(f(\mathbf{x}) - \hat{f})$. As a result, u and v are standard normally distributed and by employing the rules of probability theory one obtains $\int_{-\infty}^u f_{\tilde{Z}}(v)dv = F_{\tilde{Z}}(u)$ and $\int_{-\infty}^u v f_{\tilde{Z}}(v)dv = -f_{\tilde{Z}}(u)$ (see for example [240], Section 4.3.1.2).

The aim of the expected improvement method is to maximize the expected improvement, i.e., solve the problem

$$\max_{\mathbf{x} \in \mathbb{R}^n} E(I(\mathbf{x})) \quad (2.45)$$

in each iteration of the expected improvement method. The expected improvement can be maximized using, for example, an explorative search on the surrogate model utilizing random sampling in design space. Promising points with a large value of expected improvement are evaluated for refining the surrogate model. This is a typical procedure in Bayesian optimization and is referred to as *adaptive sampling* using the expected improvement as an infill criterion.

If further constraints $g_i(\mathbf{x}) > 0$ for $i = 1, \dots, p$ are involved¹¹, according to Schonlau [240] an alternative expected improvement can be expressed as

$$E(IC_{WEI}(\mathbf{x})) = E(I(\mathbf{x})) \cdot P(\{\mathbf{x} \mid g(\mathbf{x}) > 0\}) = E(I(\mathbf{x})) \cdot \prod_{i=1}^r F_{\tilde{Z}}\left(\frac{\hat{g}_i}{\hat{s}_{g_i}}\right). \quad (2.46)$$

Here, $P(\{\omega\})$ denotes the probability of an outcome ω . The last equality requires the statistical independence of the constraint functions g_i for $i = 1, \dots, r$. One may refer to this approach as the *constrained-weighted expected improvement* (CWEI) method.

If at least one constraint is violated for all full model evaluations, Gelbart et al. [74] propose to optimize for the probability of constraint satisfaction first. As a result, the probability of constraint satisfaction is used as a criterion for an explorative feasibility search.

Next to the CWEI method, the present work also investigates two other methods for constraint handling. Note that both methods do not include the probability of constraint satisfaction in the function to be maximized. In the naive approach only the predicted

¹⁰Note that a more general form was introduced in [240], where $f_{\min} - f(\mathbf{x})$ is replaced by $(f_{\min} - f(\mathbf{x}))^g$ and for larger g the search is more global. Here, $g = 1$ is used.

¹¹For $g_i(\mathbf{x}) \leq 0$ the probability of constraint satisfaction is $1 - F_{\tilde{Z}}((\hat{s}_g)^{-1}\hat{g})$.

constraint values are considered, such that

$$E(I_N(\mathbf{x})) = \begin{cases} E(I(\mathbf{x})) & \text{if } \hat{g}_i(\mathbf{x}) > 0 \text{ for } i = 1, \dots, r \\ 0, & \text{otherwise.} \end{cases} \quad (2.47)$$

This approach neglects the additional information of the probability distribution that is provided by the Kriging model. The constraints can be further relaxed. For the present work it is proposed to relax by a given \hat{s}_{g_i} -region for each constraint g_i with $i = 1, \dots, r$ to account for the underlying Kriging model.

Furthermore, an approach is investigated that extends the naive approach to using the probability distribution, such that

$$E(I_{N,Ext}(\mathbf{x})) = \begin{cases} E(I(\mathbf{x})) & \text{if } P(\{\mathbf{x} \mid g_i(\mathbf{x}) > 0\}) > 1 - \gamma \text{ for } i = 1, \dots, r \\ 0, & \text{otherwise.} \end{cases} \quad (2.48)$$

Here, $0 < \gamma \in \mathbb{R}$ is a parameter describing the confidence region of constraint satisfaction.

There exist several other possibilities for constraint handling in the expected improvement method. The strategy proposed in [248] and refined in [26] is to condense all expensive constraint functions into a single expression, i.e., the maximum feasibility violation v_{\max} , and to switch between maximizing the expected improvement in satisfying $v_{\max} \leq 0$ and solving the original expected improvement problem with the constraint $\hat{v}_{\max} \leq 0$. As a result, only one additional Kriging model has to be trained. However, the constraint $v_{\max} \leq 0$ can be difficult to fulfill. In [87], the authors propose to use an augmented Lagrangian method and build a Kriging model separately for all parts of the augmented Lagrangian. The use of penalty methods or augmented Lagrangian methods is an interesting option for constraint handling, which also allows for a proper handling of equality constraints. However, the adjustment of penalty parameters in this context is not well-defined, and a fixed parameter may not necessarily lead to satisfactory solutions.

2.4. Application and Results

Throughout this thesis, the proposed strategies will be applied to an exemplary shape optimization problem. It shall merely serve to demonstrate and analyze all research aspects. The chosen problem is the aerodynamic shape optimization of a two-dimensional airfoil in transonic flow. This application case comprises all needed characteristics:

- The problem can be formulated as a multi-objective optimization problem to minimize the drag and maximize the lift of the airfoil.
- The flow around the airfoil is described by a system of PDEs, resulting in a PDE-constrained optimization problem.
- The number of design variables that are usually used in this context is large enough to present the use of adjoint strategies and connected optimization methods, and also small enough to allow the use of the expected improvement method as a global optimization strategy in the hybrid approach.

2. Multi-Objective Optimization

- An important research aspect of aerodynamic shape optimization is the optimization under uncertainties.

In the following, the aerodynamic shape optimization problem is presented in Section 2.4.1. The described strategy is implemented in the context of the open-source software SU2. Details on the implementation are described in Section 2.4.2. Section 2.4.3 shows the results of the application to aerodynamic shape optimization.

The application of the multi-objective optimization strategy for the shape optimization for an internal, three-dimensional flow problem is shown in Section A.1.3 of the appendix. In the main part of the thesis, external flow problems are considered.

2.4.1. Aerodynamic Shape Optimization Problem

The present section describes the aerodynamic shape optimization problem that is used as an application test case in this thesis. The focus lies on different modeling aspects, namely the formulation of the underlying equations, the definition of objective functions and constraints, as well as aspects of parameterization of the shape. This section shall only give a short overview. The intend is not to go into any details for modeling or discretizing the underlying equations since this is not the scope of this work. Note that all matters concerning optimization with an underlying PDE constraint will be discussed in Chapter 3.

2.4.1.1. Underlying Partial Differential Equations

The balance equations for fluid flow form a system of partial differential equations. In its generalized form, it is usually referred to as the compressible Navier-Stokes equations. Without considering any body forces they can be written in their differential form using the Einstein notation as

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_i}{\partial \xi_i} = 0, \quad (2.49a)$$

$$\frac{\partial \rho v_j}{\partial t} + \frac{\partial}{\partial \xi_i} (\rho v_i v_j + \tau_{ij}) + \frac{\partial p}{\partial \xi_j} = 0, \quad (2.49b)$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial \xi_i} (\rho v_i E + v_j \tau_{ij} + v_i p + Q_i) = 0, \quad (2.49c)$$

with the viscous components τ_{ij} with $i, j \in \{1, 2, 3\}$ of the stress tensor and the heat flux Q_i for $i = 1, 2, 3$. Note that ξ_i stands for the components of the spatial coordinates $\boldsymbol{\xi}$ and v_i for the components of the velocity \mathbf{v} . The temporal variables is t and the density and the pressure are given by ρ and p . Equation (2.49a) is the equation of mass conservation, Equation (2.49b) gives the three equations that balance momentum for $j = 1, 2, 3$, and Equation (2.49c) is the equation for the balancing energy. The energy E can be described as the sum of the internal energy e and the kinetic energy, such that

$$E = e + \frac{1}{2} \|\mathbf{v}\|_2^2.$$

The system of PDEs (2.49) is usually defined on a domain with prescribed boundary conditions and initial conditions. In the present context the flow is modeled as a thermally and calorically perfect gas with $p = \rho RT$ and $e = c_v T = (c_p + R)T$. Here, T is the temperature, R is the gas constant and c_v and c_p are the specific heat capacities. The coherence of p and ρ is then given by

$$p = (\gamma - 1)\rho e,$$

with the isentropic index $\gamma = \frac{c_p}{c_v}$.

The calculated flow is often modeled under the assumption of a Newtonian fluid, which means that the viscous stresses are modeled as

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial \xi_j} + \frac{\partial v_j}{\partial \xi_i} - \frac{2}{3} \frac{\partial v_k}{\partial \xi_k} \delta_{ij} \right),$$

with the Kronecker symbol δ_{ij} .

A dimensionless number describing the flow speed is the Mach number, which is locally defined as

$$Ma = \frac{\|\mathbf{v}\|}{c}, \quad (2.50)$$

with the speed of sound $c = \sqrt{\gamma RT}$.

The flows considered in the present work are steady and transonic, i.e., flows with a velocity around the speed of sound. Such a flow can be described by the steady, compressible Euler equations, which model an inviscid flow. They are given by the conservation equations

$$\frac{\partial \rho v_i}{\partial \xi_i} = 0, \quad (2.51a)$$

$$\frac{\partial}{\partial \xi_i} (\rho v_i v_j) + \frac{\partial p}{\partial \xi_j} = 0, \quad (2.51b)$$

$$\frac{\partial}{\partial \xi_i} (\rho v_i E + v_i p) = 0, \quad (2.51c)$$

with the vector of conservative variables $\mathbf{y} := (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)^\top$.

When considering, e.g., a low subsonic flow with a Mach number below 0.3, the flow is incompressible and in the majority of cases turbulent. Because of the different length scales in turbulent flows and the resulting computational costs, usually, the Navier-Stokes equations are simplified to the Reynolds-averaged Navier Stokes equations (RANS) with an additional turbulence model.

When applied in two-dimensional aerodynamic shape optimization problems, boundary conditions are prescribed for the surface A of the airfoil, and a free-stream boundary is prescribed in the far-field. The no-slip condition is prescribed on the airfoil, i.e.,

$$\mathbf{v} \cdot \mathbf{n} = 0 \text{ on } A, \quad (2.52)$$

where \mathbf{n} is the outward-facing unit normal on the surface A .

The far-field boundary describes the free-stream flow far away from the airfoil through a free-stream Mach number Ma_∞ and an angle of attack α defining the direction of the flow.

2. Multi-Objective Optimization

2.4.1.2. Objective Functions and Constraints

The main objective of the multi-objective shape optimization problem is to minimize the drag coefficient c_d and maximize the lift coefficient c_l .

In fluid flows, the forces acting on a body can be split into two components. The component into the direction of the flow field is identified with the drag force with magnitude F_d . The orthogonal component is referred to as the lift force with a magnitude of F_l . In general, the lift force F_l and the drag force F_d can be expressed as the surface integral over the surface A , given by

$$(F_d, F_l)^\top = \oint \boldsymbol{\sigma} \mathbf{n} dA, \quad (2.53)$$

with the stress tensor $\boldsymbol{\sigma}$ with $\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$. Thus, the forces result mainly from the pressure acting on the object and, if present, from the viscous stresses.

The drag coefficient and the lift coefficient are dimensionless numbers representing these forces. The drag coefficient is defined as

$$c_d = \frac{2F_d}{\rho_\infty v_\infty^2 A_{ref}} \quad (2.54)$$

with the the free-stream density ρ_∞ , the velocity magnitude v_∞ of the free-stream and the reference area A_{ref} , which can be described by the chord length for two-dimensional airfoils.

Since one does not assume any friction forces in the present test case, the drag coefficient is described by the pressure drag. Thus, when inserting the expression (2.53) the drag coefficient is given as

$$c_d = c_{d,p} = \frac{2}{\rho_\infty v_\infty^2 A_{ref}} \oint (p - p_\infty) \mathbf{n} dA. \quad (2.55)$$

The pressure p_∞ is the pressure of the far field. The pressure drag can also be expressed with the pressure coefficient

$$c_p = \frac{2(p - p_\infty)}{\gamma * M_\infty^2 * p_\infty} = \frac{2(p - p_\infty)}{\rho v_\infty^2} \quad (2.56)$$

on a two-dimensional shape as

$$c_{d,p} = \frac{1}{A_{ref}} \oint c_p (n_1 \cos(\alpha) + n_2 \sin(\alpha)) dA. \quad (2.57)$$

The lift coefficient is defined as the drag coefficient in (2.54) with the lift force F_l instead of the drag force. The pressure lift in two dimensions can then be expressed as

$$c_l = c_{l,p} = \frac{1}{A_{ref}} \oint c_p (n_2 \cos(\alpha) - n_1 \sin(\alpha)) dA. \quad (2.58)$$

The moment coefficient c_m describes the moment produced by the forces. The moment is referred to as *pitching moment* in the context of airfoil design. In the following, the

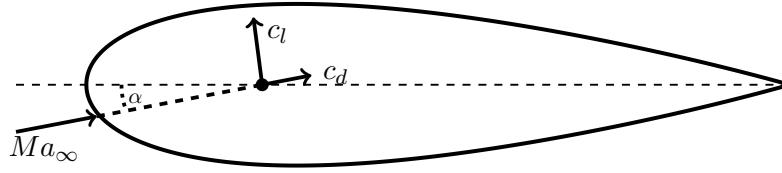


Figure 2.7.: Aerodynamic forces c_d and c_l for a two-dimensional airfoil (NACA0012).

spatial coordinates in two dimensions are described with ξ and η . While the aerodynamic forces act in the center of pressure since they are given by the above integrals, the pitching moment acts in the so-called *aerodynamic center* (ξ^m, η^m) . In the two-dimensional setting, the moment coefficient is then given as

$$c_m = \frac{1}{A_{ref}^2} \oint c_p (n_2(\xi - \xi^m) - n_1(\eta - \eta^m)) dA. \quad (2.59)$$

A negative pitching moment results in a nose-down pitching of the airfoil.

The direction of the aerodynamic forces is shown in Figure 2.7 along with the magnitude and direction of the free-stream flow indicated by the angle of attack α and the free-stream Mach number Ma_∞ .

2.4.1.3. Design Variables

The parameterization for the definition of the design variables is an important issue in the shape optimization process. A suitable parameterization should include complex geometries while at the same time controlling the smoothness of the shape. Therefore, popular parameterization approaches for the shape deformation are the application of smooth functions that perturb the geometry. In this context, *Hicks-Henne* [118] bump functions are used explicitly for two-dimensional airfoil parameterization. For other two-dimensional or three-dimensional problems, a *free form deformation* (FFD, [247]) strategy can be used. The idea of free-form deformation is to embed the geometry of interest in a simple lattice. The control points of the lattice define the design variables that serve to deform the original geometry with the help of splines. The disadvantage of free-form deformation is that the definition of constraints for deformations of the original geometry is not intuitive.

Opposed to that, in *free-node parameterization*, also referred to as CAD-free shape parameterization [196], the design variables are defined as the position of the grid points that are located on the shape of the airfoil. This type of parameterization gives the highest possible degree of freedom. However, it does not necessarily preserve any shape regularity. There is a high risk for producing uneven and wavy shapes during optimization. It can be prevented by appropriately preconditioning the design updates, e.g., with

2. Multi-Objective Optimization

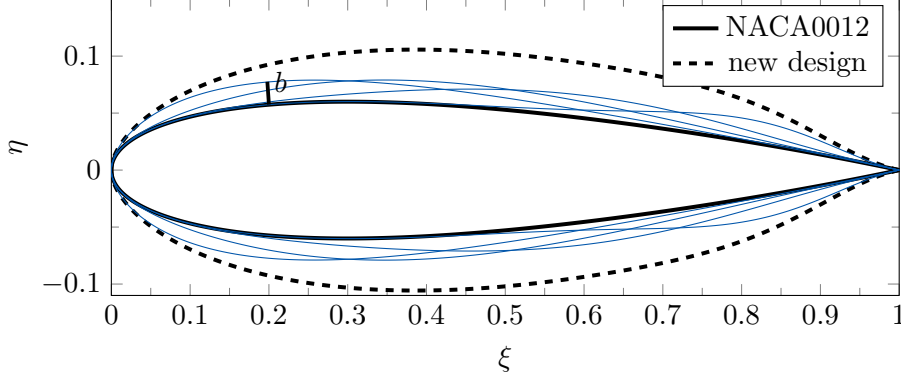


Figure 2.8.: Hicks-Henne bump functions described by the parameter b and resulting design.

the help of Sobolev smoothing originally proposed in [135]. Gradient smoothing strategies only work efficiently as a design space preconditioner if the associated parameters are correctly chosen globally or even locally. In an optimal setting, the preconditioner should mimic the reduced Hessian operator [238].

Since parameter studies shall be omitted in the present work, a parameterization based on Hicks-Henne bump functions is chosen in the following. Suppose that the shape of the airfoil is described in the ξ - η -space. The Hicks-Henne bump functions describe the displacement of the shape in the direction of η at the ξ -coordinate. The parameterized shape $\tilde{\eta}^u$ of the upper surface for an initial design with coordinates η^u is described as

$$\tilde{\eta}^u(\xi) = \eta^u(\xi) + \sum_{i=1}^{n/2} b_i^u \left(\sin(\pi \xi^{\log(0.5)/\log(\xi_{(i)})}) \right)^2 \quad (2.60)$$

with the Hicks-Henne parameters b_i^u of the upper surface and the bump locations $\xi_{(i)}$. The lower surface is parameterized analogously with parameters b_i^l for $i = 1, \dots, n/2$ in the negative η -direction. Figure 2.8 shows the initial design and the design resulting from an application of four Hicks-Henne bump functions distributed along the airfoil. They are indicated in blue and are defined by the parameter b .

The Hicks-Henne parameters are used as design variables of the optimization leading to a total number of n design variables, which will be described in the following with \mathbf{u} , such that $\mathbf{u} := (b_1^u, \dots, b_{n/2}^u, b_1^l, \dots, b_{n/2}^l)^\top$.

2.4.1.4. Shape optimization problem

The aim of airfoil design is the reduction of drag and the increase of lift under given flight conditions. This type of design is usually done after a conceptual design phase of the aircraft, which is used to determine the main configuration under given constraints. Environmental, financial, and safety aspects guide the whole design phase. The reduction of the drag especially leads to a decrease in fuel consumption to reach a desired

performance. This is advantageous for the environment and the costs. However, a reduction of drag naturally also leads to a reduction of the lift, as can be seen in the formulas. It may then decrease the overall performance. This effect is not intended. It is common in airfoil design to perform a lift-constrained drag minimization that aims at maintaining a desired lift. But it can also be of interest to find a single compromise solution between lift and drag. This is usually based on a maximization of the lift-to-drag ratio (see e.g. [118]). It is a feasible option and is the most representative value for the aerodynamic performance. It can also be interesting to consider a generating method for multi-objective optimization, which will be pursued in this work. However, the connection between lift and drag can be seen in the formulas, and we will later observe that the Pareto optimal front is convex. We will also observe that the lift-to-drag value is a representative trade-off solution, which is why a multi-objective optimization in this context is only seldomly used. It is important to note that the application was chosen without any intent to show new results for airfoil design, but as a test case to analyze the proposed strategies and show results for a realistic shape optimization problem of low complexity.

The design depends on the considered flow regime that is given by specific flight conditions, i.e., subsonic, transonic, or supersonic flow.¹² In transonic flow, e.g., depending on the shape, a strong shock wave may appear on the upper surface of the airfoil. As a result, the drag coefficient may increase significantly. Since one considers the compressible Euler flow, the drag is mainly shock-induced (*wave drag*) due to the massive, nearly discontinuous change in the pressure and partly lift-induced. As a result, aerodynamic shape optimization aims at avoiding such a strong shock by altering the shape. Note that a lift constraint is sometimes adapted by changing the angle of attack during optimization, which is not pursued in this work.

Next to optimizing the drag and lift, the design strategy is usually to keep the pitching moment of the reference design that is to be optimized to maintain the characteristics of the airfoil. The same applies to the maximum thickness of the airfoil, which is given by

$$d(\mathbf{u}) = \max_{\xi} \left(2\eta^u(\xi) + \sum_{i=1}^n u_i z_i(\xi) \right) \quad (2.61)$$

with

$$z_i(\xi) := \left(\sin(\pi \xi^{\log(0.5)/\log(\xi^{(i)})}) \right)^2 \text{ for } i = 1, \dots, n/2.$$

An improvement of the drag coefficient can naturally be obtained by decreasing the maximum thickness of the airfoil. But this is not necessarily intended since the overall stability is worsened. Therefore, it is common to prescribe a lower bound for the

¹²Note that it is also interesting to design an optimal airfoil under different flow conditions, which leads to a multi-point design problem. However, this work is restricted to specific flow conditions and certain variations are only allowed in the robust design problem in Chapter 4.

2. Multi-Objective Optimization

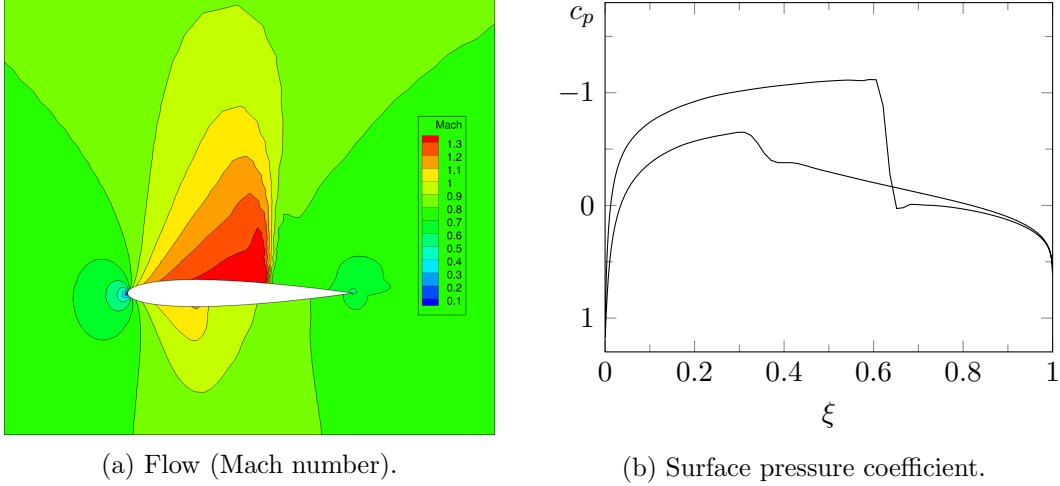


Figure 2.9.: Flow around NACA0012 airfoil depicted by the contour lines of the Mach number (a). A strong shock can be observed on the upper surface. The shock can also be observed when plotting the surface pressure coefficient (b). The upper line corresponds to the upper surface of the airfoil.

maximum thickness.¹³

The resulting multi-objective optimization problem that aims at minimizing the drag coefficient and maximizing the lift coefficient is given by

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \{c_d(\mathbf{y}, \mathbf{u}), -c_l(\mathbf{y}, \mathbf{u})\} & (2.62) \\ \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}) = \mathbf{0}, \\ & c_m(\mathbf{y}, \mathbf{u}) \geq c_m^t, \\ & d(\mathbf{u}) \geq d^t. \end{aligned}$$

Note that the underlying flow equations are introduced as a PDE constraint $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$ with the state variable \mathbf{y} . This notation will also be used in Chapter 3 and the PDE-constrained optimization problem will be analyzed. For now, it can be stated that one has to solve a partial differential equation for each design evaluation and one can obtain derivatives with a comparable cost.

The initial design for the shape optimization used in this thesis is a two-dimensional NACA0012 airfoil, which was already plotted in Figure 2.7. The NACA airfoil series was developed by the National Advisory Committee for Aeronautics. Series with different numbers of digits exist for describing the geometry of an airfoil. The NACA four-digit

¹³The thickness of the airfoil is a differentiable function in \mathbf{u} . Since the maximum is found for the ξ -coordinate, which does not depend on the design variable, the maximum thickness varies continuously with the design variable \mathbf{u} . It is important to remark that the continuous differentiability is, in general, not given. However, one may observe that the gradient-based optimization strategy works with comparable results as when using a continuously differentiable volume constraint. As a result, it is refrained from using a smooth approximation.

series was developed first. The first digit gives the maximum camber in percentage of the chord, and the next digit describes the position of the maximum camber in tenths of percentage of the chord. The last two digits give the maximum thickness of the airfoil in percentage of the chord. This means that the NACA0012 airfoil is symmetric, with a maximum thickness of 12 percent of the chord. The test case has a chord length of one meter, resulting in a maximum thickness of $d^t = 0.12$ and a moment coefficient of $c_m^t = 0.0$. In the following, the flow is transonic with an angle of attack of 1.25 degrees and a free-stream Mach number of 0.8. The shape is parameterized using 38 Hicks-Henne parameters.

Figure 2.9a shows the resulting flow for the NACA0012 airfoil by plotting the Mach number in a coarse representation using contour lines. The strong shock can be observed on the upper surface for the large changes of the Mach number. The flow over the surface can also be depicted with the help of the surface pressure coefficient. This is shown in Figure 2.9b, where the upper line corresponds to the upper surface of the airfoil. The shock can be observed for the high slope of the pressure coefficient.

2.4.2. Implementation Details

This section presents some details of the implementation of the strategy for multi-objective optimization. To summarize the strategy proposed in the present chapter:

- Constraint methods as presented in Section 2.2.3, 2.2.4 and 2.2.5 are used to solve the multi-objective optimization problem.
- The hybrid strategy presented in Section 2.3.3 is applied to the constrained single-objective optimization problem.

In the context of this work, the strategy was implemented in an existing framework for aerodynamic shape optimization, namely the multi-physics package SU2.

2.4.2.1. Simulation and Optimization with SU2

The numerical modeling of the flow is an essential factor for the success of the optimization algorithm. The success of the used optimization method highly depends on the accuracy of the discretization method to solve the flow equations and the strategy for obtaining derivatives. The presented strategy is implemented in SU2, which already provides a solver for the compressible Euler equations.

The multi-physics package SU2 [216, 54] comprises several solvers for various physical problems. It was initially developed for solving aerodynamic shape optimization problems for compressible flows. The code is open-source and is continuously improved and extended by the community. The main aim of code development is to maintain a modular structure with clearly defined interfaces. The solvers can be run in parallel based on MPI communication.

The implementation of the compressible Euler solver is based on a finite-volume method for solving Equation (2.51).¹⁴ In the finite volume method, the domain of the

¹⁴The incompressible flow solver in SU2 is based on an artificial compressibility formulation.

2. Multi-Objective Optimization

calculation is divided into non-overlapping cells of a grid. In SU2, this is a vertex-based median-dual grid, such that the cells of the dual grid are constructed with the vertices of the primal grid as centers. The variables for the state equations are stored in the centers of the cells. A system of balance equations that hold for each cell, balancing fluxes into and out of the cell, can be derived from the system of partial differential equations (2.51) in integral form by employing the Gauss divergence theorem.

Different numerical fluxes and slope limiters can be used, enabling first- and second-order spatial integration. For the specific application, a Jameson-Schmidt-Turkel scheme [137] is used, which introduces second-order and fourth-order artificial dissipation, which is controlled with a limiter. An implicit Euler strategy based on a chosen Runge-Kutta method is used to iterate the resulting residual equation in pseudo-time (see also Section 3.3.1.1). Convergence can be accelerated with the help of a multi-grid method [178].

SU2 offers the capability for a full design chain. An optimization framework is implemented in Python. The Python framework in SU2 serves to generate configuration files, run calculations with SU2, and to process results. Different types of design variables and functions of interest can be defined. The mesh deformation tool implemented in the main code can update the surface mesh and deforms the volume mesh accordingly by making use of the linear elasticity method. The derivatives of the function of interest with respect to the mesh nodes can be obtained with the help of an AD-based discrete adjoint method (see Section 3.3.1.1) and are projected back into the design space to provide gradients for optimization.

For the application in SU2, the algorithm for creating and managing the single-objective optimization problems resulting from the multi-objective method was implemented in the SU2 optimization framework. SU2 uses the SciPy optimization algorithms for shape optimization. The optimization algorithm available for constrained optimization problems of SciPy is SLSQP (Sequential Least Squares Programming), which is an SQP strategy originally proposed in [149]. For the present work, the interior-point optimizer IPOPT was additionally integrated in SU2 to provide an alternative handling of constrained optimization problems.

2.4.2.2. Hybrid Optimization Strategy

The hybrid method used in this context is based on the RoDeO (Robust Design Optimization) package, which was developed for design under uncertainties. RoDeO provides capabilities for Kriging and Bayesian optimization. The parameters of the Kriging model are found with the help of a genetic algorithm, and the maximization of expected improvement is done with the help of an explorative search using random sampling. RoDeO provides additional features that were not described in Section 2.3.3.3. Two features are presented in the following.

Optionally a linear regression model $f_{reg}(\mathbf{x})$ can be used as an additional term for predicting the response for $f(\mathbf{x})$. Opposite to universal Kriging the linear regression model is not included in the Kriging prediction. Instead, the Kriging model is built for sampled data of the output difference $f(\mathbf{x}^{(i)}) - f_{reg}(\mathbf{x}^{(i)})$ for $i = 1, \dots, N$. The regression

model is given by the formula

$$f_{reg}(\mathbf{x}) = w_0 + \sum_{i=1}^n w_i x_i \quad (2.63)$$

with the weight vector $\mathbf{w} = (w_0, \dots, w_n)^\top$, which can be determined by using the pseudoinverse, such that $\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ with the augmented data matrix

$$\mathbf{X} = \begin{pmatrix} 1 & \mathbf{x}^{(1)} \\ \vdots & \vdots \\ 1 & \mathbf{x}^{(n)} \end{pmatrix}.$$

Gradient-assisted sampling [213] can be used additionally to the samples generated from the expected improvement method. In this strategy, the gradient is evaluated at the best sample point so far. Then, the full model is evaluated in the direction of steepest descent. If no improvement is made, the step length is halved. This process is repeated up to a predefined number. Note that the gradient-assisted sampling strategy has been recently replaced by a primal-dual aggregation method [212]. The present work does not consider the use of gradients in the global search method since a hybridization is already applied on a higher level. Also, the extension to additional constraints is not straightforward and needs, for example, a suitable projection strategy.

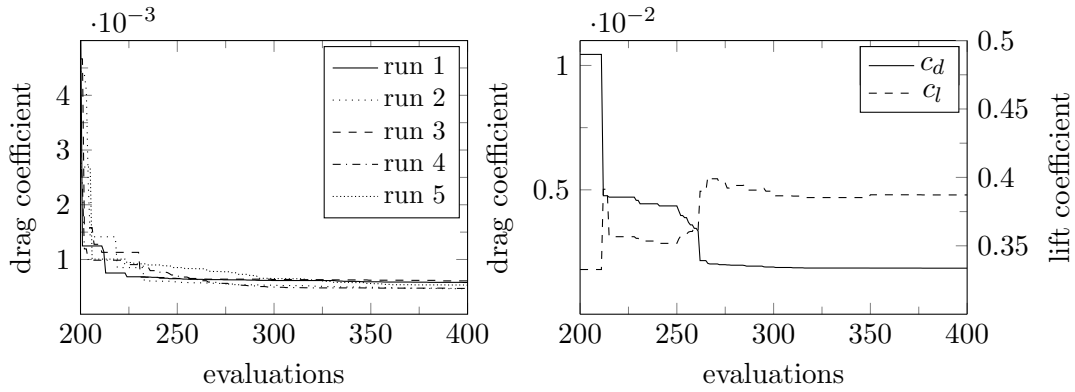
For the hybridization strategy in the present work, the package was adjusted to constrained optimization. Several features that are needed for enabling multi-objective optimization were introduced:

- **Modularity:** As the objective function to be optimized, as well as the target values for the constraints, are altered during the multi-objective optimization, the modularity of the code was increased. As a result, the objective functions and an arbitrary number of constraint functions relevant for optimization can be set in an external configuration file. Furthermore, interfaces to various functions in SU2 were provided.
- **Constraint handling:** The naive approach for constraint handling was extended, and the CWEI method was implemented.

For the presented application, the initial sampling set comprises 200 evaluations obtained from a design of experiments.

One may first observe the convergence of the expected improvement method for the NACA test case, where the drag coefficient is minimized without any additional constraints. The expected improvement (EI) method was run five times. A new initial design of experiments with 200 samples was performed for each run. The convergence history in 2.10a shows the reduction of the objective function values for each full model evaluation. It can be seen that the initial design of experiments can influence the convergence speed of the method. However, the convergence behavior does not differ significantly. After about 60 additional function evaluations, a slower convergence is observed for refining

2. Multi-Objective Optimization



(a) EI method: unconstrained problem.

(b) CWEI method: constrained problem.

Figure 2.10.: Improvement of the drag coefficient for five different runs of the expected improvement method with different initial sample sets of size 200 without additional constraints (a). Improvement of the drag coefficient for the constrained-based optimization problem (b).

the solution near the optimum. This is a typical behavior for heuristic approaches and defines a promising starting point for the gradient-based optimization.

Figure 2.10b shows a single run of the constraint-weighted expected improvement method for the constrained optimization problem with the given bounds on the maximum thickness and the moment coefficient and the constraint equation $c_l \geq 0.327$ for the lift. This corresponds to maintaining the lift coefficient of the initial NACA0012 design. Only the values for feasible evaluations with a reduction in the objective function are plotted. One can observe a similar convergence behavior as for the unconstrained optimization problem. Due to the adjustment of the constraints, the improvement in the objective function is slower. However, the number of evaluations that are needed until the slow refinement phase starts is with a value of 75 not significantly higher. Further tests show that the CWEI method, in average, gives the most promising results for constraint handling, although the general performance of the three methods is comparable. The CWEI method is the method of choice for the following results.

2.4.2.3. Multi-Objective Optimization Strategy

The framework for multi-objective optimization was implemented for the equality constraint method and the epsilon-constraint method in the Python framework of SU2. It is responsible for setting up the constrained optimization problems based on Algorithm 3. A configuration file is used to prescribe the number of objective functions k , the user-defined scanning resolution n_u , as well as the type of objective functions. Another task of the multi-objective optimization framework is to prepare and store the data for the hybrid optimization strategy. The expected improvement method and the gradient-based method are called sequentially for each single-objective optimization problem.

Before being applied to the PDE-constrained optimization problem, the strategy was

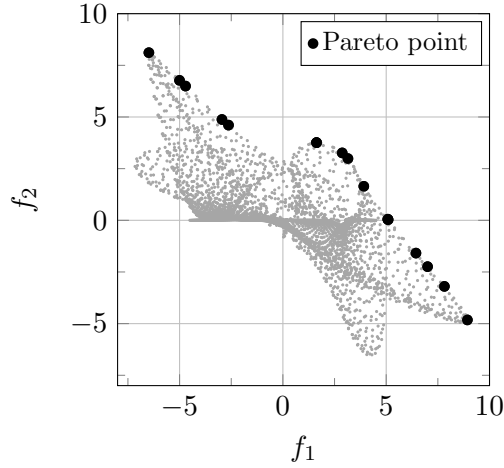


Figure 2.11.: Pareto optimal solutions for the problem (2.64) obtained with the epsilon-constraint method.

first tested for a problem presented in [144], which implies the risk of finding locally optimal solutions to the constrained single-objective optimization problems. The multi-objective optimization problem can be directly expressed as

$$\max_{\mathcal{D}} \{f_1(\mathbf{x}), f_2(\mathbf{x})\} \quad (2.64)$$

with

$$\begin{aligned} f_1(\mathbf{x}) &= 3(1 - x_1)^2 e^{-x_1^2 - (x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2 - x_2^2} \\ &\quad - 3e^{-(x_1+2)^2 - x_2^2} + 0.5(2x_1 + x_2), \\ f_2(\mathbf{x}) &= 3(1 + x_2)^2 e^{-(1-x_1)^2 - x_2^2} - 10\left(-\frac{x_2}{5} + x_2^3 + x_1^5\right) e^{-x_1^2 - x_2^2} \\ &\quad - 3e^{-x_1^2 - (2-x_2)^2}, \end{aligned}$$

and $\mathcal{D} = \{\mathbf{x} \in \mathbb{R}^2 \mid -3 \leq x_1 \leq 3, -3 \leq x_2 \leq 3\}$.

Next to the risk of finding local optima due to the highly nonlinear objective functions f_1 and f_2 , the optimization problem has got other interesting properties. The Pareto optimal front is non-convex and disconnected.

The value at the global maximum of f_1 is $\max_{\mathbf{x} \in \mathcal{D}} f_1 \approx 8.928$. When solving for the maximum with IPOPT with a tolerance of 10^{-4} , it will only find a local maximum of $f_1 \approx 3.045$. This shows the necessity to use the hybrid strategy.

When scanning the objective space with a resolution of $n_u = 7$, corresponding to a total number of 16 optimization problems, the algorithm finds the 14 points shown in Figure 2.11. Since the functions are cheap to evaluate in this specific situation, a Kriging model is directly based on a full factorial design with 360 points. The found optima are used as starting values for IPOPT. When comparing the solutions in the objective space to the samples in objective space (blue dots) that depict the feasible region, it can be observed that they are globally Pareto optimal.

2. Multi-Objective Optimization

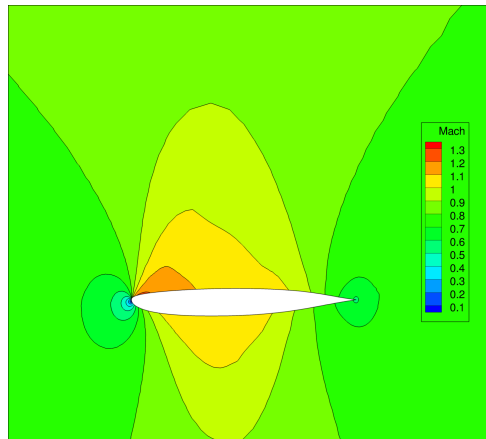


Figure 2.12.: Flow around airfoil for minimum drag coefficient depicted by the contour lines of the Mach number.

2.4.3. Results of Multi-Objective Aerodynamic Shape Optimization

The present section shows the results of applying constraint methods for multi-objective optimization based on the hybrid optimization strategy for aerodynamic shape optimization. The results of the epsilon-constraint method are presented in Section 2.4.3.1. A more detailed look at a specific visualization strategy is given in Section 2.4.3.2. For analysis purposes, Section 2.4.3.3 deals with the comparison of the epsilon-constraint method and the normal constraint method.

2.4.3.1. Results of Epsilon-Constraint Method

For the following tests, the constrained single-objective optimization problems are solved using the hybrid optimization strategy.¹⁵ The quasi-Newton strategy of IPOPT is employed for gradient-based optimization. The two-dimensional multi-objective optimization problem (2.62) is solved with the help of the epsilon-constraint method.

The anchor points are found without constraining the other objective function. The minimum drag coefficient has a value of about $7.6657 \cdot 10^{-4}$ corresponding to a lift coefficient of about 0.15920. Figure 2.12 of the flow around the optimized shape shows that the shock on the upper surface has disappeared. Since no constraint was set for the lift coefficient, the lift-induced drag is very small, and the optimal drag is only of the size of a few drag counts. The lift maximization without constraining drag gives a value of 0.80048 for the lift coefficient and $1.7406 \cdot 10^{-2}$ for the drag coefficient.

Figure 2.13 shows the results of an intensive search for Pareto optimal points. A total number of 32 Pareto optimal designs are given. The corresponding shapes are plotted for some specific designs (large dots). From bottom to top, they correspond to shapes with an increased lift coefficient.

¹⁵Note, however, that in this specific application case, the same optimal solutions will also be found when only using the gradient-based optimization strategy.

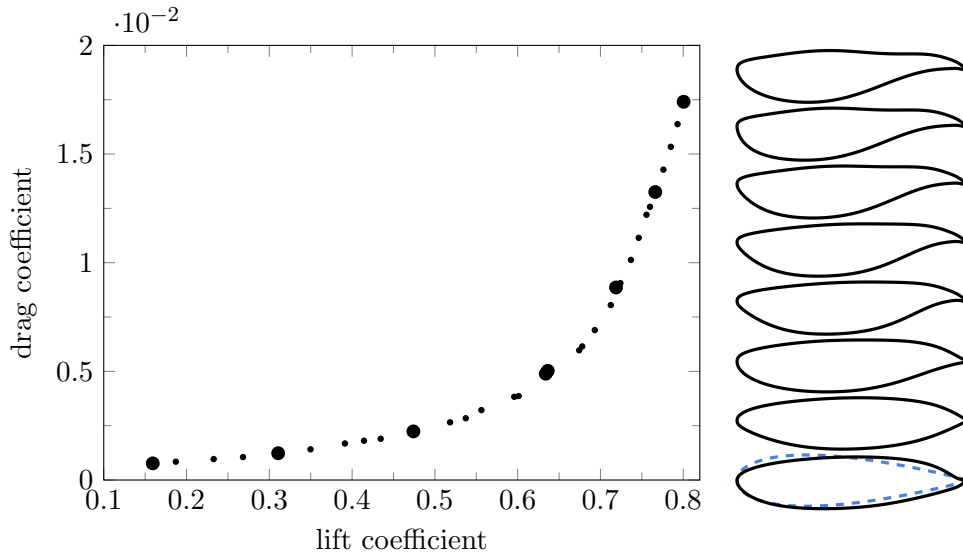


Figure 2.13.: Pareto optimal points found with the inequality constraint method. The points are shown in objective space on the left side and some designs on the left side corresponding to the large dots.

The Pareto optimal front has a convex shape and is easy to analyze concerning trade-off. The Pareto optimal point with a drag coefficient of $1.0554 \cdot 10^{-3}$ and a lift coefficient of 0.26822 has the best lift-to-drag ratio c_l/c_d . This Pareto optimal point also denotes the point that is the best compromise when looking at the slope of the approximated Pareto optimal front, i.e., it gives the best value for $\Delta c_l/\Delta c_d$ when comparing two neighboring points. This means that the lift can be increased a lot, leading only to a slight increase in the drag.

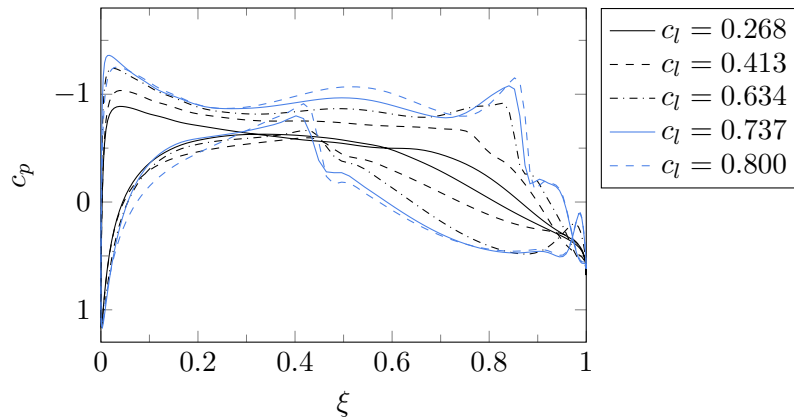


Figure 2.14.: Surface pressure coefficient plotted for five different Pareto optimal points. The upper line corresponds to the upper surface of the airfoil.

2. Multi-Objective Optimization

Furthermore, if one analyzes the flow solution of the different airfoils, one can observe that there is no shock present for low lift constraint values. For $c_l \geq 0.6$, one obtains a small shock on the upper surface near the trailing edge and on the lower surface of the airfoil. This means that for high lift values, given the flow conditions and the constraints, a shock cannot be prevented. This observation can be of interest for a decision-maker. The flow behavior is reflected in Figure 2.14, where the surface pressure coefficient is plotted for different lift constraints. The upper line represents the upper surface of the airfoil. A shock is observed for a jump in the pressure coefficient.

2.4.3.2. Visualization

The result of the MOO problem can be further visualized by using a patch plot with multi-dimensional scaling (MDS) as a projection method. The procedure was applied to the multi-objective aerodynamic shape optimization in the context of the continuation of work based on [256] for testing the strategy and comparing different projection methods. The idea is to represent the data set for each Pareto optimal point by a feature vector that comprises objective functions, design variables, and other functions of interest. The feature vector is projected into a two-dimensional space where locations are assigned for each Pareto optimal solution. A color-coded patch structure is created based on the location with the help of a Voronoi diagram. This results in a patch plot for each element of the feature vector.

The visualization using this strategy can be seen in Figure 2.15. It presents 42 diagrams, two for the objective functions representing the drag and the lift, two for the moment coefficient and the area as additional observation functions, and the rest for the 38 Hicks-Henne parameters h_i for $i = 1, \dots, 38$. The colormaps of the Hicks-Henne coefficients are set to display the same range. The patches in each diagram stand for the 32 Pareto optimal solutions. The plot directly shows the conflict between minimizing the lift and maximizing the drag. The visualization with the help of the patch plot makes it easier to compare and investigate design variable values when the number of design variables is high. It also helps to understand their direct connection to the objective space. The knowledge about similar behavior of different design variables can, for example, be used to reduce the design space. In this case, the observation functions are constraint functions of the optimization problem, but any functions of interest could be represented to observe correlations. Here, one sees the advantage of the proposed method when considering more than three functions of interest. Thus, the visualization via patch plots may be used for the analysis of higher-dimensional multi-objective optimization problems.

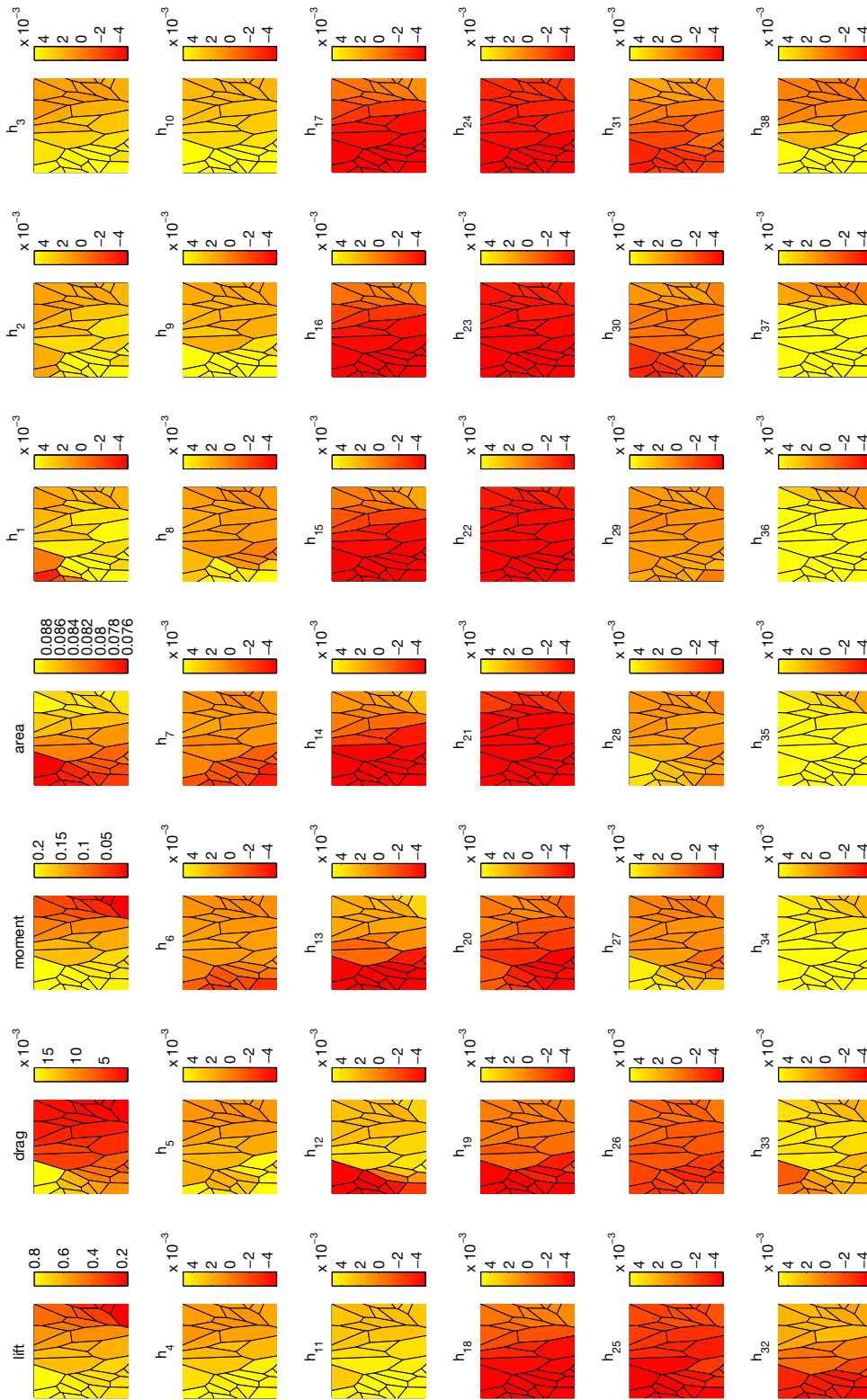


Figure 2.15.: MDS-based patch plot for the results of multi-objective airfoil design. The conflict between a maximal lift and a minimal drag is evident. With the patch plot a quick overview over the results in the 38 design variables h_1 to h_{38} is possible.

2. Multi-Objective Optimization

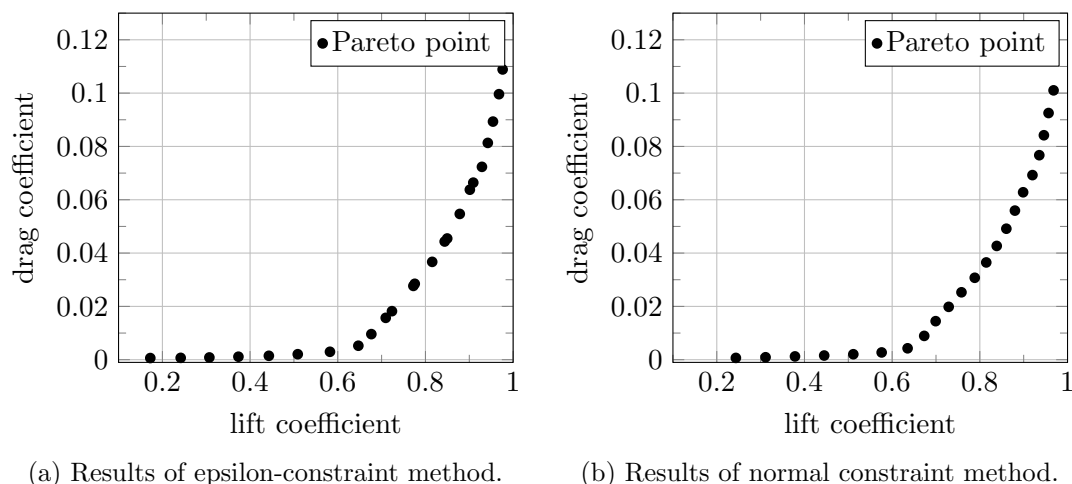


Figure 2.16.: Solutions found with the epsilon-constraint method (a) and the normal constraint method (b).

2.4.3.3. Results of the Normal Constraint Method

The multi-objective optimization is done using the epsilon-constraint method and the normal constraint method to compare both methods. The multi-objective aerodynamic shape optimization is slightly altered to consider a problem with a Pareto optimal front that does not have a similar curvature for all solutions in objective space. This is done by replacing the maximum thickness constraint by an upper bound on the airfoil area. The area of the NACA0012 initial configuration, which has a value of 0.0816925, is used as the target area to be fulfilled.

Figure 2.16a shows the solutions found with the help of the epsilon-constraint method, and the results for the normal constraint method are shown in Figure 2.16b. The epsilon-constraint method especially needs a scan in both directions of the objective space to find regions in the range of low drag coefficients. The results for the normal constraint method were obtained based on a normalization of the objective space to get coverage for low drag coefficients. We observe that the solutions obtained with the normal constraint method are more evenly distributed.

This can also be measured using a performance metric. The performance of MOO algorithms that generate a set of Pareto optimal solutions can be compared by analyzing the computational effort and the quality of the solution. For the quality, one may separate between the approximation quality, i.e., the distance of solutions to the Pareto optimal front and the distribution quality, which describes the diversity of solutions, i.e., how well the set of solutions represents the Pareto optimal front. Different performance metrics can be used to measure the respective features.

When comparing the epsilon-constraint method and the NC method, one may expect the same approximation quality and a comparable computational effort. The present work uses the spread metric [50] to measure the diversity for both algorithms. Given that the algorithm finds next to the anchor points n additional solutions, the spread

metric is defined as

$$\Delta_s := \frac{\sum_{i=1}^k d_{i,a} + \sum_{i=1}^n |d_i - \bar{d}|}{\sum_{i=1}^k d_{i,a} + n\bar{d}}. \quad (2.65)$$

The distance d_i denotes the minimum Euclidean distance of the solutions to any other solution found by the algorithm. The mean distance for all n solutions is given by \bar{d} . The distances for the anchor points, denoted as $d_{i,a}$, are treated separately. The reason for this is that the metric was introduced for evolutionary algorithms, which might not necessarily find the anchor points. This behavior leads to a higher value of the spread metric. Typically the spread metric for NSGA-II has values between 0.4 and 0.7 and for SPEA between 0.75 and 1.0 [49].

For the epsilon-constraint method one measures $\Delta_s = 0.3525$ and for the normal constraint method $\Delta_s = 0.1884$. Thus, both methods give better results than the multi-objective evolutionary approaches.

We may conclude that the normal constraint method gives promising results with a better spread metric. However, the epsilon-constraint method may be preferred due to some of the mentioned deficiencies. The normal constraint method cannot guarantee to find only (weakly) Pareto optimal solutions. Furthermore, the implementation of the construction of scanning points in the utopia plane is not very intuitive. Another reason for choosing the epsilon-constraint method is that most solvers provide the values and the gradients for the objective functions separately. The epsilon-constraint method can directly use this information, while the constraints in the normal constraint method are constructed based on all objective functions that are not minimized. Nevertheless, when a good diversity of the Pareto optimal set is of interest, it can be beneficial to use the normal constraint method.

3. One-Shot Method for PDE-constrained Optimization

The systems that are considered in engineering design problems are commonly governed by partial differential equations. For fluid flows, these are typically the Navier-Stokes equations, which may be further simplified depending on the system under consideration (see also Section 2.4.1).¹ Depending on the underlying problem, the numerical solution methods for solving such PDE constraints may require a high computational effort. If, for example, specific effects in fluid flow have to be resolved, the numerical solution strategy may need a highly-resolved computational mesh. Usually, the steady-state solution is found with the help of pseudo-time stepping strategies with several hundreds of iterations.

As already stated in Section 2.3, the use of heuristic methods is in such situations often too expensive. This can especially be observed when additionally a large number of design variables is considered.

There exist gradient-based optimization strategies that are especially suitable for PDE-constrained optimization problems with a large number of design variables. They make use of so-called *adjoint methods* for finding the sensitivities needed by gradient-based optimization algorithms. The advantage of adjoint methods is that the computational effort for computing derivatives is independent of the number of design variables. An overview of adjoint methodologies will be given in Section 3.1 of this chapter.

Classical PDE-constrained optimization problems are nested approaches such that the design update of the optimization algorithm is based on a fully converged solution of the state and the adjoint equation. The main aim of one-shot approaches is to reach state feasibility, adjoint feasibility, and optimality simultaneously in a coupled iteration. This is especially advantageous for slowly converging solvers. The single-step one-shot approach can be used if the underlying PDE constraint is solved with a contractive fixed-point solver. It is presented in Section 3.2.

A contribution of this thesis is the presentation of new theoretical and numerical results for the integration of additional equality constraints in the single-step one-shot framework. For this, the reader may refer to Section 3.2.4 and 3.2.5. The numerical results are presented in Section 3.3. As theoretical results, conditions for convergence of the one-shot method with additional equality constraints are derived. Numerical results include the application of the one-shot method to multi-objective aerodynamic shape optimization. A well-defined framework for the integration of additional constraints allows the solution of a multi-objective optimization problem with the help of constraint methods based on hybrid methodologies, as presented in Chapter 2.

¹Note that in this thesis, only steady-state processes are considered.

3.1. Fundamentals of PDE-constrained optimization

The one-shot method can be applied to optimization problems that are constrained by a set of partial differential equations. In the following, one may describe such optimization problems as

$$\begin{aligned} \min_{\mathbf{u} \in U_{ad}, \mathbf{y} \in Y} \quad & f(\mathbf{y}, \mathbf{u}) \\ \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}) = \mathbf{0} \end{aligned} \tag{3.1}$$

with the vector of design variables $\mathbf{u} \in U_{ad} \subset U$ and the vector of state variables $\mathbf{y} \in Y$ and appropriate Hilbert spaces U and Y . Here, U_{ad} denotes the admissible design space, which is assumed to be closed and convex in the following. The function $f : Y \times U \rightarrow \mathbb{R}$ is the objective function to be minimized. The equality constraint $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$, defined by the function $c : Y \times U \rightarrow Y$, describes the underlying set of partial differential equations with boundary conditions. As methods based on discrete adjoint methodologies are considered in the present work, in the following, the spaces are finite-dimensional, i.e., $Y = \mathbb{R}^m$ and $U = \mathbb{R}^n$.² Thus, one assumes that the system of PDEs is already discretized, where $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$ represents the discretized PDE constraint.

Note that it can be advantageous to replace the discretized PDE constraint $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$ by a fixed-point equation $\mathbf{y} = G(\mathbf{y}, \mathbf{u})$ that represents the solution strategy for solving the PDE constraint. Nevertheless, the following results will be presented based on the discretized PDE constraint. The reason for this is that it is the most common approach for deriving optimality conditions and adjoint methods in a discrete setting. However, the reader may keep in mind that all results can be analogously obtained for the formulation with a fixed-point equation, which is presented later in Section 3.2.1 and will be the basis of the methodologies for PDE-constrained optimization presented in this chapter.

In the following, Section 3.1.1 introduces some notations as well as assumptions for the optimization problem. These are needed for the first and second-order optimality conditions presented in Section 3.1.2. Finally, different methods for sensitivity computation are presented in Section 3.1.3, focusing on adjoint methods, which are popular methods for optimization with PDE constraints.

3.1.1. Properties and Assumptions

In the present section, some natural assumptions are made. They are required for the derivation and the analysis of methods for PDE-constrained optimization. To ease notation, one may first introduce some abbreviated forms.

The collection of partial derivatives of the function $f(\mathbf{y}, \mathbf{u})$, as described above, with

²This is also done for reasons of notation. Due to the use of finite-dimensional spaces, inner products can be written as scalar products. Elements from the dual space are described by the transpose.

3.1. Fundamentals of PDE-constrained optimization

respect to u_i for $i = 1, \dots, n$ can be defined as the gradient

$$\nabla_{\mathbf{u}} f(\mathbf{y}, \mathbf{u}) := \begin{pmatrix} \frac{\partial f}{\partial u_1}(\mathbf{y}, \mathbf{u}) \\ \vdots \\ \frac{\partial f}{\partial u_n}(\mathbf{y}, \mathbf{u}) \end{pmatrix}.$$

Accordingly, one may define the Hessian for a mixed second- order derivative of, e.g., $\nabla_{\mathbf{u}} f$ with respect to \mathbf{y} as

$$\nabla_{\mathbf{y}\mathbf{u}} f(\mathbf{y}, \mathbf{u}) = \nabla_{\mathbf{y}} \otimes \nabla_{\mathbf{u}} f = \begin{pmatrix} \frac{\partial^2 f}{\partial y_1 \partial u_1} & \frac{\partial^2 f}{\partial y_1 \partial u_2} & \cdots & \frac{\partial^2 f}{\partial y_1 \partial u_n} \\ \frac{\partial^2 f}{\partial y_2 \partial u_1} & \frac{\partial^2 f}{\partial y_2 \partial u_2} & \cdots & \frac{\partial^2 f}{\partial y_2 \partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial y_m \partial u_1} & \frac{\partial^2 f}{\partial y_m \partial u_2} & \cdots & \frac{\partial^2 f}{\partial y_m \partial u_n} \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

The Jacobian or Jacobi matrix of the function $c(\mathbf{y}, \mathbf{u})$ with respect to \mathbf{y} is written as

$$c_{\mathbf{y}}(\mathbf{y}, \mathbf{u}) := \begin{pmatrix} (\nabla c_1(\mathbf{y}, \mathbf{u}))^\top \\ \vdots \\ (\nabla c_m(\mathbf{y}, \mathbf{u}))^\top \end{pmatrix}.$$

To avoid confusion, the notation $\partial \mathbf{y} / \partial \mathbf{u}$ is used for the special case of the Jacobian of the state variables with respect to the design variables. One assumes that the objective function as well as the constraint functions are twice continuously differentiable and that

$$\lim_{\|\mathbf{y}\| + \|\mathbf{u}\| \rightarrow \infty} f(\mathbf{y}, \mathbf{u}) = +\infty, \quad (3.2)$$

such that all level sets of f are bounded to guarantee the existence of an optimal solution to (3.1).

Furthermore, assuming that the Jacobian $c_{\mathbf{y}}$ is nonsingular, the state \mathbf{y} can be uniquely determined by the design variable \mathbf{u} . This result can be directly obtained from the implicit function theorem. It is given in the following for the function $c : D \subset (U \times Y) \rightarrow Z$ in a finite-dimensional setting.³

Theorem 9 (implicit function theorem). *Let U, Y, Z be finite-dimensional Hilbert spaces. Let $c : \mathcal{X} \rightarrow Z$ be continuously differentiable on the open set $\mathcal{X} \subset Y \times U$ with $c(\mathbf{y}_0, \mathbf{u}_0) = \mathbf{0}$ for $(\mathbf{y}_0, \mathbf{u}_0) \in \mathcal{X}$. Further, suppose that $c_{\mathbf{y}}(\mathbf{y}_0, \mathbf{u}_0)$ is nonsingular. Then there exists an open neighborhood $W(\mathbf{y}_0) \times V(\mathbf{u}_0) \in \mathcal{X}$ of $(\mathbf{y}_0, \mathbf{u}_0)$ and a function $\phi : V(\mathbf{u}_0) \rightarrow W(\mathbf{y}_0)$ that is continuously differentiable on $V(\mathbf{u}_0)$, such that $\phi(\mathbf{u}_0) = \mathbf{y}_0$ and for all $u \in V(\mathbf{u}_0)$*

³The implicit function theorem can be generalized to Banach spaces in an infinite-dimensional setting, such that it is applicable to the continuous formulation of the optimization problem.

3. One-Shot Method for PDE-constrained Optimization

there is a uniquely defined $\phi(\mathbf{u})$ that solves $c(\phi(\mathbf{u}), \mathbf{u}) = \mathbf{0}$. Furthermore, the Jacobian of ϕ can be expressed as

$$\phi_{\mathbf{u}}(\mathbf{u}) = -c_{\mathbf{y}}(\phi(\mathbf{u}), \mathbf{u})^{-1}c_{\mathbf{u}}(\phi(\mathbf{u}), \mathbf{u}). \quad (3.3)$$

Proof. The proof is given in [226, p.336-339]. \square

Because of this result one can reformulate the above problem in a reduced form as

$$\min_{\mathbf{u}} f(\mathbf{y}(\mathbf{u}), \mathbf{u}). \quad (3.4)$$

3.1.2. Necessary and Sufficient Conditions for Optimality

In the following, necessary and sufficient conditions for optimality will be derived.

The Lagrange function L corresponding to the optimization problem (3.1) is

$$L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) := f(\mathbf{y}, \mathbf{u}) + c(\mathbf{y}, \mathbf{u})^{\top} \bar{\mathbf{y}}. \quad (3.5)$$

The vector of Lagrange multipliers $\bar{\mathbf{y}} \in \mathbb{R}^m$ can also be identified as the vector of adjoint variables.

Theorem 10 (First-order necessary condition for optimality). *Let $(\mathbf{y}^*, \mathbf{u}^*)$ be a locally optimal solution to (3.1) and let f, c be continuously differentiable at $(\mathbf{y}^*, \mathbf{u}^*)$ with $c_{\mathbf{y}}$ nonsingular, then there exists a unique multiplier $\bar{\mathbf{y}}^*$ such that the Karush-Kuhn-Tucker (KKT) optimality conditions*

$$\nabla_{\bar{\mathbf{y}}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*) = c(\mathbf{y}^*, \mathbf{u}^*) = \mathbf{0}, \quad (3.6a)$$

$$\nabla_{\mathbf{y}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*) = \nabla_{\mathbf{y}} f(\mathbf{y}^*, \mathbf{u}^*) + c_{\mathbf{y}}(\mathbf{y}^*, \mathbf{u}^*)^{\top} \bar{\mathbf{y}}^* = \mathbf{0}, \quad (3.6b)$$

$$\nabla_{\mathbf{u}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*) = \nabla_{\mathbf{u}} f(\mathbf{y}^*, \mathbf{u}^*) + c_{\mathbf{u}}(\mathbf{y}^*, \mathbf{u}^*)^{\top} \bar{\mathbf{y}}^* = \mathbf{0} \quad (3.6c)$$

are fulfilled.

Proof. The regularity assumption for $c_{\mathbf{y}}$ guarantees that the LICQ is fulfilled. The KKT conditions then result from the application of Theorem 1. \square

Equation (3.6a) is referred to as the *state equation*, Equation (3.6b) as the *adjoint equation*, and Equation (3.6c) as the *design equation*.⁴ The derivation of the adjoint equation allows to obtain derivatives with the help of adjoint methods, which will be introduced in the next section. One may also refer to a point fulfilling the KKT conditions as a stationary point of the optimization problem.

Let

$$H(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) := \begin{pmatrix} \nabla_{\mathbf{y}\mathbf{y}} L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) & \nabla_{\mathbf{y}\mathbf{u}} L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \\ \nabla_{\mathbf{u}\mathbf{y}} L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) & \nabla_{\mathbf{u}\mathbf{u}} L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \end{pmatrix}$$

⁴The state equation is also often referred to as the *primal equation*, while the adjoint equation is sometimes referred to as the *dual equation*.

3.1. Fundamentals of PDE-constrained optimization

be the Hessian of the Lagrange function. In the following, the second-order necessary and sufficient conditions for optimality are stated. They require positive (semi-)definiteness of the so-called *reduced Hessian* $H_r := Z^\top H Z$ evaluated at the optimal solution $(\mathbf{y}^*, \mathbf{u}^*)$. The columns of $Z \in \mathbb{R}^{(m+n) \times n}$ span the tangential space of the PDE constraint, which is given as $\mathcal{Z} := \{(\mathbf{y}, \mathbf{u}) \in \mathbb{R}^m \times \mathbb{R}^n \mid c_{\mathbf{y}}\mathbf{y} + c_{\mathbf{u}}\mathbf{u} = \mathbf{0}\}$. Due to the regularity of $c_{\mathbf{y}}$ one obtains the direct expression

$$Z(\mathbf{y}, \mathbf{u}) := \begin{pmatrix} -c_{\mathbf{y}}^{-1}(\mathbf{y}, \mathbf{u})c_{\mathbf{u}}(\mathbf{y}, \mathbf{u}) \\ I_n \end{pmatrix}. \quad (3.7)$$

Theorem 11 (Second-order necessary condition for optimality). *Let $(\mathbf{y}^*, \mathbf{u}^*)$ be a locally optimal solution to (3.1) and let f, c be twice continuously differentiable at $(\mathbf{y}^*, \mathbf{u}^*)$ with $c_{\mathbf{y}}$ nonsingular. Then the multiplier $\bar{\mathbf{y}}^*$ of Theorem 10 fulfills*

$$\mathbf{u}^\top Z^\top(\mathbf{y}^*, \mathbf{u}^*)H(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)Z(\mathbf{y}^*, \mathbf{u}^*)\mathbf{u} \geq 0 \quad \forall \mathbf{u} \in U_{ad} \quad (3.8)$$

with Z defined as in Equation (3.7).

Proof. As the reduced Hessian can be directly introduced with the help of the tangential space of the PDE constraint, Theorem 2 can be applied. \square

Theorem 12 (Second-order sufficient condition for optimality). *Let f, c be twice continuously differentiable at $(\mathbf{y}^*, \mathbf{u}^*)$ with $c_{\mathbf{y}}$ nonsingular. If the multiplier $\bar{\mathbf{y}}^*$ of Theorem 10 fulfills*

$$\mathbf{u}^\top Z^\top(\mathbf{y}^*, \mathbf{u}^*)H(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)Z(\mathbf{y}^*, \mathbf{u}^*)\mathbf{u} > 0 \quad \forall \mathbf{u} \in U_{ad} \text{ with } \mathbf{u} \neq \mathbf{0} \quad (3.9)$$

with Z defined as in Equation (3.7), then $(\mathbf{y}^*, \mathbf{u}^*)$ is a strict locally optimal solution to (3.1).

Proof. With the above results Theorem 3 proofs the statement. \square

3.1.3. Sensitivity Computation

Different approaches can be used for finding the sensitivities needed for gradient-based optimization with PDE constraints.

The total derivative $D_{\mathbf{u}}f$ of the objective function $f(\mathbf{y}(\mathbf{u}), \mathbf{u})$, which is used in a gradient-based optimization algorithm, is given by

$$D_{\mathbf{u}}f := \nabla_{\mathbf{u}}f(\mathbf{y}(\mathbf{u}), \mathbf{u}) + \left(\frac{\partial \mathbf{y}}{\partial \mathbf{u}} \right)^\top \nabla_{\mathbf{y}}f(\mathbf{y}(\mathbf{u}), \mathbf{u}). \quad (3.10)$$

It contains the Jacobian $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}$, which cannot necessarily be computed directly. The total derivative could be calculated with the help of finite differences, the forward mode of algorithmic differentiation or direct sensitivities as described in the following section or with the help of adjoint methods given in Section 3.1.3.2.

3. One-Shot Method for PDE-constrained Optimization

3.1.3.1. Finite Differences and Direct Differentiation

When using *finite differences* the total derivative is approximated with the help of a finite difference formula, for example, by one-sided differences

$$(D_{\mathbf{u}}f)_i \approx \frac{1}{\delta}(f(\mathbf{y}(\mathbf{u} + \delta \mathbf{e}_i), \mathbf{u} + \delta \mathbf{e}_i) - f(\mathbf{y}(\mathbf{u}), \mathbf{u})) \quad \forall i = 1, \dots, n \quad (3.11)$$

with the i -th unit vector \mathbf{e}_i . Thus, $(n + 1)$ calculations of the objective function are needed to find the total derivative. The error of the forward-differences formula is of the order δ . Higher-order approximations can be derived from a Taylor series expansion. The choice of δ is crucial and demands a lot of manual tuning. If it is chosen too small, cancellation errors may occur. Furthermore, the optimal choice of δ is not necessarily the same for all elements of the derivative vector.

When using *direct differentiation*, the relation

$$c_{\mathbf{u}}(\mathbf{y}(\mathbf{u}), \mathbf{u}) + c_{\mathbf{y}}(\mathbf{y}(\mathbf{u}), \mathbf{u}) \frac{\partial \mathbf{y}}{\partial \mathbf{u}} = \mathbf{0} \quad (3.12)$$

is used for directly solving for $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}$. The relation can be obtained from the implicit function theorem or by linearization of $c(\mathbf{y}(\mathbf{u}), \mathbf{u}) = \mathbf{0}$. The equation system (3.12) has to be solved for each element of \mathbf{u} and the result can be inserted in (3.10) to obtain the total derivative. This strategy is, for example, applied for aerodynamic design in [268].

Finite difference and direct differentiation, as well as the forward mode of algorithmic differentiation, presented in Section 3.2.6.2, are computationally expensive for a large number of design variables as the evaluation of the derivative depends on the number of design variables. Therefore, it is convenient to turn to adjoint approaches.

3.1.3.2. Adjoint Methods

When using adjoint approaches, the costs for an evaluation of the sensitivities of the objective function with respect to the design variables is independent of the number of design variables. This is a definite advantage in comparison to finite differences or the direct sensitivity analysis.

In the following, the adjoint methodology is derived for the discretized optimization problem (3.1), but the same ideas can be applied to the continuous problem in an infinite-dimensional setting based on function spaces. The total derivative in equation (3.10) can be expressed by eliminating $\nabla_{\mathbf{y}}f(\mathbf{y}(\mathbf{u}), \mathbf{u})$ in equation (3.10) with the help of the adjoint equation

$$\nabla_{\mathbf{y}}f(\mathbf{y}(\mathbf{u}), \mathbf{u}) + c_{\mathbf{y}}(\mathbf{y}(\mathbf{u}), \mathbf{u})^{\top} \bar{\mathbf{y}} = \mathbf{0}, \quad (3.13)$$

such that one obtains

$$D_{\mathbf{u}}f = \nabla_{\mathbf{u}}f(\mathbf{y}(\mathbf{u}), \mathbf{u}) - \left(\frac{\partial \mathbf{y}}{\partial \mathbf{u}} \right)^{\top} c_{\mathbf{y}}(\mathbf{y}(\mathbf{u}), \mathbf{u})^{\top} \bar{\mathbf{y}}.$$

Then the use of equation (3.12) results in a new expression for the total derivative, given by

$$D_{\mathbf{u}}f = \nabla_{\mathbf{u}}f(\mathbf{y}(\mathbf{u}), \mathbf{u}) + c_{\mathbf{u}}(\mathbf{y}(\mathbf{u}), \mathbf{u})^{\top} \bar{\mathbf{y}}. \quad (3.14)$$

This expression is also referred to as the *reduced gradient* since the original gradient is projected into the tangent space of the PDE constraint with the help of the linearization (3.12). Note that the adjoint equation (3.13) is equivalent to the adjoint equation in the KKT system (3.6) and a vanishing total derivative is equivalent to the design equation in the KKT system. Thus, the adjoint strategy can also be derived with the help of the necessary conditions for optimality. The result can also be obtained from the duality viewpoint (see e.g. [81] or Section A.2.1 of the appendix).

Now, the task is to find the vector of adjoint variables $\bar{\mathbf{y}}$ using the adjoint equation (3.13). Lions [169] formulated the adjoint approach for PDE-constrained optimal control problems. Later, Pironneau [221] presented the strategy in problems governed by fluid flow. The approach was first introduced in the context of aerodynamic design by Jameson in [135]. There exist two different adjoint approaches that are based on the ideas described above but make use of different strategies for deriving and solving the adjoint equation. These approaches are referred to as *continuous adjoint* and *discrete adjoint* approaches.

Continuous Adjoint Method In the continuous adjoint approach, the aim is to derive the continuous adjoint equation for the continuous formulation of the optimization problem and discretize it afterwards. The adjoint equation corresponding to the state equation is derived analytically from the given model via the variational form of the PDE constraints. The conventional approach to obtain the adjoint equation is to take the variation of the objective function and to incorporate the variations that concern the state equations with the help of an adjoint variable. Then the adjoint variable is chosen in such a way that the terms with the variation of the state variables disappear. The choice of the adjoint variable thus leads to the adjoint equation as an additional PDE that has to be solved.⁵

The continuous adjoint equation was derived and applied to various aerodynamic shape optimization problems, e.g., for the two-dimensional Euler equations in [135], and the three-dimensional Euler equations in [136].

The continuous adjoint approach is computationally efficient as it allows for the formulation of problem-dependent solution schemes. However, the derivation of the adjoint equations and the corresponding schemes can be very complex. As a result, the implementation of the adjoint equation is error-prone. For each new objective function and each new constraint function, one has to derive a new adjoint equation. Additionally, in the case of turbulent flows, for example, the continuous adjoint approach is mainly limited to freezing parameters in the turbulence model, e.g., the use of the frozen eddy viscosity assumption, as many turbulence models cannot be differentiated analytically.

⁵For the duality viewpoint, the task is to find the dual operator D^* in (A.9). Additionally, one also has to find dual boundary operators for the boundary conditions.

3. One-Shot Method for PDE-constrained Optimization

One also has to think about an appropriate discretization of the adjoint equation. Depending on the grid resolution, there can occur inconsistencies between the derivatives and the calculated objective function.

In contrast to that, when solving the discrete adjoint equation in the discrete adjoint approach, one may use the same solver as for the state equations. Depending on the numerical solution schemes and the approach for formulating the adjoint problem, the adjoint solver in the discrete adjoint approach inherits the convergence properties of the state solver. Moreover, the derivatives obtained from a converged adjoint solution may potentially be consistent with the discretized objective function if the numerical solution scheme of the state equation is taken into account.

Discrete Adjoint Method The discrete adjoint is built by first discretizing the state equation and finding the discrete adjoint variables afterwards. The general idea is, for example, formulated in [268]. Various strategies can be applied to find the discrete adjoint solution. One strategy is to collect terms corresponding to variations of the discretized state variables similarly to the continuous adjoint approach. This technique is referred to as hand differentiation. The adjoint equation can then either be solved by directly solving the linear system of equations or by using an iterative method with an appropriate preconditioner (see e.g. [27, 147] for aerodynamic shape optimization). It is common to make use of the reverse mode of algorithmic differentiation to avoid the tedious work of collecting and hand-coding the appropriate terms for solving the discrete adjoint equation and assembling the total derivative.

With the help of algorithmic differentiation, the whole computational path, from the design variables as input to the objective function as an output, or the needed parts of this path can be differentiated. Details on algorithmic differentiation are given in Section 3.2.6. For now, it suffices to state that appropriate tools for algorithmic differentiation allow a robust and semi-automatic way to obtain derivatives, and the application of the reverse mode of algorithmic differentiation can be related to the adjoint approach. The use of algorithmic differentiation also eliminates the problem of treating non-differentiable turbulence models.

There exist different strategies for applying AD to solve the adjoint equation. In the following, it is distinguished between two main strategies, which differ in the representation of the discretization of the PDE constraint:

- **Residual representation:** One strategy is to differentiate the implementation of the discretized PDE constraint in its residual form $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$, i.e., the evaluation at the stationary point of the state equation [195]. The differentiation can be done in such a way that the Jacobians $c_{\mathbf{y}}$ and $c_{\mathbf{u}}$ in (3.13) and (3.14) are obtained from matrix-vector products (see e.g. [30]). The same can be done for the implementation of the objective function. This approach is analogous to the idea of hand differentiation, which is explained above. As a result, the adjoint equation can be solved by solving the linear system of equations or by constructing a suitable solution procedure. This is advantageous for efficiently solving the adjoint equation, but requires the identification of the required routines in the source code,

e.g., the flux differentiation routine. Also, since the differentiation is based on the assumption of a steady-state, it does not represent the full solution scheme of the state equation. This decoupling of the state solver and the adjoint solver may lead to a loss of consistency, which can have negative effects on the convergence of the adjoint equation. This is especially the case if the state solution is not sufficiently converged, which often happens in real-world engineering applications. In the implementation of [205, 30], based on the exact dual approach of Giles [80], the solution procedure of the state iteration is taken into account by constructing a solution scheme that preserves discrete duality. It also requires that the steady-state solution is found. The general idea is to use the same solution scheme for the adjoint equation as for the state equation. In the exact dual approach, the strategy is to apply the transposed preconditioner of the solution scheme to the state equation to the adjoint equation. The exact dual approach gives the transition to the other type of strategy for setting up an AD-based discrete adjoint solver.

- Fixed-point representation:** The solution procedure for finding the steady-state solution of the discretized PDE-constraint can be differentiated with the help of algorithmic differentiation. Such a solution procedure can be usually described by means of a fixed-point iteration $G(\mathbf{y}, \mathbf{u}) = \mathbf{y}$. Details on formulating the optimization problem with the help of a fixed-point iteration are given in Section 3.2.1. First ideas for differentiating the iterative scheme were formulated in [148, 204] in the context of hand differentiation. In general, when differentiating the fixed-point equation, one obtains a fixed-point equation for the adjoint, and the adjoint iteration inherits the contraction properties of the state iteration. Details on the application of reverse mode to the fixed-point iteration will be given in 3.2.6.4. The main aspects are summarized in the following. Applying the reverse mode of algorithmic differentiation to the full computational path of the solution procedure gives directly the total derivative that is required for gradient-based optimization (see e.g. [73]). The resulting adjoint solution is consistent with the solution of the state equation, such that the derivative is exact for the resolution of the objective function. The advantage of this approach is that the code can be treated as a black-box and can be differentiated in a semi-automatic fashion. Hence, no detailed knowledge of the algorithm for solving the state equation is needed. However, differentiating the full iterative solution procedure with all intermediate states requires a lot of memory. This problem can be circumvented by only differentiating the application of the fixed-point iterator to the stationary point and iterating the adjoint variable based on this information [38]. This is referred to as reverse accumulation. The fixed-point iteration for the adjoint inherits the contraction property of the fixed-point iteration for the state equation at the termination state. The success of applying reverse accumulation depends on the level of accuracy of the state solution. If the state solution is converged to a stationary point and the respective fixed-point iterator is contractive, the adjoint fixed-point iteration converges. Otherwise, convergence cannot be guaranteed. However, for a sufficiently converged solution, it is often observed. The reverse accumulation

3. One-Shot Method for PDE-constrained Optimization

strategy can be altered by using a simplified recurrence as proposed in [91]. For this, a special structure of the iterator is assumed and a specific term is neglected in the differentiation. This strategy can actually be related to the exact dual approach by Giles.

In general, the use of algorithmic differentiation tools instead of hand differentiation requires less coding effort but can be less memory and runtime efficient. The advantages and disadvantages for setting up an AD-based discrete adjoint method have to be weighed up against each other for the problem at hand. Both approaches can be manually tuned to reduce issues with memory and runtime, but this requires experience and a careful analysis of the implemented routines. For the AD-based discrete adjoint method based on the discretized PDE-constraint, this is, for example, analyzed in [107, 199]. Tuning of the AD-based discrete adjoint strategy based on the fixed-point formulation is, for example, presented in [3].

Comparison of Adjoint Methods As the operations of discretizing and adjoining do not commute in general, the discrete adjoint approach and the continuous adjoint approach do not necessarily lead to the same results. But, for sufficiently smooth solutions, both approaches may converge to the same results when letting the degrees of freedom go to infinity. There are several studies that compare the continuous and the discrete adjoint approach ([5], [81]). Nadarajah and Jameson ([200],[201]) compared both approaches for the Euler equations. They concluded that, for the test cases under consideration, the discrete adjoints agree better with finite differences, but the difference between results of the discrete and continuous adjoint method is rather small and decreases as the number of degrees of freedom increases.

3.2. The One-Shot Approach

The classical optimization approach for finding a KKT optimality point that fulfills (3.6) is given by Algorithm 6.

Algorithm 6: Classical optimization approach.

```
for  $j = 1$  to  $j_{max}$  do  
    solve state equation (3.6a)  
    solve adjoint equation (3.6b)  
    perform design update using the reduced gradient (3.14)  
end for
```

As can be seen from the algorithm, the strategy of the classical approach is to reach feasibility of the state equation and the adjoint equation, also referred to as *state feasibility* and *adjoint feasibility*, in each design update in the j -th step.⁶ This is a nested

⁶Sometimes also referred to as *primal* and *dual feasibility*.

procedure. The question arises if it is always necessary to recreate feasibility, even if the solution is not optimal.

Thus, the main aim of one-shot approaches is to reach state feasibility, adjoint feasibility, and optimality simultaneously in a coupled iteration. Haftka introduced the *simultaneous analysis and design* (SAND, [101]) method for structural optimization that breaks the nested optimization structure described above. Another terminology that is used when simultaneously iterating state and design is the *all-at-once* approach [68]. The term *one-shot* was introduced in this context by Ta'asan in [262]. He developed a multigrid one-shot approach in which the design update is performed on a converged state and adjoint equation on a coarse grid. This method was applied in [264] to an aerodynamic shape optimization problem under potential flow. Later, it was extended to infinite-dimensional design spaces to optimize on different scales of the grid [6, 7]. Iollo and Ta'asan also developed a single-grid one-shot approach in a pseudo-time context [263, 131]. In their strategy, a pseudo-time equation is iterated for state and adjoint, and the design equation is described as a boundary condition. The approaches by Ta'asan make use of the continuous adjoint method. Inspired by Ta'asan, Marco and Beux [176] used a hierarchical optimization strategy in combination with a discrete adjoint approach.

More recent developments of one-shot methods can be split into two main strategies.

In [113, 116, 114] an inexact reduced SQP approach is applied to the continuous optimization problem. The needed operators are discretized afterwards. The approach can be interpreted as a pseudo-time stepping method of the full KKT system. Thus, the values for the state variable and the adjoint variable are used in the design update as soon as they are available, as done in Gauss-Seidel type iteration strategies. More details on the inexact reduced SQP approach for one-shot optimization are given in Section A.2.2 of the appendix.

The present thesis focuses on the so-called single-step one-shot strategy. Opposed to the inexact reduced SQP one-shot approach, the approach presented in [89, 72, 102, 103] uses the state and adjoint information of the old iterate for the new design update. This can be interpreted as solving the optimality system with a Jacobi-type iteration procedure. The single-step one-shot strategy is derived from the discretized version of the PDE-constrained optimization problem, which will be introduced in the following. The approach can be applied to any optimization problem that is constrained by partial differential equations that are solved numerically by a contractive fixed-point iteration.

It can be expected that the Jacobi-type style of iteration of the single-step one-shot method has a slower convergence rate than the Seidel-type style, as this is the case for reduced SQP methods in nested strategies. However, numerical studies comparing both one-shot approaches show that, for the test case under consideration, the difference in the speed of convergence is negligible [99].

Both one-shot strategies have been applied in various contexts. The approach based on an inexact reduce SQP method is, for example, used for aerodynamic shape optimization in [111] and for robust aerodynamic design in [235]. The single-step one-shot strategy was applied in the context of aerodynamic shape optimization, for example, in [72, 210, 211, 71], and in [151] for parameter studies of a climate model. Recently a variant of the

3. One-Shot Method for PDE-constrained Optimization

single-step one-shot approach was developed and applied for the design of a magnetic diverter [20]. The advantage of one-shot methods is their efficiency compared to nested approaches. When the method is carefully constructed, i.e., using a suitable design space preconditioner, one can achieve bounded retardation of the convergence rate. This means that the costs for a one-shot optimization are only up to a small multiple of the costs for solving the underlying PDE constraint. The retardation factor relates computational costs for the one-shot methods to the costs for solving the state equation. It depends on the smoothness of the problem and the construction strategy. The factor can be measured based on the runtime, the number of fixed-point iterations, or the contraction rate of the coupled one-shot iteration and the fixed-point iteration (see e.g. [24]). In [71], the authors observe a bounded retardation factor of 4.6 for the Bratu problem. Here, the retardation factor is measured in terms of iterations. A retardation factor of 4 is, for example, achieved in [210] for aerodynamic shape optimization. The runtime factor depends on the runtime of the adjoint iteration and is usually about 10 times higher.

Current research in the context of the single-step one-shot method involves different aspects. Kaland et al. [140] have extended the convergence analysis of the single-step one-shot method to a function space setting. A one-shot strategy for optimization with unsteady PDE constraints based on the one-shot approach of [102] was proposed by Günther et al. [100]. Bosse et al. [25] developed a multistep Seidel-type one-shot method based on the ideas of the single-step one-shot method. A design is updated only after several iteration steps for the state equation and the adjoint equation. The authors derive local convergence properties for an appropriate preconditioner and a lower bound for the number of steps.

In the following, the formulation based on the fixed-point iteration is presented in Section 3.2.1. In Section 3.2.2, the piggy-back approach is presented, where the state and the adjoint equation are iterated simultaneously. The single-step one-shot strategy is presented in Section 3.2.3, focusing on theoretical results on the convergence since they will be of importance for the analysis of the one-shot approach augmented with equality constraints.

The new results of extending the one-shot method to additional equality constraints are presented in Section 3.2.4. It contains an overview of existing strategies, the presentation of the overall strategy, as well as details on the convergence analysis. The strategy is to extend the single-step one-shot method to an additional update for the constraint multiplier with a suitable preconditioner. The convergence analysis gives conditions on the design space preconditioner and the preconditioner for the multiplier update. The extension to inequality constraints is discussed in Section 3.2.5 by proposing a reformulation with the help of bound constraints. Details on the use of algorithmic differentiation in the one-shot approach are given in Section 3.2.6.

3.2.1. Fixed-Point Formulation

One assumes that the solution method for the discretized PDE constraints can be represented by the fixed-point iteration

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}) \quad (3.15)$$

for a fixed design \mathbf{u} with the function $G : Y \times U \rightarrow Y$ and $Y = \mathbb{R}^m$, $U = \mathbb{R}^n$. This assumption is natural as nonlinear PDEs are, for example, solved using pseudo-time stepping approaches with $G(\mathbf{y}, \mathbf{u}) := \mathbf{y} - P^{-1}(\mathbf{y}, \mathbf{u})c(\mathbf{y}, \mathbf{u})$, where $P : Y \times U \rightarrow Y$ is an appropriate preconditioner.

As before, one assumes that f and G are twice continuously differentiable.

Furthermore, for the following derivations it is necessary to assume that the fixed-point iteration is contractive with a constant contraction rate ρ such that

$$\|G_{\mathbf{y}}\| = \|G_{\mathbf{y}}^{\top}\| \leq \rho < 1, \quad (3.16)$$

and with the mean value theorem one obtains $\|G(\mathbf{y}, \mathbf{u}) - G(\tilde{\mathbf{y}}, \mathbf{u})\| \leq \rho\|\mathbf{y} - \tilde{\mathbf{y}}\|$ for all $\mathbf{y}, \tilde{\mathbf{y}} \in Y$ and $\mathbf{u} \in U$. It follows from Banach's fixed-point theorem that the fixed-point iteration converges with a linear rate of convergence to a unique state $\mathbf{y}^* = G(\mathbf{y}^*, \mathbf{u})$ for any fixed design \mathbf{u} . As a result, the PDE constraint can be replaced by the fixed-point equation $\mathbf{y} = G(\mathbf{y}, \mathbf{u})$ that has to be fulfilled for a KKT point of the optimality system.

The resulting PDE-constrained optimization problem, when considering fixed-point iterations, is then reformulated as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & f(\mathbf{y}, \mathbf{u}) \\ \text{s.t.} \quad & \mathbf{y} = G(\mathbf{y}, \mathbf{u}). \end{aligned} \quad (3.17)$$

It will be the basis for following discussions.

All derivations done for the optimization problem with the discretized PDE constraint described by $c(\mathbf{y}, \mathbf{u})$ (see Section 3.1) can be done analogously for the new expression $G(\mathbf{y}, \mathbf{u}) - \mathbf{y}$. The main results are presented in the following, replacing all original expressions.

Due to the introduction of fixed-point iteration one can describe the Lagrangian

$$L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) = N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) - \mathbf{y}^{\top} \bar{\mathbf{y}} \quad (3.18)$$

with the help of the *shifted Lagrangian*

$$N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) := f(\mathbf{y}, \mathbf{u}) + G(\mathbf{y}, \mathbf{u})^{\top} \bar{\mathbf{y}}. \quad (3.19)$$

The first-order necessary optimality conditions are given for a KKT point $(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)$ as

$$N_{\bar{\mathbf{y}}}(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)^{\top} - \mathbf{y}^* = G(\mathbf{y}^*, \mathbf{u}^*) - \mathbf{y}^* = \mathbf{0}, \quad (3.20a)$$

$$N_{\mathbf{y}}(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)^{\top} - \bar{\mathbf{y}}^* = \nabla_{\mathbf{y}} f(\mathbf{y}^*, \mathbf{u}^*) + G_{\mathbf{y}}(\mathbf{y}^*, \mathbf{u}^*)^{\top} \bar{\mathbf{y}}^* - \bar{\mathbf{y}}^* = \mathbf{0}, \quad (3.20b)$$

$$N_{\mathbf{u}}(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*)^{\top} = \nabla_{\mathbf{u}} f(\mathbf{y}^*, \mathbf{u}^*) + G_{\mathbf{u}}(\mathbf{y}^*, \mathbf{u}^*)^{\top} \bar{\mathbf{y}}^* = \mathbf{0}. \quad (3.20c)$$

3. One-Shot Method for PDE-constrained Optimization

Note that the second-order derivatives of N and L are identical, such that the Hessian is given by

$$H(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) := \begin{pmatrix} N_{\mathbf{y}\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) & N_{\mathbf{y}\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \\ N_{\mathbf{u}\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) & N_{\mathbf{u}\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \end{pmatrix}.$$

The results for second-order optimality conditions can thus be formulated with the reduced Hessian $Z^\top H Z$ based on

$$Z = \begin{pmatrix} (I_m - G_{\mathbf{y}}(\mathbf{y}, \mathbf{u}))^{-1} G_{\mathbf{u}}(\mathbf{y}, \mathbf{u}) \\ I_n \end{pmatrix}. \quad (3.21)$$

The matrix $(I - G_{\mathbf{y}})$ is nonsingular due to the contractivity of G and the perturbation Lemma (see [84, Lemma 2.3.3]).⁷ In the following, it is assumed that the second-order sufficient condition for optimality is fulfilled, i.e., that the reduced Hessian is positive definite at a local minimizer.

As before, the equations in (3.20) can be identified as the state equation, the adjoint equation, and the design equation. The conventional approach according to Algorithm 6 for solving problem (3.17) is to fully converge the fixed-point iteration of the state equation to the fixed-point \mathbf{y}^* and use the result for fully converging a fixed-point iteration for the adjoint equation, given by $\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}^*, \bar{\mathbf{y}}_k, \mathbf{u})^\top$. Afterwards, the reduced gradient $N_{\mathbf{u}}$ is computed and used for updating the design variable. Then the whole process is repeated until the optimization algorithm has fulfilled a given convergence criterion. Before introducing the one-shot approach, one may consider the case when the state equation and the adjoint equation are solved simultaneously in an iterative fashion.

3.2.2. Piggy-Back Iteration

Solving the state equation together with the adjoint equation in a simultaneous fashion is referred to as the *piggy-back* iteration [91]. The piggy-back iteration for a fixed design \mathbf{u} and the iteration index k

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}), \quad (3.22a)$$

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u})^\top \quad (3.22b)$$

does not require any preconditioning. The adjoint $\bar{\mathbf{y}}_k$ is not an exact adjoint of \mathbf{y}_k but an approximation with the same limit (cp. [91]). It can be shown that the convergence of the adjoint iteration is guaranteed if G is contractive. Due to the fact that $\|G_{\mathbf{y}}\| = \|G_{\mathbf{y}}^\top\|$, the iteration matrix $G_{\mathbf{y}}^\top$ of the adjoint iteration $N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}) = \nabla_{\mathbf{y}} f(\mathbf{y}_k, \mathbf{u})^\top + G_{\mathbf{y}}(\mathbf{y}_k, \mathbf{u})^\top \bar{\mathbf{y}}_k$ also fulfills the contraction property (3.16). As a consequence the fixed-point iteration for the adjoint is contractive and a fixed-point $\bar{\mathbf{y}}^*$ exists due to Banach's fixed-point theorem. Furthermore, as $G_{\mathbf{y}}$ and $f_{\mathbf{y}}$ are continuous, one can guarantee that the iterates of the adjoint variable converge to the solution of the adjoint equation.

⁷In the following, the indicator for showing the dimension of the identity matrix, e.g., $I_m \in \mathbb{R}^{m \times m}$ is omitted if it is given from the context.

The state iteration and the adjoint iteration converge with the same asymptotic rate to the fixed-point $(\mathbf{y}^*, \bar{\mathbf{y}}^*)$, but with a certain time lag of the adjoint variable. This effect is referred to as *dual retardation*. The same asymptotic rate of convergence is achieved since $G_{\mathbf{y}}$ and $G_{\mathbf{y}}^\top$ have the same spectral properties. The asymptotic convergence rate ρ , i.e.,

$$\limsup_{k \rightarrow \infty} \sqrt[k]{\|\mathbf{y}_k - \mathbf{y}^*\|} \leq \rho \quad \text{and} \quad \limsup_{k \rightarrow \infty} \sqrt[k]{\|\bar{\mathbf{y}}_k - \bar{\mathbf{y}}^*\|} \leq \rho,$$

is derived in [92]. Griewank and Kressner also show the dual retardation behavior, i.e., $\|\bar{\mathbf{y}}_k - \bar{\mathbf{y}}^*\| \sim k\|\mathbf{y}_k - \mathbf{y}^*\|$. The reason for the retardation of the iterates of the adjoint variable is that their accuracy depends on the accuracy of the iterates of the state variable. The resulting inaccuracies in the iterates of the adjoint variable accumulate during the piggy-back approach and cause a time lag in the adjoint iteration. It can be shown that this dual retardation is bounded [102].

3.2.3. Single-Step One-Shot Approach

The first ideas for the single-step one-shot approach are given in [89]. In the single-step one-shot approach one iterates for the state variable, the adjoint variable and the design variable simultaneously in a coupled iteration step, which is given by

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}_k), \quad (3.23a)$$

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k)^\top, \quad (3.23b)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k - B_k^{-1} N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k)^\top. \quad (3.23c)$$

In the following, $\Delta \mathbf{y}_k = \mathbf{y}_{k+1} - \mathbf{y}_k$, $\Delta \bar{\mathbf{y}}_k = \bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k$, and $\Delta \mathbf{u}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$ denote the updates for the state iteration (3.23a), the adjoint iteration (3.23b) and the design iteration (3.23c).

The design space preconditioner B has to be positive definite and large enough to guarantee convergence to a point fulfilling the first-order necessary conditions for optimality. In [89], Griewank used an eigenvalue analysis to derive a necessary condition for a preconditioner such that the iteration procedure described above is contractive. In [72], the authors introduced the idea of a doubly augmented Lagrangian as an exact penalty function to be reduced for deriving a design space preconditioner. Later, Hamdi and Griewank [102, 103] formulated conditions for obtaining descent of the one-shot iteration on the doubly augmented Lagrangian to show the convergence of the overall strategy. An overview of theoretical results based on the doubly augmented Lagrangian will be presented in the next sections, based on the derivations in [102, 103]. This includes the results for the convergence analysis, as well as the derivation of a suitable design space preconditioner.

3. One-Shot Method for PDE-constrained Optimization

3.2.3.1. Convergence Analysis

In [72, 102, 103] the optimization problem is identified as the minimization of a merit function of a doubly augmented Lagrangian type, given by

$$L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) = \frac{\alpha}{2} \|G(\mathbf{y}, \mathbf{u}) - \mathbf{y}\|^2 + \frac{\beta}{2} \|N_{\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})^\top - \bar{\mathbf{y}}\|^2 + N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) - \bar{\mathbf{y}}^\top \mathbf{y}. \quad (3.24)$$

The gradient of L^a is given as

$$\begin{pmatrix} \nabla_{\mathbf{y}} L^a \\ \nabla_{\bar{\mathbf{y}}} L^a \\ \nabla_{\mathbf{u}} L^a \end{pmatrix} = -Ms(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \quad (3.25)$$

with

$$M = \begin{pmatrix} \alpha(I - G_{\mathbf{y}})^\top & -I - \beta N_{\mathbf{y}\mathbf{y}} & 0 \\ -I & \beta(I - G_{\mathbf{y}}) & 0 \\ -\alpha G_{\mathbf{u}}^\top & -\beta N_{\mathbf{y}\mathbf{u}}^\top & B \end{pmatrix} \quad (3.26)$$

and

$$s(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) = \begin{pmatrix} G(\mathbf{y}, \mathbf{u}) - \mathbf{y} \\ N_{\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})^\top - \bar{\mathbf{y}} \\ -B^{-1}N_{\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})^\top \end{pmatrix}. \quad (3.27)$$

The increment vector $s(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) = (\Delta\mathbf{y}, \Delta\bar{\mathbf{y}}, \Delta\mathbf{u})^\top$ is also the one-shot increment vector. It can be shown that under some conditions the doubly augmented Lagrangian is an exact penalty function and that the one-shot increment vector is a descent direction for the doubly augmented Lagrangian. This is outlined in the following by presenting the most important results of [102].

Exact Penalty Function One can show that under the above assumptions, i.e., that the necessary and sufficient conditions for optimality are fulfilled for the original problem, the stationary points of the augmented Lagrangian and the original optimization problem coincide under the conditions that are presented in the following. Additionally, the second-order sufficient condition for optimality is fulfilled for the doubly augmented Lagrangian. Then according to Definition 12, one can show that the doubly augmented Lagrangian is an exact penalty function. As a result, a strict local minimizer of the original problem is also a strict local minimizer of the doubly augmented Lagrangian.

The *correspondence condition*

$$\alpha\beta(1 - \rho)^2 > I + \beta\|N_{\mathbf{y}\mathbf{y}}\| \quad (3.28)$$

derived by Hamdi and Griewank (see [102, Corollary 3.2]) is sufficient for fulfilling

$$\det(\alpha\beta(I - G_{\mathbf{y}})^\top(I - G_{\mathbf{y}}) - I - \beta N_{\mathbf{y}\mathbf{y}}) \neq 0. \quad (3.29)$$

This ensures that M is nonsingular and, as a result, the stationary points of the augmented Lagrangian are also the stationary points of the original optimization problem.

Another condition for the two strictly positive weighting coefficients α and β gives the positive definiteness of the Hessian of L^a . This is derived in [102, Corollary 3.3].

Theorem 13 (Exact penalty function). *If the condition*

$$\alpha\beta(I - G_{\mathbf{y}})^\top(I - G_{\mathbf{y}}) \succ I + \beta N_{\mathbf{y}\mathbf{y}} \quad (3.30)$$

is fulfilled then the Hessian of L^a is positive definite in all stationary points of L^a .

Proof. The proof can be found in [102]. It is based on a diagonalization of the Hessian of L^a and requires positive definiteness of the reduced Hessian of the original optimization problem (3.17) which is given by assumption. \square

Condition (3.30) is implied by the correspondence condition (3.28). At the same time, condition (3.30) implies nonsingularity of M . Thus, given condition (3.30), L^a is an exact penalty function since the local minimizer of (3.17) is also a local minimizer of L^a .

Descent Direction Hamdi and Griewank also derive the descent direction condition (see [102, Proposition 3.4]) to show that the one-shot increment vector yields descent on the doubly augmented Lagrangian.

Theorem 14 (Descent direction condition). *If the condition*

$$\alpha\beta\Delta\bar{G}_{\mathbf{y}} \succ (I + \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}})(\Delta\bar{G}_{\mathbf{y}})^{-1}(I + \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}}) \quad (3.31)$$

with $\Delta\bar{G}_{\mathbf{y}} = \frac{1}{2}(I - G_{\mathbf{y}} + (I - G_{\mathbf{y}})^\top)$ is fulfilled, the step increment vector s yields descent on L^a for all large positive preconditioners B .

Proof. The proof can be found in [102]. The increment vector is a descent direction if $s^\top L^a < 0$, meaning that M has to be positive definite. This is shown via positive definiteness of its symmetric part

$$M_s = \frac{1}{2}(M + M^\top),$$

given by

$$M_s = \begin{pmatrix} \alpha\Delta\bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & -\frac{\alpha}{2}G_{\mathbf{u}} \\ -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & \beta\Delta\bar{G}_{\mathbf{y}} & -\frac{\beta}{2}N_{\mathbf{y}\mathbf{u}} \\ -\frac{\alpha}{2}G_{\mathbf{u}}^\top & -\frac{\beta}{2}N_{\mathbf{y}\mathbf{u}}^\top & B \end{pmatrix}. \quad (3.32)$$

For this, the matrix is subdivided into blocks $(A, C; C^\top B)$ with

$$A := \begin{pmatrix} \alpha\Delta\bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} \\ -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & \beta\Delta\bar{G}_{\mathbf{y}} \end{pmatrix}$$

and $C := (-\frac{\alpha}{2}G_{\mathbf{u}}, -\frac{\alpha}{2}G_{\mathbf{u}})^\top$. With block Gaussian elimination it remains to show that $\text{diag}(A, B - C^\top A^{-1}C)$ is positive definite. Positive definiteness of A is given by (3.31) and for the second part B has to be large enough, such that $B \succ C^\top A^{-1}C$. \square

3. One-Shot Method for PDE-constrained Optimization

If α and β satisfy the stronger condition

$$\sqrt{\alpha\beta}(1-\rho) > 1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|, \quad (3.33)$$

then condition (3.28) and (3.31) are automatically fulfilled.

In [102], the authors propose a suitable choice for the coefficients α and β such that α is large enough to ensure a monotonic reduction but not too large to avoid a slow-down of convergence. These values can be derived by minimizing α with respect to β with the condition (3.33) as a constraint. Then suitable values for α and β are⁸, for example,

$$\beta = \frac{2}{\|N_{\mathbf{y}\mathbf{y}}\|} \text{ and } \alpha > \frac{2\|N_{\mathbf{y}\mathbf{y}}\|}{(1-\rho)^2}. \quad (3.34)$$

Since the condition on B that is needed for showing that s is a descent direction is not very practicable, new results were derived by Hamdi and Griewank in [103].

Theorem 15. *If condition (3.33) is fulfilled and*

$$\left(\frac{\sqrt{\alpha}}{2}\|G_{\tilde{\mathbf{u}}}\| + \frac{\sqrt{\beta}}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \right)^2 \leq (1-\rho) - \frac{\left(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|\right)^2}{\alpha\beta(1-\rho)}, \quad (3.35)$$

with $G_{\tilde{\mathbf{u}}} := G_{\mathbf{u}}B^{\frac{1}{2}}$ and $N_{\mathbf{y}\tilde{\mathbf{u}}} := N_{\mathbf{y}\mathbf{u}}B^{-\frac{1}{2}}$, then s yields descent on the doubly augmented Lagrangian.

Proof. The proof can be found in [103, Proposition 3.2]. It is presented in this work since the results will later be important for the theoretical analysis of the extended one-shot strategy. For symmetric positive definite B the matrix M_s given in Equation (3.32) can be rescaled in such a way that

$$\tilde{M}_s = \text{diag}(I, I, B^{-\frac{\tau}{2}})M_s\text{diag}(I, I, B^{-\frac{1}{2}}) = \begin{pmatrix} \alpha\Delta\bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & -\frac{\alpha}{2}G_{\tilde{\mathbf{u}}} \\ -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & \beta\Delta\bar{G}_{\mathbf{y}} & -\frac{\beta}{2}N_{\mathbf{y}\tilde{\mathbf{u}}} \\ -\frac{\alpha}{2}G_{\tilde{\mathbf{u}}}^{\top} & -\frac{\beta}{2}N_{\mathbf{y}\tilde{\mathbf{u}}}^{\top} & B \end{pmatrix}.$$

It is shown in [103, Proposition 3.1] that for the 3×3 matrix

$$D_c := \begin{pmatrix} \alpha(1-\rho) & -1 - \frac{\beta}{2}\theta & -\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\| \\ -1 - \frac{\beta}{2}\theta & \beta(1-\rho) & -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \\ -\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\| & -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| & 1 \end{pmatrix}$$

the inequality

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}^{\top} \tilde{M}_s \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \geq \begin{bmatrix} \|v_1\| \\ \|v_2\| \\ \|v_3\| \end{bmatrix}^{\top} D_c \begin{bmatrix} \|v_1\| \\ \|v_2\| \\ \|v_3\| \end{bmatrix}$$

⁸Note that the exact choice $\alpha = \frac{2\|N_{\mathbf{y}\mathbf{y}}\|}{(1-\rho)^2}$ violates the condition (3.33).

holds. Thus, it remains to show that D_c is symmetric positive definite. This can be done by subdividing the matrix into blocks such that

$$D = \begin{pmatrix} \alpha(1 - \rho) & -1 - \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\| \\ -1 - \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\| & \beta(1 - \rho) \end{pmatrix}$$

and $d = (-\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\|, -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\|)$. The matrix D_c is positive definite if D is positive definite, which is implied by condition (3.33), and the Schur complement $1 - d^\top D^{-1}d$ is positive definite which is the case for condition (3.35). \square

Global Convergence In [103], the authors also establish a global convergence result for the single-step one-shot approach, when choosing α , β and the preconditioner B as described before. The underlying assumption is that it is made use of a line search, e.g., a standard backtracking line search, for the search direction s to get sufficient decrease of the exact penalty function L^a . Then one can show under the assumption $N_{\mathbf{u}\mathbf{u}} \succ 0$ that the angle γ between the direction of steepest descent $-\nabla L^a$ and the search direction s is bounded away from $\frac{\pi}{2}$, which means that

$$\cos(\gamma) = -\frac{s^\top \nabla L^a}{\|\nabla L^a\| \|s\|} \geq C > 0 \quad (3.36)$$

for all iterates $(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})$. Furthermore, for a starting point $(\mathbf{y}_0, \bar{\mathbf{y}}_0, \mathbf{u}_0)$ all iterates are contained in the level set \mathcal{N}_0 of L^a defined as

$$\mathcal{N}_0 := \{(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \mid L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}) \leq L^a(\mathbf{y}_0, \bar{\mathbf{y}}_0, \mathbf{u}_0)\}$$

if the conditions

$$\lim_{\|\mathbf{y}\| + \|\mathbf{u}\| \rightarrow \infty} f(\mathbf{y}, \mathbf{u}) = +\infty \quad (3.37)$$

and

$$\liminf_{\|\mathbf{y}\| + \|\mathbf{u}\| \rightarrow \infty} \frac{f(\mathbf{y}, \mathbf{u})}{\|\nabla_{\mathbf{y}} f\|^2} > 0 \quad (3.38)$$

hold. Condition (3.37) is given by assumption. Conditions (3.37) and (3.38) then ensure that all level sets of L^a are bounded and therefore L^a is bounded from below (compare [103, Theorem 2.1]). Using these results and the convergence theorem for line search procedures, which is, for example, given in [206, p.38], one obtains global convergence of the one-shot approach, which means that

$$\lim_{k \rightarrow \infty} \|\nabla L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k)\| = 0. \quad (3.39)$$

3.2.3.2. Choice of the Preconditioner

A preconditioner can be derived with the help of condition (3.35) as done in [103]. As a result, if the condition (3.33) is fulfilled, any preconditioner that satisfies

$$B \succcurlyeq B_0 := \frac{1}{\sigma} (\alpha G_{\mathbf{u}}^\top G_{\mathbf{u}} + \beta N_{\mathbf{y}\mathbf{u}}^\top N_{\mathbf{y}\mathbf{u}}) \quad (3.40)$$

3. One-Shot Method for PDE-constrained Optimization

with

$$\sigma = 1 - \rho - \frac{(1 + \frac{\|N_{\mathbf{y}\mathbf{y}}\|}{2}\beta)^2}{\alpha\beta(1 - \rho)} \quad (3.41)$$

yields descent on L^a .

The matrix B_0 is strongly related to the Hessian $\nabla_{\mathbf{u}\mathbf{u}}L^a$ with respect to the design, which is shown in [103]. A preconditioner that fulfills (3.40) can be derived by considering the optimization problem

$$\min_{\Delta \mathbf{u}} L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u} + \Delta \mathbf{u}) \quad (3.42)$$

and finding B from the condition $\Delta \mathbf{u} = -B^{-1}N_{\mathbf{u}}^\top$. When using a sequential quadratic programming approach for problem (3.42) one obtains the minimization problem

$$\min_{\Delta \mathbf{u}} \mathbf{s}^\top \nabla L^a + \frac{1}{2} \mathbf{s}^\top \nabla^2 L^a \mathbf{s},$$

which is equivalent to the problem

$$\min_{\Delta \mathbf{u}} \Delta \mathbf{u}^\top (\nabla_{\mathbf{u}}L^a + \nabla_{\mathbf{u}\mathbf{y}}L^a \Delta \mathbf{y} + \nabla_{\mathbf{u}\bar{\mathbf{y}}}L^a \Delta \bar{\mathbf{y}}) + \frac{1}{2} \Delta \mathbf{u}^\top \nabla_{\mathbf{u}\mathbf{u}}L^a \Delta \mathbf{u}$$

and can be approximated by the minimization problem

$$\min_{\Delta \mathbf{u}} \Delta \mathbf{u}^\top \nabla_{\mathbf{u}}L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u}) + \frac{1}{2} \Delta \mathbf{u}^\top \nabla_{\mathbf{u}\mathbf{u}}L^a \Delta \mathbf{u}. \quad (3.43)$$

The solution to (3.43) is given by $\Delta \mathbf{u} = -\nabla_{\mathbf{u}\mathbf{u}}^{-1}L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})\nabla_{\mathbf{u}}L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u})$. One identifies $B \approx \nabla_{\mathbf{u}\mathbf{u}}L^a$. The choice $B = \alpha G_{\mathbf{u}}^\top G_{\mathbf{u}} + \beta N_{\mathbf{y}\mathbf{u}}^\top N_{\mathbf{y}\mathbf{u}} + N_{\mathbf{u}\mathbf{u}}$ is equivalent to the Hessian $\nabla_{\mathbf{u}\mathbf{u}}L^a$ if state and adjoint feasibility are satisfied. One chooses

$$\tilde{B} = \frac{1}{\sigma} B = \frac{1}{\sigma} (\alpha G_{\mathbf{u}}^\top G_{\mathbf{u}} + \beta N_{\mathbf{y}\mathbf{u}}^\top N_{\mathbf{y}\mathbf{u}} + N_{\mathbf{u}\mathbf{u}}) \quad (3.44)$$

such that condition (3.40) is fulfilled for $N_{\mathbf{u}\mathbf{u}} \succcurlyeq 0$.⁹

In practice, B is not computed exactly, but the inverse of the Hessian is approximated by means of a BFGS update. Since one has $B \approx \nabla_{\mathbf{u}\mathbf{u}}L^a$,

$$B\Delta \mathbf{u} \approx \nabla_{\mathbf{u}}L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u} + \Delta \mathbf{u}) - \nabla_{\mathbf{u}}L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u})$$

is fulfilled. The equation can be used as a secant equation for a BFGS algorithm. Thus, the secant equation for the approximated inverse of B is given as

$$B_{k+1}^{-1} \mathbf{r}_k = \Delta \mathbf{u}_k \quad (3.45)$$

with $\mathbf{r}_k := \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k + \Delta \mathbf{u}_k) - \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k)$ and $\mathbf{s}_k := \Delta \mathbf{u}_k$. B_k^{-1} is updated with the BFGS update formula for the inverse as given in (2.26), i.e.,

$$B_{k+1}^{-1} = \left(I - \frac{\mathbf{s}_k \mathbf{r}_k^\top}{\mathbf{r}_k^\top \mathbf{s}_k}\right) B_k^{-1} \left(I - \frac{\mathbf{r}_k \mathbf{s}_k^\top}{\mathbf{r}_k^\top \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^\top}{\mathbf{r}_k^\top \mathbf{s}_k}. \quad (3.46)$$

⁹Note that $\sigma < 1 - \rho$ due to condition (3.33).

It is important to apply this update only if the positive definiteness of B is maintained which means that the curvature condition

$$\mathbf{r}_k^\top \mathbf{s}_k > 0 \quad (3.47)$$

is fulfilled. If this is not the case, it is convenient to set $B_{k+1}^{-1} = I$ or $B_{k+1}^{-1} = B_k^{-1}$. Another possibility, which is also proposed in [103], is to use a line search that guarantees to satisfy the second Wolfe condition (2.30) for a design update in the doubly augmented Lagrangian.

3.2.4. The One-Shot Approach with Additional Equality Constraints

In the above derivation, only PDE constraints were considered. When introducing additional constraints, the one-shot methodology has to be adjusted. The following approach is restricted to equality constraints.

Additional equality constraints can be integrated into optimization strategies in various ways. Here, following [111], it is mainly distinguished between a direct treatment and an indirect treatment of constraints. Examples of direct and indirect treatments of constraints were already introduced in Section 2.3.2. One refers to a direct treatment if the constraints are explicitly integrated in the optimization procedure by any projection-based method, SQP method, or with any other strategy enabling feasibility of the constraint function or a linearization of it in intermediate optimization steps. If, for example, additional variables can be defined for directly adjusting the constraint function, it is possible to achieve feasibility in each optimization step. An indirect treatment is the use of penalty methods or any similar strategy in which a new objective function is formulated, and the reduced unconstrained optimization problem is solved either with a constant multiplier or an update strategy for the multiplier.

A direct treatment of constraints in the simultaneous pseudo-time-stepping approach is proposed in [114, 111], and applied for aerodynamic shape optimization. Similarly to the procedure for the PDE constraint, the design update is projected onto the tangent space of the linearized constraints leading to a reduced system to be solved. The solution procedure involves the additional solution of the modified adjoint equation to find a reduced gradient of the additional constraints. The projection is further corrected for nonlinear constraints. Theoretical results on convergence are not given, but the numerical results for aerodynamic shape optimization show a successful optimization behavior requiring roughly ten times as much computational effort as the flow solution.

An indirect treatment of constraints in this context can be found in [115], where the additional constraint is added to the objective function with a corresponding multiplier. The reduced optimization problem is solved approximately in an inner iteration of the optimization. Furthermore, the multiplier is based on an additional variable that can be introduced to directly adjust the constraint function in the outer iteration of the optimization. In the numerical study, this is the angle of attack, which can be adjusted to reach a desired lift constraint in aerodynamic shape optimization. The multiplier is kept constant but defined with the help of sensitivities with respect to the additional variable.

3. One-Shot Method for PDE-constrained Optimization

In the single-step one-shot approach, the coordination of state, adjoint, and design iteration is well established when only the state equation is considered as an equality constraint. In numerical studies of the single-step one-shot approach, additional equality or inequality constraints are most of the time treated indirectly with the help of penalty methods (see e.g. [211]). However, although the numerical results are successful, there does not exist any theoretical or numerical analysis for the definition of penalty multipliers in the one-shot framework. As a result, an updating strategy for multipliers is not well-defined, which may lead to a reduced speed of convergence or even failure of the overall strategy.

In [23], the multistep one-shot method based on the fixed-point iteration is extended to additional equality constraints. In the analysis, the constraint is integrated via an additional term in the Lagrangian with a corresponding multiplier. However, similar to [112], the general assumption for this approach is that the state space is extended in such a way that the constraint function can be directly controlled. As a result, an adjoint equation for the additional state variables can be formulated, and the corresponding adjoint variables explicitly describe the penalty multiplier in the Lagrange function. This enables a direct treatment of the constraint function in the optimization procedure. If the solution to the constraint function can be formulated in a fixed-point form for the additional state variables with a suitable preconditioner, the original fixed-point iteration for the state equation can be extended to the augmented state space. A convergence analysis for the extended fixed-point iterator and a derivation of a lower bound for the number of steps in the multistep one-shot approach are presented in [23] alongside the first numerical results for a simple test case.

The assumption that additional variables can be introduced to enable a direct adjustment of constraints is often not realistic. In aerodynamic design, for example, it might be of interest to fulfill a lift constraint without adjusting an angle of attack. Furthermore, it is not possible to define any additional parameters that can be adjusted to satisfy, for example, a moment constraint. This inspired the development of a well-defined framework for introducing equality constraints in the single-step one-shot method in situations where no additional variables can be introduced.

This section presents the idea of the extended single-step one-shot approach as well as the results of its analysis to additional equality constraints. First ideas for introducing equality constraints in the theoretical framework of the single-step one-shot approach are formulated in a thesis by Richert [229]. The first theoretical results are published by Walther et al. in [278]. The contribution of the present thesis to this work is the provision of first numerical results as well as the identification of a suitable design space preconditioner. Furthermore, in collaboration with Andrea Walther, new theoretical results with simpler conditions are derived, which are presented in [280]. The one-shot method with additional equality constraints is firstly used in a multi-objective optimization problem in [154]. In the following, the ideas will be presented together with the results on convergence properties of the proposed method, focusing on the new results presented in [280].

One may extend the original shape optimization problem (3.17) based on the fixed-

point form and consider problems of the form

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & f(\mathbf{y}, \mathbf{u}) \\ \text{s.t.} \quad & \mathbf{y} = G(\mathbf{y}, \mathbf{u}), \\ & \mathbf{0} = h(\mathbf{y}, \mathbf{u}). \end{aligned} \quad (3.48)$$

The additional equality constraints are described by $h : Y \times U_{ad} \rightarrow V$, where V is a finite dimensional Hilbert space with $r = \dim(V)$ and $r \in \mathbb{N}$, $r \leq m$.¹⁰ Instead of introducing a new notation, the notation of Section 3.2.3 is replaced in the following.

The Lagrangian for the optimization problem (3.48) reads

$$L(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) = f(\mathbf{y}, \mathbf{u}) + (G(\mathbf{y}, \mathbf{u}) - \mathbf{y})^\top \bar{\mathbf{y}} + h(\mathbf{y}, \mathbf{u})^\top \boldsymbol{\mu} = N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) - \mathbf{y}^\top \bar{\mathbf{y}}, \quad (3.49)$$

where

$$N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) := f(\mathbf{y}, \mathbf{u}) + G(\mathbf{y}, \mathbf{u})^\top \bar{\mathbf{y}} + h(\mathbf{y}, \mathbf{u})^\top \boldsymbol{\mu} \quad (3.50)$$

is the new shifted Lagrangian and the vector of Lagrangian multipliers associated to the additional equality constraints is $\boldsymbol{\mu} \in \mathbb{R}^r$.

In the following, all assumptions of Section 3.2.3 hold. Additionally, one assumes that h is twice continuously differentiable for all components and that the full constraint Jacobian

$$A := \begin{pmatrix} I_m - G_{\mathbf{y}} & -G_{\mathbf{u}} \\ -h_{\mathbf{y}} & -h_{\mathbf{u}} \end{pmatrix} \in \mathbb{R}^{(m+p) \times (m+n)}$$

has full rank. As a result, the LICQ condition is fulfilled.

As before, a stationary point $(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*)$ of the problem must satisfy the first-order necessary optimality conditions

$$\nabla_{\bar{\mathbf{y}}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*) = G(\mathbf{y}^*, \mathbf{u}^*) - \mathbf{y}^* = \mathbf{0}, \quad (3.51a)$$

$$\nabla_{\mathbf{y}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*) = N_{\mathbf{y}}(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*)^\top - \bar{\mathbf{y}}^* = \mathbf{0}, \quad (3.51b)$$

$$\nabla_{\mathbf{u}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*) = N_{\mathbf{u}}(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*)^\top = \mathbf{0}, \quad (3.51c)$$

$$\nabla_{\boldsymbol{\mu}} L(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{u}^*, \boldsymbol{\mu}^*) = h(\mathbf{y}^*, \mathbf{u}^*) = \mathbf{0}. \quad (3.51d)$$

Again, for second-order necessary and sufficient conditions, the reduced Hessian $H_r = Z^\top H Z$ with

$$H(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) := \begin{pmatrix} N_{\mathbf{y}\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) & N_{\mathbf{y}\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) \\ N_{\mathbf{u}\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) & N_{\mathbf{u}\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) \end{pmatrix}.$$

has to be positive (semi-)definite in the locally optimal solution. Here, $Z \in \mathbb{R}^{(m+n) \times (n-r)}$ has to span the basis of the full constraint Jacobian A . One can construct Z with the help

¹⁰Note that the equality constraint $-h(\mathbf{y}, \mathbf{u}) = 0$ is chosen to guarantee for a consistent handling of multipliers in the Lagrangian. Therefore, for the equality constraint $\tilde{h} = 0$ one has to use the multiplier $\tilde{\boldsymbol{\mu}}$ with $\tilde{\boldsymbol{\mu}}_{k+1} = \tilde{\boldsymbol{\mu}}_k + \tilde{B}_k^{-1} \tilde{h}(\mathbf{y}_k, \mathbf{u}_k)$ in the Lagrangian.

3. One-Shot Method for PDE-constrained Optimization

of splitting the constraint Jacobian and expressing it in terms of the quadratic submatrix $A_1 \in \mathbb{R}^{(m+r) \times (m+r)}$ and the submatrix $A_2 \in \mathbb{R}^{(m+r) \times (n-r)}$, such that $A = (A_1, A_2)$ (see [278]). If the quadratic submatrix is regular, which is implied by assuming full rank of the full constraint Jacobian, one obtains

$$Z = \begin{pmatrix} -A_1^{-1}A_2 \\ I_{n-r} \end{pmatrix}. \quad (3.52)$$

Again, one assumes that the second-order sufficient condition for optimality is fulfilled, i.e., the reduced Hessian is positive definite at a local minimizer of (3.48).

Now, the ideas of the single-step one-shot method without additional constraints can be transferred to the problem with additional constraints with the help of the KKT system (3.51). As a result, the original single-step one-shot method can then be augmented by an iteration for the additional Lagrange multiplier μ resulting in the augmented one-shot iteration

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}_k), \quad (3.53a)$$

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top, \quad (3.53b)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k - B_k^{-1}N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top, \quad (3.53c)$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \check{B}_k^{-1}h(\mathbf{y}_k, \mathbf{u}_k), \quad (3.53d)$$

where B_k and \check{B}_k are suitably chosen symmetric and invertible preconditioners. Again, one may identify the state iteration, the adjoint iteration and the design iteration. The *augmented iteration* (3.53d) is the iteration for the Lagrange multiplier μ .

3.2.4.1. Convergence Analysis

The basic ideas of the convergence analysis of the original single-step one-shot method, which is presented in Section 3.2.3.1, can be extended to obtain convergence results for the augmented one-shot method. One introduces the doubly augmented Lagrangian

$$L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) = \frac{\alpha}{2} \left(\|G(\mathbf{y}, \mathbf{u}) - \mathbf{y}\|^2 + \|h(\mathbf{y}, \mathbf{u})\|^2 \right) + \frac{\beta}{2} \left\| N_{\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^\top - \bar{\mathbf{y}} \right\|^2 + N(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) - \mathbf{y}^\top \bar{\mathbf{y}}, \quad (3.54)$$

with positive penalty parameters $\alpha, \beta \in \mathbb{R}$. In the following, the aim is to show that L^a is an exact penalty function.

Exact Penalty Function A first step to show that L^a is an exact penalty function is to show that the stationary points of L^a and the optimization problem (3.48) coincide. This was done in [278].

For notational convenience the expression $\Delta G_{\mathbf{y}} := (I - G_{\mathbf{y}})$ is used. The gradient of L^a is given as

$$\begin{pmatrix} \nabla_{\mathbf{y}} L^a \\ \nabla_{\bar{\mathbf{y}}} L^a \\ \nabla_{\mathbf{u}} L^a \\ \nabla_{\boldsymbol{\mu}} L^a \end{pmatrix} = -Ms(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) \quad (3.55)$$

with

$$M = \begin{pmatrix} \alpha \Delta G_{\mathbf{y}}^{\top} & -(I + \beta N_{\mathbf{y}\mathbf{y}}) & 0 & \alpha h_{\mathbf{y}}^{\top} \check{B} \\ -I & \beta \Delta G_{\mathbf{y}} & 0 & 0 \\ -\alpha G_{\mathbf{u}}^{\top} & -\beta N_{\mathbf{y}\mathbf{u}}^{\top} & B & \alpha h_{\mathbf{u}}^{\top} \check{B} \\ 0 & -\beta h_{\mathbf{y}} & 0 & \check{B} \end{pmatrix}, \quad (3.56)$$

and the increment vector

$$s(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) = \begin{pmatrix} G(\mathbf{y}, \mathbf{u}) - \mathbf{y} \\ N_{\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^{\top} - \bar{\mathbf{y}} \\ -B^{-1} N_{\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^{\top} \\ -\check{B}^{-1} h(\mathbf{y}, \mathbf{u}) \end{pmatrix}. \quad (3.57)$$

One denotes the multiplier update as $\Delta \boldsymbol{\mu} = -\check{B}^{-1} h(\mathbf{y}, \mathbf{u})$. Again, one is able to observe that the increment vector $s = (\Delta \mathbf{y}, \Delta \bar{\mathbf{y}}, \Delta \mathbf{u}, \Delta \boldsymbol{\mu})^{\top}$ is also the one-shot increment vector.

Analogously to the derivation of the correspondence condition (3.28), the idea is to show that the matrix M is nonsingular. It can be shown (see [278, Proposition 3.2]) that M is nonsingular if the correspondence condition holds for the new shifted Lagrangian, i.e.,

$$\alpha \beta (1 - \rho)^2 > I + \beta \|N_{\mathbf{y}\mathbf{y}}\| \quad (3.58)$$

and the preconditioners B and \check{B} are nonsingular, which is given by assumption. The proof is based on standard rules for determinants to show, with the help of the Sherman-Morrison-Woodbury formula [84, p.63], that the determinant of M is nonzero.

The analysis of the Hessian of the augmented Lagrangian is not as straightforward. It is done in the following in correspondence with the results of [280].

The Hessian of the augmented Lagrangian at a stationary point can be represented by the summation

$$\nabla^2 L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) = H_1 + H_2 + H_3 \quad (3.59)$$

with

$$H_1 = \begin{pmatrix} H_{11} & 0 \\ 0 & \beta h_{\mathbf{y}} h_{\mathbf{y}}^{\top} - \frac{1}{\alpha} I_p \end{pmatrix},$$

$$H_{11} = \begin{pmatrix} \alpha \Delta G_{\mathbf{y}}^{\top} \Delta G_{\mathbf{y}} + (I + \beta N_{\mathbf{y}\mathbf{y}}) N_{\mathbf{y}\mathbf{y}} & -(I + \beta N_{\mathbf{y}\mathbf{y}}) \Delta G_{\mathbf{y}}^{\top} & -\alpha \Delta G_{\mathbf{y}}^{\top} G_{\mathbf{u}} + (I + \beta N_{\mathbf{y}\mathbf{y}}) N_{\mathbf{y}\mathbf{u}} \\ -\Delta G_{\mathbf{y}} (I + \beta N_{\mathbf{y}\mathbf{y}}) & \beta \Delta G_{\mathbf{y}} \Delta G_{\mathbf{y}}^{\top} & G_{\mathbf{u}} - \beta \Delta G_{\mathbf{y}} N_{\mathbf{y}\mathbf{u}} \\ -\alpha G_{\mathbf{u}}^{\top} \Delta G_{\mathbf{y}} + N_{\mathbf{u}\mathbf{y}} (I + \beta N_{\mathbf{y}\mathbf{y}}) & G_{\mathbf{u}}^{\top} - \beta N_{\mathbf{u}\mathbf{y}} \Delta G_{\mathbf{y}}^{\top} & \alpha G_{\mathbf{u}}^{\top} G_{\mathbf{u}} + \beta N_{\mathbf{u}\mathbf{y}} N_{\mathbf{y}\mathbf{u}} + N_{\mathbf{u}\mathbf{u}} \end{pmatrix},$$

$$H_2 = \begin{pmatrix} \alpha h_{\mathbf{y}}^{\top} h_{\mathbf{y}} & 0 & \alpha h_{\mathbf{y}}^{\top} h_{\mathbf{u}} & h_{\mathbf{y}}^{\top} \\ 0 & 0 & 0 & 0 \\ \alpha h_{\mathbf{u}}^{\top} h_{\mathbf{y}} & 0 & \alpha h_{\mathbf{u}}^{\top} h_{\mathbf{u}} & h_{\mathbf{u}}^{\top} \\ h_{\mathbf{y}} & 0 & h_{\mathbf{u}} & \frac{1}{\alpha} I \end{pmatrix},$$

3. One-Shot Method for PDE-constrained Optimization

and

$$H_3 = \begin{pmatrix} 0 & 0 & 0 & \beta N_{\mathbf{y}\mathbf{y}} h_{\mathbf{y}}^\top \\ 0 & 0 & 0 & -\beta \Delta G_{\mathbf{y}} h_{\mathbf{y}}^\top \\ 0 & 0 & 0 & \beta N_{\mathbf{u}\mathbf{y}} h_{\mathbf{y}}^\top \\ \beta h_{\mathbf{y}} N_{\mathbf{y}\mathbf{y}} & -\beta h_{\mathbf{y}} \Delta G_{\mathbf{y}}^\top & \beta h_{\mathbf{y}} N_{\mathbf{y}\mathbf{u}} & 0 \end{pmatrix}.$$

Note that the terms $-\frac{1}{\alpha}I$ in H_1 and $\frac{1}{\alpha}I$ in H_2 cancel each other out and are introduced for analysis purposes in the proof of the next theorem.

Theorem 16. *The Hessian of the doubly augmented Lagrangian is positive definite at all its stationary points if*

$$\alpha \beta \Delta G_{\mathbf{y}}^\top \Delta G_{\mathbf{y}} > I + \beta N_{\mathbf{y}\mathbf{y}}, \quad (3.60)$$

$$\alpha \beta h_{\mathbf{y}} h_{\mathbf{y}}^\top > I \quad (3.61)$$

and β is chosen small enough.

Proof. One may identify H_{11} as the Hessian of the augmented Lagrangian as introduced by Hamdi and Griewank in [102], but with a modified shifted Lagrangian. As a result, one can use the same ideas as in [102, Corollary 3.3] to show that the condition (3.60) ensures that H_{11} is positive definite. Since H_2 can be expressed as

$$H_2 = \begin{pmatrix} \sqrt{\alpha} h_{\mathbf{y}}^\top \\ 0 \\ \sqrt{\alpha} h_{\mathbf{u}}^\top \\ \frac{1}{\sqrt{\alpha}} I \end{pmatrix} \begin{pmatrix} \sqrt{\alpha} h_{\mathbf{y}} & 0 & \sqrt{\alpha} h_{\mathbf{u}} & \frac{1}{\sqrt{\alpha}} I \end{pmatrix},$$

one can conclude that H_2 is positive semidefinite. Therefore, the sum $H_1 + H_2$ is positive definite if

$$\beta h_{\mathbf{y}} h_{\mathbf{y}}^\top - \frac{1}{\alpha} I \succ 0$$

holds, giving condition (3.61). For the third matrix it can be shown (see e.g. [280, Lemma 1]) that λ is an eigenvalue of H_3 if and only if λ^2 is an eigenvalue of

$$\tilde{H} = \beta^2 h_{\mathbf{y}} (N_{\mathbf{y}\mathbf{y}} N_{\mathbf{y}\mathbf{y}} + N_{\mathbf{y}\mathbf{u}} N_{\mathbf{u}\mathbf{y}} + \Delta G_{\mathbf{y}}^\top \Delta G_{\mathbf{y}}) h_{\mathbf{y}}^\top.$$

Let $\lambda_{\max}(H_3)$ and $\lambda_{\min}(H_3)$ denote the largest and smallest eigenvalue of H_3 with a total number of $N = 2m + n + r$ eigenvalues. One can use the eigenvalue estimate of [84, Corollary 8.1.6], which can be derived from Weyl's inequality. As a result, one demands that

$$\begin{aligned} |\lambda_k(H_1 + H_2 + H_3) - \lambda_k(H_1 + H_2)| &\leq \max\{|\lambda_{\max}(H_3)|, |\lambda_{\min}(H_3)|\} = (\lambda_{\max}(\tilde{H}))^{\frac{1}{2}} \\ &= (\|\tilde{H}\|)^{\frac{1}{2}} \leq \beta \|h_{\mathbf{y}}\| (\|N_{\mathbf{y}\mathbf{y}}\|^2 + \|N_{\mathbf{y}\mathbf{u}}\|^2 + \|\Delta G_{\mathbf{y}}\|^2)^{\frac{1}{2}} \quad \forall k = 1, \dots, N \end{aligned}$$

to show that the Hessian of L^a at the stationary points has positive eigenvalues for β small enough, and thus is symmetric positive definite. \square

The theorem shows that under the given conditions the augmented Lagrangian is an exact penalty function.

Descent Direction The step s as defined in Equation (3.57) is a descent direction for the doubly augmented Lagrangian if M , as defined in Equation (3.56), is positive definite, i.e., has only positive eigenvalues, since then one obtains

$$\nabla L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^\top s = -s^\top M s < 0. \quad (3.62)$$

The positive definiteness of M can be proven by showing the positive definiteness of its symmetrical part

$$M_s = \frac{1}{2}(M + M^\top), \quad (3.63)$$

given by

$$M_s = \begin{pmatrix} \alpha \Delta \bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}} & -\frac{\alpha}{2} G_{\mathbf{u}} & \frac{\alpha}{2} h_{\mathbf{y}}^\top \check{B} \\ -I - \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}} & \beta \Delta \bar{G}_{\mathbf{y}} & -\frac{\beta}{2} N_{\mathbf{y}\mathbf{u}} & -\frac{\beta}{2} h_{\mathbf{y}}^\top \\ -\frac{\alpha}{2} G_{\mathbf{u}}^\top & -\frac{\beta}{2} N_{\mathbf{y}\mathbf{u}}^\top & B & \frac{\alpha}{2} h_{\mathbf{u}}^\top \check{B} \\ \frac{\alpha}{2} \check{B} h_{\mathbf{y}} & -\frac{\beta}{2} h_{\mathbf{y}} & \frac{\alpha}{2} \check{B} h_{\mathbf{u}} & \check{B} \end{pmatrix},$$

where $\Delta \bar{G}_{\mathbf{y}} = \frac{1}{2}(\Delta G_{\mathbf{y}} + \Delta G_{\mathbf{y}}^\top)$. One can define block matrices

$$\tilde{M}_s = \begin{pmatrix} \alpha \Delta \bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}} & -\frac{\alpha}{2} G_{\mathbf{u}} \\ -I - \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}} & \beta \Delta \bar{G}_{\mathbf{y}} & -\frac{\beta}{2} N_{\mathbf{y}\mathbf{u}} \\ -\frac{\alpha}{2} G_{\mathbf{u}}^\top & -\frac{\beta}{2} N_{\mathbf{y}\mathbf{u}}^\top & B \end{pmatrix}$$

and $C = \begin{pmatrix} \frac{\alpha}{2} \check{B} h_{\mathbf{y}} & -\frac{\beta}{2} h_{\mathbf{y}} & \frac{\alpha}{2} \check{B} h_{\mathbf{u}} \end{pmatrix}^\top$. Using the Schur complement, it can be argued that M_s is positive definite if \tilde{M}_s is positive definite, and $\check{B} \succ C^\top \tilde{M}_s^{-1} C$ holds. A similar strategy can be found in [278] and also for the original single-step one-shot approach in [102] in the proof of Theorem 15. Actually, one can identify \tilde{M}_s as the symmetrical part (3.32) of the Hessian used in Theorem 15 with a modified shifted Lagrangian. The same arguments as in [102] can be used to derive that \tilde{M}_s is symmetric positive definite for a large enough preconditioner B if the additional condition

$$\alpha \beta \Delta \bar{G}_{\mathbf{y}} \succ \left(I + \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}}\right) (\Delta \bar{G}_{\mathbf{y}})^{-1} \left(I + \frac{\beta}{2} N_{\mathbf{y}\mathbf{y}}\right) \quad (3.64)$$

with $\Delta \bar{G}_{\mathbf{y}} = \frac{1}{2}(I - G_{\mathbf{y}} + (I - G_{\mathbf{y}})^\top)$ is fulfilled (compare (3.31)). One can also use the same arguments as in the single-step one-shot method to formulate the stronger condition

$$\sqrt{\alpha \beta} (1 - \rho) > 1 + \frac{\beta}{2} \|N_{\mathbf{y}\mathbf{y}}\|, \quad (3.65)$$

which implies condition (3.58) and condition (3.64).

Since the additional condition on \check{B} is not practicable, alternative requirements are derived in [280] and will be presented in the following. The general idea is to consider

3. One-Shot Method for PDE-constrained Optimization

the decomposition

$$\begin{aligned}
M_s &= \begin{pmatrix} \alpha\Delta\bar{G}_{\mathbf{y}} & -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & -\frac{\alpha}{2}G_{\mathbf{u}} & 0 \\ -I - \frac{\beta}{2}N_{\mathbf{y}\mathbf{y}} & \beta\Delta\bar{G}_{\mathbf{y}} & -\frac{\beta}{2}N_{\mathbf{y}\mathbf{u}} & 0 \\ -\frac{\alpha}{2}G_{\mathbf{u}}^{\top} & -\frac{\beta}{2}N_{\mathbf{y}\mathbf{u}}^{\top} & B & 0 \\ 0 & 0 & 0 & \check{B} \end{pmatrix} \\
&\quad + \underbrace{\left(-\frac{\beta}{2}\right) \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{\mathbf{y}}^{\top} \\ 0 & 0 & 0 & 0 \\ 0 & h_{\mathbf{y}} & 0 & 0 \end{pmatrix}}_{\equiv E_1} + \underbrace{\frac{\alpha}{2} \begin{pmatrix} 0 & 0 & 0 & h_{\mathbf{y}}^{\top}\check{B} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{\mathbf{u}}^{\top}\check{B} \\ \check{B}h_{\mathbf{y}} & 0 & \check{B}h_{\mathbf{u}} & 0 \end{pmatrix}}_{\equiv E_2} \\
&= \begin{pmatrix} \tilde{M}_s & 0 \\ 0 & \check{B} \end{pmatrix} + E
\end{aligned}$$

with $E \equiv \frac{\beta}{2}(-E_1) + \frac{\alpha}{2}E_2$ and analyze the eigenvalues of these matrices. Let λ_{\min} denote the smallest eigenvalue of a given matrix, Theorem 8.1.5 of [84], which is also referred to as Weyl's inequality, yields

$$\lambda_{\min}(M_s) \geq \lambda_{\min} \begin{pmatrix} \tilde{M}_s & 0 \\ 0 & \check{B} \end{pmatrix} + \frac{\beta}{2}\lambda_{\min}(-E_1) + \frac{\alpha}{2}\lambda_{\min}(E_2). \quad (3.66)$$

The idea is now to achieve that the right-hand side of (3.66) is positive such that the smallest eigenvalue of M_s is positive. As a result, since M_s is symmetric, positive definiteness can be shown with the conditions presented in the following theorem.

Theorem 17. *The matrix M as defined in Equation (3.56) is positive definite if*

$$\min\{\tilde{\epsilon}, \lambda_2, \check{\lambda}\} > \left(\frac{\beta}{2}\|h_{\mathbf{y}}\| + \frac{\alpha}{2}\|\check{B}\| \|h_{\mathbf{y}}h_{\mathbf{y}}^{\top} + h_{\mathbf{u}}h_{\mathbf{u}}^{\top}\|^{1/2}\right) \quad (3.67)$$

with

$$\begin{aligned}
\tilde{\epsilon} &= 1 - \frac{1}{\sigma} \left(\frac{\sqrt{\alpha}}{2}\|G_{\tilde{\mathbf{u}}}\| + \frac{\sqrt{\beta}}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \right)^2, \\
\lambda_2 &= \frac{\alpha + \beta}{2}(1 - \rho) - \sqrt{\left(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|\right)^2 + \frac{(\alpha - \beta)^2}{4}(1 - \rho)^2}, \\
\check{\lambda} &= \min \lambda(\check{B}),
\end{aligned}$$

and

$$\left(\frac{\sqrt{\alpha}}{2}\|G_{\tilde{\mathbf{u}}}\| + \frac{\sqrt{\beta}}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \right)^2 < (1 - \rho) - \frac{(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|)^2}{\alpha\beta(1 - \rho)}, \quad (3.68)$$

as well as condition (3.65) are fulfilled, where $G_{\tilde{\mathbf{u}}} := G_{\mathbf{u}}B^{\frac{1}{2}}$, $N_{\mathbf{y}\tilde{\mathbf{u}}} := N_{\mathbf{y}\mathbf{u}}B^{-\frac{1}{2}}$ and

$$\sigma := 1 - \rho - \frac{(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|)^2}{\alpha\beta(1 - \rho)}. \quad (3.69)$$

Proof. First, a lower bound is derived for the smallest eigenvalue of

$$A = \begin{pmatrix} \tilde{M}_s & 0 \\ 0 & \tilde{B} \end{pmatrix}.$$

Let $\lambda(A)$ denote the set of all eigenvalues of A . Using [84, Cor. 8.1.9], one obtains that

$$\lambda(A) = \lambda(\tilde{M}_s) \cup \lambda(\tilde{B}).$$

The choice of \tilde{B} is open, such that the smallest eigenvalue $\lambda_{\min}(\tilde{B})$ can be adapted to satisfy a given condition. Inspired by the analysis in [103] one can examine the eigenvalues of \tilde{M}_s . We already identified the similarity of \tilde{M}_s and (3.32). Since the properties of the shifted Lagrangian are not explicitly used in the derivations of Theorem 15, the same derivations can be used for the modified shifted Lagrangian. As a result, \tilde{M}_s is positive definite if condition (3.68) is fulfilled. Inspired by the derivation, the eigenvalues of the 3×3 matrix

$$D_c = \begin{pmatrix} \alpha(1-\rho) & -1 - \frac{\beta}{2}\theta & -\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\| \\ -1 - \frac{\beta}{2}\theta & \beta(1-\rho) & -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \\ -\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\| & -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| & 1 \end{pmatrix}$$

can be analyzed. Due to [103, Prop. 3.1], i.e.,

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}^\top \tilde{M}_s \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \geq \begin{bmatrix} \|v_1\| \\ \|v_2\| \\ \|v_3\| \end{bmatrix}^\top D_c \begin{bmatrix} \|v_1\| \\ \|v_2\| \\ \|v_3\| \end{bmatrix},$$

the inequality

$$\lambda_{\min}(\tilde{M}_s) \geq \lambda_{\min}(D_c).$$

holds for the smallest eigenvalue of \tilde{M}_s . The eigenvalues of D_c can be obtained using the block decomposition $(D, d; d^\top, 1)$ with

$$D = \begin{pmatrix} \alpha(1-\rho) & -1 - \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\| \\ -1 - \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\| & \beta(1-\rho) \end{pmatrix}$$

and $d = (-\frac{\alpha}{2}\|G_{\tilde{\mathbf{u}}}\|, -\frac{\beta}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\|)^\top$. The Schur complement is $e = 1 - d^\top D^{-1}d$. It follows from [84, Theorem 8.1.9] that

$$\lambda(D_c) = \lambda(D) \cup \{e\}.$$

Simple calculations show that D is positive definite if the fundamental condition (3.65) is fulfilled, and the eigenvalues are given by

$$\lambda(D) = \left\{ \frac{\alpha + \beta}{2}(1-\rho) \pm \sqrt{\left(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|\right)^2 + \frac{(\alpha - \beta)^2}{4}(1-\rho)^2} \right\}.$$

3. One-Shot Method for PDE-constrained Optimization

Furthermore, one obtains with an estimate based on the condition (3.65) that

$$\begin{aligned} e &= 1 - \alpha\beta(1 - \rho)\det(D)^{-1} \left(\frac{\alpha}{4}\|G_{\tilde{\mathbf{u}}}\|^2 + \frac{1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|}{2(1 - \rho)}\|G_{\tilde{\mathbf{u}}}\|\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| + \frac{\beta}{4}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\|^2 \right) \\ &\geq 1 - \frac{1}{\sigma} \left(\frac{\sqrt{\alpha}}{2}\|G_{\tilde{\mathbf{u}}}\| + \frac{\sqrt{\beta}}{2}\|N_{\mathbf{y}\tilde{\mathbf{u}}}\| \right)^2 =: \tilde{e} > 0. \end{aligned}$$

The Schur complement is positive due to the condition (3.68). It holds for the smallest eigenvalue of \tilde{M}_s that

$$\lambda_{\min}(\tilde{M}_s) \geq \min\{\lambda(D), \tilde{e}\}.$$

This yields for the smallest eigenvalue of A that

$$\lambda_{\min}(A) \geq \min \left\{ \frac{\alpha + \beta}{2}(1 - \rho) - \sqrt{\left(1 + \frac{\beta}{2}\|N_{\mathbf{y}\mathbf{y}}\|\right)^2 + \frac{(\alpha - \beta)^2}{4}(1 - \rho)^2}, \tilde{e}, \lambda(\check{B}) \right\} =: \lambda_A.$$

Next, one may examine the eigenvalues of E_1 and E_2 . Exploiting again [280, Lemma 1], one obtains

$$\lambda_{\min}(-E_1) = - \left(\lambda_{\max}(h_{\mathbf{y}}h_{\mathbf{y}}^\top) \right)^{1/2} = -\|h_{\mathbf{y}}\|$$

and

$$\begin{aligned} \lambda_{\min}(E_2) &= - \left(\lambda_{\max} \left(\check{B}(h_{\mathbf{y}}h_{\mathbf{y}}^\top + h_{\mathbf{u}}h_{\mathbf{u}}^\top)\check{B} \right) \right)^{1/2} = -\|\check{B}(h_{\mathbf{y}}h_{\mathbf{y}}^\top + h_{\mathbf{u}}h_{\mathbf{u}}^\top)\check{B}\|^{1/2} \\ &\geq -\|\check{B}\|\|h_{\mathbf{y}}h_{\mathbf{y}}^\top + h_{\mathbf{u}}h_{\mathbf{u}}^\top\|^{1/2}. \end{aligned}$$

Hence, M_s and therefore also M is positive definite if

$$\lambda_A - \left(\frac{\beta}{2}\|h_{\mathbf{y}}\| + \frac{\alpha}{2}\|\check{B}\|\|h_{\mathbf{y}}h_{\mathbf{y}}^\top + h_{\mathbf{u}}h_{\mathbf{u}}^\top\|^{1/2} \right) > 0,$$

yielding the assertion. \square

As a result, it can be shown that under the given conditions the augmented one-shot iteration can be used together with an appropriate line search in a descent algorithm to find a stationary point of the augmented Lagrangian, and the algorithm will converge locally to a stationary point of the optimization problem (3.48).

3.2.4.2. Choice of Preconditioners

It can be shown that condition (3.68) is fulfilled for

$$B \succeq \frac{1}{\sigma}(\alpha G_{\mathbf{u}}^\top G_{\mathbf{u}} + \beta N_{\mathbf{y}\mathbf{u}}^\top N_{\mathbf{y}\mathbf{u}}). \quad (3.70)$$

The proof of these assertions can be lead analogously to the one in Hamdi and Griewank [103]. The preconditioner B is strongly related to the Hessian $\nabla_{\mathbf{u}\mathbf{u}}L^a$ with

respect to the design. This can be shown in a similar way as done in Section 3.2.3.2 by extending the derivations to the modified augmented Lagrangian. The main steps of the results of [278] are presented in the following.

One can consider the optimization problem

$$\min_{\Delta \mathbf{u}} L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u} + \Delta \mathbf{u}, \boldsymbol{\mu} + \Delta \boldsymbol{\mu}). \quad (3.71)$$

In the following, the use of L^a without any arguments implies the evaluation at $(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})$. When using a quadratic approximation of the double augmented Lagrangian one obtains the minimization problem

$$\min_{\Delta \mathbf{u}} s^\top \nabla L^a + \frac{1}{2} s^\top \nabla^2 L^a s,$$

which leads to

$$\min_{\Delta \mathbf{u}} E(\Delta \mathbf{u}) \quad (3.72)$$

with

$$\begin{aligned} E(\Delta \mathbf{u}) &= \Delta \mathbf{u}^\top (\nabla_{\mathbf{u}} L^a + \nabla_{\mathbf{u}\bar{\mathbf{y}}} L^a \Delta \bar{\mathbf{y}} + \nabla_{\mathbf{u}\mathbf{y}} L^a \Delta \mathbf{y} + \nabla_{\mathbf{u}\boldsymbol{\mu}} L^a \Delta \boldsymbol{\mu}) + \frac{1}{2} \Delta \mathbf{u}^\top \nabla_{\mathbf{u}\mathbf{u}} L^a \Delta \mathbf{u} \\ &\approx \Delta \mathbf{u}^\top \nabla_{\mathbf{u}} L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu} + \Delta \boldsymbol{\mu}) + \frac{1}{2} \Delta \mathbf{u}^\top \nabla_{\mathbf{u}\mathbf{u}} L^a \Delta \mathbf{u}. \end{aligned}$$

The solution to the optimization problem based on the last approximation is given by $\Delta \mathbf{u} = -\nabla_{\mathbf{u}\mathbf{u}}^{-1} L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) \nabla_{\mathbf{u}} L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu} + \Delta \boldsymbol{\mu})$ for a positive definite $\nabla_{\mathbf{u}\mathbf{u}} L^a$. Therefore, one identifies $B \approx \nabla_{\mathbf{u}\mathbf{u}} L^a$. The resulting choice for the preconditioner is based on

$$B = \alpha G_{\mathbf{u}}^\top G_{\mathbf{u}} + \alpha h_{\mathbf{u}}^\top h_{\mathbf{u}} + \beta N_{\mathbf{y}\mathbf{u}}^\top N_{\mathbf{y}\mathbf{u}} + N_{\mathbf{u}\mathbf{u}} \quad (3.73)$$

with $N_{\mathbf{u}\mathbf{u}} \succ 0$ which is equivalent to the Hessian $\nabla_{\mathbf{u}\mathbf{u}} L^a$ if state and adjoint feasibility as well as the equality constraints are satisfied. Again, to fulfill the condition (3.70), one uses $\tilde{B} := \frac{1}{\sigma} B$.

The choice of the constraint multiplier preconditioner \check{B} can be motivated similarly (see [280]). The optimization problem

$$\min_{\Delta \boldsymbol{\mu}} L^a(\mathbf{y} + \Delta \mathbf{y}, \bar{\mathbf{y}} + \Delta \bar{\mathbf{y}}, \mathbf{u} + \Delta \mathbf{u}, \boldsymbol{\mu} + \Delta \boldsymbol{\mu}) \quad (3.74)$$

can be formulated. Using a quadratic approximation of L^a one obtains

$$\min_{\Delta \boldsymbol{\mu}} s^\top \nabla L^a + \frac{1}{2} s^\top \nabla^2 L^a s,$$

which can be further approximated by

$$\min_{\Delta \boldsymbol{\mu}} E(\Delta \boldsymbol{\mu})$$

3. One-Shot Method for PDE-constrained Optimization

with the objective function

$$\begin{aligned} & \Delta\boldsymbol{\mu}^\top (\nabla_{\boldsymbol{\mu}} L^a + \nabla_{\boldsymbol{\mu}\mathbf{y}} L^a \Delta\mathbf{y} + \nabla_{\boldsymbol{\mu}\bar{\mathbf{y}}} L^a \Delta\bar{\mathbf{y}} + \nabla_{\boldsymbol{\mu}\mathbf{u}} L^a \Delta\mathbf{u}) + \frac{1}{2} \Delta\boldsymbol{\mu}^\top \nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a \Delta\boldsymbol{\mu} \\ & \approx \Delta\boldsymbol{\mu}^\top \nabla_{\boldsymbol{\mu}} L^a (\mathbf{y} + \Delta\mathbf{y}, \bar{\mathbf{y}} + \Delta\bar{\mathbf{y}}, \mathbf{u} + \Delta\mathbf{u}, \boldsymbol{\mu}) + \frac{1}{2} \Delta\boldsymbol{\mu}^\top \nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a \Delta\boldsymbol{\mu} =: E(\Delta\boldsymbol{\mu}) \end{aligned}$$

as a similar optimization problem. As one has $\nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a = \beta h_{\mathbf{y}} h_{\mathbf{y}}^\top$, one can only assume positive semidefiniteness of $\nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a$. Therefore, the above optimization problem is modified by adding a quadratic penalty term for $\boldsymbol{\mu}$ such that the problem is bounded.

The solution of the modified optimization problem

$$\begin{aligned} \min_{\Delta\boldsymbol{\mu}} & \Delta\boldsymbol{\mu}^\top \nabla_{\boldsymbol{\mu}} L^a (\mathbf{y} + \Delta\mathbf{y}, \bar{\mathbf{y}} + \Delta\bar{\mathbf{y}}, \mathbf{u} + \Delta\mathbf{u}, \boldsymbol{\mu}) \\ & + \frac{1}{2} \Delta\boldsymbol{\mu}^\top \nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a \Delta\boldsymbol{\mu} + \frac{\varepsilon}{2} \|\Delta\boldsymbol{\mu}\|^2 \end{aligned}$$

with $\varepsilon > 0$ being sufficiently small is defined by

$$\Delta\boldsymbol{\mu} = -(\nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} L^a + \varepsilon I)^{-1} \nabla_{\boldsymbol{\mu}} L^a (\mathbf{y} + \Delta\mathbf{y}, \bar{\mathbf{y}} + \Delta\bar{\mathbf{y}}, \mathbf{u} + \Delta\mathbf{u}, \boldsymbol{\mu}).$$

From $\Delta\boldsymbol{\mu} = -\check{B}^{-1}h$ one can identify the positive definite preconditioner

$$\check{B} = \beta h_{\mathbf{y}} h_{\mathbf{y}}^\top + \varepsilon I \quad (3.75)$$

if adjoint feasibility is fulfilled.

In practice, if the inversion of \check{B} is computationally expensive, a low-rank update to \check{B}^{-1} can be used as an approximation. One can modify the double augmented Lagrangian by adding a quadratic penalty term for $\boldsymbol{\mu}$, such that

$$\tilde{L}^a = L^a + \frac{\varepsilon}{2} \|\boldsymbol{\mu}\|^2 \quad (3.76)$$

and $\nabla_{\boldsymbol{\mu}\boldsymbol{\mu}} \tilde{L}^a = \beta h_{\mathbf{y}} h_{\mathbf{y}}^\top + \varepsilon I$. Then the approximation

$$\check{B} \Delta\boldsymbol{\mu} \approx \nabla_{\boldsymbol{\mu}} \tilde{L}^a (\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu} + \Delta\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}} \tilde{L}^a (\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) \quad (3.77)$$

can be used as a secant equation for a BFGS update of \check{B}^{-1} .

The above derivations of [280] neglect the condition (3.67) involving the norm and the eigenvalues of \check{B} . The present paragraph gives a closer look at one condition on \check{B} , namely

$$\lambda_{\min}(\check{B}) > \frac{\beta}{2} \|h_{\mathbf{y}}\| + \frac{\alpha}{2} \|\check{B}\| \|h_{\mathbf{y}} h_{\mathbf{y}}^\top + h_{\mathbf{u}} h_{\mathbf{u}}^\top\|^{1/2}. \quad (3.78)$$

Since \check{B} is symmetric positive definite, one has $\|\check{B}\| = \lambda_{\max}(\check{B})$. One may observe that the ratio of the largest and the smallest eigenvalue, i.e., the condition number, has to fulfill the property

$$\kappa(\check{B}) = \frac{\lambda_{\max}(\check{B})}{\lambda_{\min}(\check{B})} < \frac{2 - \beta(\lambda_{\min}(\check{B}))^{-1} \|h_{\mathbf{y}}\|}{\alpha \|h_{\mathbf{y}} h_{\mathbf{y}}^\top + h_{\mathbf{u}} h_{\mathbf{u}}^\top\|^{1/2}}. \quad (3.79)$$

Additionally, since the condition number has to be positive it is required that the inequality

$$\lambda_{\min}(\check{B}) > \frac{\beta}{2}\|h_{\mathbf{y}}\| \quad (3.80)$$

holds, giving an additional condition for the minimum eigenvalue of \check{B} . Both conditions on \check{B} can be used to derive a specific choice for \check{B} .

For the specific choice of the preconditioner given by (3.75) one has $\lambda_{\min}(\check{B}) = \varepsilon$ and $\lambda_{\max}(\check{B}) = \beta\|h_{\mathbf{y}}\|^2 + \varepsilon$. Thus, one may choose a specific ε to fulfill the above conditions. Since this will not necessarily result in a sufficiently small ε , it leads to a different interpretation of the preconditioner. Alternatively, one can derive new conditions on α and β depending on ε such that condition (3.79) and (3.80) are fulfilled.

The first option, for example, gives

$$\varepsilon > (1 - \frac{\alpha}{2}\|h_{\mathbf{y}}h_{\mathbf{y}}^{\top} + h_{\mathbf{u}}h_{\mathbf{u}}^{\top}\|^{1/2})^{-1}(\frac{\beta}{2}\|h_{\mathbf{y}}\| + \beta\|h_{\mathbf{y}}\|^2\frac{\alpha}{2}\|h_{\mathbf{y}}h_{\mathbf{y}}^{\top} + h_{\mathbf{u}}h_{\mathbf{u}}^{\top}\|^{1/2}),$$

for $(2 - \alpha\|h_{\mathbf{y}}h_{\mathbf{y}}^{\top} + h_{\mathbf{u}}h_{\mathbf{u}}^{\top}\|^{1/2}) > 0$, which means that ε has to be large enough and α has to be small enough. This gives further restrictions on an upper bound for α .

When looking at the remaining conditions on α and β , one can observe from Theorem 16 that, next to condition (3.68) for the design space preconditioner B and the fundamental condition (3.65), β has to be small enough and condition (3.61) is implied by

$$\alpha\beta\|h_{\mathbf{y}}\|^2 > 1. \quad (3.81)$$

Thus, if β has to be small, α has to be large enough. This is already observed for the fundamental condition. As a result, for the specific choice of the preconditioner, one obtains an upper and a lower bound on α . This bound might be too restrictive, depending on the choice of ε and β .

Note that it might be advantageous to introduce an additional parameter in the penalty term of the augmented Lagrangian associated with the equality constraints instead of sharing the parameter α with the state constraint. This helps to avoid the direct influence of the constraint preconditioner on the choice of α , and implicitly on the convergence of the state equation. Introducing the additional parameter would give more freedom to the choice of parameters. A thorough investigation is omitted in the following, but one may observe that this would result in the fact that conditions (3.61) and (3.79) are independent of α .

3.2.5. Bound Constraints in the One-Shot Approach

The feasible design space for most shape optimization problems is, among other constraints, constrained by lower and upper bounds. In this section, the aim is to transfer the general procedure to tackle bound constraints in iterative optimization methods to one-shot methods. In [150], the limited BFGS update with a treatment of bound constraints is used for building the preconditioner in the one-shot method. No line search

3. One-Shot Method for PDE-constrained Optimization

strategy is used, and no details of the implementation are given. In this section, it is intended to describe a strategy for integrating bound constraints. The integration of bound constraints is also of importance when using the one-shot approach in the hybrid strategy described in Chapter 2. Furthermore, the handling of bound constraints allows the extension of the augmented one-shot approach to problems with inequality constraints.

Therefore, before turning towards the introduction of bound constraints, one may also have a quick look at the extension of the one-shot strategy to inequality constraints. The question arises if the multiplier update can be adjusted in such a way that it does not penalize constraint fulfillment. This requires, for example, that, when using inequality constraints $g(\mathbf{y}, \mathbf{u}) \leq 0$, the corresponding multiplier is not negative. This means that $\eta_i^{k+1} = \min(\tilde{\eta}_i^{k+1}, 0)$ for $i = 1, \dots, s$ with $\tilde{\eta}_i^{k+1}$ obtained from the original multiplier update. This is a variant of augmented Lagrangian methods (see e.g. [206, p.523f]) and is also applied in the context of one-shot methods in [210]. However, the corresponding augmented Lagrangian in its general form is non-smooth. For a rigorous convergence analysis, this would require a suitable reformulation. In general, it is not clear how the modified multiplier update influences the choice of the design space preconditioner. The strategy is not pursued in the following. Instead, this work presents another option via bound constraints.

Inequality constraints can be transformed into equality constraints with the help of slack variables as introduced in Section 2.3.2.4 in the formulation (2.36). This leads to additional bound constraints for the slack variables, which can either be handled by augmenting the objective functions with a barrier function, as done in interior-point strategies, or by using *gradient projection methods*, which will be introduced in the following.

It is important to remark that the introduction of bound constraints in the one-shot strategy based on gradient projection does not interfere with the theoretical results and the construction of solution strategies based on the doubly augmented Lagrangian. The reason for that is that one can explicitly include the bound constraints in the original optimization problem (3.48) and in the optimization problem of the doubly augmented Lagrangian and treat the subproblem with fixed multipliers given by

$$\min_{\mathbf{u} \in \tilde{U}_{ad}} L^a(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \tilde{\boldsymbol{\mu}}). \quad (3.82)$$

Here, the original admissible design space U_{ad} is extended by the slack variables and, accordingly, all bound constraints are defined by \tilde{U}_{ad} . The multiplier $\tilde{\boldsymbol{\mu}} \in \mathbb{R}^{r+s}$ acts on the collection of original equality constraints and the equality constraints arising from the reformulation of the inequality constraints.

First approaches towards bound constraints for gradient-based search algorithms can be found in [18], where gradient projection methods were proposed. One may consider the simplified bound constrained minimization problem

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}), \quad (3.83)$$

with $\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid L_i \leq x_i \leq U_i \text{ for } i = 1, \dots, n\}$ and finite or infinite L_i and U_i . The projection onto the bound constrained design space Ω is then given as

$$P(\mathbf{x})_i = \begin{cases} L_i & x_i \leq L_i, \\ x_i & L_i < x_i < U_i, \\ U_i & x_i \geq U_i. \end{cases}$$

Furthermore, $\mathcal{A}(\mathbf{x})$ is the active set with all indices for which the bound is reached, i.e., $x_i = L_i$ or $x_i = U_i$, and its complement $\mathcal{I}(\mathbf{x})$ the inactive set, as introduced in Section 2.1.1. In the following, a set projection $P_{\mathcal{S}}$ onto a given set \mathcal{S} is defined as

$$P_{\mathcal{S}}(\mathbf{x})_i := \begin{cases} x_i & \text{if } i \in \mathcal{S}, \\ 0 & \text{else.} \end{cases}$$

As the gradient does not necessarily vanish at the bound constraint, an alternate necessary condition for bound constrained problems is given.

Theorem 18. *Let f be continuously differentiable in Ω . The first-order necessary condition for a local minimum at a point \mathbf{x}^* is*

$$\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \leq 0 \quad \forall \mathbf{x} \in \Omega. \quad (3.84)$$

Proof. A proof can be found in [142][p.88]. \square

Second-order necessary and sufficient conditions can also be found in [142].

For a Quasi-Newton or Newton scheme it is necessary to consider a reduced (approximate) Hessian R . The design is then updated by

$$\mathbf{x}^R(\gamma) := P(\mathbf{x} - \gamma R^{-1} \nabla f(\mathbf{x}))$$

and $\gamma > 0$ is chosen such that the condition for sufficient decrease

$$f(\mathbf{x}^R(\gamma)) - f(\mathbf{x}) \leq -c_1 \nabla f(\mathbf{x})^\top (\mathbf{x} - \mathbf{x}^R(\gamma)) \quad (3.85)$$

holds and ideally also the curvature condition, i.e.,

$$\nabla f(\mathbf{x}^R(\gamma))(\mathbf{x}^R(\gamma) - \mathbf{x}) \leq c_2 \|\nabla f(\mathbf{x})(\mathbf{x}^R(\gamma) - \mathbf{x})\|. \quad (3.86)$$

It is not feasible to use a standard reduced Hessian or standard Hessian approximations for constructing a descent direction. Instead following the propositions of [18], an underestimated inactive set \mathcal{I}^ε is used, which is a complement of the ε -active set \mathcal{A}^ε with

$$\mathcal{A}^\varepsilon := \{i \mid U_i - x_i \leq \varepsilon \text{ or } x_i - L_i \leq \varepsilon\}$$

Typically

$$\varepsilon < \min(\tilde{\varepsilon}, \|\mathbf{x} - \mathbf{x}^R(1)\|) \quad (3.87)$$

3. One-Shot Method for PDE-constrained Optimization

is chosen. The variable $\tilde{\varepsilon} > 0$ is a small constant. For finite bound constraints, it is proposed in [142] to choose $0 \leq \tilde{\varepsilon} < \min_i [(U_i - L_i)/2]$. Accordingly, the reduced (approximate) Hessian R of the (approximate) Hessian H is

$$R = P_{\mathcal{A}^\varepsilon} + P_{\mathcal{I}^\varepsilon} H P_{\mathcal{I}^\varepsilon} = \begin{cases} \delta_{ij} & \text{if } i \text{ or } j \in \mathcal{A}^\varepsilon, \\ H_{ij} & \text{otherwise.} \end{cases} \quad (3.88)$$

The update of the generalized inverse of R can be formulated as

$$R_{k+1}^{-1} = \left(I - \frac{\mathbf{s}_P \mathbf{r}_P^\top}{\mathbf{r}_P^\top \mathbf{s}_P} \right) P_{\mathcal{I}^\varepsilon} R_k^{-1} P_{\mathcal{I}^\varepsilon} \left(I - \frac{\mathbf{r}_P \mathbf{s}_P^\top}{\mathbf{r}_P^\top \mathbf{s}_P} \right) + \frac{\mathbf{s}_P \mathbf{s}_P^\top}{\mathbf{r}_P^\top \mathbf{s}_P} \quad (3.89)$$

with $\mathbf{s}_P = P_{\mathcal{I}^\varepsilon}(\mathbf{s})$ and $\mathbf{r}_P = P_{\mathcal{I}^\varepsilon}(\mathbf{r})$.

The updated algorithm to build the inverse of the approximated Hessian with the help of a BFGS update is given in Algorithm 7 (see e.g. [142]). The initial value for the approximated inverse is given by I but can be chosen differently.

Algorithm 7: BFGS method with bound constraints.

Input: x_0, μ_1, μ_2

Function:

Find ε using Equation (3.87) based on $\mathbf{x}_0 - P(\mathbf{x}_0 - \nabla f(\mathbf{x}_0))$

$R_0^{-1} = P_{\mathcal{I}^\varepsilon(\mathbf{x}_0)} I P_{\mathcal{I}^\varepsilon(\mathbf{x}_0)}$

while $\|\mathbf{x}_k - P(\mathbf{x}_k - \nabla f(\mathbf{x}_k))\| \geq \mu_1 + \mu_2 \|\mathbf{x}_0 - P(\mathbf{x}_0 - \nabla f(\mathbf{x}_0))\|$ **do**

$\mathbf{d}_k = -P_{\mathcal{A}^\varepsilon(\mathbf{x}_k)} \nabla f(\mathbf{x}_k) - P_{\mathcal{I}^\varepsilon(\mathbf{x}_k)} R_k^{-1} P_{\mathcal{I}^\varepsilon(\mathbf{x}_k)} \nabla f(\mathbf{x}_k)$

 Find γ such that Equation (3.85) holds.

 Find ε using Equation (3.87) based on $\mathbf{x}_k - P(\mathbf{x}_k - \nabla f(\mathbf{x}_k))$

$\mathbf{x}_{k+1} = P(\mathbf{x}_k + \gamma \mathbf{d}_k)$

$\mathbf{r}_{k+1} = P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})} (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))$

$\mathbf{s}_{k+1} = P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})} (\mathbf{x}_{k+1} - \mathbf{x}_k)$

if $\mathbf{r}_{k+1}^\top \mathbf{s}_{k+1} > 0$ **then**

 Find R_{k+1}^{-1} with the help of the update rule (3.89) using $\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})$, \mathbf{r}_{k+1} , and

\mathbf{s}_{k+1}

else

$R_{k+1}^{-1} = P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})} I P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})}$

end if

$k = k + 1$

end while

The line search strategy in Algorithm 7 does not guarantee for fulfilling the curvature condition. Instead, if this condition is not satisfied, one resets R_k^{-1} to the identity matrix. Note that the approximation R_k^{-1} is built using old iterates that were based on different estimates of the inactive set. Therefore, it has to be projected before applying the update for R_{k+1}^{-1} . It can be advantageous in terms of consistency and memory requirements to rebuild the approximation with the help of a limited-memory BFGS method for bound constraints, which is presented in Section A.2.5 of the appendix.

When applied in the one-shot approach, the idea is to use Algorithm 7 or the limited-memory version for the BFGS update based on the gradient of the doubly augmented Lagrangian.

The design variable \mathbf{u} is then updated with the formula

$$\mathbf{u}_{k+1} = P(\mathbf{u}_k + \gamma \mathbf{d}_k), \quad (3.90)$$

with

$$\mathbf{d}_k := -P_{\mathcal{A}^\varepsilon(\mathbf{u}_k)} N_{\mathbf{u}} - P_{\mathcal{I}^\varepsilon(\mathbf{u}_k)} B_k^{-1} P_{\mathcal{I}^\varepsilon(\mathbf{u}_k)} N_{\mathbf{u}},$$

where $N_{\mathbf{u}}$ is evaluated at $(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top$ and

$$\varepsilon = \min(\tilde{\varepsilon}, \|\mathbf{u}_k - P(\mathbf{u}_k - N_{\mathbf{u}})\|).$$

In the following, $L_k^a(\tilde{\mathbf{u}})$ denotes an evaluation at $(\mathbf{y}_k, \bar{\mathbf{y}}_k, \tilde{\mathbf{u}}, \boldsymbol{\mu}_k)^\top$. The line search for γ is based on the condition

$$L_k^a(\mathbf{u}_{k+1}) \leq L_k^a(\mathbf{u}_k) - c_1 \nabla_{\mathbf{u}} L_k^a(\mathbf{u}_k)^\top (\mathbf{u}_{k+1} - \mathbf{u}_k). \quad (3.91)$$

The inverse of the preconditioner B of the one-shot approach is constructed from

$$\mathbf{y}_k = P_{\mathcal{I}^\varepsilon(\mathbf{u}_{k+1})} (\nabla L_{\mathbf{u}}^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k + \Delta \mathbf{u}_k, \boldsymbol{\mu}_k) - \nabla L_{\mathbf{u}}^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k))$$

and $\mathbf{s}_k = P_{\mathcal{I}^\varepsilon(\mathbf{u}_{k+1})}(\mathbf{u}_{k+1} - \mathbf{u}_k)$.

3.2.6. Algorithmic Differentiation

As already explained in Section 3.1.3.2, it is advantageous to use the reverse mode of *algorithmic differentiation* (Griewank and Walther, [93]) for obtaining the needed derivatives in gradient-based optimization methods using the discrete adjoint method. Algorithmic differentiation is sometimes also referred to as *automatic differentiation*. A feature of algorithmic differentiation is that the function gradients can be calculated accurately to working precision, which is advantageous for the convergence of gradient-based optimization methods. Accurate derivatives are, for example, needed in projection-based algorithms for constrained optimization problems or for enabling a robust Hessian approximation in quasi-Newton methods.

The general idea is to differentiate a function or algorithm implemented as a computer program. The implementation requires that the algorithm can be broken down into a concatenation of elementary operators (e.g. +, *, sin). A differentiated version of the algorithm can then be established by using the chain rule of differentiation.

To present the basic ideas of algorithmic differentiation, one can consider a function

$$F : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m, \mathbf{y} = F(\mathbf{x}).$$

Suppose F describes the function that evaluates a concatenation of elementary functions Φ_i with $i = 1, \dots, p$ that are differentiable at the point $\mathbf{x} \in \mathbb{R}^n$, such that

$$F(\mathbf{x}) = \Phi_p \circ \Phi_{p-1} \circ \dots \circ \Phi_1(\mathbf{x}). \quad (3.92)$$

3. One-Shot Method for PDE-constrained Optimization

Then the Jacobian of F in \mathbf{x} , which is denoted as $F'(\mathbf{x})$, can be obtained by using the chain rule of differentiation

$$F'(\mathbf{x}) = (\Phi'_p \circ \Phi_{p-1} \circ \dots \circ \Phi_1(\mathbf{x})) \cdot (\Phi'_{p-1} \circ \Phi_{p-2} \circ \dots \circ \Phi_1(\mathbf{x})) \cdot \dots \cdot \Phi'_1(\mathbf{x}), \quad (3.93)$$

or, when defining intermediate values $\mathbf{x}^0 := \mathbf{x}$ and $\mathbf{x}^i = \Phi_i(\mathbf{x}^{i-1})$ for $i = 1, \dots, p$, by

$$F'(\mathbf{x}^0) = \Phi'_p(\mathbf{x}^{p-1}) \cdot \dots \cdot \Phi'_1(\mathbf{x}^0). \quad (3.94)$$

There exist two different approaches (or modes) for AD, namely the *tangent-linear* and the *reverse* approach.¹¹

This section gives a brief introduction to algorithmic differentiation, focusing on the application in the one-shot method. In the following, both approaches, the tangent-linear and the reverse mode, are presented alongside with additional information on implementation strategies and second-order derivative computations (see Section 3.2.6.3). To formalize the robust discrete adjoint strategy based on algorithmic differentiation, Section 3.2.6.4 presents different methodologies for the application of AD to the fixed-point iterations. Finally, the calculation of the needed derivatives of the one-shot approach is discussed.

3.2.6.1. Tangent-linear Mode

In the tangent-linear mode of AD the product in Equation (3.93) is evaluated from right to left. When applying the tangent-linear mode, instead of directly evaluating the Jacobian one ends up with the evaluation of

$$\dot{\mathbf{y}} = F'(\mathbf{x})\dot{\mathbf{x}}, \quad (3.95)$$

where $\dot{\mathbf{y}} = \partial \mathbf{y}(t) / \partial t$ and $\dot{\mathbf{x}} = \partial \mathbf{x}(t) / \partial t$ are directional derivatives, which can be interpreted as tangents of $\mathbf{x}(t)$ and $\mathbf{y}(t)$. This evaluation results from applying the chain rule when considering the change of \mathbf{y} under an input t , as

$$\frac{\partial \mathbf{y}(t)}{\partial t} = \frac{\partial}{\partial t} F(\mathbf{x}(t)) = F'(\mathbf{x}(t)) \frac{\partial \mathbf{x}(t)}{\partial t}.$$

The Jacobian can be obtained from $\dot{\mathbf{y}}$ by evaluating Equation (3.95) with $\dot{\mathbf{x}} = \mathbf{e}_i$ for $i = 1, \dots, n$ resulting in n evaluations of the tangent-linear mode.

A tangent-linear version of an algorithm can, for example, be generated by augmenting the original code. Each statement in the code is differentiated according to the rule of Equation (3.95). To show the general procedure one can make use of Single Assignment Code (SAC), where each elementary function ϕ_j depending on some values v_i ($i \prec j$, where " \prec " denotes the dependence relation) is assigned to a variable v_j . The n independent input variables are assigned to v_0, \dots, v_{n-1} , and the applications of elementary functions are assigned to q intermediate variables v_n, \dots, v_{n+q-1} and m variables

¹¹The tangent-linear mode is also referred to as the *forward* mode of AD. The notation pendant to the tangent-linear mode is to refer to the reverse mode as the *adjoint* mode. It reflects the idea of the general adjoint approach to find sensitivities but is not to be confused with it.

$v_{n+q}, \dots, v_{n+q+m-1}$, which are then assigned to the dependent output variables.¹² The original SAC is:

$$\begin{aligned} v_j &= x_j & j &= 0, \dots, n-1 & \text{(independent input variables)} \\ v_j &= \phi_j(v_i)_{i \prec j} & j &= n, \dots, n+q+m-1 & \text{(q intermediate variables)} \\ y_j &= v_{n+q+j} & j &= 0, \dots, m-1 & \text{(dependent output variables),} \end{aligned}$$

and the augmented tangent-linear version is:

$$\begin{aligned} \dot{v}_j &= \dot{x}_j & j &= 0, \dots, n-1 \\ v_j &= x_j & j &= 0, \dots, n-1 \\ \dot{v}_j &= \sum_{i \prec j} \frac{\partial \phi_j}{\partial v_i} \dot{v}_i & j &= n, \dots, n+q+m-1 \\ v_j &= \phi_j(v_i)_{i \prec j} & j &= n, \dots, n+q+m-1 \\ \dot{y}_j &= \dot{v}_{n+q+j} & j &= 0, \dots, m-1 \\ y_j &= v_{n+q+j} & j &= 0, \dots, m-1. \end{aligned}$$

3.2.6.2. Reverse Mode

The idea of the adjoint mode of AD is to evaluate the product in Equation (3.93) in reverse order from left to right. Thus, it produces a product of the transposed Jacobian and a vector that is given by

$$\bar{\mathbf{x}}^\top = \bar{\mathbf{y}}^\top F'(\mathbf{x}) \quad \text{or} \quad \bar{\mathbf{x}} = F'(\mathbf{x})^\top \bar{\mathbf{y}}, \quad (3.96)$$

where $\bar{\mathbf{x}}^\top = \partial a / \partial \mathbf{x}$ and $\bar{\mathbf{y}}^\top = \partial a / \partial \mathbf{y}$ can be interpreted geometrically as normals and $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are referred to as the vector of adjoint variables or adjoint vector. The evaluation can be derived by looking at the change of an output a under the input \mathbf{x} , as

$$\frac{\partial a}{\partial \mathbf{x}} = \frac{\partial a}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial a}{\partial \mathbf{y}} F'(\mathbf{x}).$$

The Jacobian can be obtained from $\bar{\mathbf{x}}$ by evaluating Equation (3.96) with $\bar{\mathbf{y}} = \mathbf{e}_i$ for $i = 1, \dots, m$ resulting in m evaluations of the adjoint mode. As a result, this approach is more efficient for $n \gg m$.

For reasons of completeness, the procedure for the reverse chain rule evaluation

$$\bar{\mathbf{x}}^\top = \bar{\mathbf{y}}^\top \Phi'_p(\mathbf{x}^{p-1}) \cdot \dots \cdot \Phi'_1(\mathbf{x}^0) \quad (3.97)$$

is to evaluate intermediate values

$$(\bar{\mathbf{x}}^{p-1})^\top = \bar{\mathbf{y}}^\top \Phi'_p(\mathbf{x}^{p-1}), \dots, \bar{\mathbf{x}}^\top = (\bar{\mathbf{x}}^0)^\top = (\bar{\mathbf{x}}^1)^\top \Phi'_1(\mathbf{x}^0).$$

¹²Similarly to Equation (3.92), one may express the SAC as a concatenation of functions, where Φ_1 is identified as the mapping of x to the variable space, Φ_p is the projection of the variables to y , and $\Phi_2, \dots, \Phi_{p-1}$ are mappings in the variable space, that can be related to the elementary operations $\phi_n, \dots, \phi_{n+q+m-1}$, i.e., $q = p - m - 2$.

3. One-Shot Method for PDE-constrained Optimization

This means that the calculated values (e.g. \mathbf{x}^{p-1}) have to be present at the beginning of the evaluation. The general strategy is to execute the original code in a first run and store all needed intermediate values. Afterwards, the augmented part is executed in a second (or reverse) run. In SAC this reads:

$$\begin{aligned}
 v_j &= x_j & j &= 0, \dots, n-1 \\
 v_j &= \phi_j(v_i)_{i \prec j} & j &= n, \dots, n+q+m-1 \\
 y_j &= v_{n+q+j} & j &= 0, \dots, m-1 \\
 \bar{v}_{n+q+j} &= \bar{y}_j & j &= m-1, \dots, 0 \\
 \bar{v}_i &= \sum_{j: i \prec j} \bar{v}_j \frac{\partial \phi_j}{\partial v_i} & i &= n+q-1, \dots, 0 \\
 \bar{x}_j &= \bar{v}_j & j &= n-1, \dots, 0.
 \end{aligned}$$

For programming it is more convenient to relate each assignment to an adjoint assignment, which is why the fifth line of the above SAC can be rewritten as the incremental statement:

$$(\bar{v}_i)_{i \prec j} + = \bar{v}_j \frac{\partial \phi_j}{\partial v_i} \quad j = n+q+m-1, \dots, n.$$

As a result, using the reverse mode of algorithmic differentiation requires only one run of the augmented code for calculating the state and the adjoint variables. The time that is needed for running the reverse mode of AD is only a small multiple of the time needed for running the original code. The utilization of intermediate values is a drawback of the reverse mode since intermediate values might be overwritten during the original run. As a result, these have to be either stored during the execution of the original statements or recomputed when they are needed. Storing values or statements is sometimes also referred to as *recording* or *taping* and will increase the overall memory consumption. On the other hand, when recomputing values, the computational costs are increased. *Checkpointing* strategies are often used as a compromise between time and memory. The code is separated into blocks, for which only the input values that are needed to execute the block from a given point are stored. This storage point is also referred to as a *snapshot*. When the reverse run reaches such a block, the intermediate values for the individual block are recalculated using the snapshot and stored to be used in the reverse run.

There exist two different strategies for the implementation of algorithmic differentiation. These are the application of *source code transformation* or *operator overloading*. In source code transformation, the original source code is augmented by the corresponding statements for the tangent-linear or the reverse mode as done before for the SAC. There exist tools like Tapenade [106] that enable the code augmentation in an automated fashion. Tapenade is applicable for code in Fortran or C. The main drawback of such an approach is that code augmentation has to be repeated when the original code is changed, causing a maintenance overhead. Furthermore, code optimization concerning runtime and memory is rather difficult and often has to be done manually. Operator

overloading does not require as much code transformation as source transformation, but it can only be implemented in languages that support operator overloading, e.g., C++. In operator overloading, the original data type and all respective operators are augmented, and the original data type is replaced with the augmented data type. In tangent-linear mode, the augmented data type is simply used in the overloaded operators to compute the tangent-linear derivative according to Equation (3.95) in addition to the primal computation. When using the reverse mode of AD in this context, a graph (*tape*) is set up at run time. It is reversed or *interpreted* in the adjoint run to accumulate derivatives. AD via operator overloading can be enabled with tools like ADOL-C [279] or CoDiPack [231]. The augmented graph generally causes a high memory consumption. The advantage of operator overloading is that it is easily maintainable since, due to the use of the augmented data type, all code changes are automatically integrated into the derivative computation. Furthermore, since only the data type is changed and some access functions are added in the augmented version, the code remains easily accessible for users.

In this work, it is made use of algorithmic differentiation via operator overloading with the help of CoDiPack (see Section 3.3.1.1).

3.2.6.3. Second-Order Derivatives

Second-order derivatives can be obtained with algorithmic differentiation by any possible combination of modes. This gives four possible options, namely the tangent-linear mode over the tangent-linear mode, the tangent-linear mode over the reverse mode, the reverse mode over the tangent-linear mode, or the reverse mode over the reverse mode. The method of choice is often to use tangent-linear over reverse mode, or in short *tangent over reverse*. This means that the tangent-linear mode is applied to the result of applying the reverse mode to a function. It is preferred as, for example, the reverse over tangent-linear mode would involve the additional taping of all tangent variables. For tangent over reverse mode, on the other hand, the taping itself and the tape evaluation are done in tangent-linear mode [202].

One considers again $y = F(\mathbf{x})$ and denotes the Hessian of F as F'' . Then the tangent over reverse mode results in the evaluation of

$$\dot{\mathbf{y}} = F'(\mathbf{x})\dot{\mathbf{x}} \quad (3.98a)$$

$$\mathbf{y} = F(\mathbf{x}) \quad (3.98b)$$

$$\dot{\mathbf{x}}^\top = \dot{\mathbf{y}}^\top F'(\mathbf{x}) + \bar{\mathbf{y}}^\top F''(\mathbf{x})\dot{\mathbf{x}} \quad (3.98c)$$

$$\bar{\mathbf{x}}^\top = \bar{\mathbf{y}}^\top F'(\mathbf{x}), \quad (3.98d)$$

where the second step, which is the application of the tangent-linear mode, corresponds to Equations (3.98a) and (3.98c). The product of the Hessian with a vector \mathbf{v} can be obtained from $\dot{\mathbf{x}}$ by setting $\dot{\mathbf{y}} = 0$, $\bar{\mathbf{y}} = 1$ and $\dot{\mathbf{x}} = \mathbf{v}$.

3. One-Shot Method for PDE-constrained Optimization

3.2.6.4. AD-Based Discrete Adjoint Approach

The present section intends to show the application of algorithmic differentiation in the context of the optimization problem (3.17) based on the fixed-point iteration.

Algorithmic differentiation can be applied to an implementation of an iteration of the fixed-point equation and an evaluation of the objective function, i.e.,

$$\begin{aligned}\tilde{\mathbf{y}} &= G(\mathbf{y}, \mathbf{u}), \\ z &= f(\mathbf{y}, \mathbf{u}).\end{aligned}\tag{3.99}$$

The fixed-point iteration and the objective function are evaluated at \mathbf{y} . The result of an evaluation of G is denoted as $\tilde{\mathbf{y}}$ and the result of an evaluation of f as z .

Using the reverse mode of algorithmic differentiation one can write the resulting evaluation as

$$\begin{pmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} G_{\mathbf{y}}(\mathbf{y}, \mathbf{u}) & G_{\mathbf{u}}(\mathbf{y}, \mathbf{u}) \\ f_{\mathbf{y}}(\mathbf{y}, \mathbf{u}) & f_{\mathbf{u}}(\mathbf{y}, \mathbf{u}) \end{pmatrix}^{\top} \begin{pmatrix} \bar{z} \\ \bar{\mathbf{y}} \end{pmatrix}.\tag{3.100}$$

When setting $\bar{z} = 1$, one obtains the derivative of the shifted Lagrangian $N(\mathbf{y}, \tilde{\mathbf{y}}, \mathbf{u})$ with respect to \mathbf{y} and \mathbf{u} as

$$\begin{pmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} \nabla_{\mathbf{y}} f(\mathbf{y}, \mathbf{u}) + G_{\mathbf{y}}(\mathbf{y}, \mathbf{u})^{\top} \bar{\mathbf{y}} \\ \nabla_{\mathbf{u}} f(\mathbf{y}, \mathbf{u}) + G_{\mathbf{u}}(\mathbf{y}, \mathbf{u})^{\top} \bar{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} N_{\mathbf{y}}(\mathbf{y}, \tilde{\mathbf{y}}, \mathbf{u})^{\top} \\ N_{\mathbf{u}}(\mathbf{y}, \tilde{\mathbf{y}}, \mathbf{u})^{\top} \end{pmatrix}.\tag{3.101}$$

This shows the connection of the reverse mode of algorithmic differentiation to the necessary conditions for optimality. For the fixed-point $(\mathbf{y}^*, \tilde{\mathbf{y}}^*)^{\top}$ with $\mathbf{y}^* = G(\mathbf{y}^*, \mathbf{u})$, one obtains the adjoint equation, which has to be fulfilled at the stationary point, from $\bar{\mathbf{y}} = \tilde{\mathbf{y}}^*$.

The above considerations require that the fixed-point equation can be differentiated at a converged state and that the corresponding fixed-point iterator is contractive. If this is possible, the fixed-point equation for the adjoint converges to a fixed-point with the same contraction rate as the state iteration. The resulting adjoint solution can be used to provide a derivative that is consistent with the derivative of the objective function up to machine precision. However, often it cannot be assumed that the state equation is fully converged. As explained before, there exist different strategies to set up and iterate the adjoint equation.

Suppose that starting values \mathbf{y}_0 and $\tilde{\mathbf{y}}_0$ are given. One strategy is to differentiate the whole computational graph for solving the state equation with all intermediate results, and to iterate the adjoint equation by using the corresponding statements and intermediate results in reverse order, i.e.,

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}) \text{ for } k = 0, \dots, S\tag{3.102a}$$

$$\tilde{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_{S-k}, \tilde{\mathbf{y}}_k, \mathbf{u})^{\top} \text{ for } k = 0, \dots, S.\tag{3.102b}$$

The derivative with respect to the design \mathbf{u} is then build by differentiating the evaluation of the objective function f . The corresponding adjoint variable is given by

$$\bar{\mathbf{u}} = \nabla_{\mathbf{u}} f(\mathbf{y}_S, \mathbf{u}) + \sum_{i=0}^S G_{\mathbf{u}}(\mathbf{y}_i, \mathbf{u})^{\top} \bar{\mathbf{y}}_{S-i}.$$

One may refer to this as the *black-box adjoint* approach. The black-box adjoint approach is robust in the sense that it will always provide derivatives that are consistent with the solution strategy for the state equation. Even if the state equation is not fully converged, derivative information will be provided. However, it is unclear to what extent such information is useful. Additionally, the reversal of the state iteration may require a significant amount of memory. Thus, the approach only becomes feasible when using, e.g., checkpointing techniques.

If the state equation is converged to a solution \mathbf{y}_S with sufficient accuracy, i.e., $\mathbf{y}_S \approx \mathbf{y}^*$, in the *reverse accumulation* (see [38]) strategy only the computational graph for evaluating $G(\mathbf{y}_S, \mathbf{u})$ is stored. Note that the above considerations based on the stationary point \mathbf{y}^* can be seen as the ideal case. The iteration procedure for the adjoint variable reads

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_S, \bar{\mathbf{y}}_k, \mathbf{u})^\top \text{ for } k = 0, \dots, \tilde{S}. \quad (3.103)$$

Since contractivity of the iterator G is expected near the solution of the state equation, the adjoint iteration converges to a fixed-point. The number of iteration steps can be chosen independently of S . The adjoint of the design variable is given by

$$\bar{\mathbf{u}} = N_{\mathbf{u}}(\mathbf{y}_S, \bar{\mathbf{y}}_{\tilde{S}}, \mathbf{u})^\top.$$

It approximates the derivative of the objective function with the same order of accuracy as the objective function itself.

In the piggy-back approach and also in the one-shot approach, the state and the adjoint equation are iterated simultaneously resulting in the procedure

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}) \quad (3.104a)$$

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u})^\top \quad (3.104b)$$

for $k = 0, \dots, S$. Properties of the piggy-back approach were already mentioned in Section 3.2.2. If needed, e.g., for the one-shot approach, the adjoint variable of \mathbf{u} can be obtained in each step as

$$\bar{\mathbf{u}}_{k+1} = N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k)^\top. \quad (3.105)$$

When using the tangent over reverse mode, the second-order derivative can be derived by applying the tangent-linear mode to Equations (3.99) and (3.100), resulting in

$$\begin{pmatrix} \dot{\bar{\mathbf{y}}} \\ \dot{\bar{\mathbf{z}}} \\ \dot{\bar{\mathbf{y}}} \\ \dot{\bar{\mathbf{u}}} \end{pmatrix} = \begin{pmatrix} G_{\mathbf{y}} & G_{\mathbf{u}} & 0 & 0 \\ f_{\mathbf{y}} & f_{\mathbf{u}} & 0 & 0 \\ \bar{\mathbf{y}}^\top G_{\mathbf{y}\mathbf{y}} + \bar{\mathbf{z}}^\top f_{\mathbf{y}\mathbf{y}} & \bar{\mathbf{y}}^\top G_{\mathbf{y}\mathbf{u}} + \bar{\mathbf{z}}^\top f_{\mathbf{y}\mathbf{u}} & G_{\mathbf{y}}^\top & f_{\mathbf{y}}^\top \\ \bar{\mathbf{y}}^\top G_{\mathbf{u}\mathbf{y}} + \bar{\mathbf{z}}^\top f_{\mathbf{u}\mathbf{y}} & \bar{\mathbf{y}}^\top G_{\mathbf{u}\mathbf{u}} + \bar{\mathbf{z}}^\top f_{\mathbf{u}\mathbf{u}} & G_{\mathbf{u}}^\top & f_{\mathbf{u}}^\top \end{pmatrix} \begin{pmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{u}} \\ \dot{\bar{\mathbf{y}}} \\ \dot{\bar{\mathbf{z}}} \end{pmatrix}. \quad (3.106)$$

Note that, when used in a piggy-back fashion, the convergence of the second-order adjoint variables lags behind the convergence of the state variables by a factor of k^2 (see [92]).

3. One-Shot Method for PDE-constrained Optimization

3.2.6.5. AD in the One-Shot Approach

This section presents the use of AD to obtain the derivatives for the one-shot approach.

The piggy-back iteration given in (3.104) is the basis for the single-step one-shot approach. One can extend the framework to integrate additional equality constraints h by introducing the evaluation of the equality constraints denoted as

$$\mathbf{w} = h(\mathbf{y}, \mathbf{u})$$

in the system (3.99).

If the objective function and the constraint functions are evaluated after one fixed-point iteration, the adjoint output \bar{z} of the objective function is set to $\bar{z} = 1$, and the adjoint output of the constraint functions is set to $\bar{\mathbf{w}} = \boldsymbol{\mu}_k$, one obtains the modified piggy back iteration

$$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}) \quad (3.107a)$$

$$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}, \boldsymbol{\mu}_k)^\top. \quad (3.107b)$$

The derivatives $N_{\mathbf{u}}$, which are needed for the design variable update in each iteration, are then given by

$$\bar{\mathbf{u}}_{k+1} = N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top. \quad (3.108)$$

For constructing the preconditioner B , one needs the gradient of the doubly augmented Lagrangian with respect to the design variables. For now, it is assumed that one wants to compute the doubly augmented Lagrangian evaluated with all variables at the iterate k . It is given by

$$\nabla_{\mathbf{u}} L^a = \alpha G_{\mathbf{u}}^\top (G - \mathbf{y}_k) + \beta N_{\mathbf{y}\mathbf{u}}^\top (N_{\mathbf{y}}^\top - \bar{\mathbf{y}}_k) + N_{\mathbf{u}}^\top + \alpha h_{\mathbf{u}}^\top h, \quad (3.109)$$

where all functions without arguments are evaluated at the iterate k . When taping the state iteration, the calculation of the objective function and the calculation of the equality constraints, the tape evaluation gives

$$\bar{\mathbf{u}}_{k+1} = \nabla_{\mathbf{u}} f(\mathbf{y}_k, \mathbf{u}_k) \bar{z} + G_{\mathbf{u}}(\mathbf{y}_k, \mathbf{u}_k)^\top \bar{\bar{\mathbf{y}}} + h_{\mathbf{u}}(\mathbf{y}_k, \mathbf{u}_k)^\top \bar{\mathbf{w}}.$$

The term $N_{\mathbf{u}}^\top$ is obtained from a tape evaluation with $\bar{z} = 1$, $\bar{\mathbf{w}} = \boldsymbol{\mu}_k$ and $\bar{\bar{\mathbf{y}}} = \bar{\mathbf{y}}_k$ (see Equation (3.108)). To obtain $G_{\mathbf{u}}^\top (G - \mathbf{y}_k) + h_{\mathbf{u}}^\top h$, one may evaluate the same tape a second time. One has to set $\bar{\bar{\mathbf{y}}}$ to $\Delta \mathbf{y} := G(\mathbf{y}_k, \mathbf{u}_k) - \mathbf{y}_k$ and the corresponding adjoint of h , i.e., $\bar{\mathbf{w}}$, to $h(\mathbf{y}_k, \mathbf{u}_k)$, as well as $\bar{z} = 0$.

The second-order derivative term $\beta N_{\mathbf{y}\mathbf{u}}^\top (N_{\mathbf{y}}^\top - \bar{\mathbf{y}}_k)$ can either be obtained from second-order adjoints or can be approximated with one-sided finite differences for the direction \mathbf{y} , such that

$$N_{\mathbf{y}\mathbf{u}}^\top \Delta \bar{\mathbf{y}}_k = \frac{1}{\delta} (N_{\mathbf{u}}(\mathbf{y}_k + \delta \Delta \bar{\mathbf{y}}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k) - N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)) + O(\delta) \quad (3.110)$$

with $\Delta\bar{\mathbf{y}}_k = \bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k$. This requires an additional taping and tape evaluation with the updated state variable.

With the help of second-order adjoints one can obtain the second-order derivative term from the second-order adjoint $\dot{\hat{\mathbf{u}}}$ as described in (3.106) with the additional evaluation of h , such that

$$\begin{aligned} \dot{\hat{\mathbf{u}}}^\top &= (\bar{\mathbf{y}}^\top G_{\mathbf{u}\mathbf{y}} + \bar{z} f_{\mathbf{u}\mathbf{y}} + \bar{\mathbf{w}}^\top h_{\mathbf{u}\mathbf{y}}) \dot{\mathbf{y}} \\ &\quad + (\bar{\mathbf{y}}^\top G_{\mathbf{u}\mathbf{u}} + \bar{z} f_{\mathbf{u}\mathbf{u}} + \bar{\mathbf{w}}^\top h_{\mathbf{u}\mathbf{u}}) \dot{\mathbf{u}} + \dot{\bar{\mathbf{y}}}^\top G_{\mathbf{u}} + \dot{\bar{z}} f_{\mathbf{u}} + \dot{\bar{\mathbf{w}}}^\top h_{\mathbf{u}}. \end{aligned}$$

To obtain $N_{\mathbf{u}\mathbf{y}}(N_{\mathbf{y}}^\top - \bar{\mathbf{y}})$ from the first term one can set $\bar{z} = 1$, $\bar{\mathbf{y}} = \bar{\mathbf{y}}$ and $\bar{\mathbf{w}} = \boldsymbol{\mu}$ as usual and the direction $\dot{\mathbf{y}} = (N_{\mathbf{y}}^\top - \bar{\mathbf{y}})$. It is furthermore necessary to set $\dot{\bar{z}}$, $\dot{\bar{\mathbf{w}}}$, $\dot{\mathbf{u}}$ and $\dot{\bar{\mathbf{y}}}$ to zero. As the direction $\dot{\mathbf{y}}$ contains the update for the adjoint state, which is only available after evaluating the tape of the state iteration and the respective functions evaluations, one has to set up a new tape. Similar to the use of finite differences, one ends up with an additional tape recording and tape evaluation. If the preconditioner B is constructed based on a second evaluation of $\nabla_{\mathbf{u}} L^a$ at a modified design, all steps have to be repeated.

The preconditioner \check{B} for the multiplier update is based on the derivative $h_{\mathbf{y}}$. The derivative of the i -th constraint function can be obtained by evaluating the original tape setting $\bar{\mathbf{z}}$ to \mathbf{e}_i , i.e., the i -th unit vector, and all other values to zero. The tape evaluation has to be repeated for all $i = 1, \dots, r$. If the preconditioner is obtained using updates based on the secant equation (3.77), the derivative of the modified doubly augmented Lagrangian \tilde{L}^a with respect to $\boldsymbol{\mu}$ can be found with AD. One has, e.g.,

$$\nabla_{\boldsymbol{\mu}} \tilde{L}^a(y_k, \bar{\mathbf{y}}_k, u_k, \boldsymbol{\mu}_k) = \beta \Delta \bar{\mathbf{y}}_k h_{\mathbf{y}}(\mathbf{y}, \mathbf{u}) + h(\mathbf{y}, \mathbf{u}) + \varepsilon \boldsymbol{\mu},$$

and the first term requires a tape evaluation with $\bar{\mathbf{z}} = \beta \Delta \bar{\mathbf{y}}_k$. The term

$$\nabla_{\boldsymbol{\mu}} \tilde{L}^a(y_k, \bar{\mathbf{y}}_k, u_k, \boldsymbol{\mu}_k + \Delta \boldsymbol{\mu}_k)$$

requires an additional taping and evaluation with the modified variables.

3.3. Application and results

In this section, the proposed strategy for the integration of constraints in the one-shot approach is applied to the aerodynamic shape optimization problem, as presented in Section 2.4.1. Section 3.3.1 comments on the implementation of the strategy in SU2. Finally, in Section 3.3.2, the results of various studies are shown, focusing on the application for multi-objective aerodynamic shape optimization. A further application to the topology optimization of structures is shown in Section A.2.3 of the appendix.

3.3.1. Implementation in SU2

This section provides details of the realization of the extended approach that was proposed in Section 3.2.4 and 3.2.5 in SU2. The section starts with the presentation of the discrete adjoint methodology implemented in SU2. Afterwards, an algorithmic summary and further details on the implementation of the one-shot strategy are given.

3. One-Shot Method for PDE-constrained Optimization

3.3.1.1. Discrete adjoint in SU2

The discrete adjoint approach in SU2 is enabled by the differentiation of the fixed-point iteration with the help of algorithmic differentiation (see e.g. [4]).

The discretization of the original partial differential equation is, for example, in the case of fluid flow based on a finite volume method on a vertex-based median-dual grid resulting in a given number of cells for a given mesh resolution. Let Ω_i represent the volume of the cell i and let $R(\mathbf{y}, \mathbf{u})$ be the residual equation comprising the residuals R_i of the spatial discretization of each state equation (see e.g. (2.49)) in the respective cell i . The vector \mathbf{y} comprises the state variables y_i in each cell i . Then one can identify the discretized PDE constraint $c(\mathbf{y}, \mathbf{u}) = \mathbf{0}$ of problem (3.1) as

$$R(\mathbf{y}, \mathbf{u}) = \mathbf{0}. \quad (3.111)$$

Commonly, this equation is solved iteratively using a pseudo-time integration method. By using an implicit Euler scheme for the pseudo-time integration one ends up with the linear system

$$\left(D^k(\mathbf{u}) + R_{\mathbf{y}}(\mathbf{y}^k, \mathbf{u}) \right) \Delta \mathbf{y}^k = -R(\mathbf{y}^k, \mathbf{u}) \quad (3.112)$$

to be solved in each iteration k , with the update $\Delta y_i^k := y_i^{k+1} - y_i^k$ and

$$(D^k)_{ij} := \frac{|\Omega_i|}{\Delta t_i^k} \delta_{ij}.$$

Note that Δt_i^k is the pseudo-time-step that may be different in each cell due to the local time-stepping technique.

Usually, the flow Jacobian $R_{\mathbf{y}}$ is approximated using a first-order approximation. Thus, the implicit Euler discretization (3.112) naturally leads to a damped Newton method for solving the discretized PDE constraint $R(\mathbf{y}, \mathbf{u}) = \mathbf{0}$, given by

$$\mathbf{y}^{k+1} = \mathbf{y}^k - P(\mathbf{y}^k, \mathbf{u})R(\mathbf{y}^k, \mathbf{u}) =: G(\mathbf{y}^k, \mathbf{u}), \quad (3.113)$$

with a preconditioner $P \approx (D + R_{\mathbf{y}})^{-1}$. By introducing the operator G one obtains a fixed-point iteration.

It is assumed that the fixed-point iteration is contractive and, thus, converges to a unique stationary point. Furthermore, it is natural to assume that G is stationary only at feasible points of the residual formulation, i.e.,

$$R(\mathbf{y}^*, \mathbf{u}) = \mathbf{0} \Leftrightarrow \mathbf{y}^* = G(\mathbf{y}^*, \mathbf{u}).$$

The discrete adjoint technique can now be derived in several ways. The approach applied in SU2 is explained in detail in [4, 2]. The adjoint iteration is based on the reverse accumulation strategy making use of the reverse mode of algorithmic differentiation. Some simplifications are made to increase the overall efficiency of the AD-based discrete adjoint method. In SU2, the so-called *simplified recurrence* can be utilized for updating

the adjoint (see e.g. [90, 91]). The derivative of G , as defined in Equation (3.113), with respect to \mathbf{y} is given by

$$G_{\mathbf{y}}(\mathbf{y}, \mathbf{u}) = I - P_{\mathbf{y}}(\mathbf{y}, \mathbf{u})R(\mathbf{y}, \mathbf{u}) - P(\mathbf{y}, \mathbf{u})R_{\mathbf{y}}(\mathbf{y}, \mathbf{u}).$$

The idea of the simplified recurrence is to neglect the term $P_{\mathbf{y}}R$. If $P_{\mathbf{y}}$ is uniformly bounded the term vanishes when \mathbf{y} converges to the steady state solution \mathbf{y}^* . This is, for example, given if the preconditioner is differentiable in a neighborhood of \mathbf{y} (see [91]). The simplification can then be used when updating the adjoint variable to result in a simplified recurrence. It introduces a negligible error if the residual equation is sufficiently reduced (see [4]). For more complex situations of G , where P cannot be expressed explicitly, e.g., in multigrid methods, one has to be careful. However, the use of the simplified recurrence is often still allowed [93].

For reasons of completeness, it is to note that an additional constraint $\mathbf{x} = M(\mathbf{u})$ has to be considered in the optimization problem (3.48) when shape optimization is used. The reason for that is that the computational mesh is subject to change. Let $M : \mathbb{R}^n \rightarrow \mathbb{R}^{n_x}$ be a linear function that formally contains the surface and mesh deformation. In SU2, a mesh deformation routine using the Linear Elasticity method creates new mesh point coordinates $\mathbf{x} \in \mathbb{R}^{n_x}$. This introduces a dependency on the variable $\mathbf{x}(\mathbf{u})$ for all objective and constraint functions. As a result, the modified Lagrangian with the modified shifted Lagrangian \tilde{N} associated to this problem reads

$$\tilde{L}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}, \bar{\mathbf{x}}, \mathbf{u}, \boldsymbol{\mu}) = \tilde{N}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}, \boldsymbol{\mu}) - \mathbf{y}^\top \bar{\mathbf{y}} + (M(\mathbf{u}) - \mathbf{x})^\top \bar{\mathbf{x}}, \quad (3.114)$$

where $\bar{\mathbf{x}}$ is the Lagrange multiplier associated with the mesh deformation constraint. Analogously to (3.53), one obtains the fixed-point iterations for $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}$ as

$$\bar{\mathbf{y}}_{k+1} = \tilde{N}_{\mathbf{y}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{x}_k, \boldsymbol{\mu}_k)^\top, \quad (3.115)$$

$$\bar{\mathbf{x}}_{k+1} = \tilde{N}_{\mathbf{x}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{x}_k, \boldsymbol{\mu}_k)^\top, \quad (3.116)$$

and the update $\mathbf{u}_{k+1} = \mathbf{u}_k - B_k^{-1}M_{\mathbf{u}}(\mathbf{u}_k)^\top \bar{\mathbf{x}}_k$ can be obtained from $\bar{\mathbf{x}}_k$. Equation (3.116) is just an evaluation of $\tilde{N}_{\mathbf{x}}$ at the current state.

The introduction of the additional constraint $M(\mathbf{u}) - \mathbf{x}$ can be done similarly to the state constraint by introducing a contractive fixed-point iteration with a carefully chosen preconditioner \tilde{B} for the mesh point coordinates \mathbf{x} , namely

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tilde{B}_k^{-1}(M(\mathbf{u}_k) - \mathbf{x}_k). \quad (3.117)$$

Equation (3.117) can then be iterated alongside the state variables, adjoint variables and the design in a one-shot iteration similarly to (3.53). This necessitates the derivation of new convergence properties based on a modified augmented Lagrangian \tilde{L}^a with an additional penalty term for the mesh update. Furthermore, a suitable preconditioner that also ensures a good mesh quality for each update has to be derived.

As it is not the intention to focus on this kind of integration in the present work, it is restricted to the situation where the linear system is solved in each one-shot iteration,

3. One-Shot Method for PDE-constrained Optimization

and \mathbf{x}_k denotes the mesh corresponding to the respective design \mathbf{u}_k . This is a feasible choice as, usually, the computational effort for solving the linear system is negligible in comparison to the effort for solving the state equation. As a result, the solution of the linear system can be interpreted as a projection step. This projection has to be considered in the implementation of the method. Note that the projection step and its corresponding adjoint version also have to be considered when constructing the design space preconditioner B . To simplify and generalize the notation, these steps in the design chain are again neglected in the following.

In previous works, the open-source tool CoDiPack was already applied to SU2 [4, 2] to enable the robust AD-based discrete adjoint strategy presented above. The runtime factor when comparing a state iteration of a three-dimensional flow solution with an iteration of the adjoint equation is around 2.3 without any manual tuning [2]. Usually, the factor for memory consumption is around 10 – 12. CoDiPack [231] is based on the operator overloading approach and is specially designed for high-performance computing and industrial applications. It provides different taping strategies, namely Jacobi taping and primal value taping. In CoDiPack, algorithmic differentiation is applied to the statement level instead of the operation level. The idea is to store only the information for each statement instead of the information for all operations involved in the statement. This usually increases the overall efficiency.

3.3.1.2. Optimization algorithm

As established before, the augmented Lagrangian (3.54) can be minimized with an appropriate line search for the determination of the step length γ using the descent direction

$$\mathbf{s}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu}) = \begin{pmatrix} G(\mathbf{y}, \mathbf{u}) - \mathbf{y} \\ N_{\mathbf{y}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^{\top} - \bar{\mathbf{y}} \\ -\gamma B^{-1} N_{\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{u}, \boldsymbol{\mu})^{\top} \\ -\check{B}^{-1} h(\mathbf{y}, \mathbf{u}) \end{pmatrix} \quad (3.118)$$

of equation (3.53).

The following steps outline the basic framework of the adapted one-shot algorithm:

0. **Initialization.** Choose initial points $\mathbf{y}_0 \in Y$ and $\mathbf{u}_0 \in U_{ad}$, Lagrange multipliers $\bar{\mathbf{y}}_0$ and $\boldsymbol{\mu}_0$, $\varepsilon > 0$, parameters α and β and preconditioners B_0 and \check{B}_0 . Set $k = 0$.
1. **Step computation.** Compute a descent direction $\mathbf{s}_k(\mathbf{y}_k, \bar{\mathbf{y}}, \mathbf{u}_k, \boldsymbol{\mu}_k)$ using (3.118).
2. **Choose step length.** Compute an efficient step length γ_k .
2. **Updates.** Update \mathbf{y}_{k+1} , $\bar{\mathbf{y}}_{k+1}$, \mathbf{u}_{k+1} and $\boldsymbol{\mu}_{k+1}$.
3. **Stationarity test.** If $\|N_{\mathbf{u}}(\mathbf{y}_{k+1}, \bar{\mathbf{y}}_{k+1}, \mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1})\|_2 < \varepsilon$: STOP.
4. **Prepare next iteration.** Set $k = k + 1$, update B_k and \check{B}_k , go to Step 1.

However, the success of the optimization depends on the details, i.e., the construction of the preconditioners as well as the line search strategy. The following paragraphs consider the respective implementation details.

Preconditioner for design update The preconditioner B can be approximated using the BFGS method with the secant equation

$$B\mathbf{s}_k = \mathbf{r}_k \quad (3.119)$$

with

$$\mathbf{r}_k := \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k + \Delta\mathbf{u}_k, \boldsymbol{\mu}_k) - \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k) \quad (3.120)$$

and $\mathbf{s}_k := \Delta\mathbf{u}_k$, as proposed in [103]. This means that \mathbf{r}_k needs an additional evaluation of the gradient of the doubly augmented Lagrangian, which causes the computational effort for obtaining $\nabla_{\mathbf{u}}L^a$ twice in each iteration of the optimization.

The use of a modified secant equation with

$$\tilde{\mathbf{r}}_k := \nabla_{\mathbf{u}}L^a(\mathbf{y}_{k+1}, \bar{\mathbf{y}}_{k+1}, \mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) - \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k) \quad (3.121)$$

has become a convenient alternative in one-shot methods. In general, this makes sense since the result of the analysis gives that the preconditioner shall resemble (3.73) for $\Delta\mathbf{y}_k = 0$ and $\Delta\bar{\mathbf{y}}_k = 0$. The modified secant equation can also be motivated by observing that one can derive a preconditioner without using the approximation in the optimization problem (3.72). The resulting optimal solution

$$B\Delta\mathbf{u} = \nabla_{\mathbf{u}\bar{\mathbf{y}}}L^a\Delta\bar{\mathbf{y}} + \nabla_{\mathbf{u}\mathbf{y}}L^a\Delta\mathbf{y} + \nabla_{\mathbf{u}\boldsymbol{\mu}}L^a\Delta\boldsymbol{\mu} + \nabla_{\mathbf{u}\mathbf{u}}L^a\Delta\mathbf{u}$$

directly suggests the use of $\tilde{\mathbf{r}}_k$ as an approximation (see [19]). Note that one has to choose the modified preconditioner $\frac{1}{\sigma}B_k$ to ensure that it is large enough. This factor can be included in the step length γ . Furthermore, the update rule (3.89) is used to construct the inverse of B_k .

For reasons of efficiency, the present work uses a backtracking line search for finding γ in the following and one sets $B = I$ when the curvature condition $\mathbf{r}_k^\top \mathbf{s}_k > 0$ is not fulfilled, which is common practice. The condition for sufficient decrease

$$L^a(\mathbf{y}_{k+1}, \bar{\mathbf{y}}_{k+1}, \mathbf{u}_{k+1}, \boldsymbol{\mu}_{k+1}) \leq L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k) + c_1\gamma\mathbf{d}_k^\top \nabla_{\mathbf{u}}L^a(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k) \quad (3.122)$$

is consistent with the strategy for the BFGS update with

$$\mathbf{d}_k := -B_k^{-1}N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k).$$

For the BFGS update step, $\nabla_{\mathbf{u}}L^a$ is obtained for each iterate k with the reverse mode of AD using the discrete adjoint of SU2, as explained in Section 3.3.1.1. As already presented in Section 3.2.6.5, the second-order derivative term in the doubly augmented Lagrangian can be either obtained with the help of finite differences or with the help second-order adjoints. Both approaches involve a second taping of the solver iteration, as well as the corresponding tape evaluation. In the following, a finite difference approximation is used. Studies for second-order adjoints can be found in Section 3.3.2.2.

3. One-Shot Method for PDE-constrained Optimization

Preconditioner for multiplier update For the first results, a constant positive definite diagonal matrix is used as a preconditioner. The values of the diagonal matrix are obtained based on a parameter study. It is important to note that one can guarantee descent if the preconditioner is large enough. A carefully chosen preconditioner will result in an improved speed of convergence of the overall method. The preconditioner proposed in Section 3.2.4.2 is tested later in Section 3.3.2.3. A meaningful choice for the starting value of the multiplier is, e.g., given by

$$(\boldsymbol{\mu}_0)_i = \|\nabla_{\mathbf{u}} f(\mathbf{y}_0, \mathbf{u}_0)\| / \|\nabla_{\mathbf{u}} h_i(\mathbf{y}_0, \mathbf{u}_0)\| \text{ for } i = 1, \dots, r.$$

Bound Constraints Bound constraints are considered throughout the optimization. Numerical tests show that the BFGS method with bound constraints and the limited-memory BFGS method with bound constraints behave similarly. The reason for that is that the backtracking line search without ensuring the curvature condition will trigger a resetting of the Hessian after a few updates for both updating strategies.

When using bound constraints, the scaling of the design variables, the objective function, and the constraint functions becomes more important. This avoids that the first design updates are too large and would reach the bounds of the design space, since this may slow down the convergence of the algorithm due to oscillations at the beginning. Throughout, the objective functions and the constraint functions will be scaled with a factor of 10^{-2} . Note that some authors in the literature suggest starting the one-shot method after a certain number of iterations to avoid too large updates at the beginning (see e.g. [210]). The present study shows that the effects of both strategies, scaling or a delayed start, is similar.

Implementation in SU2 As SU2 has a highly modular structure, the same routines can be called for different areas of application. This modularity can be easily transferred to the implementation of the one-shot approach. Existing access routines, which enable the implementation of the discrete adjoint strategy, can be reused or replaced by similar routines to enable the calculation of sensitivities for the one-shot approach. This all happens outside the main solver level, such that other types of solvers can be easily introduced. Any needed parameters are defined in an external configuration file.

The implemented algorithm is presented in Algorithm 8. It is presented for the case of bound constraints, the proposed preconditioner for the multiplier update, and finite differences for obtaining the second-order derivative term in the doubly augmented Lagrangian. Note that the projection of the gradient onto the design space has been omitted. It can be observed that each line search step involves an update of the state and the adjoint iteration. The outer optimization steps need additional computational effort for obtaining the gradient of the doubly augmented Lagrangian, i.e., an additional tape evaluation and an update of the state and the adjoint involving additional taping. One counts a number of r tape evaluations for the multiplier preconditioner. As the state and adjoint variables are overwritten during the algorithm, they need to be stored in memory slots A and B.

Algorithm 8: One-shot optimization in SU2 (pseudo-code).

Input:

γ_0 : initial step length (estimate of σ)
 $\boldsymbol{\mu}_0$: starting value for constraint multiplier
 $\mathbf{y}_0, \bar{\mathbf{y}}_0, \mathbf{u}_0, \varepsilon, \varepsilon_1, \varepsilon_2, \delta$

Function:

$k \leftarrow 0, B_0 \leftarrow I, c_1 \leftarrow 10^{-4}$

repeat

$\gamma \leftarrow \gamma_0$

Shift A to B, Store \mathbf{u}_{k-1}

repeat

if $k \neq 0$ **then**

$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k-1} + \Delta\boldsymbol{\mu}_{k-1}$

if $\gamma \neq 1.0$ **then**

Load $\mathbf{y}_k, \bar{\mathbf{y}}_k$ from B, Load \mathbf{u}_{k-1}

$\gamma = 0.5\gamma$

end if

$\mathbf{u}_k = P(\mathbf{u}_{k-1} + \gamma\mathbf{d}_{k-1}), \Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$

end if

$\mathbf{y}_{k+1} = G(\mathbf{y}_k, \mathbf{u}_k), f \leftarrow f(\mathbf{y}_k, \mathbf{u}_k), h \leftarrow h(\mathbf{y}_k, \mathbf{u}_k)$

$\bar{\mathbf{y}}_{k+1} = N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top$

$\Delta\mathbf{y}_k = \mathbf{y}_{k+1} - \mathbf{y}_k, \Delta\bar{\mathbf{y}}_k = \bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k$

$L_k^a = \frac{\alpha}{2}\|\Delta\mathbf{y}_k\|^2 + \frac{\beta}{2}(\|\Delta\bar{\mathbf{y}}_k\|^2 + \|h\|^2) + f + \boldsymbol{\mu}_k^\top h + \Delta\mathbf{y}_k^\top \bar{\mathbf{y}}_k$ (Use $\bar{\mathbf{y}}_k$ from B)

until $L_k^a \leq L_{k-1}^a + c_1\Delta\mathbf{u}_k^\top \nabla_{\mathbf{u}}L_{k-1}^a$

Store $\mathbf{y}_{k+1}, \bar{\mathbf{y}}_{k+1}$ in A, Store $N_{\mathbf{u}}$

$\nabla_{\mathbf{u}}L_k^a \leftarrow N_{\mathbf{u}}^\top$ (in $\bar{\mathbf{u}}_{k+1}$)

Obtain $h_{\mathbf{u}}$ from tape evaluations

$\Delta\boldsymbol{\mu}_k = [\beta h_{\mathbf{y}} h_{\mathbf{y}}^\top + \varepsilon I]^{-1} h$

Obtain $\alpha G_{\mathbf{u}}(\mathbf{y}_k, \mathbf{u}_k)^\top \Delta\mathbf{y}_k + \alpha h_{\mathbf{u}}^\top h$ from tape evaluation

$\nabla_{\mathbf{u}}L_k^a \leftarrow \nabla_{\mathbf{u}}L_k^a + \alpha G_{\mathbf{u}}(\mathbf{y}_k, \mathbf{u}_k)^\top \Delta\mathbf{y}_k + \alpha h_{\mathbf{u}}^\top h$

Load $\mathbf{y}_k, \bar{\mathbf{y}}_k$ from B

$\mathbf{y} = \mathbf{y}_k + \delta\Delta\bar{\mathbf{y}}_k$

$\tilde{\mathbf{y}} \leftarrow G(\mathbf{y}, \mathbf{u}_k), f \leftarrow f(\mathbf{y}, \mathbf{u}_k), h \leftarrow h(\mathbf{y}, \mathbf{u}_k)$

$\bar{\mathbf{u}} = N_{\mathbf{u}}(\mathbf{y}, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)^\top$

$\nabla_{\mathbf{u}}L_k^a \leftarrow \nabla_{\mathbf{u}}L_k^a + \frac{\beta}{\delta}(\bar{\mathbf{u}} - N_{\mathbf{u}})$

if $k \neq 0$ **then**

$\mathbf{r}_k = P_{\mathcal{I}^\varepsilon}(\nabla_{\mathbf{u}}L_k^a - \nabla_{\mathbf{u}}L_{k-1}^a), \mathbf{s}_k = P_{\mathcal{I}^\varepsilon}(\Delta\mathbf{u}_k)$

Determine B_k using Equation (3.89) for the inverse

end if

$\mathbf{d}_k = -(P_{\mathcal{A}^\varepsilon} + P_{\mathcal{I}^\varepsilon}B_k^{-1}P_{\mathcal{I}^\varepsilon})N_{\mathbf{u}}(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k, \boldsymbol{\mu}_k)$

Store $\nabla_{\mathbf{u}}L_k^a, L_k^a$

until $\|N_{\mathbf{u}}\|_2 < \varepsilon_1$ or $\|\Delta\mathbf{u}_k\| < \varepsilon_2$

3.3.2. Results of One-Shot Aerodynamic Shape Optimization

The present section serves to present the results for applying the one-shot strategy for solving the aerodynamic shape optimization problem.

In Section 3.3.2.1 and 3.3.2.2, different constrained optimization problems with a single and multiple equality constraints are solved using the proposed augmented one-shot method. A multi-objective aerodynamic shape optimization problem involving the lift constraint is considered. Finally, Section 3.3.2.3 shows the investigation of the use of second-order adjoints for the second-order derivative term in the augmented Lagrangian, and Section 3.3.2.4 presents results based on a specific choice of the multiplier preconditioner.

3.3.2.1. Results for Aerodynamic Shape Optimization with Lift Constraint

The first application example for the extended one-shot method is the multi-objective optimization of the introduced aerodynamic shape optimization, neglecting the additional constraints for the moment coefficient and the thickness. As a result, the optimization task is to minimize the drag coefficient c_d and maximize the lift coefficient c_l without any additional equality constraints. The flow properties and the parameterization is chosen as in Chapter 2. Bound constraints are prescribed for the design variables such that $-0.005 \leq u_i \leq 0.005$ for $i = 1, \dots, n$.

The application of the equality constraint method to the multi-objective optimization problem results in several constrained optimization problems of the form (3.48). As explained in Section 2.2.3, the outlines of the Pareto optimal front can be found by minimizing the objective functions individually without imposing additional constraints using the original single-step one-shot approach. Having observed in Chapter 2 that the Pareto optimal front is connected and convex, it is feasible to use the equality constraint method. The extended one-shot approach is used for all constrained optimization problems.¹³

Figure 3.1a shows the optimization history for the minimization of the drag coefficient without additional constraints, and the corresponding common logarithm of the pressure residual and the adjoint pressure residual are shown in Figure 3.1b. Both residuals reach a desired level of accuracy during the optimization. The flow solution of the steady Euler equations would need around 200 state iterations to achieve a pressure residual of the order of 10^{-7} . The optimization needs around 270 outer iterations and 300 additional line search steps to reach a design with a drag coefficient of 0.00033 (3 drag counts) and a lift coefficient of 0.098. When comparing the number of state iterations in the one-shot procedure with the number of iterations for a flow solution, the retardation factor is around 2.9. When using a nested strategy with a BFGS method, one needs 31 outer iterations to converged to a similar level of accuracy. The overall computational time is about 5.15 times higher. The multipliers are chosen as $\alpha = 20$ and $\beta = 2$ with the help of a parameter study. The higher value for α is based on the condition (3.65)

¹³Tests show that the one-shot strategy without hybridization is able to find the same solutions as for the hybridization with stricter bound constraints.

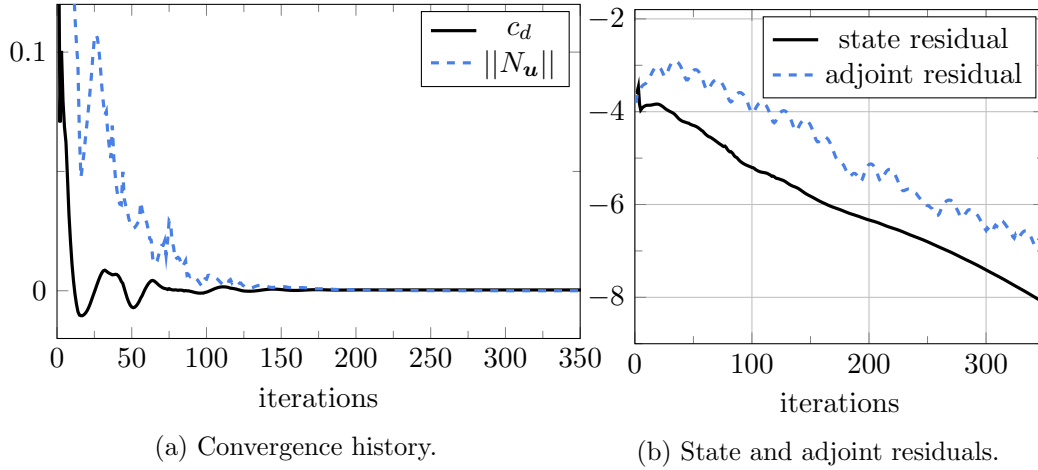


Figure 3.1.: Optimization history of single-step one-shot approach (a) and corresponding state and adjoint residuals (b).

using estimates for ρ and $\|N_{\mathbf{y}\mathbf{y}}\|$ (see [102]).

In the present study, the Pareto front is only scanned in one direction. The reason for this is that the maximization of the lift coefficient with an additional constraint for the drag coefficient requires an additional parameter study. Furthermore, one may observe a bad convergence behavior for large values of the lift coefficient ($c_l \geq 0.6$). This may be explained by the missing additional constraints and the high nonlinearities due to shocks. As a result, the Pareto front is only scanned in the region $0.098 \leq c_l \leq 0.6$.

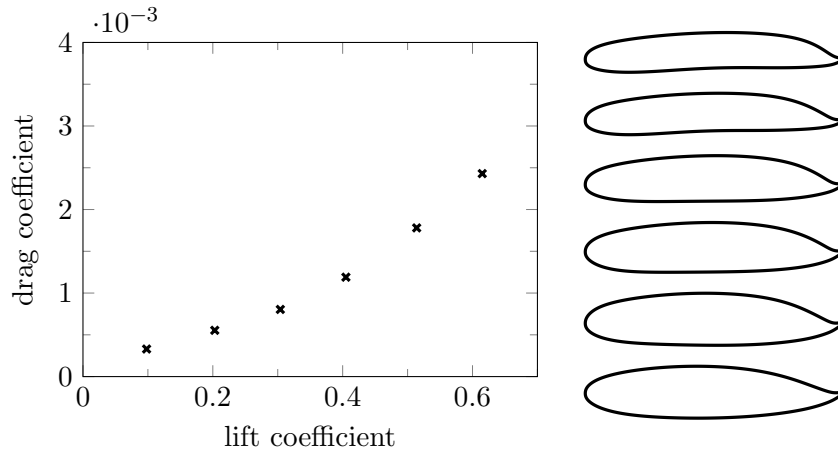


Figure 3.2.: Obtained points in objective space (left) and corresponding designs (right) using the one-shot approach with additional constraints.

The solutions marked with crosses in Figure 3.2 in the objective space are possible Pareto optimal solutions and were found by minimizing the drag coefficient while varying the target lift coefficient between the bound of $c_{l,t}^0 = 0.098$, which is determined by the

3. One-Shot Method for PDE-constrained Optimization

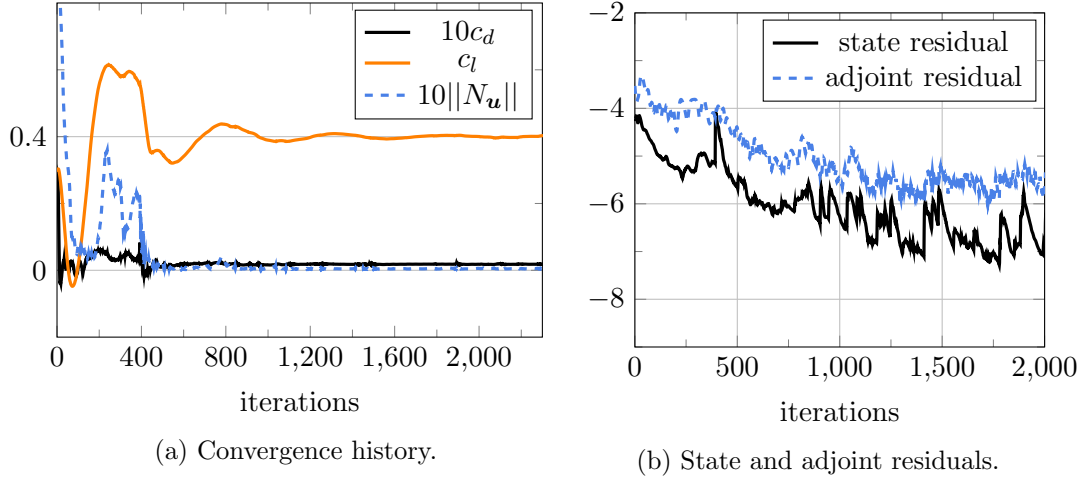


Figure 3.3.: Exemplary convergence history of one-shot optimization with lift constraint of 0.4 (a) and the corresponding logarithmic plot of the state and adjoint residuals (b).

minimum drag coefficient, and $c_{l,t}^5 = 0.6$. The constraint function for the lift coefficient is then given as

$$h(\mathbf{y}, \mathbf{u}) = c_l(\mathbf{y}, \mathbf{u}) - c_{l,t}^i$$

for each optimization problem $i = 1, \dots, 5$. The designs on the right side of the figure correspond to the solutions from top ($c_l = 0.6$) to bottom ($c_l = 0.098$).

Exemplary, the optimization history for the minimization of the drag coefficient with a target lift of 0.4 is shown in Figure 3.3a. The preconditioner $\check{B} = 20$ is constant throughout the optimization, and the corresponding multiplier μ is zero at the beginning. The drag coefficient of 0.0012 (12 drag counts) is reached after around 1400 iterations. The red line in the figure shows the lift coefficient, which approaches the desired value of 0.40 during the optimization.

The common logarithm of the pressure residual and the adjoint pressure residual corresponding to the optimization history are shown in Figure 3.3b. Both residuals reach a desired level of accuracy during the optimization. Towards the end of the optimization, they do not stagnate but drop at a slower rate. All shown iteration steps still involve an update of the design variable. When the updates stop due to the abort criterion, the usual piggy-back convergence behavior can be observed.

The same multipliers and the same preconditioner \check{B} were used for all six constrained optimization problems showing the robustness of the implemented approach. The optimization runs need around 1200 to 3000 outer optimization steps to converge. The additional equality constraint does not affect the costs of a single one-shot iteration as the number of tape recordings and evaluations is the same when compared to the classical one-shot approach. Note, however, that this is only the case for a constant multiplier preconditioner. Still, it can be observed for the specific test case that the solution of

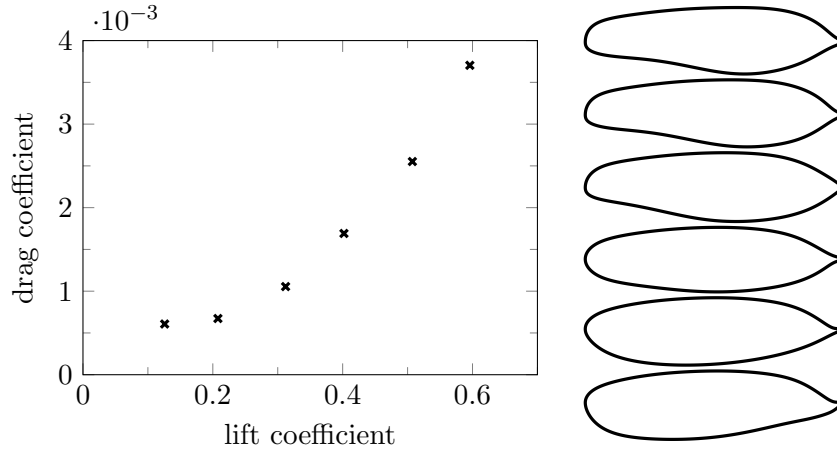


Figure 3.4.: Obtained points in objective space (left) and corresponding designs (right) for test case with multiple constraints solved with the adapted one-shot method.

the constrained optimization problem requires more optimization steps. Based on the number of iterations in the inner and outer optimization loop, one obtains retardation factors of about 10 up to 13.

3.3.2.2. Results for Aerodynamic Shape Optimization with Multiple Constraints

To obtain more realistic designs in comparison to the ones resulting from the optimization with a lift constraint, the original constraints of the aerodynamic shape optimization problem (2.62) are re-introduced, resulting in the constraint vector

$$\mathbf{h}(\mathbf{y}, \mathbf{u}) = \begin{pmatrix} c_l(\mathbf{y}, \mathbf{u}) - c_{l,t}^i \\ c_m(\mathbf{y}, \mathbf{u}) \\ d(\mathbf{u}) - 0.12 \end{pmatrix}.$$

Note that the thickness constraint is treated exactly like the other constraints, although it is a geometric constraint and does, therefore, not depend on the state \mathbf{y} . This is done to study the behavior of the method for several constraints. In a more general setting, it can be of advantage to use a different strategy, e.g., the use of an additional optimization problem in the design space for a suitable projection.

Again, one considers the minimization of the drag coefficient and the maximization of the lift coefficient. The found Pareto optimal points are shown in Figure 3.4. The corresponding designs on the right side represents the designs for an increasing lift constraint from the bottom to the top. It can be observed that the upper surface close to the leading edge of the airfoil flattens for a higher lift constraint, which is the typical result that was also obtained in Chapter 2. In a direct comparison with the results of this chapter, one may observe that the designs and the objective values for low lift constraints are similar. Notably, for the one-shot method the target value for the mo-

3. One-Shot Method for PDE-constrained Optimization

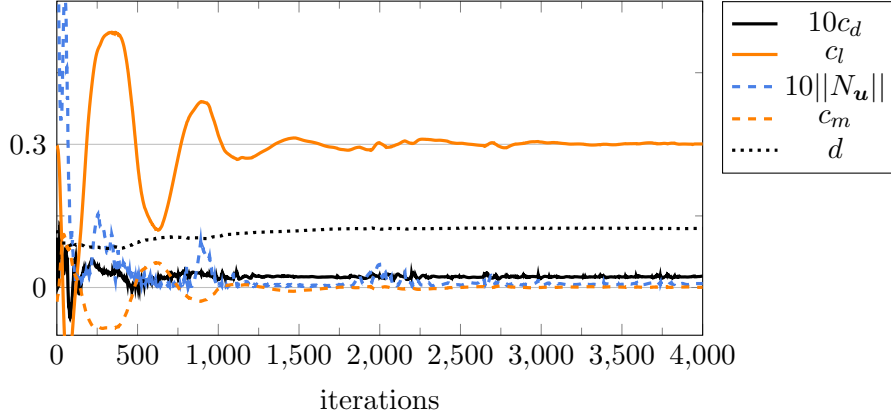


Figure 3.5.: History of the adapted one-shot optimization for test case with multiple constraints.

ment coefficient is prescribed as an equality constraint. A difference can be observed for higher lift constraints. The objective values are similar, but the one-shot strategy converges to a different design. A comparison of a design is shown in Section A.2.4 of the appendix. Interestingly, the one-shot strategy tends to find designs similar to the ones that will be found in Chapter 4 when considering robustness with respect to geometrical uncertainties.

An exemplary optimization history for a lift constraint of 0.3 is shown in Figure 3.5. The diagonal matrix $\mathbf{diag}(20, 20, 100)$ is used as preconditioner for the equality constraints. As before, $\alpha = 20$ and $\beta = 2$. A drag coefficient of 0.0012 is reached for the optimized design. Again, all Pareto optimal designs could be obtained with similar configurations of the optimization algorithm. As the constrained optimization tests need around 1700 to 3800 outer optimization steps to converge, the number of iterations and the overall computational costs of the optimization are comparable to the test case with a single constraint. This is quite understandable, as the convergence behavior of the moment constraint is very similar to the one of the lift constraint. A faster convergence is observed for the thickness constraint. This can be explained by the independence concerning the convergence of the state variables.

3.3.2.3. Using Second-Order Adjoints for Design Space Preconditioner

For the results presented above, the second-order derivative term $\beta N_{\mathbf{y}\mathbf{u}}^\top (N_{\mathbf{y}}^\top - \boldsymbol{\mu})$ in the reduced gradient of the doubly augmented Lagrangian was approximated using finite differences in the direction of the state \mathbf{y} with a step size of $\delta = 10^{-5}$. This is common practice and works for the specific application. Nevertheless, the introduction of second-order derivatives in the given SU2 framework is quite simple and, therefore, it is worthwhile testing it. The general strategy for obtaining the second-order derivative term with second-order adjoints was explained in Section 3.2.6.5.

In the implementation of second-order adjoints, the old data type

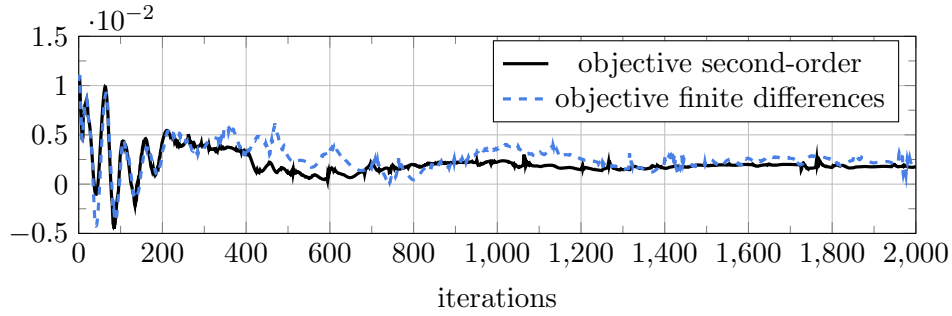


Figure 3.6.: Comparison of the use of second-order adjoints versus finite differences. The convergence history for the objective function is shown for the original preconditioner and the preconditioner making use of second-order adjoints.

```
RealReverseGen<double> RReverse
```

is substituted by the second-order data type

```
RealReverseGen<RForward> RTanORev.
```

The values are now accessed via first getting the reverse part and then the tangent-linear part, i.e., `tape.getGradient(x[adjPos++]).getGradient()`.

	RReverse	RTanORev	SOTerm
run 1	1	1.403	1.557
run 2	1	1.392	1.514
run 3	1	1.360	1.461
mean	1	1.385	1.511

Table 3.1.: Runtime factors for the introduction of second-order adjoints. The factor is scaled to the runtime for building the finite difference approximation with the RReverse datatype (first column). The change of datatype will increase the runtime (second column), as well as the use of exact derivatives instead of finite differences (third column).

Table 3.1 gives the runtime factor when introducing the new data type compared to the old data type. The runtime is measured for the evaluation of the second-order derivative term. The test was performed by taking the mean of 100 evaluations for three different runs. The overall mean is 1.385. When using second-order adjoints instead of finite differences, the mean factor is 1.511 in comparison to the use of the old datatype (see last column). In general, the use of second-order adjoints is computationally more expensive than the use of finite difference, but the additional costs are not significantly high.

When looking at the performance of second-order adjoints for the one-shot optimization in Figure 3.6, it can be seen that the convergence of, e.g., the objective function throughout the optimization appears to be smoother and slightly faster. A similar behav-

3. One-Shot Method for PDE-constrained Optimization

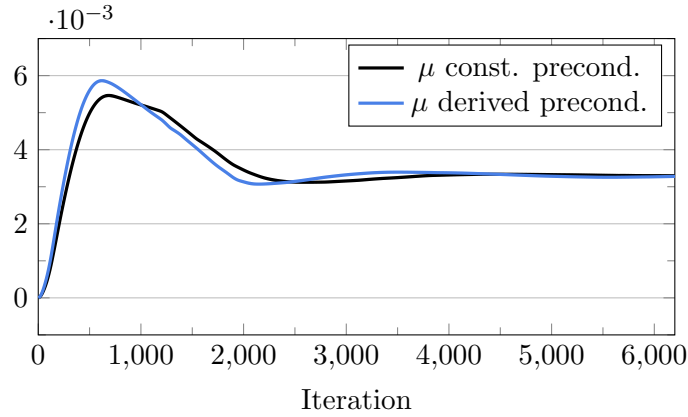


Figure 3.7.: Convergence history of multiplier for constant preconditioner for the multiplier update and for the preconditioner given by Equation (3.75).

ior can be observed for the constraint functions. But, based on experience, this behavior can be counterbalanced by a higher number of inner line search steps, although these are cheaper than the outer iterations. In this case, one has 50 percent more line search steps. Also, for some tests, no clear difference could be seen in the convergence behavior. As a result, for the present application, no clear advantage was determined. However, in general, the smoother convergence may make the use of second-order adjoints advantageous for test cases with bad convergence behaviors.

3.3.2.4. Convergence Results for Multiplier Preconditioner

The results of the previous sections were based on a constant preconditioner for the multiplier update. The preconditioner was chosen large enough to achieve convergence. When using more than one constraint, the preconditioner is constructed as a diagonal matrix with large enough entries. However, this requires a parameter study for finding the respective values for different constraints. We have seen that the convergence results are satisfying, but it remains to analyze if the preconditioner proposed in Section 3.2.4.2 enables a faster convergence. It is given by $\tilde{B} = \beta h_{\mathbf{y}} h_{\mathbf{y}}^{\top} + \varepsilon I$ (see Equation (3.75)). In the following, the preconditioner is calculated exactly, and the inverse is applied in the multiplier update.

Numerical tests show that, when keeping the parameters α and β of the original calculations, the choice of \tilde{B} only converges for a high value of ε . However, this was not the intention when deriving the preconditioner since the information of the first term does not have an influence anymore. When increasing β and using a small value for ε , it is not possible to find a suitable parameter α to obtain convergence. This was already predicted by the analysis of the choice of the preconditioner. The idea is to scale the first term of the preconditioner with a constant factor of $c_{\tilde{B}}$, such that $\tilde{B} = c_{\tilde{B}} \beta h_{\mathbf{y}} h_{\mathbf{y}}^{\top} + \varepsilon I$. As a result, the influence of the preconditioner can be maintained without changing the values for α and β . Similarly to the factor based on σ for the design space preconditioner,

the factor $c_{\check{B}}$ may serve to ensure that the preconditioner is large enough. The procedure can also be interpreted as using a smaller step size for the multiplier update if the value of ε is scaled appropriately.

Figure 3.7 shows the application for lift-constrained drag minimization with a target lift of $c_l^t = 0.327$. The convergence history of the multiplier based proposed preconditioner with $c_{\check{B}} = 100$ is compared to using the constant multiplier $\check{B} = 20$. Again, the penalty parameters are given by $\alpha = 20$ and $\beta = 2$. It can be observed that the speed of convergence of the multiplier is similar, such that the proposed preconditioner does not show any clear improvements. Similar results can be observed for the test case with additional constraints for the moment coefficient and the thickness when directly setting up the inverse of the preconditioner for the multiplier update. In this case, the choice of $c_{\check{B}} = 100$ results in a slower speed of convergence compared to using a constant multiplier. A similar convergence behavior is obtained with a different scaling of the multiplier update for the thickness constraint.

4. Robustness in Multi-Objective Optimization

Considering uncertainties in a problem formulation is a significant step towards realistic design. The performance of a design that is only optimized for specific nominal conditions may deviate under real-world conditions. In engineering design, it is then of general interest to find designs that are robust and optimal under uncertainty. Existing formulations of robustness are presented in this chapter, and shortcomings are discussed. The primary goal is to discuss the use of different robustness measures for multi-objective optimization and to formulate a new measure for robustness in a multi-objective context. The main idea for introducing a new measure for robustness in the context of multi-objective optimization is to account for the structure of the uncertainty in objective space. Especially, solely judging robustness based on the expected value or the variance of the individual objective functions cannot necessarily account for such a structure.

When considering uncertainties, an important aspect is how to model them and how to propagate them. There exist different approaches to describe uncertainty depending on the available information and the underlying problem. This work is restricted to a probabilistic description of uncertainties instead of, for example, using fuzzy set theory. The non-intrusive polynomial chaos method, which can be interpreted as a particular type of collocation method inspired by spectral methods, is employed for uncertainty propagation.

In the following, some basic notations for probability theory are introduced in Section 4.1. Furthermore, different concepts in uncertainty quantification are presented in Section 4.2, focusing on spectral methods. The sections on probability theory and quantification of uncertainties are based on [251]. For details on the non-intrusive polynomial chaos approach the reader may refer to Sections 4.2.1 and 4.2.2.3. A short overview of robust design formulations in single-objective optimization is given in Section 4.3. The section focuses on robust design in the context of multi-objective optimization, and new measures for robustness are introduced in Section 4.3.3. Finally, the application of the introduced concepts is shown in Section 4.4.

4.1. Probability Theory

When looking at random differential equations, one has to differentiate them from stochastic differential equations, where the realization of a random process is not smooth. Random differential equations can be written with parameters that either depend on the elements from the probability space or on its realizations. The present work is restricted

4. Robustness in Multi-Objective Optimization

to random differential equations and the uncertainty in the parameters is modeled with a probability distribution. Thus, in the following, the uncertainty will be expressed by making use of *random variables*. It is helpful to introduce some terms and notations for probability theory in the following to provide a basis for further discussion.

Random variables are measurable functions $Z : \Omega \rightarrow \mathbb{R}$ that map an outcome of a random experiment onto a value. Furthermore, it holds that $\{\omega \in \Omega \mid Z(\omega) \leq z\} \in \mathcal{F}$ for all realizations $z \in \mathbb{R}$. The set Ω is the set of all possible outcomes and \mathcal{F} its power set. Together with the probability measure P , they form the *probability space* (Ω, \mathcal{F}, P) .

A *random process* is a collection of random variables with an ordered index set, e.g., for a time-dependent random process $Z_t = \{Z(t, \omega), t \in \mathcal{T}\}$. A *random field* is the generalization for vector-valued indices without any ordering.

Definition 13 (Cumulative distribution function). *The cumulative distribution function (CDF) is a function*

$$F_Z : \mathbb{R} \rightarrow [0, 1], \quad F_Z(z) = P(\{\omega \in \Omega \mid Z(\omega) \leq z\}) = \int_{-\infty}^z f_Z(s) ds, \quad (4.1)$$

with the probability density function (PDF) f_Z of Z .

The moments of a PDF are described by the formula

$$E(Z^n) := \int_{\mathbb{R}} z^n f_Z(z) dz, \quad (4.2)$$

where $\mu = E(Z)$ is the *expected value* and $\sigma^2 = E((Z - \mu)^2)$ is the *variance*.

A joint CDF for a random vector $\mathbf{Z} \in \mathbb{R}^N$ is given by

$$F_{\mathbf{Z}} : \mathbb{R}^N \rightarrow [0, 1], \quad F_{\mathbf{Z}}(z_1, \dots, z_N) = P(\{\omega \in \Omega \mid Z_j(\omega) \leq z_j, j = 1, \dots, N\}). \quad (4.3)$$

Marginal densities $f_{Z_i}(z_i)$ of one variable Z_i are obtained by integrating $f_{\mathbf{Z}}(z_1, \dots, z_N)$ against all other variables z_j with $j = 1, \dots, N$ and $j \neq i$.

An important aspect of the quantification of uncertainties is the *independence* of random variables, also referred to as *statistical independence*.

Definition 14 (Independent random variables). *The random variables Z_1, \dots, Z_N are independent iff $F_{\mathbf{Z}}(z_1, \dots, z_N) = F_{Z_1}(z_1) \cdot \dots \cdot F_{Z_N}(z_N)$. The probability density function and the expected value can also be factorized.*

Identically and independent distributed variables (iid) are independent, and the marginal PDF for each random variable is the same function. If two random variables are independent the *covariance* is zero.

Definition 15 (Covariance). *The covariance of two random variables Z_1 and Z_2 is given as*

$$\text{cov}(Z_1, Z_2) = E((Z_1 - E(Z_1))(Z_2 - E(Z_2))). \quad (4.4)$$

The covariance matrix \mathbf{C} for a vector $\mathbf{Z} \in \mathbb{R}^N$ is defined as $\mathbf{C}_{i,j} = \text{cov}(Z_i, Z_j)$ for all $i = 1, \dots, N, j = 1, \dots, N$.

If the covariance of two random variables is zero, they are *uncorrelated* but not necessarily independent.

Later, the operational uncertainties will be modeled using a normal distribution and the geometrical uncertainties will be modeled by means of a Gaussian random field, which is why some definitions based on the normal distribution are of importance.

The distribution of a random vector $\mathbf{Z} \in \mathbb{R}^N$ is referred to as the *multivariate normal distribution* if any linear combination of the components of \mathbf{Z} is normally distributed. Its probability density function can be expressed in terms of the covariance matrix \mathbf{C} and the vector of expected values $\boldsymbol{\mu}$ as

$$f_{\mathbf{Z}}(\mathbf{z}; \boldsymbol{\mu}, \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right). \quad (4.5)$$

For $N = 1$, one obtains the *univariate normal distribution* $f_Z(z; \mu, \sigma^2)$. The notation that is often used in this context is $z \sim N(\mu, \sigma^2)$. The *standard normal distribution* uses $\mu = 0$ and $\sigma = 1$. A nice property is that the normal distribution of an affine linear combination of independent normally distributed random variables is normally distributed with the linear combinations of the mean and the variance. This means that a Gaussian random variable can be reduced to a linear combination of the mean and the variance, such that $Z = \mu + \sigma Y$ where Y is standard normally distributed. Another frequently distribution function is the uniform distribution with the probability density function

$$f_Z(z; a, b) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b, \\ 0 & \text{for } x < a \text{ or } x > b. \end{cases} \quad (4.6)$$

The notation $z \sim U(a, b)$ denotes that z is uniformly distributed.

For second-order random processes ($E(Z_t^2) < \infty$), one can define functions for the expected value and the covariance, i.e., $\mu(t) = E(Z_t)$ and $C(t, s) = \text{cov}(Z_t, Z_s)$. One may talk of a stationary random process if the randomness is stationary which means that two random vectors that are shifted in time have the same distribution. It follows that $\mu(t)$ is constant and $C(t, s) = C(t - s)$.

Definition 16 (Gaussian Process). *If all collections of finite-dimensional vectors \mathbf{Z}_t of random variables of a continuous random process have a multivariate normal distribution $\mathbf{Z}_t \sim N(\boldsymbol{\mu}(t), \mathbf{C}(t))$, it is called a Gaussian process.*

4.2. Uncertainty Quantification

The aim of uncertainty quantification is to quantify the uncertainties in the input and the response (or output) of a specific model. Next to that, the aim is to identify uncertainties and to reduce them.

4. Robustness in Multi-Objective Optimization

One distinguishes between two types of uncertainties. *Aleatoric uncertainties* cannot be reduced by further improving a model with experiments or simply knowledge. They are inherent to a given problem and may, for example, occur as input uncertainties in initial conditions or boundary conditions of a PDE. In the context of aerodynamics, this comprises uncertainties in the operational conditions or geometrical uncertainties. Geometrical uncertainties may, for example, be present due to icing on the wing. *Epistemic uncertainties*, on the opposite, arise out of simplified modeling assumptions or missing aspects in the models. This may involve modeling errors due to approximations or resolutions of a numerical model, input uncertainty due to missing knowledge in the modeling parameters, or numerical uncertainties due to numerical errors. The present work focuses on aleatoric uncertainties.

There exist different strategies for representing uncertainties and for propagating them through a model. The complexity of a simulation often requires the use of specific modeling approaches, for example, stochastic spectral methods or sparse grid methods, in this context. In this thesis, it is made use of non-intrusive polynomial chaos for uncertainty quantification, and the computational costs are reduced by using dimension-adaptive sparse grid techniques. In Section 4.2.1, two strategies for representing uncertainties are presented. Different methods for uncertainty propagation are given in Section 4.2.2, focusing on stochastic spectral methods. Finally, the need for suitable integration strategies in stochastic spectral methods requires the introduction of quadrature techniques in Section 4.2.3.

The topic of inverse uncertainty quantification is also an interesting aspect of uncertainty quantification. It deals with the question of parameter estimation given noisy data, which can be treated using a frequentist point of view, where the unknown parameter is fixed, or a Bayesian point of view, where the unknown parameter is assumed to be a random variable. This is, for example, done when modeling the error term in Kriging (see Section 2.3.3.2). This topic will be omitted in the following discussions, restricting the present work to direct problems.

4.2.1. Representation of Random Inputs

In the following, it will be explained how to decorrelate random variables with the Karhunen-Loève (KL) expansion and how to represent known distributions using Generalized Polynomial Chaos (GPC). The general idea is to approximate random processes with finite-dimensional expansions. The main difference between KL and GPC is that for KL, the basis functions are generated, while for GPC, they are assumed to be known for a given distribution. In both representations, the basis functions depend on the random variables, and the coefficients do not have a random dependency, such that the deterministic and the stochastic part are separated. The KL expansion is applied in Section 4.4 for representing the random field that is used to model geometrical uncertainties.

4.2.1.1. Karhunen-Loève Expansion

The *Karhunen-Loève expansion* [141, 170] is used as a finite-dimensional representation of random fields and for decorrelating random variables. For finite dimensions, it is simply a Singular Value Decomposition. Let $\psi(t, \omega)$ be a random field with mean $\psi_0(t)$ and covariance function $C(t, s)$, where $t, s \in \mathcal{T}$ may be spatial or temporal variables and $\omega \in \Omega$ is a random event. Furthermore, let $\mathbf{Z} = (Z_1(\omega), \dots, Z_N(\omega))^\top$ be a vector of random variables. By definition C is a symmetric positive definite operator. Since it is continuous and bounded, Mercer's theorem [186] gives the spectral decomposition

$$C(t, s) = \sum_{j=1}^{\infty} \lambda_j v_j(t) v_j(s).$$

The orthonormal eigenfunctions v_j of the operator C and the corresponding eigenvalues λ_j can be obtained from

$$\int_{\mathcal{T}} C(t, s) v_j(s) ds = \lambda_j v_j(t). \quad (4.7)$$

The eigenvalues are real and non-negative. In the following, they are sorted in decreasing order, i.e., $\lambda_1 \geq \lambda_2 \geq \dots$. The eigenfunctions form a complete orthonormal basis, such that the KL expansion of ψ can be expressed as an infinite linear combination of the orthonormal eigenfunctions. The spectral expansion reads

$$\psi(t, \omega) = \psi_0(t) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} v_i(t) Z_i(\omega), \quad (4.8)$$

and the uncorrelated random variable Z_i is described by

$$Z_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_{\mathcal{T}} (\psi(t, \omega) - \psi_0(t)) v_i(t) dt. \quad (4.9)$$

The KL expansion is truncated to get an approximation of the random field in the form of a finite-dimensional random space \mathbf{Z} with uncorrelated random variables with zero mean and unit variance. In the case of a Gaussian field, the property that the variables are uncorrelated implies mutual independence of the random variables. One denotes the truncated expansion of ψ at the K -th term as ψ_K , such that

$$\psi_K(t, \omega) = \psi_0(t) + \sum_{i=1}^K \sqrt{\lambda_i} v_i(t) Z_i(\omega). \quad (4.10)$$

The basis v_i is optimal such that the mean square error of the truncation after the last term $E \left(\|\psi - \psi_K\|_{L_2(\mathcal{T})}^2 \right)$ is minimized (see [78]). The approximation error can be obtained as

$$E \left(\|\psi - \psi_K\|_{L_2(\mathcal{T})}^2 \right) = \sum_{i=K+1}^{\infty} \lambda_i.$$

4. Robustness in Multi-Objective Optimization

Thus, the important properties for a good approximation are a good choice of K and a fast decay of eigenvalues towards zero. The rate of decay depends on the smoothness of the covariance matrix and is subexponential for a Gaussian covariance function (see Equation (4.64)) [246].

4.2.1.2. Generalized Polynomial Chaos

The original *polynomial chaos* was introduced by Wiener [285] as homogeneous chaos using Hermite polynomials and Gaussian random variables. Convergence was proven in [29] for second-order random processes, i.e., processes with a finite variance. The polynomial chaos approach and its theory were generalized (GPC) in [78], where GPC was applied together with a KL expansion in solid mechanics. Polynomial chaos stands for a spectral expansion with specific basis functions $\phi_i : \mathbb{R} \rightarrow \mathbb{R}$. The GPC basis functions ϕ_i for a single random variable Z with a PDF f_Z all satisfy

$$E(\phi_i(Z)\phi_j(Z)) = \langle \phi_i(Z), \phi_j(Z) \rangle_{f_Z} = \gamma_i \delta_{ij}, \quad (4.11)$$

i.e., they are orthogonal under a given probability density function f_Z , and $\phi_0(Z) = 1$. Here, the expression $\langle \cdot, \cdot \rangle_{f_Z}$ describes the expected value in the form of a weighted scalar product. For $Z \sim N(0, 1)$ these are, for example, the Hermite polynomials and for uniformly distributed $Z \sim U(-1, 1)$ the Legendre polynomials.

A GPC approximation for a function ψ with a PDF f_Z is the projection with respect to the L_2 -Norm, which means

$$\psi_K(Z) := \sum_{k=0}^K \hat{\psi}_k \phi_k(Z), \quad (4.12)$$

with coefficients

$$\hat{\psi}_k = \frac{1}{\gamma_k} \langle \psi, \phi_k \rangle_{f_Z}. \quad (4.13)$$

Note that ψ_K is a truncation of the infinite-dimensional expansion

$$\psi(Z) = \sum_{k=0}^{\infty} \hat{\psi}_k \phi_k(Z). \quad (4.14)$$

In multi-dimensional cases the approximation can also be written as before with the basis functions $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}$ and a one-to-one correspondence between functions and coefficients of

$$\psi(\mathbf{Z}) = \sum_{k=0}^{\infty} \hat{\psi}_k \Phi_k(\mathbf{Z}) \quad (4.15)$$

and the summation

$$\psi(\mathbf{Z}) = \hat{\psi}_0 I_0 + \sum_{i_1=1}^{\infty} \hat{\psi}_{i_1} I_1(Z_{i_1}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \hat{\psi}_{i_1, i_2} I_2(Z_{i_1}, Z_{i_2}) + \dots, \quad (4.16)$$

where $I_p(Z_{i_1}, \dots, Z_{i_p})$ are interaction terms representing the polynomial chaos of order p . A truncation using N random variables with p summation terms, where p is the highest order of polynomials I_p , corresponds to a summation over

$$K := ((N + p)!)/(N!p!) - 1$$

terms in (4.15). Alternatively, (4.15) can be expressed as

$$\psi(\mathbf{Z}) = \sum_{|\mathbf{k}|=0}^{\infty} \hat{\psi}_{\mathbf{k}} \Phi_{\mathbf{k}}(\mathbf{Z}). \quad (4.17)$$

Here, \mathbf{k} denotes a N -dimensional multi-index with $|\mathbf{k}| = \sum_{i=1}^N |k_i|$, such that the multi-variate basis function is $\Phi_{\mathbf{k}}(\mathbf{Z}) = \prod_{i=1}^N \phi_{k_i}(Z_i)$. The orthogonality assumption (4.11) holds analogously with $\gamma_{\mathbf{k}} = \prod_{i=1}^N \gamma_{k_i}$. In the following, one may stick to the numbering in (4.15) and denote the approximation as ψ_K with a number of K summation terms.

The GPC expansion converges exponentially for an optimal basis (e.g. Hermite polynomials for Gaussian processes) [292], which is denoted as spectral convergence. In general, the convergence rate depends on the smoothness of the data in random space. If a function is non-smooth, oscillations may occur in the expansion, and a slow-down in the convergence rate may be observed (see e.g. [86]).

A major advantage of the GPC expansion is that statistical quantities like the mean and variance can be determined from the coefficients, i.e., $E(\psi_K) = \hat{\psi}_0$ and $\text{var}(\psi_K) = \sum_{k=1}^K \hat{\psi}_k^2 \gamma_k$. This is due to the choice of the GPC polynomials, which form a complete orthogonal basis, such that, for example, for the expectation value one obtains

$$E(\psi_K) = \sum_{k=0}^K \hat{\psi}_k E(\Phi_k(\mathbf{Z})) = \sum_{k=0}^K \hat{\psi}_k E(\Phi_k(\mathbf{Z})\Phi_0(\mathbf{Z})) = \hat{\psi}_0. \quad (4.18)$$

For the variance, when using the result for $E(\psi_K)$, one obtains

$$\begin{aligned} \text{var}(\psi_K) &= E(\psi_K^2) - E(\psi_K)^2 = E\left[\left(\sum_{k=0}^K \hat{\psi}_k \Phi_k(\mathbf{Z})\right)^2\right] - \hat{\psi}_0^2 \\ &= E\left[\sum_{k=0}^K \hat{\psi}_k^2 \Phi_k(\mathbf{Z})^2 + 2 \sum_{i=1}^K \sum_{j=0}^{K-i} \hat{\psi}_j \Phi_j(\mathbf{Z}) \hat{\psi}_{j+i} \Phi_{j+i}(\mathbf{Z})\right] - \hat{\psi}_0^2 \\ &= \sum_{k=1}^K \hat{\psi}_k^2 E(\Phi_k(\mathbf{Z})^2) = \sum_{k=1}^K \hat{\psi}_k^2 \gamma_k. \end{aligned} \quad (4.19)$$

The second summation in the derivation vanishes due to the orthogonality of polynomials. Note that for $\mathbf{Z} \in \mathbb{R}^N$ the values for γ_k are given by $\langle \Phi_k(\mathbf{Z}), \Phi_k(\mathbf{Z}) \rangle_{f_{\mathbf{Z}}}$.

4.2.2. Methods for Uncertainty Propagation

Given an uncertain input with a known probability distribution, the task is to propagate uncertainties through the model. One may distinguish between three types of methods for uncertainty propagation: sampling methods, perturbation methods, and spectral methods. In this thesis, the focus is on spectral methods, and it is made use of a non-intrusive form of spectral methods, also referred to as the pseudo-spectral approach. In the following, the different propagation methods are described.

4.2.2.1. Sampling Methods

The most prominent sampling method is *Monte Carlo sampling*. The general idea of sampling strategies is to select finitely many iid input parameters and calculate the quantities of interest from the output variables. An estimator for the mean in Monte Carlo sampling, e.g., is

$$E(\psi_M) = \frac{1}{M} \sum_{i=1}^M \psi(\mathbf{z}^i) \quad (4.20)$$

and the variance is estimated by means of the formula

$$var(\psi_M) = \frac{1}{M-1} \sum_{i=1}^M (\psi(\mathbf{z}^i) - E(\psi_M))^2 \quad (4.21)$$

for a number of M experiments with realizations \mathbf{z}^i of the input random variable \mathbf{Z} . The method converges due to the central limit theorem (see e.g. [251], p.86) with a rate of $M^{-1/2}$, which is a slow convergence rate, but it is independent of the number of random input parameters. A significant advantage is the possibility of parallelization, which makes the whole method independent from the number of parameters. There exist different approaches to sample from a known distribution. A computer can generate uniform pseudo-random sequences in the interval $[0, 1]$, which is made use of in strategies for sampling non-uniform or non-Gaussian parameters.

There exist alternative sampling methods that have faster convergence rates, e.g., Latin Hypercube sampling or quasi-Monte Carlo methods.

4.2.2.2. Perturbation Methods

In *perturbation methods* (see for example [219]), the random output is expanded in terms of a Taylor series using the sensitivity information of the output with respect to the random input parameters. A first-order expansion results in a first-order perturbation method, a second-order expansion in a second-order method. Sometimes perturbation methods are referred to as second-moment method or methods of moments since the first and second moments can be expressed using the expansion. In this case, the terms *first-order second moment* (FOSM) method or *second-order second moment* (SOSM) method are used. Expressions for statistical quantities of the output like the mean

value and the variance can be derived from the sensitivities and the statistical quantities of the random input variables (including covariances in the case of dependence). The underlying assumption is that the density for each random input variable is symmetric around a mean value. As a result, each random input variable can be represented as $Z_i = E(Z_i) + \delta Z_i$ for $i = 1, \dots, N$, where the perturbation δZ_i could, for example, be represented by the standard deviation σ_i in the case of normally distributed random variables.

For a first-order expansion of a function ψ and $\boldsymbol{\mu} \in \mathbb{R}^N$ with $\mu_i = E(Z_i)$, such that

$$\psi(\mathbf{Z}) = \psi(\boldsymbol{\mu}) + \sum_{i=1}^N \frac{\partial \psi}{\partial Z_i}(\boldsymbol{\mu}) \delta Z_i + \mathcal{O}(\delta Z_i^2),$$

one obtains, for example,

$$E(\psi(\mathbf{Z})) \approx \psi(\boldsymbol{\mu}) \quad (4.22)$$

and for a given variance $\text{var}(Z_i)$ and covariance $\text{cov}(Z_i, Z_j)$ the variance of the function ψ can be expressed as

$$\text{var}(\psi(\mathbf{Z})) \approx \sum_{i=1}^N \left(\frac{\partial \psi}{\partial Z_i}(\boldsymbol{\mu}) \right)^2 \text{var}(Z_i) + \sum_{i=1}^N \sum_{j=1, j \neq i}^N \frac{\partial \psi}{\partial Z_i}(\boldsymbol{\mu}) \frac{\partial \psi}{\partial Z_j}(\boldsymbol{\mu}) \text{cov}(Z_i, Z_j). \quad (4.23)$$

A significant aspect of perturbation methods is that the resulting approximation is only locally defined for small variations δZ_i since this is an assumption for the Taylor series expansion.

The efficiency of algorithmic differentiation can be exploited in this context, which was, for example, done by Su and Renaud for robust design [257].

4.2.2.3. Stochastic Spectral Methods

Stochastic spectral methods are based on the spectral expansion introduced in Section 4.2.1. Thus, in the following, one assumes second-order random variables and processes. General stochastic spectral methods require either mutually independent parameters or a representation of the joint density of the random input parameters. In comparison to Monte-Carlo sampling exponential convergence can potentially be observed for spectral methods. But they suffer from the curse of dimensionality. As a result, they are only feasible for a moderate number of random variables. The recommended maximum number of random variables is problem-dependent, but, usually, up to ten variables are realistic. One distinguishes between intrusive and non-intrusive spectral methods. In this context, the term intrusive means that an existing solution method has to be modified. The intrusive stochastic Galerkin method and the non-intrusive polynomial chaos method will be introduced in the following. Additionally, stochastic collocation, which is based on the GPC approach as well, is presented. In contrast to the non-intrusive polynomial chaos method, the stochastic collocation method is built using interpolation instead of projection.

4. Robustness in Multi-Objective Optimization

Stochastic Galerkin Method Given a PDE $c(y, t, \mathbf{Z}) = 0$ with the solution variable y , spatial and/or temporal variables t and the vector of random variables \mathbf{Z} , the *stochastic Galerkin method* is based on a spectral ansatz (4.12) for $y(t, \mathbf{Z})$, which is

$$y_K(t, \mathbf{Z}) = \sum_{k=0}^K \hat{y}_k(t) \Phi_k(\mathbf{Z}). \quad (4.24)$$

The ansatz uses random basis functions and deterministic coefficients. When the solution y is vector-valued the ansatz is made for all components individually. This ansatz is inserted into the PDE and the residual is projected onto the ansatz space, such that

$$\langle c(y_K, t, \mathbf{Z}), \Phi_k \rangle_{f_{\mathbf{Z}}} = 0, \quad \text{for } k = 0, \dots, K. \quad (4.25)$$

As can be seen in the equation, in contrast to standard Galerkin methods the projection is done using the expected value. As a result one obtains a system of new deterministic equations. The system can often be simplified by exploiting the orthogonality of the polynomials. For example, applying the stochastic Galerkin method to a scalar random ODE results in a system of deterministic ODEs for the expansion coefficients.

As explained before in Chapter 4.2.1, the expectation value and the variance can directly be identified from the expansion coefficients and the use of a polynomial chaos expansion guarantees for an optimal projection of residuals onto the space of approximating polynomials in a L_2 sense. Furthermore, depending on the smoothness of the underlying problem, spectral convergence may be observed, if the integrals over orthogonal polynomials are integrated exactly (see e.g. [291] for steady-state diffusion problems). Following [287], spectral convergence may also be observed for sufficiently smooth functions, if the order of quadrature is high enough. For discontinuities in random space, one may observe the typical behavior of the spectral expansion, i.e. a slower convergence. In such cases, locally defined expansions like the use of multi-element GPC [281] can be advantageous for convergence.

A disadvantage is that the stochastic Galerkin method is intrusive. On top of that, the original solution procedure or even parts of it cannot necessarily be reused when applying the methodology. Additionally, when using multi-variate inputs, one runs into the curse of dimensionality either because the system becomes too large or the expansion coefficients are high-dimensional integrals. Furthermore, for applying the stochastic Galerkin method, it is required that the random variables are independent with a known distribution for which orthogonal polynomials can be defined. Extensions to multivariate distributions can be made (see e.g. [203]) with restrictions on the dimensionality of the problem and the uniqueness of the basis.

The method was applied for different probability distributions in [292] and to the Navier-Stokes-equations in [293].

Stochastic Collocation Method The idea of general collocation methods is quite simple. To stay non-intrusive one uses deterministic or stochastic methods to generate samples \mathbf{z}^m for $m = 1, \dots, M$ from random space. To obtain an expression for the

output, (4.24) shall hold for all collocation points, such that

$$\sum_{k=0}^K \hat{y}_k(t) \Phi_k(\mathbf{z}^m) = y(t, \mathbf{z}^m). \quad (4.26)$$

As a result a system has to be solved to find the coefficients $\hat{y}_k(t)$ for $k = 0, \dots, K$. It may be of size $K + 1$ (see e.g. [123]) or overdetermined leading to a linear least squares problem [122]. This approach is often referred to as *point collocation*. Depending on the choice of collocation points, the system matrix may be ill-conditioned.

In the simplest case with $K + 1$ collocation points, where the chosen polynomials are Lagrange polynomials such that $\Phi_k(\mathbf{z}^m) \equiv L_k(\mathbf{z}^m) = \delta_{km}$, one can identify $\hat{y}_m(t) = y(t, \mathbf{z}^m)$. This is equivalent to employing Lagrange polynomials in the formulation (4.25) of the stochastic Galerkin method and using the quadrature points, which are used for the approximation of the resulting integrals, as collocation points. As a result, the underlying equations are decoupled and can be solved for each collocation point on a grid. Then the random variable y is constructed using Lagrange interpolation, such that

$$y(t, \mathbf{Z}) \approx \sum_{m=0}^K L_m(\mathbf{Z}) y(t, \mathbf{z}^m). \quad (4.27)$$

Quantities like the mean and the variance are calculated by integration of the interpolated output on the grid assuming a probability distribution for the input uncertainty is given, such that, for example,

$$E(y(t, \mathbf{Z})) \approx \sum_{m=0}^K y(t, \mathbf{z}^m) \int L_m(\mathbf{z}) f_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z}. \quad (4.28)$$

For complicated probability density functions, the integral over the Lagrange basis has to be integrated using quadrature techniques or approximated using, e.g., sampling techniques if a direct representation of the PDF is not given. If the collocation points were chosen as quadrature points, the integral reduces to the corresponding quadrature weight offset by the PDF, since $L_m(\mathbf{z}^q) = \delta_{mq}$. The methodology is referred to as *stochastic collocation* and was first proposed in [179, 180] and further analyzed and generalized in [12] and [173].

The stochastic collocation method is non-intrusive by construction, which is a major advantage in comparison to the stochastic Galerkin method, as the underlying problem can be used in a black-box manner. Additionally, the evaluations in random and deterministic space can be decoupled, such that the efficiency in comparison to the Stochastic Galerkin method may be increased. Opposed to the stochastic Galerkin method and the non-intrusive polynomial chaos method, stochastic collocation is an interpolation method and not a projection method. Thus, it neither requires the independence of random variables nor the construction of orthogonal polynomials. As a result, any kind of distribution is compatible, although a description of a density function is required for obtaining the quantities of interest.

4. Robustness in Multi-Objective Optimization

The problem of using an interpolation method instead of a projection method is that orthogonality assumptions are, in general, not valid outside of the set of collocation points. Thus, simplifications for the integral quantities like (4.18) and (4.19) cannot be used. Also, the stochastic collocation method requires the construction of Lagrange polynomials. Furthermore, it needs an appropriate choice of collocation points to guarantee a certain accuracy since, in general, the choice of collocation points is not unique. The optimal choice of collocation points in terms of accuracy is to use the zeros of the orthogonal polynomial of respective order of the underlying distribution, thus corresponding to the quadrature points for Gauss quadrature rules. This applies especially to the accuracy of the integral quantities like the expected value since one makes use of exact evaluations at the quadrature points. The convergence behavior of stochastic collocation is based on interpolation theory. The interpolation error for K collocation points in N -dimensional random space behaves like $\mathcal{O}(K^{-\alpha/N})$ for functions with bounded derivatives up to the order α , i.e., $f \in C^\alpha(\Omega)$ with Ω being the domain of interpolation. The method is less accurate than the stochastic Galerkin approach since it introduces an additional interpolation error. In contrast to the Monte Carlo method, stochastic collocation suffers from the curse of dimensionality and is only feasible for moderate dimensions in random space. Since the interpolation polynomials are globally defined in random space, discontinuities in random space may again decrease the accuracy and the efficiency of the method. Some approaches like the use of sparse grid interpolation methods based on locally defined basis function [174] or different interpolation techniques, as, for example, in simplex stochastic collocation [286], can enhance the overall efficiency of the method.

An application example for stochastic collocation may be found in [172] for airfoil analysis. In [21], a sparse grid collocation approach is used for robust optimal control.

Non-Intrusive Polynomial Chaos Method The *non-intrusive polynomial chaos method* is often also referred to as a discrete projection or pseudo-spectral method. As for the stochastic Galerkin method, it is required that the PDF of every independent input random variable is known, and an orthogonal polynomial system is constructed for each one to get a basis for the multi-dimensional input. The solution y is again expanded in terms of the constructed orthogonal polynomials. Then a quadrature rule is employed to approximate the Fourier coefficients $\hat{y}_k(t)$ with the known formula

$$\hat{y}_k(t) = \frac{1}{\gamma_k} \sum_{r=1}^R y(t, \mathbf{z}^r) \Phi_k(\mathbf{z}^r) f_{\mathbf{Z}}(\mathbf{z}^r) w^r, \quad (4.29)$$

where \mathbf{z}^r are the quadrature points and w^r the quadrature weights of a N -dimensional quadrature rule and R is the total number of quadrature points. Matching quadrature rules can be chosen for integration weighted by the PDF (e.g. Gauss-Legendre quadrature for a uniform distribution or a transformed Gauss-Hermite quadrature for a normal distribution), such that the evaluation $f_{\mathbf{Z}}(\mathbf{z}^r)$ in (4.29) is omitted. The quantities of interest can be obtained from the expressions (4.18) and (4.19) by using the orthogonality of polynomials. Note that the expansion coefficients may also be approximated using different techniques than quadrature rules, e.g., random sampling. The approach

using quadrature methods is described and analyzed in [289]. When employing Lagrange polynomials as basis functions with unit density and weights, the method is equivalent to stochastic collocation. Note that in some literature (e.g. [290]), the non-intrusive polynomial chaos approach is seen as an extension of stochastic collocation methods to the ideas of GPC. As a result, it is sometimes referred to as a collocation method with quadrature points as collocation points. Vice versa, point collocation methods based on the polynomials of the PC expansion are sometimes described as non-intrusive polynomial chaos methods.

As the method is non-intrusive, the original PDE solver can be used as a black-box. Since the deterministic and the stochastic part are fully decoupled, the evaluation at different quadrature points can be easily parallelized. As a result, the efficiency in comparison to the intrusive approach is potentially increased.

Additionally to the error introduced by the series truncation, a quadrature error is present [289]. Similar to the interpolation error, one observes for a tensorized grid an error of $\mathcal{O}(R^{-\alpha/N})$ for functions with bounded derivatives up to the order α . Thus, if the functions are smooth, exponential convergence can potentially be achieved. For an exact integration, the method inherits the convergence properties of the GPC expansion. As it is the case for stochastic collocation, especially for larger dimensional random spaces, the non-intrusive polynomial chaos method usually needs more degrees of freedom than the intrusive approach to achieve a comparable accuracy. Sparse grid quadrature rules (see Section 4.2.3) can be used to increase the overall efficiency.

An early application of non-intrusive polynomial chaos can be found in [161]. A comparison to stochastic collocation can be found in [58] and in [228]. The computational effort and the accuracy of both methods are comparable.

In the present work, the non-intrusive approach is chosen to avoid a derivation and implementation of the solution procedure for the intrusive method. Furthermore, for cases where the calculation of expectation is of importance, the non-intrusive approach allows to compute only a single coefficient of the PC expansion. The non-intrusive polynomial chaos approach is preferred over stochastic collocation since the input density function is assumed to be known, and the construction of a suitable quadrature rule is straightforward.

4.2.3. Quadrature

Most of the above methods require quadrature or interpolation techniques. In the following, it will be focused on quadrature techniques. However, the methods introduced below can also be applied for interpolation using interpolation points instead of quadrature points and evaluations of interpolation functions instead of quadrature weights. Integrals can be solved with Monte-Carlo simulations, full tensor grid quadrature, or sparse grid quadrature.

4. Robustness in Multi-Objective Optimization

4.2.3.1. Tensor Product Quadrature

A full tensor grid quadrature rule is constructed from the tensor product of the one-dimensional quadrature rule

$$Q_i^{(1)}(f) = \sum_{r=1}^{R_i} f(q_r^i) w_r^i \quad (4.30)$$

for an integration of the function $f : \Omega_i \rightarrow \mathbb{R}$ with a level of integration $i \geq 1$ with nodes q_r^i and weights w_r^i . The set of nodes $\mathcal{Q}_i^{(1)} = \{q_1^i, \dots, q_{R_i}^i\}$ lies in Ω_i , and R_i indicates the number of quadrature nodes for the level i with $R_i < R_{i+1}$. A rule of dimension N is then built as

$$Q_{\mathbf{i}}^{(N)} f = (Q_{i_1}^{(1)} \otimes \dots \otimes Q_{i_N}^{(1)})(f) = \sum_{r_1=1}^{R_{i_1}} \dots \sum_{r_N=1}^{R_{i_N}} f(q_{r_1}^1, \dots, q_{r_N}^N) w_{r_1}^{i_1} \dots w_{r_N}^{i_N} \quad (4.31)$$

with the multi-index $\mathbf{i} = (i_1, \dots, i_N)$. The resulting set of nodes is given by

$$\mathcal{Q}_{\mathbf{i}}^{(N)} = \mathcal{Q}_{i_1}^{(1)} \times \dots \times \mathcal{Q}_{i_N}^{(1)}.$$

For one-dimensional quadrature rules (4.30), one commonly observes an error of $\mathcal{O}(R_i^{-\alpha})$ for functions with bounded derivatives up to the order α . The error in a N -dimensional quadrature rule then behaves like $\mathcal{O}(R^{-\alpha/N})$. Here, $R = \prod R_i$ is the total number of quadrature nodes. As a result, the growth in required nodes for achieving a particular error is exponential in the dimension N , which is the curse of dimensionality.

4.2.3.2. Sparse Grid Quadrature

Sparse grids were constructed to integrate polynomials of a certain total degree exactly with less nodes than the number of nodes needed for tensorized grids. Based on Smolyak [252], one can define

$$\Delta_i^{(1)} = Q_i^{(1)} - Q_{i-1}^{(1)}, \quad Q_0^{(1)}(f) = 0 \quad (4.32)$$

as a new operator for defining quadrature rules. The tensorized quadrature formula (4.31) of level $k \geq N$ can be rewritten as

$$Q_k^{(N)}(f) = \sum_{\mathbf{i} \in \mathbb{R}^N: \|\mathbf{i}\|_{\infty} \leq k} (\Delta_{i_1}^{(1)} \otimes \dots \otimes \Delta_{i_N}^{(1)})(f). \quad (4.33)$$

In comparison, the sparse quadrature formula of level k is given by

$$S_k^{(N)}(f) = \sum_{\mathbf{i} \in \mathbb{R}^N: \|\mathbf{i}\|_1 \leq k+N-1} (\Delta_{i_1}^{(1)} \otimes \dots \otimes \Delta_{i_N}^{(1)})(f) \quad (4.34)$$

and can be written recursively as

$$S_k^{(N)}(f) = S_{k-1}^{(N)}(f) + \sum_{\mathbf{i} \in \mathbb{R}^N: \|\mathbf{i}\|_1 = k+N-1} (\Delta_{i_1}^{(1)} \otimes \cdots \otimes \Delta_{i_N}^{(1)})(f). \quad (4.35)$$

Equation (4.34) can be equivalently written in terms of the original operator and using $q = k + N - 1$ [283] as

$$S_k^{(N)}(f) = \sum_{\mathbf{i} \in \mathbb{R}^N: q-N+1 \leq \|\mathbf{i}\|_1 \leq q} (-1)^{q-\|\mathbf{i}\|_1} \cdot \binom{N-1}{q-\|\mathbf{i}\|_1} \cdot (Q_{i_1}^{(1)} \otimes \cdots \otimes Q_{i_N}^{(1)})(f). \quad (4.36)$$

The set of grid points of the sparse grid is

$$\mathcal{Q}_k^{(N)} = \bigcup_{\mathbf{i} \in \mathbb{R}^N: \|\mathbf{i}\|_1 \leq k+N-1} \mathcal{Q}_{i_1}^{(1)} \times \cdots \times \mathcal{Q}_{i_N}^{(1)}.$$

Nested quadrature rules have to be used to enable a sparse construction with less function evaluations and allow for an efficient approximation of integrals. Nestedness means that the nodes in the lower-order formula are also used in the higher-order formula, i.e., $\mathcal{Q}_i^{(1)} \subset \mathcal{Q}_{i+1}^{(1)}$. As a result, one can reuse the function values of low orders and the set of quadrature points for a nested rule of level k can be expressed as

$$\mathcal{Q}_k^{(N)} = \bigcup_{\|\mathbf{i}\|_1 = k+N-1} \mathcal{Q}_{i_1}^{(1)} \times \cdots \times \mathcal{Q}_{i_N}^{(1)}.$$

Also, for nested rules $\Delta_i^{(1)}$ defines a quadrature rule where the nodes and the weights are the same as for $Q_i^{(1)}$, but instead of only evaluating the function itself the differences of the function values and the estimated values at the lower order rule are evaluated at the quadrature points. Due to the fact that the difference is zero for all quadrature points of the lower order rule, one obtains

$$\Delta_i^{(1)} = \sum_{x_i \in \mathcal{Q}_i^{(1)} \setminus \mathcal{Q}_{i-1}^{(1)}} w_{x_i} \left(f - Q_{i-1}^{(1)}(f) \right) (x_i). \quad (4.37)$$

Here, w_{x_i} denotes the weight associated to the quadrature point x_i . The difference in (4.37) is referred to as the one-dimensional hierarchical surplus and can be used as an error estimator. Based on the one-dimensional expression, a hierarchical strategy can be expressed for higher-dimensional rules (see e.g. [145]).

An example for a nested quadrature rule is the Clenshaw-Curtis rule [40], where the quadrature nodes for the integration domain $[0, 1]$ and a level of integration $i > 0$ are given by the transformed location of extrema of the Chebyshev polynomials as

$$q_r^i = \frac{1}{2} \left[1 - \cos \left(\frac{\pi(r-1)}{2^{i-1}} \right) \right] \quad \text{for } r = 1, \dots, 2^{i-1} + 1 \text{ and } i > 1 \quad (4.38)$$

4. Robustness in Multi-Objective Optimization

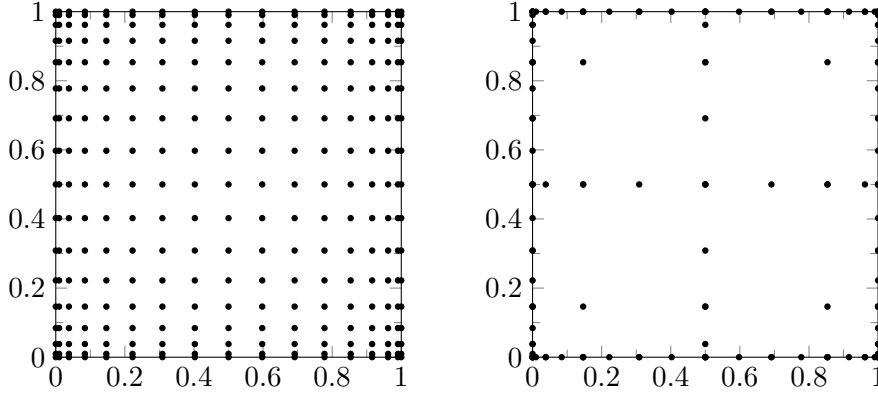


Figure 4.1.: Full tensor grid of Clenshaw-Curtis rule of level $k = 4$ (left) and sparse grid rule of level $k = 4$ (right)

and $q_1^1 = 0.5$. Figure 4.1 shows the points of a two-dimensional tensorized Clenshaw-Curtis rule of level $k = 4$ (left) and the corresponding points when using a sparse grid (right). The tensorized rule uses $17 \times 17 = 289$ points while the sparse grid rule only needs function values at 81 points.

The error for sparse grid quadrature using a rule $S(k, N)$ with a number of nodes R_i of the order 2^i is $O(R^{-\alpha}(\log_2 R)^{(N-1)(\alpha+1)})$ [283] for functions with bounded mixed derivatives up to the order α . This shows that the onset of the curse of dimensionality is delayed in comparison to tensorized grids.

4.2.3.3. Adaptive Sparse Grids

The grid used in the sparse grid rule (4.34) is isotropic since all directions are equally important. When, for example, using a KL decomposition for a stochastic process, the dimensions are not equally important. Thus, it is more efficient to use adaptive strategies to further reduce the number of grid points.

In general, the refinement might be dimension-adaptive or spatially adaptive. A spatially adaptive strategy using the hierarchical surplus as an error indicator can, for example, be found in [174]. An intuitive way to define anisotropic sparse grids for adaptivity in the dimension is to choose an index set $\{\mathbf{i} \in \mathbb{R}^N \mid \mathbf{a}^\top \mathbf{i} \leq k + N - 1\}$ instead of $\{\mathbf{i} \in \mathbb{R}^N \mid \mathbf{1}^\top \mathbf{i} \leq k + N - 1\}$. The vector $\mathbf{a} \in \mathbb{R}^N$ gives a direction in index space and has to be chosen according to the analysis of the underlying problem.

With the dimension-adaptive strategy proposed in [77], generalized sparse grids can be produced. The important dimensions are identified with the help of error estimators to refine the grid in these dimensions. The method can be understood as a sparse grid approach with a generalized admissible index set \mathcal{I} , such that the quadrature formula reads

$$\tilde{Q}_{\mathcal{I}}^{(N)} f = \sum_{\mathbf{i} \in \mathcal{I}} (\Delta_{i_1}^{(1)} \otimes \cdots \otimes \Delta_{i_N}^{(1)})(f). \quad (4.39)$$

The generalized formula includes the formula for the full tensor grid with

$$\mathcal{I}^t = \{\mathbf{i} \in \mathbb{R}^N \mid \|\mathbf{i}\|_\infty \leq k\},$$

as well as the formula for the original sparse grid with

$$\mathcal{I}^s = \{\mathbf{i} \in \mathbb{R}^N \mid \|\mathbf{i}\|_1 \leq k + N - 1\}.$$

An error indicator of an index $\mathbf{i} \in \mathcal{I}$ can, for example, be computed from the term $(\Delta_{i_1}^{(1)} \otimes \cdots \otimes \Delta_{i_N}^{(1)})(f)$ [77]. It estimates the correction that is obtained for the integral when adding an index. Starting from the coarsest sparse grid, if the index sets remains admissible, the algorithm adaptively adds the index with the maximum error indicator. An index set \mathcal{I} is admissible, if for all elements $\mathbf{i} \in \mathcal{I}$ the condition

$$\mathbf{i} - \mathbf{e}_j \in \mathcal{I} \quad \forall 1 \leq j \leq N \quad (4.40)$$

holds. As a result, index sets are only admissible if for all indices of the set the indices with lower entries in at least one dimension are also contained in the index set.

An advantage of the dimension-adaptive strategy based on generalized index sets is that it enables the use of problem-dependent quadrature formulas like the Gauss-Hermite formula. In Schillings et al. [235], the dimension-adaptive approach is applied for the calculation of the expected value of the drag coefficient under the consideration of geometrical uncertainties.

4.3. Robust Design

Robust design, sometimes also referred to as *robust optimal design* or *robust design optimization*, is mainly used in the context of single-objective optimization problems. A design is *robust optimal* if it is optimal and robust under perturbations. A design is *robust* if it is not very sensitive under perturbations, or, in the best case, insensitive. A basic modeling aspect in robust design is how to measure robustness and how to achieve robust optimality.

There exist different formulations for robust design that can be distinguished based on the definition of robustness itself. In Section 4.3.1, some formulations for single-objective optimization problems are introduced. Multi-objective optimization problems are of interest in Section 4.3.2. The specific section presents existing formulations and points out the differences to single-objective optimization problems, as well as the necessity for a measure that accounts for the structure of uncertainty in objective space. In Section 4.3.3, a new robustness measure is introduced, and different corresponding formulations of robust design problems are presented.

Note that robust design is different from reliability-based design, but both can be combined. *Reliability-based design* aims at finding reliable designs with a significantly low failure rate, which is usually measured in terms of probabilities of constraint satisfaction. As a result, reliability-based design is concerned with avoiding extreme cases of uncertain events that may lead to failure of a design. In contrast, robust design aims at reducing performance losses for small perturbations.

4.3.1. Robust Design in a Single-Objective Context

The first ideas for robust design originated from Taguchi's work [265, 266]. He introduced the concept of optimizing the expected value of a quality loss function. Originally, the quality loss was represented by the mean statistical variation found with the help of a design of experiments, and optimized by means of statistical data analysis. The work of Taguchi initiated the discussion of robust design in general and motivated the use of the mean and the variance of an objective function as measures for robustness for finding robust solutions and robust optimal solutions.

In early works for robust design, it was more common to use perturbation methods. However, the use of sophisticated strategies for uncertainty propagation, as presented in Section 4.2.2, has also become interesting for robust design.

In the following, PDE-constrained optimization problems of the form (3.1) are considered with additional uncertainties with outcomes $\omega \in \Omega$, which are represented by the vector of random variables $\mathbf{Z}(\omega) \in \mathbb{R}^N$ and may occur, for example, in the initial condition or parameters of the PDE constraint. In the following, for ease of notation, the additional constraints are expressed as the inequality constraint function g . The variable $\bar{\mathbf{z}}$, which is also referred to as the vector of *nominal values*, will be used when no uncertainties are present. One can classify robust design formulations according to the robustness measures used. In general, robustness measures can be included in the optimization procedure in various ways:

- By replacing the original objective function,
- as an additional constraint,
- or as an additional objective function leading to a MOO problem.

The following formulations focus on replacing the original objective function, but most of the formulations can be altered to represent any of the strategies mentioned above.

4.3.1.1. Worst-Case Robustness

The worst case of uncertain events for an objective function $f(\mathbf{y}_z, \mathbf{u}, \mathbf{Z}(\omega))$, which can be expressed as

$$\max_{\omega \in \Omega} f(\mathbf{y}_z, \mathbf{u}, \mathbf{Z}(\omega)), \quad (4.41)$$

can be used for measuring the robustness of a solution [219]. Each realization \mathbf{z} of $\mathbf{Z}(\omega)$ has an individual state \mathbf{y} , which is why the notation \mathbf{y}_z for the state variable is used. In the following, the subscript is omitted for the sake of clarity. Again, as done in Chapter 3, one assumes that the state constraint $c(\mathbf{y}, \mathbf{u}, \mathbf{Z}) = 0$ will implicitly define a state \mathbf{y} depending on the design \mathbf{u} and the random vector \mathbf{Z} . The *min-max formulation* is used for optimization in a worst-case scenario and is defined as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \max_{\omega \in \Omega} f(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \\ \text{s.t. } c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = \mathbf{0}, \quad g(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq \mathbf{0} \quad \forall \omega \in \Omega. \end{aligned} \quad (4.42)$$

As a probability measure is not needed, this approach is often employed if the probability density of the random variable is unknown, e.g., when only the ranges of variation of the uncertain parameters are given. The case of optimization when considering worst-case robustness is also referred to as *robust optimization* and is typically analyzed for specific set-based descriptions of uncertainty. The formulation (4.42) can be particularly interesting for situations in which a finite, discrete set of random events is considered. In general, the approach is quite conservative since an optimal design is found to reduce the worst-case behavior. As a result, the overall performance can be significantly worse than the performance without considering uncertainties. Note that in the formulation (4.42) constraint feasibility is required for each realization. For the robustness measure presented in the following section, different strategies for constraint feasibility are discussed, based on the classification in [244].

4.3.1.2. Robustness based on Expectation

The expectation or expected value

$$E(f(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))) = \int_{\mathbb{R}^N} f(\mathbf{y}, \mathbf{u}, \mathbf{z}) f_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} \quad (4.43)$$

of a function can be used for measuring robustness. As a result, the mean performance is optimized.

In the *semi-infinite formulation* (see also [129]) the expected value of the objective function is optimized while requiring feasibility for all random outcomes, such that the robust design problem reads

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \int_{\mathbf{Z}} f(\mathbf{y}, \mathbf{u}, \mathbf{z}) f_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = \mathbf{0}, \quad g(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq \mathbf{0} \quad \forall \omega \in \Omega. \end{aligned} \quad (4.44)$$

The formulation is referred to as semi-infinite as it is based on infinitely many inequality constraints. Thus, a direct treatment is only feasible for small problems with specific assumptions (see e.g. [66] for the solution of a problem with a single constraint and a one-dimensional bounded random variable).

When numerically approximating the expected value, e.g., with a quadrature rule with points \mathbf{z}^i and weights w^i as it is the case for the non-intrusive polynomial chaos methods (4.29), the problem may be expressed as a multiple set-point problem

$$\begin{aligned} \min_{\mathbf{y}^i, \mathbf{u}} \quad & \sum_{i=1}^R f(\mathbf{y}^i, \mathbf{u}, \mathbf{z}^i) w^i \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}^i, \mathbf{u}, \mathbf{z}^i) = \mathbf{0}, \quad g(\mathbf{y}^i, \mathbf{u}, \mathbf{z}^i) \leq \mathbf{0} \quad \forall i \in \{1, \dots, R\}. \end{aligned} \quad (4.45)$$

Such a problem is well-suited for parallelization, and, due to the numerical approximation, the constraints are only evaluated at a finite number of points. However, this way of approximating the semi-infinite formulation depends on the level of discretization of the random space. Alternatively, if this is possible, one may define finitely many local maximizers of the constraints g for each design \mathbf{u} during the optimization, and reformulate

4. Robustness in Multi-Objective Optimization

corresponding reduced problems based on defining constraints for all local maximizers (see e.g. [243]).

There exist alternatives to requiring constraint feasibility for every realization of the random variables. Instead, defining feasibility robustness can be of interest. Therefore, another approach is the *chance-constrained formulation*, which differs from the semi-infinite formulation in the treatment of the constraints. Here, the inequality constraints are more flexible such that they have to hold with a certain probability $P_0 \in (0, 1]$. They are referred to as chance constraints [32]. The corresponding formulation reads

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \int_{\mathbf{Z}} f(\mathbf{y}, \mathbf{u}, \mathbf{z}) f_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z} \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = \mathbf{0} \quad \forall \omega \in \Omega, \\ & P(\{\omega \in \Omega \mid g_i(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq \mathbf{0}\}) \geq P_0 \quad \text{for } i = 1, \dots, s. \end{aligned} \quad (4.46)$$

Chance constraints were originally proposed for reliability-based design. Thus, the incorporation of chance constraints results in a combination of robust design and reliability-based design strategies. It is also possible to formulate a joint chance constraint (see e.g. [1]). The constraints can be either evaluated by sampling or by approximating the cumulative distribution function F_{g_i} for $i = 1, \dots, s$ using standard propagation techniques. In the case of a scalar-valued constraint and when additionally assuming that the constraint is normally distributed or can be transformed appropriately, one may, for example, use the the moments μ_g and σ_g to describe the chance constraint. One obtains

$$\begin{aligned} P(\{\omega \in \Omega \mid g(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq 0\}) &= P(\{\tilde{Z} \mid \mu_g + \sigma_g \tilde{Z} \leq 0\}) \\ &= P(\{\tilde{Z} \mid \tilde{Z} \leq -\frac{\mu_g}{\sigma_g}\}) = F_{\tilde{Z}}(-\frac{\mu_g}{\sigma_g}), \end{aligned} \quad (4.47)$$

where $\tilde{Z} \sim N(0, 1)$ and $F_{\tilde{Z}}$ is the cumulative distribution function of the standard normal distribution. The inequality constraint can then be simplified by substituting (4.47) in (4.46) (see e.g. [219, 52]), such that

$$\mu_g + F_{\tilde{Z}}^{-1}(P_0)\sigma_g \leq 0.$$

This procedure is referred to as *moment matching method*.

In [224], the chance-constrained formulation is used in combination with a first-order perturbation method for the moments of the constraints and the objective function for a simple CFD application.

Other approaches for feasibility robustness include constraining the expected value of h or the evaluation of h at the nominal input, the corner space evaluation method [259] or most probable point-based importance sampling [52]. The worst-case analysis in [219] assumes that the perturbations are simultaneously at their worst, which can, for example, be represented by

$$h(\boldsymbol{\mu}) + \sum_{i=1}^N \left| \frac{\partial f}{\partial Z_i}(\boldsymbol{\mu}) \delta Z_i \right| \leq 0, \quad (4.48)$$

when using a perturbation method and assuming small variations δZ_i around the mean vector $\boldsymbol{\mu}$ of the random variables \mathbf{Z} . Reliability analysis approaches like the reliability index approach [37] or the performance measure approach [271] can also be integrated in robust design formulations (see e.g. [53]) but require additional computational effort due to an optimization subproblem.

4.3.1.3. Robustness based on Variance

The expected value reflects the mean behavior of a function. Thus, it does not account for reducing any variations. An optimal design using the expectation may, therefore, still result in large variations of function values from the mean. Therefore, using the variance or the standard deviation as a robustness measure can be of interest for robust design. Since only minimizing the variations in the objective function does not have any influence in the performance of the objective function, it is essential to include the robustness measure as a constraint or as an additional objective function to the deterministic problem.

4.3.1.4. Trading off Expectation and Variance

In contrast to using the expected value for measuring robustness, the use of the variance or the standard deviation σ as a robustness measure neglects the optimization for the overall expected performance concerning the objective function. Therefore, it is recommended to use both measures, i.e., the expected value and the standard deviation, for robust design formulations.

Commonly, both measures are concatenated in an additive way to form a single robustness measure that replaces the original measure, e.g., in the semi-infinite formulation. This leads to the robust design formulation

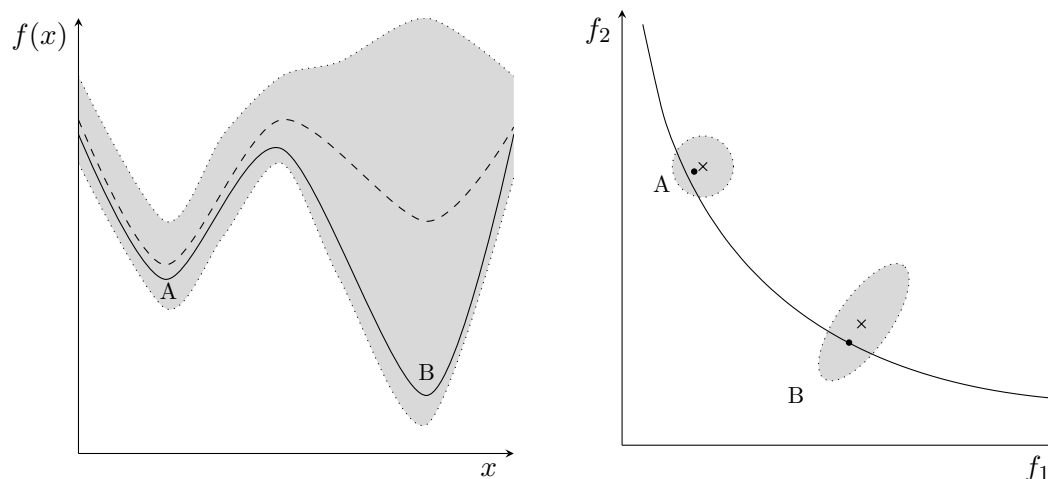
$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \mu_f + k_f \sigma_f \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = \mathbf{0}, \quad g(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq \mathbf{0} \quad \forall \omega \in \Omega, \end{aligned} \quad (4.49)$$

where μ_f and σ_f denote the expected value and the standard deviation of $f(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))$. The factor k_f can either be interpreted as a weight to the importance of the standard deviation or, if an assumption on the distribution function of f is given, can be chosen such that minimizing $\mu_f + k_f \sigma_f$ is equivalent to minimizing the worst case user-defined probability quantile threshold for the objective function or a similar measure like the conditional value at risk (compare [214]).

Since expectation and variance are commonly conflicting objectives, a more general strategy is to formulate a multi-objective problem that aims to minimize the expected value as well as the variance (or the standard deviation). This naturally leads to a robust design problem with multiple objectives leading to trade-off solutions for robustness, i.e.,

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & (E[f(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))], \text{var}[f(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))]) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = \mathbf{0}, \quad g(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \leq \mathbf{0} \quad \forall \omega \in \Omega. \end{aligned} \quad (4.50)$$

4. Robustness in Multi-Objective Optimization



(a) Robustness: Single-objective optimization. (b) Robustness: Multi-objective optimization.

Figure 4.2.: Robustness in single-objective optimization (a) and multi-objective optimization. In both figures design A is more robust than design B. In multi-objective optimization, uncertainty results in a combined effect in objective space.

Note, however, that the underlying formulation is still only considering the effect of uncertainties for a single performance measure and shall not be confused with robust design for multi-objective problems. Early works of this type are [36] and [35], where compromise programming is used to solve the multi-objective optimization problem. In [225], three objective functions, the objective function evaluated at the nominal value, the expected value and the variation, are minimized using a genetic algorithm.

4.3.2. Robust Design in a Multi-Objective Context

In a generalized form with an arbitrary number of objective functions, one can also speak of robust solutions as solutions that are not very sensitive under uncertainty. In this context, robust optimal solutions are solutions that fulfill a specific type of optimality (e.g. Pareto optimality) and are robust under the consideration of uncertainties.

The main difference between robustness in single-objective optimization context and multi-objective optimization context is depicted in Figure 4.2a and 4.2b.

In Figure 4.2a, the case of single-objective optimization is shown. The line indicates the objective function evaluated at the nominal values and the dashed line indicates the expected value of the objective functions for the different designs x . The dotted line shows the range of variations of the objective function that are obtained with a fixed level of probability P_0 for each design x . Judging from the expected value and the variations for a fixed level of probability, one may observe that point A is more robust than point B, and can be denoted as robust optimal.

Figure 4.2b shows the situation for multi-objective optimization with the help of two

exemplary solutions in objective space. The dot indicates the objective function evaluated at the nominal values and the cross indicates the expected value in objective space. The dotted line shows the contours of the set of random outcomes in objective space for a fixed level of probability P_0 , which one may refer to as *probability region*. Thus, the level of probability indicates that the probability of outcomes being observed in the given region in the multi-objective space is P_0 . The region of probability shows that the uncertainty has a combined effect on both objective functions. The Pareto optimal front that is obtained without considering any uncertainties is represented by the line. When looking only at the expected values, no dominance can be observed. Judging from the probability region, one may denote Point A as more robust than point B. Usually, similarly to MOO without considering uncertainties, the aim is to find a set of robust optimal points, i.e., there does not exist a unique, robust optimal solution. Thus, the difference to single-objective robust design is that one obtains a set of solutions in objective space and, also, one has to consider a combined effect in objective space [48].

Robust design in the context of multi-objective optimization also has its origin in Taguchi methods, which were extended to multiple objective functions by transforming the loss function into an aggregated loss function (see e.g. [59, 220, 270]). All of these strategies find robust design using a design of experiment, and thus remain purely statistical approaches instead of using numerical optimization. Nevertheless, they form a basis for general ideas on robust design.

The prior aggregation of the objective functions into a single-objective function (e.g. by using a weighted sum of objective functions) is a rather basic approach to multi-objective optimization under uncertainties. In general, the choice of a suitable aggregation strategy is unclear, and the multi-objective nature of the problem is disregarded. The aggregated objective function is simply treated using robust design techniques in the context of single-objective optimization.

Also, quantities of interest like the expected value or the variance of the objective functions can be aggregated into a single objective function. In [189, 187], for example, the authors use physical programming methods to trade off an aggregation function for the expected value and an aggregation function for the standard deviation. Hard constraints are treated using worst-case feasibility robustness.

In the following, existing strategies for introducing robustness and robust optimality in a multi-objective context are reviewed, focusing on their application in engineering design. First, a short overview of the formulation of robustness measures in multi-object evolutionary algorithms is given, and restrictions to these ideas are pointed out. It is then turned to approaches that are applicable in a more general setting. They include the sensitivity analysis in the context of multi-objective optimization and robustness measures inspired from single-objective robust design like the worst case of events or the use of quantities of interest like the expected value and the variance. Additional constraints are neglected in the multi-objective robust design formulations since, for most of the proposed strategies, they can be handled in the same way as in single-objective robust design.

4.3.2.1. Robustness in Multi-Objective Evolutionary Algorithm

The first approaches to combine robustness and Pareto optimality for finding multiple robust designs were made in the context of multi-objective evolutionary algorithms. The idea originally proposed in [269, 124] is to extend the concept of dominance to robust design by defining a probability of dominance. In the single-objective case, the probability that a design \mathbf{u}_A is better than a design \mathbf{u}_B can be found using the probability density functions f_A^{pr} and f_B^{pr} and the probabilities P_A and P_B based on the respective CDFs for describing the distribution of the objective function values f for the designs \mathbf{u}_A and \mathbf{u}_B . The probability that a design A is preferred over B for a minimization can then be described as

$$P(\{\omega \in \Omega \mid f(\mathbf{u}_A, \omega) \leq f(\mathbf{u}_B, \omega)\}) = \int_{-\infty}^{\infty} f_A^{pr}(z) P_B(\{\omega \in \Omega \mid f(\mathbf{u}_B, \omega) \geq z\}) dz. \quad (4.51)$$

For the multi-objective case, the idea is to assume independence and formulate the probability of dominance based on the marginal probability density functions as

$$P(\{\omega \in \Omega \mid \mathbf{u}_A \text{ dominates } \mathbf{u}_B\}) = \prod_{i=1}^k P(\{\omega \in \Omega \mid f_i(\mathbf{u}_A, \omega) \leq f_i(\mathbf{u}_B, \omega)\}). \quad (4.52)$$

When having an explicit formulation of the respective density function, the formulation can be used for Pareto ranking in evolutionary methods (see [269] for uniformly distributed objective functions and [124] for normally distributed objective functions). However, there are several restrictions to this strategy:

- If the density function is not known, it has to be sampled or approximated, which requires extensive computation effort.
- The concept of Pareto dominance is usually used in direct Pareto approaches like evolutionary algorithms. Scalarization approaches do not explicitly compare solutions based on this criterion since it is implicitly stated in the formulation itself.
- One assumes independence of the objective functions and does not consider a joint distribution. The assumption that objective functions are not correlated is usually not applicable in real-world situations.

To summarize the literature review on robustness measures in multi-objective evolutionary algorithms, it can be stated that they either do not necessarily account for the effect of uncertainty in objective space or they are tailored to specific concepts that can only be used in evolutionary methods. This applies, e.g., to clustering and ranking variation [108] or expected indicator functions [14].

The last point mentioned as a restriction to the concept of probabilistic dominance is important for the following discussions. In an ideal scenario, one could obtain a joint probability density function and could reformulate Equation (4.51) for robust Pareto dominance in objective space. One could even go one step further and use a set-valued optimization strategy for finding designs with optimal distributions (see e.g. [57]). This approach is not further pursued in the present work.

4.3.2.2. Sensitivity Analysis in MOO

The size and direction of a sensitivity region introduced in [98] can be used as a robustness measure. The sensitivity region is the region in random space such that a variation compared to the nominal parameter value will not cause the variation in objective space to exceed a certain limit. The measure was first introduced in single-objective optimization in [96]. The sensitivity region can be approximated locally by using derivative information resulting in a local sensitivity analysis. Or, it can be estimated using the worst-case sensitivity region, which can be found by solving an additional optimization problem for each design point and is able to account for directions of the sensitivity region in parameter space. The information on the sensitivity region can then be included as an additional constraint in the optimization problem based on the nominal values to account for robustness. Related approaches for feasibility robustness can be found in [97].

A similar concept is used in [126]. Here, the authors use a local sensitivity analysis to obtain the influence of perturbations in objective space. The region of sensitivity is approximated in objective space using, e.g., a rectangle in two-dimensions, which is constructed based on the normal and tangent direction to the nominal Pareto optimal front. Thus, opposed to the sensitivity region described above, it accounts for directions in objective space. It is worth noting that a good approximation of the nominal Pareto optimal front has to be constructed to find representative directions. For higher-dimensional objective spaces, it is harder to build the region of sensitivity. Some strategies for coping with these difficulties are discussed in [125].

A different strategy based on a local sensitivity analysis is pursued in Augusto et al. [8]. The authors make use of local sensitivity analysis by describing robustness with the help of the maximum singular value of the Jacobian of the objective function with respect to the uncertainties.

4.3.2.3. Worst-Case Analysis in MOO

Several robustness measures for MOO are inspired by single-objective robustness definitions based on statistical quantities. One may distinguish between worst-case, expectation-based, and variance-based measures. Also, the quantities can either be objectives or set as additional constraints.

As in the case of single-objective optimization, a worst-case analysis of a design can be used for measuring robustness, such that a robust worst-case multi-objective optimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \max_{\omega \in \Omega} \mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \\ \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = 0 \quad \forall \omega \in \Omega. \end{aligned} \quad (4.53)$$

Usually the worst case of objective function values

$$\max_{\omega \in \Omega} \mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) \quad (4.54)$$

4. Robustness in Multi-Objective Optimization

is considered as a single, worst-case ideal solution comprising the worst cases of all individual objective functions, such that it is not accounted for the effect of uncertainties in multi-objective space. In practice, there exists a worst-case Pareto front for each design point, which can be found by solving the multi-objective optimization problem of worst cases (4.54). First ideas for using such a representation for defining a dominance relationship for robustness to be applied in a multi-objective evolutionary algorithm were proposed in [10, 9]. Since the worst-case Pareto front was based on discrete set scenarios, the additional effort was negligible. In a scenario-based setting, different concepts of robustness have been analyzed in a multi-objective context, e.g., based on set order relations in [130]. In a more general setting in random space, however, finding a worst-case Pareto front for each design point is computationally expensive. Hence, the use of metamodeling techniques in random space is essential (see e.g. [163]). Similarly to the situation in single-objective robust design, the worst-case approach is rather conservative for finding robust optimal solutions and is especially of interest in reliability-based design.

4.3.2.4. Expectation- and Variance-Based Measures in MOO

In [47], Deb and Gupta introduce the concept of the mean effective objective function as a robustness measure, which can be generalized as a sampled expected value. The authors consider uncertainties in the design variables and sample in a ball in design space. Although the concepts of robust design were formulated for uncertainties in the design variables based on the mean effective objective function, they can be generalized for arbitrary multi-objective optimization problems under uncertainty using expectation-based measures. Thus, the work of Deb and Gupta can be seen as the basis for the use of expectation- and variance-based measures in MOO. The authors introduce two different concepts of robust solutions based on expected values.

Inspired from the expectation-based formulations in single-objective optimization one concept is to minimize the expected values of the individual objective function, which can be formulated as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & E[\mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))] \\ \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = 0 \quad \forall \omega \in \Omega. \end{aligned} \quad (4.55)$$

The expectation operator is to be understood component-wise. The resulting Pareto optimal front is sometimes referred to as the expected Pareto optimal front.¹

The second formulation for multi-objective robust design, given by

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) \\ \text{s.t.} \quad & \|E[\mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))] - \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}})\| \leq \nu, \\ & c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = 0 \quad \forall \omega \in \Omega, \end{aligned} \quad (4.56)$$

¹This terminology shall not be confused with the description of a distribution of the Pareto optimal front since the order of minimization and applying a statistical description is not necessarily interchangeable.

uses the expected deviations of the individual objective functions from the nominal values as a robustness measure. This formulation introduces the robustness measure as an additional constraint to the original problem based on nominal values $\bar{\mathbf{z}}$. The deviation is understood component-wise, such that

$$(E[\mathbf{F}(\mathbf{Z}(\omega))] - \mathbf{F}(\bar{\mathbf{z}}))_i = (E[\mathbf{F}(\mathbf{Z}(\omega)) - \mathbf{F}(\bar{\mathbf{z}})])_i = E[f_i(\mathbf{Z}(\omega)) - f_i(\bar{\mathbf{z}})]$$

for $i = 1, \dots, k$ and $\omega \in \Omega$.

The existence of a solution for the second formulation (4.56) depends on the value of maximum perturbation ν , which has to be chosen by the designer. Note that in a worst-case scenario, the worst solution can be chosen instead of the expected value. Furthermore, different norms (e.g. $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$) can be used in objective space. The original definition of the constraint in [47] uses the relative expected deviations, i.e., the absolute deviations divided by $\|\mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}})\|$. The formulation (4.56) will be further analyzed when introducing the new robust design formulation. For now, one may observe that while the first formulation reduces the influence of uncertainties to a single point in objective space (expected value), the second formulation accounts for component-wise deviations in objective space (expected deviations).

Deb and Gupta use both formulations in an evolutionary context. In [13], the degree of robustness is used as an additional measure for ranking solutions in multi-objective evolutionary optimization. Inspired by the maximum perturbation in formulation (4.56), the solutions are ordered based on the size of the design region such that a maximum perturbation in objective space is not violated.

The ideas of Deb and Gupta can be extended to variance-based measures using the individual variances of the objective function as robustness measures. This leads, for example, to the robust design problem

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) \\ \text{s.t.} \quad & \|\text{var}[\mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega))]\| \leq \mu, \\ & c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = 0 \quad \forall \omega \in \Omega. \end{aligned} \tag{4.57}$$

The variance is to be understood component-wise and can be obtained from the distributions of the individual objective functions. It accounts for the expected squared deviation from the expected value instead of an expected absolute deviation from a nominal value. The vector of variances can be treated as an additional vector of objective functions as well, thus leading to an extended multi-objective optimization problem, e.g.,

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \{E(\mathbf{F})^\top, \text{var}(\mathbf{F})^\top\} \\ \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) = 0 \quad \forall \omega \in \Omega. \end{aligned} \tag{4.58}$$

This formulation is, for example, applied in [193] for the design of an unmanned space vehicle and in [94, 46] for conceptual aircraft design.

In [182], a formulation similar to (4.49), which uses a combined objective function of the expected value and the standard deviation, is proposed in a multi-objective context.

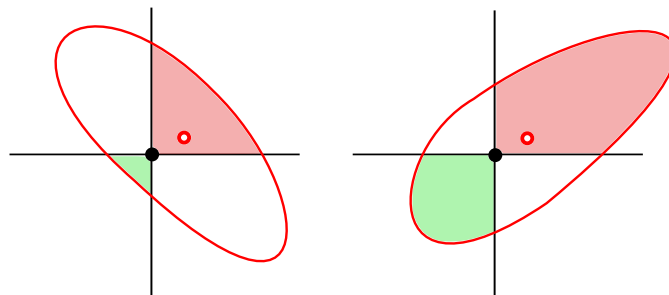


Figure 4.3.: Two different probability regions in two-dimensional objective space with definite gains (green) and losses (red) compared to the deterministic outcome (black dot). The red circle indicates the expected value.

It is applied in [109] for the design of a vehicle suspension system using an epsilon-constraint method and point collocation for uncertainty propagation and in [258] for crashworthiness design of a vehicle. In [197] the formulation based on expected values and variances is combined with the robustness measure of formulation (4.56).

Analogously to single-objective robust design, introducing the variance as an additional quantity allows for a better way of taking perturbations of the objective functions into account. Nevertheless, the individual variances of objective functions are not necessarily representative for analyzing the influence of uncertainties in objective space since the correlation between objective functions is not captured. This problem will be discussed in the following section.

4.3.3. Expected Losses in Objective Space

The intend of this section is to introduce a new measure for robust optimality that can be used in a scalarization approach to find robust optimal solutions and that is able to account for the effects of uncertainty in objective space. The general idea is to measure the expected signed distance of an outcome from the deterministic Pareto optimal front. Using the deterministic Pareto optimal front, one can state if an outcome of random samples in objective space is better or worse in the sense of Pareto optimality. In the present work, when comparing the effect of uncertainties of a design to the deterministic Pareto optimal front, this is denoted as *gains* and *losses* in objective space.

Figure 4.3 shows the importance of considering the combined effect of uncertainties in objective space. The probability region, as introduced at the beginning of Section 4.3.2, is used in the following to represent the distribution of solutions for a given design in objective space.² It is shown for two different designs in a two-dimensional objective space (red line). The designs are assumed to be close to each other. Thus, both designs have a similar deterministic value (black dot), where the objective function is evaluated at the nominal values. Also, the variances and the expected values (red circle) are

²The depiction for a fixed level of probability P_0 is used for illustration purposes. Gains and losses are defined for all outcomes and not only for outcomes in a probability region.

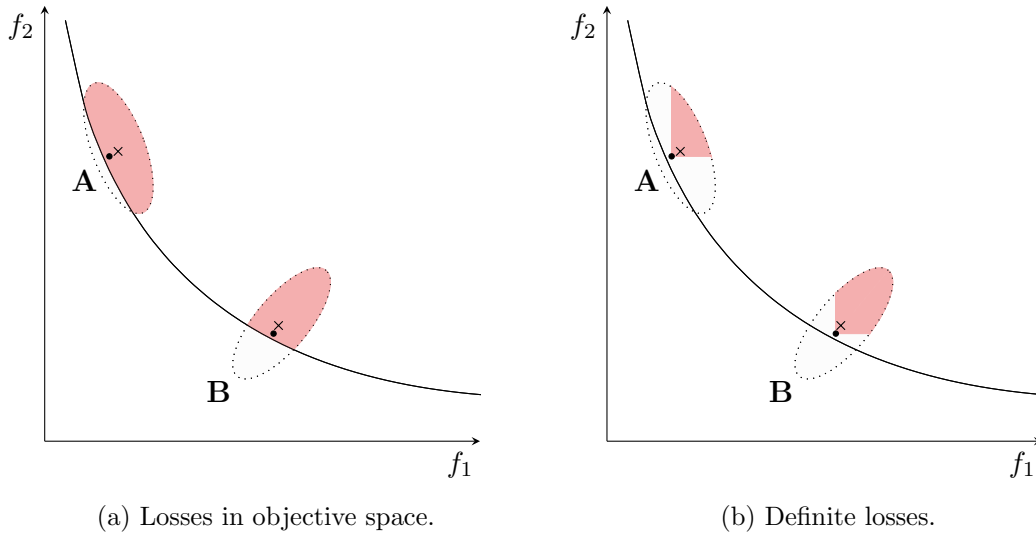


Figure 4.4.: Region of losses in objective space with respect to the Pareto optimal front (a) and definite losses with respect to the deterministic outcome indicated by the black dot (b). The red-colored probability regions indicate these regions. The design corresponding to point A is more robust than the one corresponding to point B.

similar.

Note that for a minimization problem, the points found in the lower-left region are definitely better (region of definite gains depicted by the green color), and the points in the upper right region definitely worse (region of definite losses depicted by the red color) in comparison to the deterministic outcome. The vertical and horizontal lines in the figure indicate these regions. The design on the right side has a large region of losses. When comparing both designs, it can be noticed that most outcomes of the left design dominate the outcomes of the right design. Thus, although both designs have similar expected values and the variances of the individual objective functions are similar, one would prefer the left design over the right one due to the shape of the probability region. Additionally, the fact that the gains outweigh the losses for the right design shows that a robustness measure should be defined based on losses in objective space.

As indicated in Figure 4.3, definite losses as well as possible losses can be measured based on the signed distance to the deterministic value of the specific design, i.e. the vector of objective functions evaluated at the nominal value. This is necessary if a deterministic Pareto optimal front is not known. However, if the deterministic Pareto front is known, it is possible to indicate gains and losses by using, for example, the signed distance of every outcome to the Pareto optimal front as an indicator. Both cases are depicted in Figure 4.4a and 4.4b. Figure 4.4a shows the region of losses, indicated in red and restricted to a specific probability region, for two different designs in objective space. The value for losses is higher when the outcomes are further away from the Pareto optimal front. When measuring the distance to the Pareto optimal front, it can be seen

4. Robustness in Multi-Objective Optimization

that point A can be denoted as more robust than point B. The region of definite losses compared to a deterministic design (dot) is shown in Figure 4.4b. The corresponding distances to the deterministic design can be used to measure definite losses if the Pareto optimal front is not known.

To summarize, the example in Figure 4.3 shows that the use of individual expected values and variances of objective functions for robust design in a multi-objective setting, as presented in Section 4.3.2, is not necessarily representative for measuring robustness. It can be observed that the distribution in objective space, that may be represented by a probability region, can be used for judging if a design is robust. Notably, the losses with respect to the deterministic Pareto optimal front or the deterministic outcome are of interest for robust optimality. Since it is in general too expensive to obtain a description of the probability region itself, the idea is to measure the losses based on the expected value of a signed distance function.³

In the following, robustness measures are introduced for both situations described above. In the first section, one assumes that the deterministic Pareto optimal front is known or approximated to measure the expected gains and losses. The robustness measure is used to find robust optimal points in a two-phase approach. Then the next section introduces the case when the Pareto optimal front is unknown and proposes a formulation for finding optimal robust designs in a direct approach.

4.3.3.1. Two-Phase Approach

In the two-phase approach, it is assumed that a given set of Pareto optimal points has been determined for the deterministic optimization problem (2.1) in a first phase. Additionally, one assumes to have an approximation of the Pareto optimal front in objective space. This can, for example, be done using local approximation techniques based on Taylor expansions (see [273] for application in MOO), using spline interpolation in a two-dimensional case or using other sophisticated interpolation methods for higher dimensions. Note that the approximation can become non-trivial for disconnected Pareto optimal fronts, although a distance to the front can still be defined. The representation of the Pareto optimal front is denoted as the interface Γ_0 for determining gains and losses.

The losses can then be expressed by means of a signed distance function $\delta : \mathbb{R}^k \rightarrow \mathbb{R}$ in objective function space, such that

$$\delta(\mathbf{F}, \Gamma_0) := \pm \min_{\mathbf{F}^* \in \Gamma_0} \|\mathbf{F} - \mathbf{F}^*\|. \quad (4.59)$$

The signed distance function measures the distance of a point in objective space to the representation of the deterministic Pareto optimal front Γ_0 with a positive sign indicating that the point is located in the region of losses.

³Reducing the representation of the losses to the expected value is a simplification with a loss of information. Additionally, the variance of the losses can be of interest. If the distribution of the loss function itself can be obtained with reasonable computational effort, e.g., using a surrogate model, it can be used to assess robust optimality.

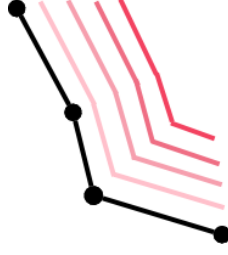


Figure 4.5.: Two-phase approach: Linear approximation of the Pareto optimal front (zero level set, $t = 0$) and different propagation steps (e.g. $t = 0.1, 0.2, 0.3$) of the level set function $\varphi(\mathbf{F}, t)$ representing level sets for the signed distance function.

The task is now to calculate the signed distance. Different definitions of the norm $\|\cdot\|$ can be used. In the following, the Euclidean distance is used. The problem is that, in general, a closed expression for $\delta(\mathbf{F}, \Gamma_0)$ is not available. Instead, the minimum distance has to be recalculated for every sample in objective space. This can be, for example, circumvented by approximating the signed distance function using strategies based on the level-set method (see Section A.3.1 in the appendix for further details). The general idea is to represent the signed distance function using a level set function with Γ_0 as the zero level set at a pseudo-time $t = 0$. Then the zero level set is propagated in pseudo-time using a transport equation.

In the exemplary Figure 4.5 the front is approximated using linear splines. The corresponding level sets with positive values of the signed distance function are shown in the figure as red lines.

The optimization problem to be solved in the second step aims at minimizing the performance while constraining the expected losses in objective space and can be formulated as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) & (4.60) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) = \mathbf{0}, \\ & E(\max(0, \delta(\mathbf{F}(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)), \Gamma_0))) \leq \delta_{max}. \end{aligned}$$

The evaluation at the nominal value $\bar{\mathbf{z}}$ denotes the deterministic case, which is the value prescribed when not considering any uncertainties. The expected losses are constrained by an upper bound δ_{max} . The choice of the upper bound is important for the existence of a solution to the above problem. The $\max(0, \cdot)$ -function ensures that only losses are considered.⁴ Note that the expected losses can also be used as an objective function to avoid the definition of δ_{max} . However, the constraint-based formulation will be applied in

⁴Note that for gradient-based optimization, the problem has to be transformed to obtain constraint functions that are continuously differentiable. This can be done by either reformulating the problem with the help of additional variables or by approximating the maximum function with a smooth function. The latter is done in the context of the present work.

4. Robustness in Multi-Objective Optimization

the following as it extends the original deterministic optimization problem. Furthermore, the definition of an upper bound for expected losses is less conservative.

4.3.3.2. Direct Approach

In the direct approach, the deterministic Pareto optimal front is not needed. As a result, a solution of the deterministic multi-objective optimization problem is not required. This can be advantageous if the computational effort for solving the MOO problem is high, and the deterministic Pareto optimal front is not of interest. It is also necessary to use the direct approach if the approximation of the Pareto optimal front is not representative enough. Instead, the distance of the samples to the objective function evaluated at $\bar{\mathbf{z}}$ for a given design \mathbf{u} is used to describe losses. This approach is similar to the constrained expectation-based formulation (4.56) using expected deviations in objective space. However, there are significant differences to the proposed formulation:

- Perturbations in objective space consider gains and losses equally. This can be used when the design variables are considered as uncertain. However, in a general setting, this penalizes gains in objective space.
- The expected distances are considered component-wise, and a single measure is defined afterwards using a suitable vector norm. As a result, this measure is not equivalent to an expected distance in objective space, i.e., in general,

$$\|E[\mathbf{F}(\mathbf{Z}(\omega)) - \mathbf{F}(\bar{\mathbf{z}})]\| \neq E[\|\mathbf{F}(\mathbf{Z}(\omega)) - \mathbf{F}(\bar{\mathbf{z}})\|].$$

For the proposed direct approach, different assumptions for the local estimation of losses can be made. When considering expected possible losses one may formulate the optimization problem as

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}) = \mathbf{0}, \\ & \sum_{i=1}^k E(\max(0, f_i(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) - f_i(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}}))) \leq \mu_1. \end{aligned} \quad (4.61)$$

Another assumption is to approximate the losses based on a local linear approximation of the Pareto optimal front in the deterministic outcome of the objective function for a design \mathbf{u} . The local front can then be represented as the zero level set of the signed distance function $\delta(\mathbf{F}) = \sum_{i=1}^k f_i(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) - f_i(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}})$. The corresponding constraint replaces the original constraint in (4.61), and is given by

$$E\left(\max\left(0, \sum_{i=1}^k f_i(\mathbf{y}, \mathbf{u}, \mathbf{Z}(\omega)) - f_i(\mathbf{y}, \mathbf{u}, \bar{\mathbf{z}})\right)\right) \leq \mu_2. \quad (4.62)$$

Figure 4.6 depicts the signed distance functions for the two different approaches. The expected possible losses and the linear approximation are always obtained locally for the particular deterministic outcome.

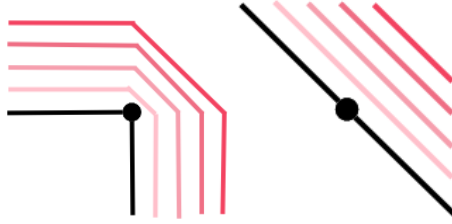


Figure 4.6.: Distance function for expected possible losses (left) and linear approximation (right).

Other expressions may be based on the expected definite losses or a better local approximation of the front. If, for example, the MOO problem is convex, a convex approximation, e.g., $\varphi_0 = f_1 + f_2 + f_1 \cdot f_2$ in two dimensions, can be used.⁵

4.4. Application and Results

This section serves to present the application of strategies for multi-objective robust design to the aerodynamic shape optimization problem described in Section 2.4.1, using the optimization strategies of Sections 2.3.3 and 3.2. After giving an overview of the literature on the application of uncertainty quantification and robust design in aerodynamics in Section 4.4.1, details on the implementation of the above-described methodologies are given in Section 4.4.2. In Section 4.4.3, multi-objective robust design is performed using the expectation-based approach, as well as both proposed approaches based on expected losses, considering uncertainties in the operational conditions. Uncertainties in the geometry are considered separately in Section 4.4.4.

4.4.1. Uncertainty Quantification and Robust Design in Aerodynamics

Uncertainties in aerodynamic applications may arise in different forms due to various effects. In general, one may distinguish between uncertainties in the operational conditions and geometrical uncertainties. Both types of uncertainties, operational and geometrical, may be included in aerodynamic design processes. In the following results, they are introduced separately in different studies to investigate the different effects. The present section shall give an overview of strategies for uncertainty quantification and robust design in aerodynamic design based on both types of uncertainties.

Uncertainties in the operational conditions are caused by general spatial or temporal variations in the flow regime. They can arise due to atmospheric conditions or certain flight maneuvers. It can also be of interest to include operational uncertainties in aerodynamic design when trying to cope with sensor uncertainties in real flight or corrections in wind tunnel experiments due to walls or inhomogeneous conditions [228]. Geometrical uncertainties can be used to represent variations due to manufacturing and wear over

⁵Note that an appropriate scaling has to be considered. Especially for the proposed convex approximation one can expect values to be in the range of -1 and 1 .

4. Robustness in Multi-Objective Optimization

time. During flight, geometrical uncertainties may occur due to loading or icing effects. Icing, for example, occurs when droplets at a temperature below the freezing point hit the airfoil surface and freeze, which can significantly impact the performance and may lead to accidents (see [79]).

Uncertainty Quantification in Aerodynamics Uncertainties in the operational conditions are commonly modeled by assuming a uniform or a normal distribution for the free-stream Mach number (see e.g. [127, 171]). Furthermore, it can be made use of truncated distributions to prohibit unphysical effects [242]. Additionally, the angle of attack is sometimes assumed to be independently distributed (see e.g. [244]).

For uncertainties in the operational conditions, it is common (see e.g. [243, 34]) to assume a Gaussian random process $\psi(\boldsymbol{\xi}, \omega)$ with a mean function

$$E(\psi(\boldsymbol{\xi}, \omega)) = 0 \quad (4.63)$$

and a covariance function

$$Cov(\boldsymbol{\xi}, \boldsymbol{\xi}') = b^2 \exp\left(-\frac{1}{l^2} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_2^2\right). \quad (4.64)$$

Here b stands for the variability in the manufacturing error represented by the magnitude of the bumps. The parameter l describes the frequency of the bumps, which is proportional to $1/l$ and indicates how fast the correlation is reduced along the airfoil. The variables $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ are points on the airfoil surface A . The effects of uncertainty act in the outward normal direction \boldsymbol{n} of the airfoil, such that the uncertain shape \tilde{s} is expressed as

$$\tilde{s}(\boldsymbol{\xi}, \omega) = \boldsymbol{\xi} + \psi(\boldsymbol{\xi}, \omega)\boldsymbol{n}(\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in A, \omega \in \Omega. \quad (4.65)$$

An early overview of methods in uncertainty quantification and robust design in engineering is given in [298], pointing out the general need for UQ-based strategies and the resulting opportunities for engineering design. In the following years, different methods for uncertainty propagation have been tested for general CFD applications and, in particular, for aerodynamic applications.

In [172], geometrical variations are modeled using global shape parameters that are assumed to be distributed with a truncated normal distribution. The effects of uncertainties in a subsonic and transonic RANS flow are studied using stochastic collocation. The influence of geometrical uncertainties on an Euler flow around an airfoil is investigated in [34]. The manufacturing error is modeled using a the Gaussian random process with the covariance matrix (4.64). The infinite-dimensional process is represented using finitely many modes of the KL expansion, and the dimension is further reduced by using a global sensitivity analysis requiring a sampling strategy. In [228], stochastic collocation and the pseudo-spectral approach are applied to an airfoil in subsonic RANS flow under the consideration of operational and geometrical uncertainties with uniform distributions. The general observation, in this case, is that the effort of both approaches is comparable with a slight advantage for the pseudo-spectral approach when the underlying problem is smooth.

Robust Design in Aerodynamics Robust airfoil design dates back to Huyse et al. [127, 128]. Huyse used a discretized semi-infinite robust design formulation based on minimizing the expected value of the drag coefficient. The mean drag coefficient was optimized under an uncertain Mach number in a transonic Euler flow. The Mach number was assumed to be uniformly distributed, and the expected value was evaluated by using a quadrature method. A min-max formulation was used in [164] for the same setting. Different studies for robust airfoil design have been conducted afterwards. Some of them are listed in the following.

In [95] robust design is performed using the simplified assumption of independent normally distributed geometrical parameters of a 3D wing. The uncertainties are propagated using a first-order perturbation method. In [207] geometrical and operational uncertainties are considered separately for a 2D airfoil in Euler flow. A worst-case formulation is used, and the inner optimization problem considering the worst performance of a design under uncertainties is solved on a surrogate model. The overall optimization is carried out using a genetic algorithm. Lee et al. [162] optimize a three-dimensional wing under uncertainties in the Mach number. The sampled mean and the variance of the drag coefficient are considered as objective functions for a multi-objective evolutionary algorithm. In [244], the formulation (4.45) resulting from a semi-infinite formulation and the chance-constrained formulation (4.46), using a first-order and second-order perturbation method, are compared for lift-constrained robust design with uncertainties in the operating conditions. The Mach number and angle of attack are distributed using a truncated normal distribution. The use of the first-order and second-order perturbation method does not make a significant difference for the specific test case of an airfoil in a 2D Euler flow. The semi-infinite formulation leads to a better design than the chance-constrained formulation when considering the performance of the range of the uncertain parameters. The work in [243] focuses on geometrical uncertainties, which are modeled using a Gaussian random process with the covariance function. The KL basis is chosen goal-oriented, making use of the derivative of the objective function with respect to the single basis functions. The semi-infinite formulation is discretized using non-intrusive polynomial chaos based on spatially-adaptive sparse grid rules. In [235], a dimension-adaptive sparse grid method based on the work of Gerstner and Griebel [77] is used. In all works [244, 243, 235], the one-shot method based on approximate reduced SQP iterations is used for optimization. Ghisu et al. [79] study robust aerodynamic design considering the effects of icing. The unknown position of a single ice shape is propagated using non-intrusive polynomial chaos. In [214, 215], the geometrical variations are modeled as uncertainties in the design variables with independent Gaussian distributions and a robust design of a two-dimensional transonic airfoil is performed using a proposed quadrature technique for uncertainty propagation. The model of using design parameter uncertainties stems from the idea of accounting for uncertainties in the conceptual design phase.

Robust airfoil design with multiple objectives is performed in [39] for a 2D airfoil in viscous flow using game theory. Uncertainties are assumed in the Mach number and the angle of attack, and the expectation-variance-based approach (see problem (4.58)) is pursued using four objective functions, namely the expected value and the variances

4. Robustness in Multi-Objective Optimization

of the drag and lift coefficient.

4.4.2. Implementation Details

The test case introduced in Chapter 2 is considered for robust airfoil design with multiple objectives. As before, the objectives are the drag and the lift coefficient. Additional inequality constraints for the pitching moment and the thickness of the airfoil are imposed. To summarize the strategy:

- Different formulations for multi-objective robust design are analyzed. This includes the new formulations based on expected losses proposed in Section 4.3.3.
- The resulting multi-objective optimization problems are solved with the help of constraint methods that make use of the hybrid strategy presented in Section 2.3.3.
- Uncertainties are propagated with the help of a non-intrusive polynomial chaos method (see Section 4.2.2). When considering geometrical uncertainties it is made use of a KL expansion and sparse grids, as introduced in Section 4.2.1 and 4.2.3.

Details on the optimization procedure, methods for uncertainty quantification as well as robustness measures that are specific for the application are given in the following.

4.4.2.1. Hybrid Optimization

For the hybrid optimization, the robust design formulation has to be integrated into the metamodeling approach. There exist two possible options.

1. One possibility is to build a metamodel based on the design space and the random space. Since the metamodel is cheap to evaluate, Monte Carlo sampling can be applied for propagating uncertainties.
2. The other possibility is to build a metamodel for the approximated statistical quantities like expectation, variance, or losses.

The first possibility is, for example, successfully applied in [139] based on different metamodeling strategies, where Kriging turned out to be the most promising. However, since it shall be avoided to increase the approximation space of the metamodel, the present work sticks to the second possibility and also makes use of a non-intrusive polynomial chaos approximation for building the Kriging models.

4.4.2.2. Scaling

The objective functions have to be scaled appropriately to compare losses in objective space and to find meaningful thresholds for the variance. In the application to aerodynamic shape optimization, for example, the lift coefficient is about 100 times bigger than the drag coefficient.

For the possible losses in objective space in the direct approach, as well as for the variance, $c_d(\bar{z})$ and $c_l(\bar{z})$ are used as constant scaling values. The same scaling strategy is chosen for the dimension-adaptive sparse grid quadrature to allow for a multi-objective error indicator.

For the two-level approach, the objective function values are normalized in objective space based on the ideal and the nadir point of the deterministic Pareto optimal front.

4.4.2.3. Quadrature

All stochastic quantities are calculated using non-intrusive polynomial chaos. Since the uncertainties are expressed assuming normally distributed random variables, it is made use of Hermite polynomials. Gauss-Hermite quadrature [85] is used to approximate integrals in random space. As the polynomial chaos expansion is based on Hermite polynomials, Gauss-Hermite quadrature is a natural choice. The reason for that is that the Hermite polynomials $H_i(x)$, which are described by the recursive relation

$$H_{i+1}(x) = xH_i(x) - iH_{i-1}(x) \quad \text{with} \quad H_0(x) = 1, \quad H_1(x) = x, \quad (4.66)$$

are orthogonal when using the probability density function of the standard normal distribution

$$w(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (4.67)$$

as a weight function. One obtains $\langle H_i(x), H_j(x) \rangle_{w(x)} = i! \delta_{ij}$. This definition of Hermite polynomials is the probabilistic definition. There also exists a physical definition with $\tilde{H}_{i+1}(x) = 2x\tilde{H}_i(x) - 2i\tilde{H}_{i-1}(x)$ and $\tilde{H}_0(x) = 1, \tilde{H}_1(x) = 2x$.

The Gauss-Hermite quadrature rule of order k approximates the value of an integral based on the weight function w as

$$\int_{-\infty}^{\infty} f(x)w(x)dx = \frac{1}{\sqrt{\pi}} \sum_{i=1}^k w_i f(\sqrt{2}x_i) \quad (4.68)$$

with the quadrature points x_i being the zeros of the physical Hermite polynomial $\tilde{H}_k(x)$ and weights

$$w_i = \frac{2^{k-1}k!\sqrt{\pi}}{k^2[\tilde{H}_{k-1}(x_i)]^2}. \quad (4.69)$$

Unfortunately, Gauss-Hermite quadrature rules are only weakly nested, i.e., the rules of odd orders all include the grid point $x = 0.0$. Figure 4.7 shows the points of a two-dimensional tensorized Gauss-Hermite quadrature rule of level $k = 14$ (left). The accuracy and the number of points given by $15 \times 15 = 225$ is comparable to the Clenshaw-Curtis grid presented in Figure 4.1. When constructing the corresponding sparse grid from the full Gauss-Hermite rule of level $k = 14$, the resulting number of grid points is 1233 (right). As a result, although the accuracy of the Gauss-Hermite sparse grid

4. Robustness in Multi-Objective Optimization

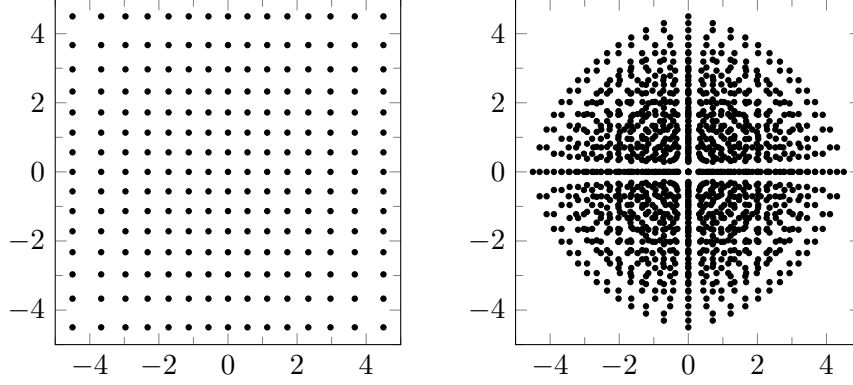


Figure 4.7.: Full tensor grid of Gauss-Hermite rule of level $k = 14$ (left) and corresponding sparse grid rule of level $k = 14$ (right).

is higher, the number of points in the sparse grid is significantly higher for the weakly nested rule.

For trading off accuracy and the number of grid points, the idea is to construct a sparse grid rule that is only based on certain levels of Gauss-Hermite quadrature. A possible choice is to use levels 1, 3, 7, 15, 31, \dots of the one-dimensional Gauss-Hermite quadrature rule resulting in a new sparse grid Gauss-Hermite rule. The corresponding sparse grid of level $k = 3$ (up to level 15 of the Gauss-Hermite quadrature rule) is shown in 4.8a and consists of 73 points.

In the present work, when considering uncertainties in the operational conditions, it is made use of four quadrature points. For the initial design, one observes a relative error of the order of 10^{-4} due to the high nonlinearities that result from the shock. Near the optimal design, where a better approximation quality is important, the relative error is of the order of 10^{-5} . Throughout the optimization, the resulting approximation is representative enough to guide the optimization algorithm, especially when compared to the numerical approximation quality of the objective function.

For the geometrical uncertainties, the present work uses the implementation of a dimension-adaptive sparse grid construction by Schillings, which was applied to Bayesian inverse problems in [237]. It is implemented in C++. When used in the Python framework of SU2, the executable can be called. The interface for evaluating functions of interested was extended to the SU2 environment. For the one-shot method, the implementation in C++ was directly integrated in the respective routines in SU2.

For the present work, the implementation of the dimension-adaptive sparse grid was extended to non-nested and weakly nested Gauss-Hermite rules, and the error estimation was adapted for a multi-objective setting. To avoid the generation of different index sets for the objective functions f_j with $j = 1, \dots, k$, it is made use of the maximum L_2 -norm of the individual error indicators

$$(\Delta_{i_1}^{(1)} \otimes \dots \otimes \Delta_{i_N}^{(1)})(f_j) \quad \text{for } j = 1, \dots, k. \quad (4.70)$$

This requires a proper scaling of the objective functions such that the errors can be

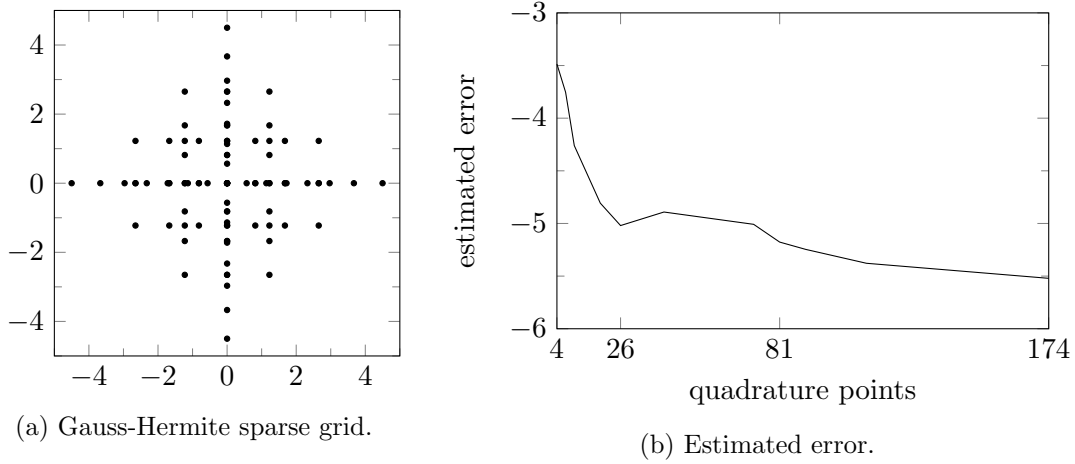


Figure 4.8.: Sparse grid Hermite rule constructed from levels 1, 3, 7, 15 of Gauss-Hermite quadrature. (a). Estimated error of dimension-adaptive sparse grid Hermite rule in (three-dimensional geometrical uncertainties) (b).

compared with each other. In the dimension-adaptive sparse grid strategy, it is made use of the Gauss-Hermite sparse grid rule. The estimated error is shown in Figure 4.8b for a three-dimensional random space assuming geometrical uncertainties for the initial profile. A value of 10^{-5} for the estimated error is chosen as an abort criterion.

4.4.2.4. Goal-Oriented Karhunen-Loève expansion

The KL-expansion is employed to represent the random process with the help of a construction based on the largest eigenvalues. When using this type of construction, the input random process can be approximated in a representative form. However, the modes corresponding to the largest eigenvalues can have different impacts on the random outputs, i.e., the objective functions. Therefore, it can be of interest to further reduce the dimensionality by only choosing the eigenvectors with a high impact on the outputs. This strategy is referred to as a *goal-oriented* KL expansion.

There exist different strategies to measure this impact, inspired by methods for model reduction or adaptive refinement. In [34], e.g., the dimension is reduced with the help of a global sensitivity analysis based on sampling. In [243], local sensitivity information of the objective function is used to measure the impact of an eigenvector v_i as a KL basis function. Suppose that $\tilde{\boldsymbol{\xi}}$ are the surface mesh points describing the perturbed shape \tilde{s} , then the total derivative of an objective function f with respect to Z_i measures the impact of the respective eigenvector. When the perturbed shape is evaluated at a fixed $Z_i = \hat{Z}$ in random space, one obtains the local impact. The total derivative can be obtained with algorithmic differentiation, namely

$$D_{Z_i} f(\tilde{s}(\hat{Z})) = D_{\tilde{\boldsymbol{\xi}}} f(\tilde{s}(\hat{Z})) \cdot \frac{\partial \tilde{\boldsymbol{\xi}}}{\partial Z_i}(\hat{Z}) = D_{\tilde{\boldsymbol{\xi}}} f \cdot \sqrt{\lambda_i} v_i(\boldsymbol{\xi}) \mathbf{n}(\boldsymbol{\xi}), \quad (4.71)$$

4. Robustness in Multi-Objective Optimization

where λ_i is the corresponding eigenvalue and \mathbf{n} is the direction of perturbation for the unperturbed points $\boldsymbol{\xi}$ (see Equation (4.65)). The total derivative with respect to the surface mesh points can be obtained efficiently using an AD-based discrete adjoint strategy. The basis is reduced by keeping indices i with large sensitivity. The goal-oriented KL expansion based on local sensitivity information is implemented in the context of the current work. For multi-objective optimization, the L_2 -norm of the sensitivity analysis for the objective functions is used to measure the impact. Unfortunately, for the application in the present work, the KL basis cannot be further reduced in a goal-oriented manner. This is due to the fact that the first eigenvectors always have a considerable impact on one of the objective functions.

4.4.2.5. Signed Distance Function

The signed distance function is used for obtaining the expected losses as a measure for robustness. In the following, a linear spline approximation of the deterministic Pareto optimal front is constructed. For the linear approximation in an objective space with moderate dimensions it is computationally feasible to calculate the exact signed distance (4.59) from each part of the front by means of linear algebra. The presented results are restricted to a two-dimensional objective space. Given a set of dimension P of deterministic Pareto optimal points, which are described by $(f_1^i, f_2^i)^\top$ for $i = 1, \dots, P$, the signed distance δ_i of a sample point $(f_1, f_2)^\top$ to the linear approximation of the Pareto optimal front based on the point $(f_1^i, f_2^i)^\top$ can be obtained by inserting the sample point in the Hesse normal form, which is

$$\delta_i = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \left[\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} - \begin{pmatrix} f_1^i \\ f_2^i \end{pmatrix} \right],$$

where n_i is the normal of the linear approximation pointing into the direction of losses in objective space. The signed distance to the Pareto optimal front is the minimum over all individual distances δ_i .

The expected losses are then defined as the expected value of the positive signed distance, which is obtained from $\max\{0, \delta\}$. As already explained above, for gradient-based optimization the maximum is approximated using a smooth function. The approximation

$$\max\{0, \delta\} \approx \frac{1}{2} \left(\delta + \sqrt{\delta^2 + \varepsilon} \right) \quad (4.72)$$

with $\varepsilon > 0$ is chosen.

4.4.3. Results for Uncertainties in the Operational Conditions

In a first application, different robust design formulations are analyzed for the multi-objective aerodynamic shape optimization under the consideration of uncertainties in the operational conditions.

The scalar-valued uncertainties in the operational conditions of flight are modeled by using random variables with an assumed probability density function. In the following,

it is assumed that the free-stream Mach Number Ma_∞ is uncertain and can be described with the normal distribution $Ma_\infty \sim N(\mu, \sigma^2)$ with $\mu = 0.8$ and $\sigma = 0.01$, which can also be expressed as $Ma_\infty = \mu + \sigma Z$ with $Z \sim N(0, 1)$. The associated orthogonal polynomials for the standard normal distribution with probability function f_Z are the Hermite polynomials H_k defined by (4.66).

Following the derivation in (4.18), the expected value of c_d , for example, is then given by

$$\begin{aligned} E(c_d(\mathbf{y}, \mathbf{u}, Z)) &\approx E\left(\sum_{k=0}^m c_{d,k}(\mathbf{y}, \mathbf{u}) H_k(Z)\right) \\ &= c_{d,0}(\mathbf{y}, \mathbf{u}) = \int_{-\infty}^{\infty} c_d(\mathbf{y}, \mathbf{u}, z) H_0(z) f_z(z) dz \\ &\approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^M c_d(\mathbf{y}, \mathbf{u}, \sqrt{2}z^i) w^i. \end{aligned}$$

Note that it was made use of the orthogonality of the polynomials and that the resulting integral was approximated with a Gauss-Hermite quadrature formula (4.68) with weights w^i and points z^i for $i = 1, \dots, 4$.

For the variance, one obtains analogously

$$\begin{aligned} var(c_d(\mathbf{y}, \mathbf{u}, Z)) &\approx E\left(\left(\sum_{k=0}^m c_{d,k}(\mathbf{y}, \mathbf{u}) H_k(Z) - c_{d,0}\right)^2\right) \\ &= \sum_{k=1}^m k! c_{d,k}(\mathbf{y}, \mathbf{u})^2 \tag{4.73} \\ &= \sum_{k=1}^m k! \left(\int_{-\infty}^{\infty} c_d(\mathbf{y}, \mathbf{u}, z) H_k(z) f_z(z) dz\right)^2 \\ &\approx \sum_{k=1}^m \frac{k!}{\pi} \left(\sum_{i=1}^M c_d(\mathbf{y}, \mathbf{u}, \sqrt{2}z^i) H_k(\sqrt{2}z^i) w^i\right)^2. \end{aligned}$$

The results in the following sections make use of the expected value of the objective functions or loss functions. Furthermore, only the Mach number is considered to be uncertain. Further numerical results are shown in Section A.3.3 of the appendix. The variances of the objective functions are analyzed. Furthermore, the angle of attack is considered as an additional uncertain variable.

4. Robustness in Multi-Objective Optimization

4.4.3.1. Expectation-Based Approach

The robust multi-objective optimization problem based on the expected value is given by

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \{E(c_d(\mathbf{y}, \mathbf{u}, Z(\omega))), E(-c_l(\mathbf{y}, \mathbf{u}, Z(\omega)))\} \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, Z(\omega)) = \mathbf{0} \quad \forall \omega \in \Omega, \\ & g(\mathbf{y}, \mathbf{u}, Z(\omega)) \geq \mathbf{0} \quad \forall \omega \in \Omega. \end{aligned} \quad (4.74)$$

When applying the above expression for the expected values and the ε -constraint method and when discretizing the semi-infinite formulation, one obtains the multiple set-point problem

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{u}} \quad & \frac{1}{\sqrt{\pi}} \sum_{i=1}^M c_d(\mathbf{y}, \mathbf{u}, z_i) w_i \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{y}, \mathbf{u}, z_i) = \mathbf{0} \quad \forall i = 1, \dots, M, \\ & g(\mathbf{y}, \mathbf{u}, z_i) \geq \mathbf{0} \quad \forall i = 1, \dots, M, \\ & -\frac{1}{\sqrt{\pi}} \sum_{i=1}^M c_l(\mathbf{y}, \mathbf{u}, z_i) w_i \leq c_{l,j}, \end{aligned} \quad (4.75)$$

for each iteration step j for the constraint $c_{l,j}$ on the lift coefficient. The same applies to the constraints on the drag coefficient when minimizing the expected value of the lift coefficient. The additional inequality constraints and the PDE constraint shall be feasible for all quadrature points z_i . The inequality constraints are the same as for the original optimization problem without any uncertainties, namely a lower bound on the moment coefficient and a lower bound on the maximum thickness of the airfoil. Note that in this specific situation the maximum thickness does not depend on the random variable. Furthermore, since the moment coefficient is monotonously increasing for low Mach numbers, the constraints can be reduced to a single constraint

$$c_m(\mathbf{y}, \mathbf{u}, \min_{i=1, \dots, M} z_i) \geq 0.$$

For minimizing drag and maximizing lift, the robust multi-objective optimization problem is solved for finding eight Pareto-optimal points. In a first approach, the epsilon-constraint method is applied in combination with the hybrid approach, using IPOPT for gradient-based optimization.

In Figure 4.9, the found points on the deterministic Pareto optimal front are marked as blue dots. The expected values can be found in a post-processing step for each of the designs and are marked with a blue cross.⁶ The points found using the expectation-based approach are marked in black. The black crosses denote the expected Pareto optimal front, and black dots indicate the corresponding nominal values in objective space. The

⁶The values for the evaluation at the nominal parameters and the expected values are plotted in the same objective space. It is, however, important to remark that the expected values do not define a direct solution in objective space.

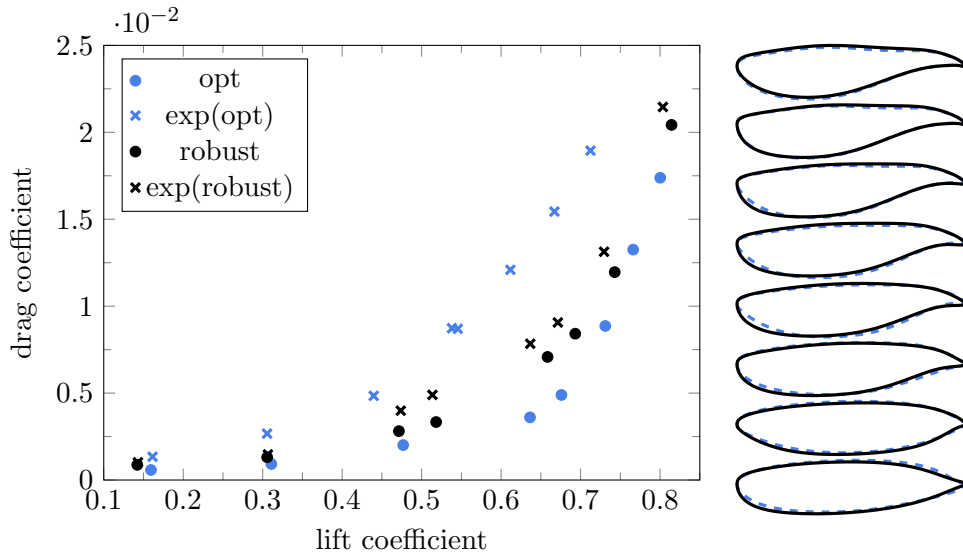


Figure 4.9.: Solutions found for the expectation-based approach. The black crosses indicate the expected values in objective space. The points correspond to an evaluation at the nominal parameters. The optimal solutions without considering uncertainties are indicated by the blue points and their respective expected values by blue crosses. The designs corresponding to the solutions are presented on the right side.

points indicate a convex Pareto optimal front for each problem. One can clearly see the improvement gained by the robust optimization. The expected values corresponding to the deterministic Pareto optimal front tend to be worse than the respective minimum objective values leading to high deviations from the nominal values when considering uncertainties. On the right side of the figure are the designs that correspond to the found points from top to bottom. The dashed shapes belong to the optimal designs, and the solid shapes belong to the robust optimal designs.

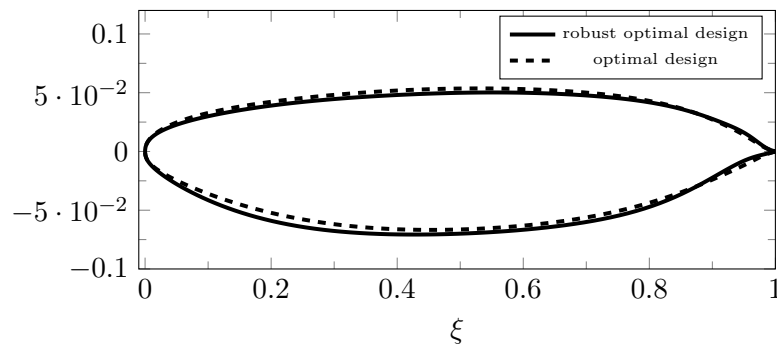


Figure 4.10.: Exemplary robust optimal airfoil design (solid line) and optimal airfoil design (dashed line).

4. Robustness in Multi-Objective Optimization

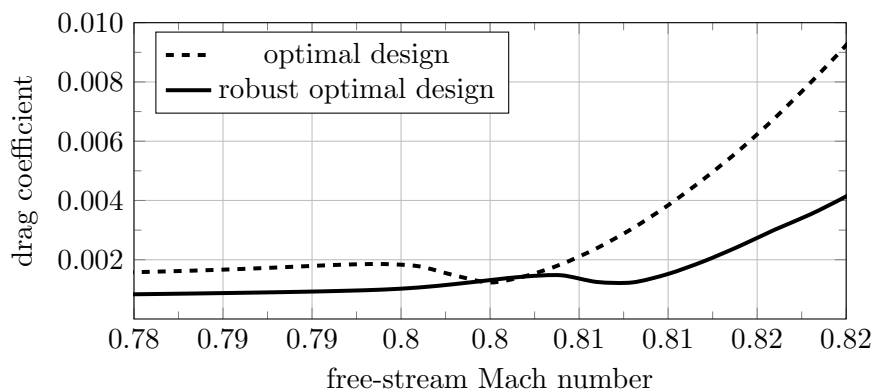


Figure 4.11.: Behavior of the drag coefficient for the deterministic and the robust optimal design. The increase in the drag coefficient is shifted to larger Mach numbers.

The robustness is shown exemplary for one robust optimal design, which is given in Figure 4.10. The expected value of the optimal design are $E(c_d) = 0.00274$ and $E(c_l) = 0.306$ and the expected values of the robust optimal design are $E(c_d) = 0.00147$ and $E(c_l) = 0.307$. The differences of the two shapes are apparent near the trailing edge and on the upper and lower side.

Figure 4.11 shows the behavior of the drag coefficient around the mean Mach number. While the drag coefficient of the optimal design (dashed line) is sensitive around the Mach number of 0.8, it is not very sensitive for the robust optimal design. The strong increase of the drag coefficient is also shifted to a higher Mach number. For the lift coefficient, a weaker increase in the lift coefficient with respect to the Mach number is observed for the robust optimal design.

For further analysis purposes, the hybrid strategy using the one-shot strategy for gradient-based optimization is applied to solve the multi-objective robust design problem. To present new results, the optimization problem under consideration is the multi-objective aerodynamic shape optimization problem without considering constraints on the moment coefficient and the thickness. Figure 4.12 shows the obtained robust optimal solutions in objective space, as well as the corresponding designs.

During the one-shot strategy, the flow solutions and adjoint solutions for the different Mach numbers are iterated simultaneously, and a combined objective and augmented Lagrange function is built based on the corresponding quadrature weights. It can be observed that the one-shot method based on expected values needs fewer iterations in comparison to the results of Chapter 3. This is a typical result for robust design based on expectation values since the integration has a smoothing effect on the objective and constraint functions. Nevertheless, the computational costs for finding a robust optimal design are naturally higher than the costs for finding an optimal design since the consideration of the Mach number as a random variable increases the dimension. For the following results, IPOPT is used as a gradient-based optimization algorithm to

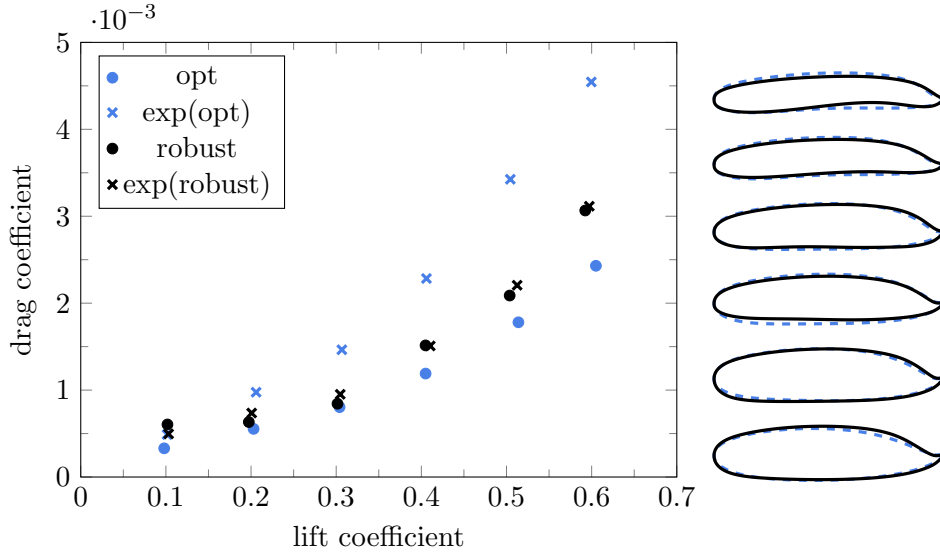


Figure 4.12.: Robust optimal solutions found with the one-shot method when only considering bound constraints (black dots, crosses for expectation), and comparison to optimal designs (blue dots, crosses for expectation). Corresponding designs are compared on the right side.

present and analyze results for higher lift coefficients.

For further numerical results, including the consideration of an additional uncertainty in the angle of attack, one may have a look at Section A.3.3 of the appendix.

4.4.3.2. Two-Phase Approach

In the following, the two-phase approach is used with a prescribed constraint on the distance $\delta_{max} = 0.15$ in normalized objective space. The additional constraint for the moment coefficient is also evaluated at the nominal value \bar{z} . Note, however, that any type of feasibility robustness could be applied. The expected losses are calculated with the help of non-intrusive polynomial chaos using Gauss-Hermite quadrature with four quadrature points z^i and weights w^i for $i = 1, \dots, n$. The resulting multi-objective optimization problem

$$\begin{aligned}
 \min_{\mathbf{y}, \mathbf{u}} \quad & \mathbf{F}(\mathbf{y}, \mathbf{u}, \bar{z}) := (c_d(\mathbf{y}, \mathbf{u}, \bar{z}), c_l(\mathbf{y}, \mathbf{u}, \bar{z}))^\top \\
 \text{s.t.} \quad & c(\mathbf{y}, \mathbf{u}, \bar{z}) = 0, \\
 & c_m(\mathbf{y}, \mathbf{u}, \bar{z}) \geq 0, \\
 & t(\mathbf{u}) \geq 0.12, \\
 & \sum_{i=1}^n w^i (\max(0, \delta(\mathbf{F}(\mathbf{y}, \mathbf{u}, z^i), \phi_0))) \leq \delta_{max}
 \end{aligned} \tag{4.76}$$

is solved for eight Pareto optimal points using the ε -constraint method. The signed distance δ is based on the linear approximation presented in Section 4.4.2.

4. Robustness in Multi-Objective Optimization

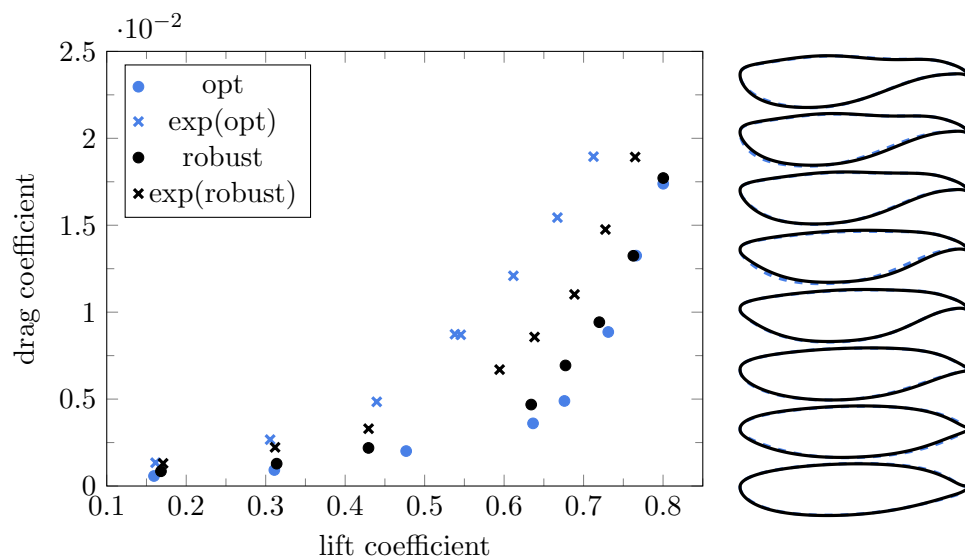


Figure 4.13.: Pareto optimal points for two-phase approach (robust) and deterministic Pareto optimal points (opt). The design are given on the right-hand side.

Figure 4.13 shows the optimization result in objective function space. The black dots indicate the robust optimal designs evaluated for the Mach number $\bar{z} = 0.8$, which is used in the deterministic optimization. The crosses indicate the expected values. For reasons of comparison, the deterministic values and the expected values of the multi-objective optimization without considering uncertainties are shown by the black dots and crosses. The corresponding designs are plotted on the right side of the figure. The upper design corresponds to the maximum lift coefficient and the lower design to the minimum drag coefficient. It can be observed that the designs are very similar, while the expected values for the robust design approach are significantly improved.

In Figure 4.14 random samples are shown for a chosen design to depict the probability region. The blue-colored dots denote the probability region for a comparable design that was obtained using the expectation-based approach, i.e. solving problem (4.74). The probability regions differ significantly as the result obtained by the expectation-based approach leads to higher losses in objective space. In particular, the probability region based on the expected losses is close to the deterministic Pareto optimal front. As an interesting side-effect for the application under consideration, the perturbations of the resulting solution are less prone to result in a strong shock wave in the flow solution. For the robust optimal design based on expectation, the occurrence of strong shocks for specific perturbations can be observed due to the sudden increase of the drag coefficient and decrease of the lift coefficient.

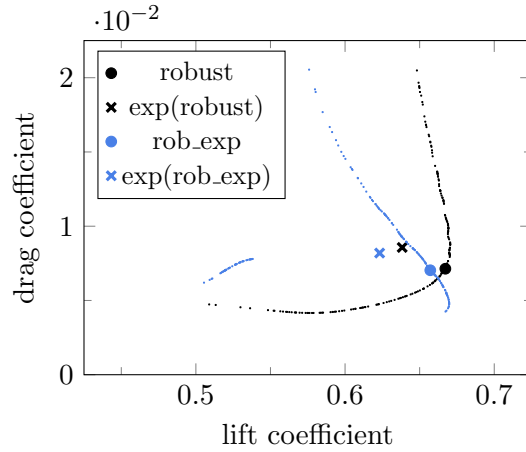


Figure 4.14.: Sampled probability region for the expected losses (robust) and the expectation-based approach (rob_exp).

4.4.3.3. Direct Approach

The direct approach can be used if a construction of the Pareto optimal front is computationally too expensive. Exemplary, it is applied to the given test case by constraining the expected possible losses presented in Equation (4.61) with $\mu_1 = 0.15$. The results in objective space are shown in Figure 4.15. The constraining of expected possible losses leads to designs with improved expected values.

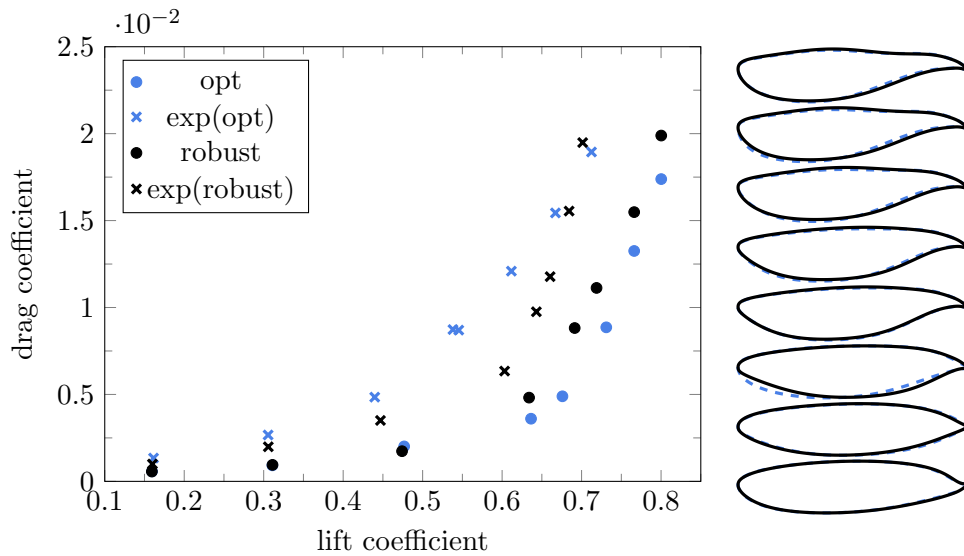


Figure 4.15.: Pareto optimal solutions for direct approach using expected possible losses. The robust optimal solutions (robust) are compared to optimal solutions (opt). The corresponding designs are given on the right-hand side.

4. Robustness in Multi-Objective Optimization

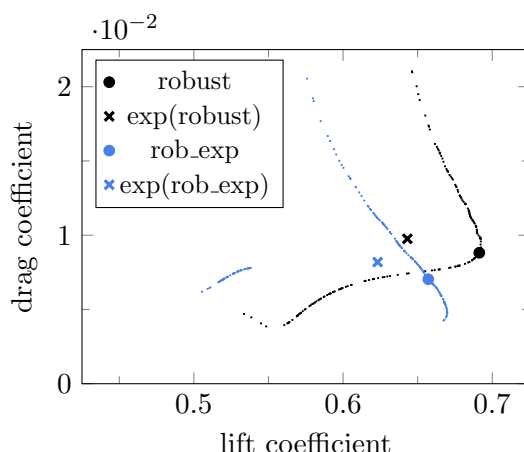


Figure 4.16.: Sampled probability region for the direct approach (robust) and the expectation-based approach (rob_exp).

In Figure 4.16, the probability region for a specific design is compared to the probability region of the design obtained using the expectation-based approach. Again, the expectation-based approach leads to higher losses in objective space. The samples of the probability region of the expectation-based approach are dominated by the samples for the design obtained using the direct approach.

When comparing the results of the direct approach with the results of the two-phase procedure, it can be seen that the two-phase approach results in designs with a better probability region. Nevertheless, the direct approach is a good compromise when the additional construction of the deterministic Pareto optimal front is too expensive. It can be expected that the results of the direct approach can be improved using a different approximation of the local Pareto optimal front, as presented in Section 4.3.3.

4.4.4. Results for Geometrical Uncertainties

As a further application for robust design with multiple objectives, geometrical uncertainties of the airfoil are assumed. Given the spatial coordinates ξ , the uncertainties are restricted to 80 percent of the airfoil surface A defined by $\tilde{A} = \{\xi \in A \mid \xi_1 \leq 0.8\}$, neglecting the trailing edge. This is a modeling assumption that avoids unreasonable geometrical representations like intersections of the lower and upper surface. Note that the shape A is defined by the design variables \mathbf{u} in each iteration of the optimization algorithm. The coordinates of the perturbed airfoil \tilde{s} are given by Equation (4.65) using a random process ψ . The expected value of the random process is $E(\psi) = 0$ and the covariance function is assumed to be the Gaussian covariance function (4.64) with $b = 0.001$ and $l = 0.1$. The covariance function models the correlation between different points on the airfoil. Note that the perturbations represented by the random field are chosen independently of the parameterization that is used for optimization. This makes sense since they cannot be represented by the chosen design variables.

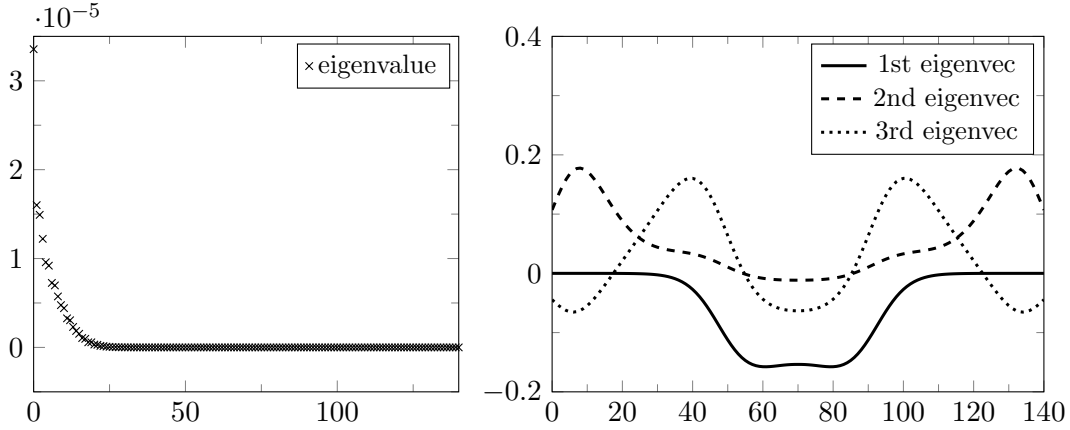


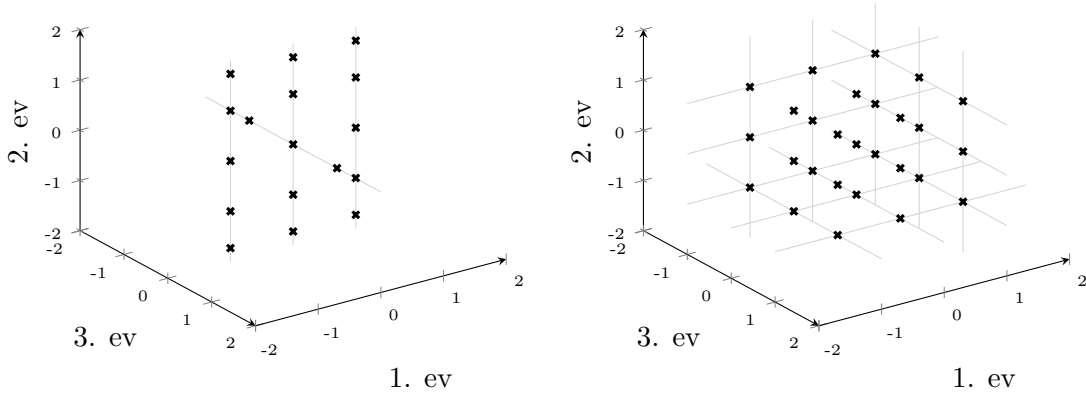
Figure 4.17.: Eigenvalues (left) and first three eigenvectors (right) of the covariance matrix of the original design.

Describing the geometrical uncertainties with the help of a random process results in an infinite-dimensional random space. For applying the non-intrusive polynomial chaos approach, it is necessary to provide finite-dimensional and uncorrelated random variables. This can be done by making use of a KL expansion, as described in Section 4.2.1. The eigenvalue problem (4.7) is discretized based on the mesh points on the surface, leading to the solution of a discrete eigenvalue problem with a covariance matrix of size (141×141) for the discretized airfoil. The problem is solved using a standard linear algebra package (linalg in Python, Eigen in C++). Figure 4.17 shows the decaying eigenvalues on the left side and the first three eigenvectors on the right side for the NACA airfoil, which is the initial shape for the optimization. During the optimization, the eigenvalue problem is solved for each perturbed shape. In each optimization step, the basis constructed based on the first three eigenvectors is chosen as an approximation of the random process. The random process is approximated using a KL expansion ψ_K as given in Equation (4.10) with $K = 3$. As a result the infinite-dimensional random space is reduced to a finite-dimensional random space with uncorrelated variables $Z_1, Z_2, Z_3 \sim N(0, 1)$. Thus, a non-intrusive polynomial chaos approach can be applied, such that the expected value of the drag coefficient c_d , for example, is expressed as

$$E(c_d(\mathbf{y}, \mathbf{u}, \psi_3(\boldsymbol{\xi}, \boldsymbol{\omega})) = \int_{\mathbb{R}^3} c_d(\mathbf{y}, \mathbf{u}, \sum_{i=1}^3 \sqrt{\lambda_i} \mathbf{v}_i(\boldsymbol{\xi}) z_i(\boldsymbol{\omega})) f_{Z_1}(z_1) f_{Z_2}(z_2) f_{Z_3}(z_3) dz_1 dz_2 dz_3. \quad (4.77)$$

The integral in the three-dimensional random space is approximated with the help of dimension-adaptive sparse grids based on Gauss-Hermite quadrature using the sparse Gauss-Hermite rule. A grid that was obtained with the help of the dimension-adaptive strategy is shown in Figure 4.18a. It was produced for the initial design. For illustration purposes, the original Gauss-Hermite quadrature rule was used. With a prescribed tolerance of $5 \cdot 10^{-5}$ for the estimated error, the dimension-adaptive strategy leads to 17

4. Robustness in Multi-Objective Optimization



(a) Grid: One important dimension.

(b) Grid: Equally important dimensions.

Figure 4.18.: Dimension-adaptive sparse grid for the original design with 17 grid points (a) and a different design with 27 grid points (b).

grid points. The dimension belonging to the third eigenvector is less important and the dimension belonging to the second eigenvector is the most important. The grid in Figure 4.18b consists of 27 grid points and is an example of a design in which all eigenvectors are found to be equally important. The number of grid points obtained for the sparse Gauss-Hermite rule with a tolerance of 10^{-5} for the estimated error is usually between 19 and 41.

For the multi-objective optimization, one may prescribe the same constraints for the objective functions as for the deterministic optimization. The geometric constraint shall hold for the unperturbed design. The same way of constraint handling is applied to the moment coefficient to avoid the high number of constraints resulting from a discretized semi-infinite formulation.⁷ In the hybrid optimization strategy, the Kriging surrogate model is built for the expected values. The expected values are obtained with the help of the dimension-adaptive sparse grid rule based on the truncated KL expansion. In the gradient-based strategy, the gradient of the expected function is evaluated for the respective sparse grid. When using the one-shot strategy for the gradient-based optimization, the eigenvectors and eigenvalues of the KL expansion and the dimension-adaptive sparse grid are set up for the modified design in each optimization step. Since the state and adjoint are not fully converged, one may choose the threshold for the estimated approximation quality of the sparse grid rule depending on the state and adjoint residuals. However, the adaption has to be chosen carefully to increase the computational efficiency. Otherwise, it may deteriorate the convergence of the overall methodology. In the nested gradient-based optimization strategy using IPOPT, the expansion and the dimension-adaptive sparse grid are set up for each design based on the converged state and adjoint solution with a fixed tolerance for the estimated error.

⁷Alternatively one may constrain the local minima for the distribution of c_m , which can be, for example, approximated based on the non-intrusive PC expansion [236]. An approximation of the local minima is possible when only finite perturbations of the design are allowed.

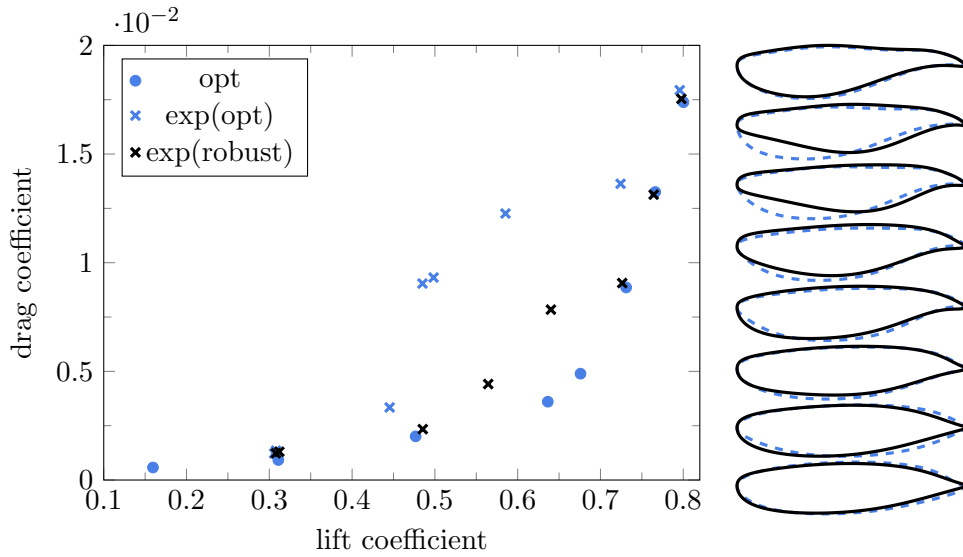


Figure 4.19.: Expected values of the robust Pareto optimal points (black cross) and solutions of the deterministic optimization (blue point), with corresponding expected values (blue crosses) when considering geometrical uncertainties. Corresponding designs are given on the right side.

Figure 4.19 shows the resulting expected objective function values represented by the black crosses. The blue crosses are the expected outcomes of the designs found by the deterministic optimization. The blue dots mark the corresponding deterministic objective function values. When evaluating the objective functions of the unperturbed design for the robust optimal designs, the resulting objective function values are close to the respective expected values.

The found robust Pareto optimal points are, in general, close to the original deterministic front. One may observe that the robust designs when considering geometrical uncertainties differ from the deterministic designs, which can be seen in the right part of the figure. The solid shapes represent the robust designs, and the dashed shapes are the designs resulting from the optimization without considering any uncertainties.

5. Conclusion and Outlook

5.1. Conclusion

In the present thesis, a strategy for multi-objective robust design was proposed and applied in the context of aerodynamic shape optimization.

A constraint method that makes use of a hybrid strategy for the single-objective optimization methods was used to solve the multi-objective optimization problem. The hybrid optimization strategy combines a global search on a surrogate model with the help of the expected improvement method and a gradient-based optimization strategy that makes use of derivatives obtained from a robust discrete adjoint approach supported by algorithmic differentiation. The hybridization was performed with the help of a sequential approach that uses the solution of the optimization on the surrogate model as a starting point for a gradient-based optimization on the full model.

The efficiency of the gradient-based strategy was further improved by using a one-shot approach for solving the constrained optimization problems. The single-step one-shot approach was extended to additional equality constraints. The proposed method was analyzed theoretically and numerically. Furthermore, a way of treating inequality constraints was proposed by introducing the handling of bound constraints in the framework.

Suitable robustness measures were introduced to enable multi-objective robust design. The losses in objective space were used to account for the combined effect of uncertainties in objective space. Different formulations of optimization problems based on the expected losses were introduced. For obtaining the functions of interest, the uncertainties were propagated through the model with the help of a non-intrusive polynomial chaos approach. It was made use of dimension-adaptive sparse grids to reduce the computational costs of the quadrature method needed in the polynomial chaos approach.

The proposed strategy was implemented in the optimization framework of SU2 and the outer structure of the solver. For analysis purposes, it was applied for the shape optimization of a two-dimensional airfoil in a transonic flow regime. The multi-objective optimization task was to minimize the drag coefficient and maximize the lift coefficient under additional constraints on the moment coefficient and the thickness of the airfoil. For robust design, uncertainties in the flow conditions and the geometry were considered separately.

To present a detailed conclusion for the results of the analysis, it is referred back to the original research goals defined in the introduction.

Hybrid Strategies for Constraint Methods in MOO Three different types of constraint methods for multi-objective optimization were applied, the equality constraint, the epsilon-constraint method, and the normal constraint method. The epsilon-constraint

5. Conclusion and Outlook

method has shown to be the most reliable when it comes to finding only Pareto optimal solutions and is easy to implement in a given constraint-based optimization framework. When it is of interest to find a good distribution of points on the containing surface, it can be advantageous to use the normal constraint method.

When defining the research goals, it was pointed out that the success of constraint methods for multi-objective optimization depends on the choice of the algorithm for solving the constrained single-objective problems. With success, it is referred to both, the success in finding a Pareto optimal point, which is a global solution to the single-objective optimization problem, and the success of finding a solution to the optimization problem in a robust and computationally efficient manner. The hybrid strategy was especially suggested for this purpose and could be successfully applied for multi-objective aerodynamic shape optimization. The found set of solutions shows that the strategy is robust and can find Pareto optimal solutions with a reasonable computational effort for the presented application. The number of function evaluations needed for the heuristic strategy could be controlled by using a surrogate model. The use of the surrogate model is advantageous for multi-objective optimization since the model can be reused during the optimization.

For finding a global solution, the proposed strategy will outperform a scalarization approach based on a purely gradient-based strategy. Concerning efficiency, it will outperform a scalarization approach based on a global search due to the use of the surrogate model. It can be expected that the computational effort is, in the worst case, comparable to the costs of conventional direct Pareto approaches like a multi-objective evolutionary algorithms. In general, one may expect to obtain a better performance, especially for higher-dimensional objective spaces.

Constraint Handling in One-Shot Strategies The single-step one-shot strategy was extended to include additional equality constraints. In the convergence analysis, it was shown that the resulting strategy could give descent on an exact penalty function under specific conditions. The conditions were formulated for the penalty parameters, the design space preconditioner B , and the preconditioner for the multiplier update \tilde{B} . The analysis has shown that the conditions for the preconditioners B and \tilde{B} can be separated via the choice of suitable penalty parameters. As a result, one may use the design space preconditioner for the single-step one-shot method without additional constraints. Furthermore, a preconditioner for the multiplier update was proposed.

A projection strategy was proposed to handle bound constraints in the one-shot strategy and to enable the handling of inequality constraints. The formulation of bound constraints guarantees that a local search can be conducted in the vicinity of a promising starting point. As a result, the one-shot strategy could be easily integrated into the hybrid strategy for solving MOO problems with the help of constraint methods.

Numerical results have shown the success of the proposed strategy for aerodynamic shape optimization with multiple objectives. Several Pareto optimal points could be found with the help of the equality constraint method. The one-shot method with additional equality constraints and bound constraints converged with a retardation factor of

10 to 13 when related to the iterations of the state equation. Thus, the additional handling of equality constraints resulted in an increase of the computational costs by a factor of 3 to 4 compared to the single-step one-shot method without additional constraints. The computational costs could not be improved by using the proposed preconditioner \tilde{B} instead of a constant factor for the multiplier update. In general, a slower speed of convergence can be expected for the handling of equality constraints in gradient-based optimization, especially when using an indirect treatment of constraints.

Robust Design with Multiple Uncertain Objectives Losses in objective space were introduced as a new measure for robust optimality when considering multiple objectives. The robustness measure was constructed to account for the effects of uncertainty in multi-objective space. Two different formulations for representing losses in objective space were proposed. In the first formulation, the losses could be defined with respect to the deterministic Pareto optimal front. The second formulation was based on different types of estimated losses, including possible losses and definite losses. The estimates can be used if the deterministic Pareto optimal front is not known. As strategies for robust design, both measures were introduced in a robust design problem by using the respective expected values, i.e., expected losses and expected estimated losses.

The strategies were applied for multi-objective robust design of the aerodynamic shape optimization problem considering uncertainties in the operational conditions. The results of the previous considerations for using constraint methods based on a hybrid optimization strategy could be successfully transferred to solve the multi-objective robust design problem. The uncertainties were propagated with the help of a non-intrusive polynomial chaos approach. The numerical analysis of the different strategies to multi-objective robust design has shown the advantage of using expected losses as a robustness measure in comparison to only using the expected values of the individual objective functions. The random outcomes in objective space for the robust optimal design were closer to the deterministic Pareto optimal front. Results have shown that the formulation based on estimated losses can be used as a compromise if the deterministic Pareto optimal front is not available.

Furthermore, geometrical uncertainties were analyzed using an expectation-based approach for multi-objective robust design. A Karhunen-Loève expansion was used to approximate the random process describing the perturbation of the shape. A dimension-adaptive sparse grid strategy was adapted to the multi-objective setting and Gauss-Hermite quadrature rules. The combination of efficient strategies for multi-objective optimization and uncertainty quantification was successfully applied to find robust Pareto optimal points in objective space.

To summarize, one may conclude that the research goals defined in the introduction were met. The present work has shown that it is advantageous to use robustness measures for multi-objective robust design that account for the effect of uncertainty in objective space. Furthermore, the proposed strategies for robust multi-objective design with PDE constraints was successfully applied.

5.2. Outlook

The present work has introduced a strategy for robust design with multiple objectives and applied it to a two-dimensional aerodynamic shape optimization problem. As an outlook, several future tasks will be formulated in the following.

Due to the modular structure of the implemented strategies, i.e., the multi-objective optimization framework, the one-shot approach, and the handling of uncertainties, the strategy can be applied to other interesting applications within SU2:

- The strategy can be applied to three-dimensional shape optimization problems in the same aerodynamic shape optimization setting. Furthermore, it can be interesting to look at other problems in fluid dynamics, e.g., internal flow problems. In this context, unsteady flow problems introduce an additional dimension and need the adjustment of the implementation of the one-shot strategy.
- The multi-objective optimization approach is predestined to be used for problems with objective functions arising from multiple disciplines. Here, one might think of aero-structural design problems with objective functions from aerodynamics and structural mechanics. In this context, it can also be interesting to look at topology optimization problems (see also Section A.2.3 of the appendix).

For three-dimensional internal flow in a U-shaped bend, the robust design formulation based on the expectation value was tested in the present work. However, the obtained results were not of scientific value for the given setting. The optimization of expected values resulted in the same Pareto optimal designs as for the deterministic problem (see Section A.3.4 in the appendix). This shows that it is favorable to solve problems that naturally arise in engineering design instead of constructing an artificial problem.

Aspects of high-performance computing were not explicitly considered in the present thesis. However, there is a lot of potential for parallelization. Most of the underlying tasks can be parallelized without significant manipulation:

- The constrained single-objective optimization problems arising from multi-objective optimization are fully decoupled. When optimizing several problems in parallel, the Kriging surrogate model cannot be reused. A strategy for communication is, in general, not needed.
- The training of the Kriging model in the expected improvement method is already parallelized with OpenMP. This type of parallelization was utilized in the present work.
- The flow solver in SU2 makes use of MPI parallelization, which was also employed for obtaining the presented results.
- The evaluation at quadrature points in the non-intrusive polynomial chaos method can be performed in parallel. The values need to be communicated to construct the expected value or the variance. For dimension-adaptive sparse grid rules, however,

the distribution of tasks is not straightforward since specific information is needed during the construction of the sparse grid.

- In the multi-step one-shot method, the state and adjoint equation can be iterated asynchronously in a parallel fashion. However, this requires a strategy for load balancing.

It remains to discuss where parallelization is the most fruitful for efficiency and, based on this discussion, how to distribute tasks.

In the following, an outlook is given for the specific topics of the present work.

Hybrid Strategies for Constraint Methods in MOO The presented approach to multi-objective optimization has shown to provide optimal solutions with a reasonable computational effort. The number of design variables gives the only restriction due to the use of a surrogate model. However, there exist other types of hybrid strategies designed for multi-objective optimization. A task of future work will be to benchmark the algorithm by comparing it to other hybrid strategies. In this context, it can be especially interesting to look at hybridization or surrogate modeling strategies on the level of the multi-objective solution strategy. This can be pursued differently:

- **Hybridization strategy:** The hybridization can be done before scalarization. One may realize this by applying a direct Pareto method based on a heuristic strategy and refining the solutions by formulating appropriate scalarization problems, which are solved with a gradient-based strategy. In [275], members of the population of a multi-objective algorithm are refined based on a weighted sum of objective functions. An application of the epsilon-constraint method can be found in [288], where the a local search is applied to intermediate populations by randomly selecting members and setting the inequality constraints according to the objective function values of the members. The resulting optimization problem is solved with a sequential quadratic programming approach.

Such a method can potentially decrease the computational effort. However, Bosman and de Jong [22] conclude from their studies that improving the efficiency of multi-objective evolutionary approaches with local search techniques requires a carefully constructed hybridization and is, in general, not as fruitful as for single-objective optimization. Furthermore, it is not as easy to maintain diversity as when directly applying a scalarization approach. A systematic approach for scanning the Pareto optimal front is not possible. Therefore, a lot of thought has to be put in the construction of the evolutionary algorithm, as well as in the selection of members. Only then, a fair comparison is possible. An idea can be to systematically subdivide the objective function space and build separate populations that are refined with the help of a hybrid strategy.

- **Surrogate modeling:** The optimization on a surrogate model can also be done on the level of the multi-objective optimization strategy. Jeong et al. [138] construct a Kriging model for each fitness function of a multi-objective evolutionary

5. Conclusion and Outlook

strategy. A promising strategy is given in [61, 60]. The multi-objective evolutionary method aims at maximizing the hypervolume, which is a performance metric for MOO, and the idea of a multi-objective expected improvement is proposed for searching on the surrogate model. It can be interesting to apply a hybridization in this context. Similarly to applying a hybrid method in a direct Pareto approach, running a direct approach on a surrogate model can be highly beneficial for computational performance. However, a systematic search for Pareto optimal points is not straightforward.

When applying the hybridization on the level of the single-objective optimization problem, it can be of interest to further investigate the constraint handling in the expected improvement method. The constraint-weighted expected improvement method gives promising results. However, since an optimization on the surrogate model is cheap, it can be of interest to use a projection-based method for constraint handling. Furthermore, it can be interesting to add a further level of hybridization for the expected improvement method by using gradients. As already explained in Section 2.4.2, the extension of a gradient-based local search to additional constraints is not straightforward. One idea can be to use a suitable projection strategy and take one or more steps into the obtained direction. Also, when using a penalty method for constraint handling, the respective derivative gives a descent direction.

The scalarization approach was chosen to make use of the derivatives obtained from an AD-based discrete adjoint method. For future work, it can be interesting to use these derivatives for a search method in multi-objective space. One example is the directed search method proposed in [245]. The general idea is that a descent direction in objective space at a design \mathbf{x} can be described by a direction \mathbf{d} fulfilling

$$D\mathbf{F}(\mathbf{x})\mathbf{d} \leq \mathbf{0} \text{ and } D\mathbf{F}(\mathbf{x})\mathbf{d} \neq \mathbf{0}, \quad (5.1)$$

where $D\mathbf{F}(\mathbf{x})$ describes the Jacobian of the vector of objective functions in \mathbf{x} . The strategy for choosing a descent direction is then to solve the underdetermined linear system of equations $D\mathbf{F}(\mathbf{x})\mathbf{d} = -\mathbf{a}$ with a prescribed negative right hand side $-\mathbf{a}$, and to iteratively update the design with the known formula for descent

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma D\mathbf{F}(\mathbf{x}_k)^\dagger \mathbf{a}. \quad (5.2)$$

The choice of \mathbf{a} defines the strategy. When solving the optimization problem

$$\max_{\mathbf{a}} \left\| \sum a_i (JF)_i(\mathbf{x}_k) \right\|^2 \text{ s.t. } \sum a_i = 1, \mathbf{a} \geq 0$$

in each iteration k , the method gives steepest descent in objective space. The problem is inspired from the Kuhn-Tucker conditions for optimality of Theorem (4), but a similar problem can be derived when formulating the dual problem of $\min_{\mathbf{d}} \max_i (JF(\mathbf{x})\mathbf{d})_i + 0.5\|\mathbf{d}\|^2$ [65]. Different choices of \mathbf{a} lead to the interpretation of search directions for specific scalarization approaches. Furthermore, the idea of descent directions in objective space can be used in a continuation strategy to direct the search in objective space along a continuous Pareto optimal front. Continuation strategies are based on finding a search direction tangent to the Pareto optimal front linearized in the current Pareto optimal solution.

Constraint Handling in One-Shot Strategies In the present thesis, the theoretical and numerical analysis of the one-shot method with additional equality constraints has been brought to a stage where it is ready for further application. However, some interesting research aspects that can be of interest for future work have emerged during its development:

- **Mesh deformation:** In Section 3.3.1.1, the additional constraint for the transformation of the computational mesh according to the design variables has been introduced, i.e., $\mathbf{x} = M(\mathbf{u})$. Usually, the mesh is transformed by solving a linear system of equations. Depending on the mesh size and the definition of design variables, this can lead to a significant computational effort. One may expect the mesh deformation to be cheaper than a solution of the state equation. However, when done alongside a single iteration of the state and adjoint equation, the computational costs become more significant. Multi-step one-shot methods can be constructed in such a way that the computational costs are less significant. However, the question arises if the mesh has to be deformed exactly in each iterative step of the one-shot method. Instead, as proposed in Section 3.3.1.1, one could introduce a fixed-point iteration for the mesh deformation that uses a suitable preconditioner for ensuring a good mesh quality. Then the fixed-point iteration can be treated in the one-shot approach like an additional state equation with corresponding multipliers $\bar{\mathbf{x}}$.
- **Convergence Analysis:** Although more practicable conditions were derived for the parameters of the one-shot method, it has proven to be difficult to apply them for the construction of a preconditioner for the constraint multipliers. For the specific precondition proposed in Section 3.2.4.2, the penalty parameter α for fulfilling the state equation and the constraints can potentially be restricted too much, leading to a slow convergence or in the worst case there does not exist a feasible value for α . In any case, due to the given conditions, the construction of the preconditioner has to ensure a small condition number. If, for example, low-rank modifications are used for the construction, the spectrum has to be controlled (see e.g. [88]). Future research in this direction can be interesting. The question is whether one can find more practicable conditions or a better alternative for constructing B_k . Also, the convergence analysis of the proposed method is still missing the aspect of global convergence, which usually results in additional conditions on the objective and constraint functions.
- **Inequality Constraints:** Ideas for handling inequality constraints based on a modified multiplier update or the introduction of slack variables were already formulated in the present thesis. For the specific application to aerodynamic design, the inequality constraints resulting from the epsilon-constraint method will be active in an optimal solution, such that a numerical analysis does not give representative results. For future work, it is important to analyze the proposed strategy for different types of problems, where the use of inequality constraints is obligatory.

5. Conclusion and Outlook

It can be interesting to extend the ideas of the augmented one-shot iteration to the interesting topics of the current research in the context of the one-shot method:

- **Unsteady problems:** Since a lot of applications in engineering design are unsteady, it can be interesting to apply the one-shot strategy with additional equality constraints to optimization problems with unsteady PDE constraints. The developments in [100] can be used as a basis for numerical results. The idea is to differentiate a modified time marching scheme that uses a reduced number of pseudo-time iterations in each time step. Thus, the inner pseudo-time iterations and the outer iterations are reversed, and for implementing such a strategy, the solvers have to be further adjusted.
- **Variants:** In [20], a variant of the single-step one-shot method that neglects the penalty term of the adjoint residual in the doubly augmented Lagrangian is introduced, i.e., $\beta = 0$. The strategy is more efficient since the gradient of the doubly augmented Lagrangian does not contain any second-order derivatives. Convergence results can be shown with the help of the upper bound L for the adjoint time lag such that $\|\bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k\| \leq L\|\mathbf{y}_{k+1} - \mathbf{y}_k\|$ for all iterations k . A lower bound on α that depends on L can be derived, namely $\alpha > 2L/(1 - \rho)$. It may be of interest to analyze how this strategy can be integrated for the one-shot method augmented with equality constraints. This requires the derivation of new conditions on the penalty parameters and the preconditioners.

Robust Design with Multiple Uncertain Objectives The proposed robustness measure based on losses in objective space was constructed to provide solutions that are expected to show less deviation from the Pareto optimal front. For the presented application, this could be successfully shown. However, some important remarks can be made:

- **Robustness measure:** It can be interesting to search for solutions that are close to the deterministic Pareto optimal front. However, the definition of robustness is, in general, a matter of own preferences. Depending on the problem at hand, it could also be of interest to maximize gains or to reduce variations in objective space. It remains to be evaluated how well the proposed strategies perform for different applications and what kind of results can be observed. However, the main takeaway is that in a multi-objective setting, it is of importance to consider a combined effect of uncertainties.
- **Robust design formulation:** In the proposed formulations for robust design, the expected losses are introduced as a constraint function with a prescribed upper bound. Thus, they replace the role of the standard deviation. This is a natural choice since the standard deviation is also based on the expected value of a distance measure. However, since the aim is to reduce the distance to the Pareto optimal front, it can also be used in future work as a performance measure to be optimized.
- **Level of approximation** In the present work, the information on losses was reduced to the expected value of losses. For future work, it can be interesting to

additionally consider higher-order moments, e.g., the variance of losses. In general, the loss function describes a distribution in objective space. If it can be sufficiently approximated with feasible computational effort, e.g., with the help of a polynomial chaos expansion, the probability density function itself or a cumulative distribution function can be sampled. For this specific case, it will be possible to use losses or similar distance measures to compare distributions. An interesting research aspect can be to use probability quantiles of the distribution for minimizing risks, like a minimization of the worst-case threshold defined with the help of the (conditional) value at risk (see e.g. [230]).

Furthermore, concerning the methodology for propagating uncertainties, future work may involve the following aspects:

- **Representation of random input:** Following the suggestions in literature, in the present work it was assumed that the random input variables are normally distributed. The quadrature rules used to approximate the quantities of interest evaluate the respective functions at physically meaningful solutions \mathbf{y} . To prevent the occurrence of nonphysical input values, one might want to choose a truncated distribution for uncertainties in the operational conditions and geometrical uncertainties. However, this requires the construction of proper orthogonal polynomials. Or, as an approximation, suitable transformations of the original distribution have to be formulated. When introducing the angle of attack as an additional uncertainty in the operating conditions one might think of an alternative strategy. A representation of uncertainties in the magnitude and the direction of the free-stream velocity could be a smoother description and the assumption of correlation can be included. Furthermore, when a distribution can be described based on measurements, it should be used for uncertainty propagation in a more realistic setting.
- **Adaptivity:** The level of approximation of the non-intrusive polynomial chaos method was kept constant during the optimization. Only adaptivity with respect to the dimension was considered when using dimension-adaptive sparse grids. However, one may pursue an adaptive approach to increase the approximation quality during the optimization algorithm. This is implicitly achieved when choosing a fixed order of quadrature for the operational conditions since the objective function becomes smoother near the optimal solution. But this effect cannot be guaranteed for general application cases.
- **Inverse Uncertainty Quantification:** Further developments of the overall strategy can be the application for inverse uncertainty quantification with multiple objectives. Here, one might think of a typical Bayesian inverse problem of fitting model parameters to different types of outputs, given only noisy measurements (see e.g. [237]).

To summarize the outlook, it can be stated that the concepts and strategies proposed in the present thesis offer the potential for further developments and applications that go beyond the scope of this work.

Bibliography

- [1] S. Ahmed and A. Shapiro. Solving Chance-Constrained Stochastic Programs via Sampling and Integer Programming. In Z. L. Chen and S. Raghavan, editors, *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, Tutorials in Operations Research, chapter 12, pages 261–269. INFORMS, Catonsville, MD, USA, 2008.
- [2] T. Albring, M. Sagebaum, and N. R. Gauger. Development of a Consistent Discrete Adjoint Solver in an Evolving Aerodynamic Design Framework. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2015-3240, Dallas, TX, USA, 2015. American Institute of Aeronautics and Astronautics.
- [3] T. Albring, M. Sagebaum, and N. R. Gauger. Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2016-3518, Washington, DC, USA, 2016. American Institute of Aeronautics and Astronautics.
- [4] T. Albring, B. Zhou, N. R. Gauger, and M. Sagebaum. An Aerodynamic Design Framework based on Algorithmic Differentiation. *ERCRAFT Bulletin*, 102:10–16, 2015.
- [5] W. K. Anderson and V. Venkatakrishnan. Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation. *Computers & Fluids*, 28(4-5):443–480, 1999.
- [6] E. Arian and S. Ta’asan. Multigrid One Shot Methods for Optimal Control Problems: Infinite Dimensional Control. Technical Report ICASE Report No. 94-52, NASA Langley Research Center, Hampton, VA, USA, 1994.
- [7] E. Arian and S. Ta’asan. Shape Optimization in One Shot. In *Optimal Design and Control*, pages 23–40, Boston, MA, USA, 1995. Birkhäuser.
- [8] O. B. Augusto, F. Bennis, and S. Caro. Multiobjective Engineering Design Optimization Problems: A Sensitivity Analysis Approach. *Pesquisa Operacional*, 32(3):575–596, 2012.
- [9] G. Avigad and J. Branke. Embedded Evolutionary Multi-Objective Optimization for Worst Case Robustness. In M. Keijzer, editor, *Proceedings of the 10th annual Conference on Genetic and Evolutionary Computation*, pages 617–624, New York, NY, USA, 2008. ACM.

Bibliography

- [10] G. Avigad, A. Moshaiiov, and N. Brauner. MOEA-Based Approach to Delayed Decisions for Robust Conceptual Design. In F. Rothlauf, J. Branke, S. Cagnoni, D. Wolfe Corne, R. Drechsler, Y. Jin, P. Machado, E. Marchiori, J. Romero, G. D. Smith, and G. Squillero, editors, *Applications of Evolutionary Computing: EvoWorkshops 2005*, Lecture Notes in Computer Science 3449, pages 584–589, Lausanne, Switzerland, 2005. Springer Berlin Heidelberg.
- [11] S. Azarm, B. J. Reynolds, and S. Narayanan. Comparison of Two Multiobjective Optimization Techniques With and Within Genetic Algorithms. In *Proceedings of the 25th ASME Design Engineering Technical Conference*, pages 1–10, Las Vegas, NV, USA, 1999. ASME.
- [12] I. Babuška, F. Nobile, and R. Tempone. A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.
- [13] C. Barrico and C. H. Antunes. A New Approach to Robustness Analysis in Multi-Objective Optimization. In *7th International Conference on Multi-Objective Programming and Goal Programming*, Tours, France, 2006.
- [14] M. Basseur and E. Zitzler. Handling Uncertainty in Indicator-Based Multiobjective Optimization. *International Journal of Computational Intelligence Research*, 2(3):255–272, 2006.
- [15] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear Programming with Multiple Objective Functions: Step Method (STEM). *Mathematical Programming*, 1(1):366–375, 1971.
- [16] M. P. Bendsøe. Optimal Shape Design as a Material Distribution Problem. *Structural Optimization*, 1(4):193–202, 1989.
- [17] H. P. Benson. Existence of Efficient Solutions for Vector Maximization Problems. *Journal of Optimization Theory and Applications*, 26(4):569–580, 1978.
- [18] D. P. Bertsekas. Projected Newton Methods for Optimization Problems with Simple Constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.
- [19] M. Blommaert. *Automated Magnetic Divertor Design for Optimal Power Exhaust*. Phd thesis, RWTH Aachen, Aachen, Germany, 2016.
- [20] M. Blommaert, W. Dekeyser, M. Baelmans, N. R. Gauger, and D. Reiter. A practical globalization of one-shot optimization for optimal design of tokamak divertors. *Journal of Computational Physics*, 328:399–412, 2017.
- [21] 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(3):2172–2192, 2009.

- [22] P. A. N. Bosman and E. D. de Jong. Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 755–762, Washington, DC, USA, 2005. ACM Press.
- [23] T. Bosse. Augmenting the One-Shot Framework by Additional Constraints. *Optimization Methods and Software*, 31(6):1132–1148, 2016.
- [24] T. Bosse, N. R. Gauger, A. Griewank, S. Günther, and V. Schulz. One-Shot Approaches to Design Optimization. In *Trends in PDE Constrained Optimization*, International Series of Numerical Mathematics 165, pages 43–66. Springer International Publishing Switzerland, 2014.
- [25] T. Bosse, L. Lehmann, and A. Griewank. Adaptive Sequencing of Primal, Dual, and Design Steps in Simulation Based Optimization. *Computational Optimization and Applications*, 57(3):731–760, 2014.
- [26] F. Boukouvala and M. G. Ierapetritou. Derivative-Free Optimization for Expensive Constrained Problems Using a Novel Expected Improvement Objective Function. *AIChE Journal*, 60(7):2462–2474, 2014.
- [27] G. Burgreen and O. Baysal. Three-Dimensional Aerodynamic Shape Optimization of Wings Using Sensitivity Analysis. In *32nd Aerospace Sciences Meeting and Exhibit*, AIAA 94-0094, Reston, VA, USA, 1994. American Institute of Aeronautics and Astronautics.
- [28] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [29] R. H. Cameron and W. T. Martin. The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals. *Annals of Mathematics*, 48(2):385–392, 1947.
- [30] G. Carpentieri, B. Koren, and M. J. L. van Tooren. Adjoint-Based Aerodynamic Shape Optimization on Unstructured Meshes. *Journal of Computational Physics*, 224(1):267–287, 2007.
- [31] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making – Theory and Methodology*. North-Holland Publishing Company, New York, NY, USA, 1983.
- [32] A. Charnes and W. W. Cooper. Chance-Constrained Programming. *Management Science*, 6(1):73–79, 1959.
- [33] A. Charnes, W. W. Cooper, and R. O. Ferguson. Optimal Estimation of Executive Compensation by Linear Programming. *Management Science*, 1(2):138–151, 1955.

Bibliography

- [34] H. Chen, Q. Wang, R. Hu, and P. Constantine. Conditional Sampling and Experiment Design for Quantifying Manufacturing Error of Transonic Airfoil. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2011-658, pages 1–17, Reston, VA, USA, 2011. American Institute of Aeronautics and Astronautics.
- [35] W. Chen, J. K. Allen, K.-L. Tsui, and F. Mistree. A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors. *ASME Journal of Mechanical Design*, 118(4):478, 1996.
- [36] W. Chen, M. M. Wiecek, and J. Zhang. Quality Utility – A Compromise Programming Approach to Robust Design. *Journal of Mechanical Design*, 121(2):179, 1999.
- [37] K. K. Choi and B. D. Youn. On Probabilistic Approaches for Reliability-Based Design Optimization (RBDO). In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA 2002-5472, Atlanta, GA, USA, 2002. American Institute of Aeronautics and Astronautics.
- [38] B. Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326, 1994.
- [39] M. Ciprian, V. Pediroda, and C. Poloni. Multi Criteria Decision Aiding Techniques to Select Designs After Robust Design Optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Proceedings of the 4th International Conference on Evolutionary Multicriterion Optimization*, Lecture Notes in Computer Science 4403, pages 619–632, Matsushima, Japan, 2007. Springer-Verlag Berlin Heidelberg.
- [40] C. W. Clenshaw and A. R. Curtis. A Method for Numerical Integration on an Automatic Computer. *Numerische Mathematik*, 2(1):197–205, 1960.
- [41] H. Corley. A New Scalar Equivalence for Pareto Optimization. *IEEE Transactions on Automatic Control*, 25(4):829–830, 1980.
- [42] R. Courant. Variational Methods for the Solution of Problems of Equilibrium and Vibrations. *Bulletin of the American Mathematical Society*, 49(1):1–24, 1943.
- [43] N. A. C. Cressie. *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., revised edition, 1993.
- [44] I. Das and J. E. Dennis. Normal-Boundary Intersection: An Alternate Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems. Technical Report NASA Contractor Report 201616, ICASE Report No. 96-62, NASA Langley Research Center, Hampton, VA, USA, 1996.
- [45] I. Das and J. E. Dennis. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, 14(1):63–69, 1997.

- [46] M. J. Daskilewicz, B. J. German, T. T. Takahashi, S. Donovan, and A. Shajanian. Effects of Disciplinary Uncertainty on Multi-Objective Optimization in Aircraft Conceptual Design. *Structural and Multidisciplinary Optimization*, 44(6):831–846, 2011.
- [47] K. Deb and H. Gupta. Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Proceedings of 3rd International Conference on Evolutionary Multi-Objective Optimization*, Lecture Notes in Computer Science 3410, pages 150–164, Guanajuato, Mexico, 2005. Springer.
- [48] K. Deb and H. Gupta. Introducing Robustness in Multi-Objective Optimization. *Evolutionary Computation*, 14(4):463–494, 2006.
- [49] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [50] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [51] G. Debreu. Definite and Semidefinite Quadratic Forms. *Econometrica*, 20(2):295, 1952.
- [52] X. Du and W. Chen. Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design. *Journal of Mechanical Design*, 122(4):385, 2000.
- [53] X. Du, A. Sudjianto, and W. Chen. An Integrated Framework for Optimization Under Uncertainty Using Inverse Reliability Strategy. *Journal of Mechanical Design*, 126(4):562, 2004.
- [54] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 54(3):828–846, 2016.
- [55] M. Ehrgott and S. Ruzika. Improved ϵ -Constraint Method for Multiobjective Programming. *Journal of Optimization Theory and Applications*, 138(3):375–396, 2008.
- [56] M. Ehrgott and D. Tenfelde-Podehl. Computation of Ideal and Nadir Values and Implications for Their Use in MCDM Methods. *European Journal of Operational Research*, 151(1):119–139, 2003.
- [57] G. Eichfelder, J. Niebling, and S. Rocktäschel. An algorithmic approach to multi-objective optimization with decision uncertainty. *Journal of Global Optimization*, 2019.

Bibliography

- [58] M. S. Eldred and J. Burkardt. Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, AIAA 2009-976, Orlando, FL, US, 2009. American Institute of Aeronautics and Astronautics.
- [59] E. A. Elsayed and A. Chen. Optimal Levels of Process Parameters for Products with Multiple Characteristics. *International Journal of Production Research*, 31(5):1117–1132, 1993.
- [60] M. T. M. Emmerich. *Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels*. PhD thesis, University of Dortmund, Dortmund, Germany, 2005.
- [61] M. T. M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Third International Conference (EMO 2005)*, Lecture Notes in Computer Science 3410, pages 62–76, Guanajuato, Mexico, 2005. Springer Berlin Heidelberg.
- [62] T. Erfani and S. V. Utyuzhnikov. Directed Search Domain: A Method for Even Generation of the Pareto Frontier in Multiobjective Optimization. *Engineering Optimization*, 43(5):467–484, 2011.
- [63] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, New York, NY, USA, 1968.
- [64] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe. Many-Objective Optimization: An Engineering Design Perspective. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Third International Conference (EMO 2005)*, Lecture Notes in Computer Science 3410, pages 14–32, Guanajuato, Mexico, 2005. Springer Berlin Heidelberg.
- [65] J. Fliege and B. F. Svaiter. Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
- [66] C. A. Floudas and O. Stein. The Adaptive Convexification Algorithm: A Feasible Point Method for Semi-Infinite Programming. *SIAM Journal on Optimization*, 18(4):1187–1208, 2008.
- [67] C. M. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA, USA, 1993. Morgan Kaufman Publishers.

- [68] P. D. Frank and G. R. Shubin. A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem. *Journal of Computational Physics*, 98(1):74–89, 1992.
- [69] A. Gambier. Performance Evaluation of Several Multi-objective Optimization Methods for Control Purposes. In *Proceedings of the 8th Asian Control Conference*, pages 1067–1071, Kaohsiung, Taiwan, 2011. IEEE.
- [70] N. R. Gauger. Semi-Automatic Transition from Simulation to Optimization. In P. Neittaanmaeki, J. Periaux, and T. Tuovinen, editors, *Evolutionary Methods for Design, Optimization and Control Eurogen 2007*, Jyväskylä, Finland, 2008. CIMNE.
- [71] N. R. Gauger, A. Griewank, A. Hamdi, C. Kratzstein, E. Özkaya, and T. Slawig. Automated Extension of Fixed Point PDE Solvers for Optimal Design with Bounded Retardation. In *Constrained Optimization and Optimal Control for Partial Differential Equations*, volume 160, pages 99–122. Springer Basel, Basel, Switzerland, 2012.
- [72] N. R. Gauger, A. Griewank, and J. Riehme. Extension of Fixed Point PDE Solvers for Optimal Design by One-Shot Method. *European Journal of Computational Mechanics*, 17(1-2):87–102, 2008.
- [73] N. R. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm. Automatic Differentiation of an Entire Design Chain for Aerodynamic Shape Optimization. In C. Tropea, S. Jakirlic, H. J. Heinemann, R. Henke, and H. Hönlinger, editors, *New Results in Numerical and Experimental Fluid Mechanics VI*, pages 454–461. Springer Berlin Heidelberg, 2007.
- [74] M. A. Gelbart, J. Snoek, and R. P. Adams. Bayesian Optimization with Unknown Constraints. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 250–259, Quebec City, Quebec, Canada, 2014. AUAI Press Arlington, VA, US.
- [75] F. W. Gembicki. *Vector Optimization for Control with Performance and Parameter Sensitivity Indices*. Ph.d. thesis, Case Western Reserve University, Cleveland, OH, USA, 1974.
- [76] F. W. Gembicki and Y. Y. Haimes. Approach to Performance and Sensitivity Multiobjective Optimization: The Goal Attainment Method. *IEEE Transactions on Automatic Control*, 20(6):769–771, 1975.
- [77] T. Gerstner and M. Griebel. Dimension-Adaptive Tensor-Product Quadrature. *Computing*, 71(1):65–87, 2003.
- [78] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer New York, New York, NY, USA, 1991.

Bibliography

- [79] T. Ghisu, J. P. Jarrett, and G. T. Parks. Robust Design Optimization of Airfoils with Respect to Ice Accretion. *Journal of Aircraft*, 48(1):287–304, 2011.
- [80] M. B. Giles. On the Iterative Solution of Adjoint Equations. In G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann, editors, *Automatic Differentiation of Algorithms*, pages 145–151. Springer New York, New York, NY, USA, 2002.
- [81] M. B. Giles and N. A. Pierce. An Introduction to the Adjoint Approach to Design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000.
- [82] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1):99–131, 2005.
- [83] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, USA, 1989.
- [84] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, fourth edition, 1983.
- [85] G. H. Golub and J. H. Welsch. Calculation of Gauss Quadrature Rules. *Mathematics of Computation*, 23(106):221–221, 1969.
- [86] D. Gottlieb and D. Xiu. Galerkin Method for Wave Equations with Uncertain Coefficients. *Communications in Computational Physics*, 3(2):505–518, 2008.
- [87] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. H. Lee, P. Ranjan, G. Wells, and S. M. Wild. Modeling an Augmented Lagrangian for Blackbox Constrained Optimization. *Technometrics*, 58(1):1–11, 2016.
- [88] C. Greif and J. M. Varah. Minimizing the Condition Number for Small Rank Modifications. *SIAM Journal on Matrix Analysis and Applications*, 29(1):82–97, 2006.
- [89] A. Griewank. Projected Hessians for Preconditioning in One-Step One-Shot Design Optimization. In G. Di Pillo and M. Roma, editors, *Large Scale Nonlinear Optimization*, pages 151–171. Springer US, Boston, MA, USA, 2006.
- [90] A. Griewank, C. Bischof, G. Corliss, A. Carle, and K. Williamson. Derivative Convergence for Iterative Equation Solvers. *Optimization Methods and Software*, 2(3-4):321–355, 1993.
- [91] A. Griewank and C. Faure. Reduced Functions, Gradients and Hessians from Fixed-Point Iterations for State Equations. *Numerical Algorithms*, 30(2):113–139, 2002.
- [92] A. Griewank and D. Kressner. Time-lag in Derivative Convergence for Fixed Point Iterations. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, 3:87–102, 2005.

- [93] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2008.
- [94] M. D. Guenov, M. Nunez, and A. Gondhalekar. Comparing Design Margins in Robust and Deterministic Design Optimization. In *Evolutionary and Deterministic Methods for Design, Optimization and Control*, pages 530–542, Capua, Italy, 2011. CIRA.
- [95] C. R. Gumbert, P. A. Newman, and G. J.-W. Hou. High-Fidelity Computational Optimization for 3-D Flexible Wings: Part II – Effect of Random Geometric Uncertainty on Design. *Optimization and Engineering*, 6(1):139–156, 2005.
- [96] S. Gunawan and S. Azarm. On a Combined Multi-Objective and Feasibility Robustness Method for Design Optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, USA, 2004. American Institute of Aeronautics and Astronautics.
- [97] S. Gunawan and S. Azarm. A Feasibility Robust Optimization Method Using Sensitivity Region Concept. *Journal of Mechanical Design*, 127(5):858–865, 2005.
- [98] S. Gunawan and S. Azarm. Multi-objective Robust Optimization Using a Sensitivity Region Concept. *Structural and Multidisciplinary Optimization*, 29(1):50–60, 2005.
- [99] S. Günther. *Ein Vergleich zweier One-Shot-Methoden*. Master thesis, University of Trier, Trier, Germany, 2012.
- [100] S. Günther, N. R. Gauger, and Q. Wang. Simultaneous Single-Step One-Shot Optimization with Unsteady PDEs. *Journal of Computational and Applied Mathematics*, 294:12–22, 2016.
- [101] R. T. Haftka. Simultaneous Analysis and Design. *AIAA Journal*, 23(7):1099–1103, 1985.
- [102] A. Hamdi and A. Griewank. Properties of an Augmented Lagrangian for Design Optimization. *Optimization Methods and Software*, 25(4):645–664, 2010.
- [103] A. Hamdi and A. Griewank. Reduced Quasi-Newton Method for Simultaneous Design and Optimization. *Computational Optimization and Applications*, 49(3):521–548, 2011.
- [104] S. P. Han. A Globally Convergent Method for Nonlinear Programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, 1977.
- [105] Z.-H. Han, S. Görtz, and R. Zimmermann. On Improving Efficiency and Accuracy of Variable-Fidelity Surrogate Modeling in Aero-data for Loads Context. In *Proceedings of European Air and Space Conference*, Manchester, UK, 2009.

Bibliography

- [106] L. Hascoët and V. Pascual. The Tapenade Automatic Differentiation Tool. *ACM Transactions on Mathematical Software*, 39(3):1–43, 2013.
- [107] L. Hascoët, M. Vázquez, and A. Dervieux. Automatic Differentiation for Optimum Design, Applied to Sonic Boom Reduction. In V. Kumar, M. L. Gavrilova, C. J. K. Tan, and P. L’Ecuyer, editors, *International Conference on Computational Science and its Applications*, pages 85–94, Montreal, Canada, 2003. Springer Berlin Heidelberg.
- [108] G. Hassan and C. D. Clack. Multiobjective Robustness for Portfolio Optimization in Volatile Environments. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1507–1514, New York, NY, USA, 2008. ACM Press.
- [109] J. Hays, A. Sandu, C. Sandu, and D. Hong. Parametric Design Optimization of Uncertain Ordinary Differential Equation Systems. *Journal of Mechanical Design*, 134(8), 2012.
- [110] S. B. Hazra. An Efficient Method for Aerodynamic Shape Optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, USA, 2004. American Institute of Aeronautics and Astronautics.
- [111] S. B. Hazra. Direct Treatment of State Constraints in Aerodynamic Shape Optimization Using Simultaneous Pseudo-Time-Stepping. *AIAA Journal*, 45(8):1988–1997, 2007.
- [112] S. B. Hazra. Multigrid One-shot Method for Aerodynamic Shape Optimization. *SIAM Journal on Scientific Computing*, 30(3):1527–1547, 2008.
- [113] S. B. Hazra and V. Schulz. Simultaneous Pseudo-Timestepping for PDE-Model Based Optimization Problems. *BIT Numerical Mathematics*, 44(3):457–472, 2004.
- [114] S. B. Hazra and V. Schulz. Simultaneous Pseudo-Timestepping for Aerodynamic Shape Optimization Problems with State Constraints. *SIAM Journal on Scientific Computing*, 28(3):1078–1099, 2006.
- [115] S. B. Hazra, V. Schulz, and J. Brezillon. Simultaneous Pseudo-Time Stepping for 3D Aerodynamic Shape Optimization. *Journal of Numerical Mathematics*, 16(2):139–161, 2008.
- [116] S. B. Hazra, V. Schulz, J. Brezillon, and N. R. Gauger. Aerodynamic Shape Optimization using Simultaneous Pseudo-Timestepping. *Journal of Computational Physics*, 204(1):46–64, 2005.
- [117] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.
- [118] R. M. Hicks and P. A. Henne. Wing Design by Numerical Optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

- [119] J. H. Holland. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [120] J. Horn and N. Nafpliotis. Multiobjective Optimization Using the Niche Pareto Genetic Algorithm. Technical report, IlliGAL (Illinois Genetic Algorithms Laboratory), Urbana-Champaign, IL, USA, 1993.
- [121] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, Orlando, FL, USA, 1994. IEEE.
- [122] S. Hosder, R. W. Walters, and M. Balch. Efficient Sampling for Non-Intrusive Polynomial Chaos Applications with Multiple Uncertain Input Variables. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2007-1939, Honolulu, HI, USA, 2007. American Institute of Aeronautics and Astronautics.
- [123] S. Hosder, R. W. 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, Reno, NV, USA, 2006. American Institute of Aeronautics and Astronautics.
- [124] E. J. Hughes. Constraint Handling with Uncertain and Noisy Multi-Objective Evolution. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 2, pages 963–970, Seoul, South Korea, 2001. IEEE.
- [125] T.-C. Hung and K.-Y. Chan. Uncertainty Quantification of Pareto Optimum in Multiobjective Problems. In *9th World Congress on Structural and Multidisciplinary Optimization*, Shizuoka, Japan, 2011.
- [126] T.-C. Hung and K.-Y. Chan. Multi-Objective Design and Tolerance Allocation for Single- and Multi-Level Systems. *Journal of Intelligent Manufacturing*, 24(3):559–573, 2013.
- [127] L. Huyse. Solving Problems of Optimization Under Uncertainty as Statistical Decision Problems. In *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, AIAA 2001-1519, Seattle, WA, USA, 2001. American Institute of Aeronautics and Astronautics.
- [128] L. Huyse and R. M. Lewis. Aerodynamic Shape Optimization of Two-Dimensional Airfoils Under Uncertain Conditions. Technical report, NASA Langley Research Center, 2001.
- [129] L. Huyse, S. L. Padula, R. M. Lewis, and W. Li. Probabilistic Approach to Free-Form Airfoil Shape Optimization Under Uncertainty. *AIAA Journal*, 40(9):1764–1772, 2002.

Bibliography

- [130] J. Ide and A. Schöbel. Robustness for Uncertain Multi-Objective Optimization: A Survey and Analysis of Different Concepts. *OR Spectrum*, 38(1):235–271, 2016.
- [131] A. Iollo, G. Kuruvila, and S. Ta’asan. Pseudotime Method for Shape Design of Euler Flows. *AIAA Journal*, 34(9):1807–1813, 1996.
- [132] A. Ismail-Yahaya and A. Messac. Effective Generation of the Pareto Frontier: The Normalized Normal Constraint Method. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2002-1232, Denver, CO, USA, 2002. American Institute of Aeronautics and Astronautics.
- [133] A. Ismail-Yahaya and A. Messac. Effective Generation of the Pareto Frontier Using the Normal Constraint Method. In *40th Aerospace Sciences Meeting and Exhibit*, AIAA 2002-0178, Reno, NV, USA, 2002. American Institute of Aeronautics and Astronautics.
- [134] K. Ito, K. Kunisch, V. Schulz, and I. Gherman. Approximate Nullspace Iterations for KKT Systems. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1835–1847, 2010.
- [135] A. Jameson. Aerodynamic Design via Control Theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.
- [136] A. Jameson. Optimum Aerodynamic Design Using CFD and Control Theory. In *AIAA 12th Computational Fluid Dynamics Conference*, AIAA 95-1729, pages 926–949, San Diego, CA, USA, 1995. American Institute of Aeronautics and Astronautics.
- [137] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes. In *14th Fluid and Plasma Dynamics Conference*, AIAA 81-1259, Palo Alto, CA, USA, 1981. American Institute of Aeronautics and Astronautics.
- [138] S. Jeong, K. Suzuki, S. Obayashi, and M. Kurita. Improvement of Nonlinear Lateral Characteristics of Lifting-Body Type Reentry Vehicle Using Optimization Algorithm. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, AIAA 2007-2893, Rohnert Park, CA, USA, 2007. American Institute of Aeronautics and Astronautics.
- [139] R. Jin, X. Du, and W. Chen. The Use of Metamodeling Techniques for Optimization Under Uncertainty. *Structural and Multidisciplinary Optimization*, 25(2):99–116, 2003.
- [140] L. Kaland, J. C. De Los Reyes, and N. R. Gauger. One-Shot Methods in Function Space for PDE-Constrained Optimal Control Problems. *Optimization Methods and Software*, 29(2):376–405, 2014.

- [141] K. Karhunen. Zur Spektraltheorie Stochastischer Prozesse. *Annales Academiae Scientiarum Fennicae*, 34, 1946.
- [142] C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
- [143] I. Y. Kim and O. L. de Weck. Adaptive Weighted Sum Method for Multiobjective Optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2004-4322, Albany, NY, USA, 2004. American Institute of Aeronautics and Astronautics.
- [144] I. Y. Kim and O. L. de Weck. Adaptive Weighted-Sum Method for Bi-Objective Optimization: Pareto Front Generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158, 2005.
- [145] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: Piecewise Multilinear Hierarchical Sparse Grid Interpolation in Matlab. *ACM Transactions on Mathematical Software*, 31(4):561–579, 2005.
- [146] J. D. Knowles. ParEGO: A hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- [147] V. M. Korivi, A. C. Taylor III, P. A. Newman, G. J.-W. Hou, and H. E. Jones. An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives. *Journal of Computational Physics*, 113(2):336–346, 1994.
- [148] V. M. Korivi, A. C. Taylor III, P. A. Newman, and G. W. Hou. An Incremental Strategy for Calculating Consistent Discrete CFD Sensitivity Derivatives. In *4th Symposium on Multidisciplinary Analysis and Optimization*, AIAA 92-4746. NASA Langley Research Center, 1992.
- [149] D. Kraft. A Software Package for Sequential Quadratic Programming. Technical report, DFVLR, Oberpfaffenhofen, Germany, 1988.
- [150] C. Kratzenstein. *Assessment of the One-Shot Strategy for the Calibration of Marine Ecosystem Models*. Phd thesis, Christian-Albrecht-Universität zu Kiel, Kiel, Germany, 2015.
- [151] C. Kratzenstein and T. Slawig. Simultaneous Model Spin-Up and Parameter Identification with the One-Shot Method in a Climate Model Example. *An International Journal of Optimization and Control: Theories & Applications*, 3(2):99–110, 2013.
- [152] D. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, 52(6):119–139, 1952.

Bibliography

- [153] H. W. Kuhn and A. W. Tucker. Nonlinear Programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, CA, USA, 1951. University of California Press.
- [154] L. Kusch, T. Albring, A. Walther, and N. R. Gauger. A One-Shot Optimization Framework with Additional Equality Constraints Applied to Multi-Objective Aerodynamic Shape Optimization. *Optimization Methods and Software*, 33(4-6):694–707, 2018.
- [155] L. Kusch and N. R. Gauger. The One-Shot Method in SU2. *Proceedings in Applied Mathematics and Mechanics*, 16(1):699–700, 2016.
- [156] L. Kusch and N. R. Gauger. Robust Airfoil Design in the Context of Multi-objective Optimization. *Computational Methods in Applied Sciences*, 48:391–403, 2019.
- [157] L. Kusch and N. R. Gauger. Robustness Measures for Multi-Objective Robust Design. *Computational Methods in Applied Sciences (Accepted for Publication)*, 2020.
- [158] L. Kusch, N. R. Gauger, and M. Spiller. Efficient Calculation of Pareto-Optimal Points for Shape Optimization. In *Full Paper Compilation: Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems - EUROGEN 2013, ISBN 978-84-617-2141-2*, Universidad de Las Palmas de Gran Canaria, Spain, 2014.
- [159] J. Laurenceau and P. Sagaut. Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging. *AIAA Journal*, 46(2):498–507, 2008.
- [160] B. S. Lazarov and O. Sigmund. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86:765–781, 2011.
- [161] 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*, 181:9–44, 2002.
- [162] D. S. Lee, L. F. Gonzalez, J. Périaux, and K. Srinivas. Robust Design Optimisation Using Multi-Objective Evolutionary Algorithms. *Computers & Fluids*, 37(5):565–583, 2008.
- [163] M. Li, R. Silva, F. Guimaraes, and D. Lowther. A New Robust Dominance Criterion for Multiobjective Optimization. *IEEE Transactions on Magnetics*, 51(3):1–4, 2015.
- [164] W. Li, L. Huyse, and S. L. Padula. Robust Airfoil Optimization to Achieve Drag Reduction Over a Range of Mach Numbers. *Structural and Multidisciplinary Optimization*, 24:38–50, 2002.

- [165] J. G. Lin. Circuit Design under Multiple Performance Objectives. In *Proceedings of 1974 IEEE International Symposium on Circuits and Systems*, pages 549–552, San Francisco, CA, USA, 1974. IEEE.
- [166] J. G. Lin. Multiple-Objective Problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints. *IEEE Transactions on Automatic Control*, 21(5):641–650, 1976.
- [167] J. G. Lin. Proper Equality Constraints and Maximization of Index Vectors. *Journal of Optimization Theory and Applications*, 20(2):215–244, 1976.
- [168] J. G. Lin. Proper Inequality Constraints and Maximization of Index Vectors. *Journal of Optimization Theory and Applications*, 21(4):505–521, 1977.
- [169] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer (Translated by Mitter, S.K.), New York, NY, USA, 1971.
- [170] M. Loève. *Probability Theory II*. Springer Verlag New-York, New York, NY, USA, 1978.
- [171] G. J. Loeven, J. A. Witteveen, and H. Bijl. Probabilistic Collocation: An Efficient Non-Intrusive Approach for Arbitrarily Distributed Parametric Uncertainties. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2007-317, Reno, NV, USA, 2007. American Institute of Aeronautics and Astronautics.
- [172] G. J. A. Loeven and H. Bijl. Airfoil Analysis with Uncertain Geometry Using the Probabilistic Collocation Method. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference*, AIAA 2008-2070, Schaumburg, IL, USA, 2008. American Institute of Aeronautics and Astronautics.
- [173] G. J. A. Loeven, J. A. S. Witteveen, and H. Bijl. Efficient Uncertainty Quantification Using a Two-Step Approach with Chaos Collocation. In *Proceedings of the European Conference on Computational Fluid Dynamics*, Egmond aan Zee, The Netherlands, 2006.
- [174] X. Ma and N. Zabaras. An Adaptive Hierarchical Sparse Grid Collocation Algorithm for the Solution of Stochastic Differential Equations. *Journal of Computational Physics*, 228(8):3084–3113, 2009.
- [175] B. Maar and V. Schulz. Interior Point Multigrid Methods for Topology Optimization. *Structural and Multidisciplinary Optimization*, 19(3):214–224, 2000.
- [176] N. Marco and F. Beux. Multilevel Optimization: Application to One-Shot Shape Optimum Design. Technical report, Inria, Rocquencourt, France, 1993.
- [177] S. A. Marglin. *Public Investment Criteria*. MIT Press, Cambridge, MA, USA, 1967.

Bibliography

- [178] L. Martinelli, A. Jameson, and F. Grasso. A Multigrid Method for the Navier Stokes Equations. In *24th Aerospace Sciences Meeting*, AIAA 86-0208, Reston, VA, USA, 1986. American Institute of Aeronautics and Astronautics.
- [179] L. Mathelin, M. Y. Hussaini, and T. A. Zang. A Stochastic Collocation Algorithm for Uncertainty Analysis. Technical report, NASA, Hampton, VA, USA, 2003.
- [180] L. Mathelin, M. Y. Hussaini, and T. A. Zang. Stochastic Approaches to Uncertainty Quantification in CFD Simulations. *Numerical Algorithms*, 38(1-3):209–236, 2005.
- [181] G. Matheron. Principles of Geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
- [182] C. A. Mattson and A. Messac. Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. *Optimization and Engineering*, 6(1):85–115, 2005.
- [183] G. Mavrotas. Effective Implementation of the ϵ -Constraint Method in Multi-Objective Mathematical Programming Problems. *Applied Mathematics and Computation*, 213(2):455–465, 2009.
- [184] M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245, 1979.
- [185] F. Menter. Zonal Two Equation $k\text{-}\omega$ Turbulence Models for Aerodynamic Flows. In *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, AIAA 93-2906, Orlando, FL, USA, 1993. American Institute of Aeronautics and Astronautics.
- [186] J. Mercer. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 209:415–446, 1909.
- [187] A. Messac and A. Ismail-Yahaya. Multiobjective Robust Design Using Physical Programming. *Structural and Multidisciplinary Optimization*, 23(5):357–371, 2002.
- [188] A. Messac and C. A. Mattson. Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. *AIAA Journal*, 42(10):2101–2111, 2004.
- [189] A. Messac and G. J. Sundararaj. A Robust Design Approach Using Physical Programming. In *38th Aerospace Sciences Meeting and Exhibit*, AIAA 2000-0562, Reno, NV, USA, 2000. American Institute of Aeronautics and Astronautics.
- [190] K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, MA, USA, 1999.

- [191] K. Miettinen and M. M. Mäkelä. Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. *Optimization*, 34(3):231–246, 1995.
- [192] N. Milickovic, M. Lahanas, D. Baltas, and N. Zamboglou. Comparison of Evolutionary and Deterministic Multiobjective Algorithms for Dose Optimization in Brachytherapy. In E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science 1993, pages 168–180, Zurich, Switzerland, 2001. Springer Berlin Heidelberg.
- [193] E. A. Minisci, H. Liquiang, and M. Vasile. Multidisciplinary Design of a micro-USV for Re-entry Operations. In *AAS/AIAA Astrodynamics Specialists Conference*, AIAA 2010-7968, Toronto, Ontario, Canada, 2010. American Institute of Aeronautics and Astronautics.
- [194] J. Mockus, V. Tiesis, and A. Zilinskas. The Application of Bayesian Methods for Seeking the Extremum. In L. Dixon and G. Szego, editors, *Towards global optimisation*, volume 2, pages 117–129. North-Holland, Amsterdam, 1978.
- [195] B. Mohammadi. A New Optimal Shape Design Procedure for Inviscid and Viscous Turbulent Flows. *International Journal for Numerical Methods in Fluids*, 25(2):183–203, 1997.
- [196] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, England, UK, 2001.
- [197] A. Molina-Cristobal, G. T. Parks, and P. J. Clarkson. Finding Robust Solutions to Multi-Objective Optimisation Problems Using Polynomial Chaos. In *Proceedings of the 6th ASMO UK/ISSMO Conference on Engineering Design Optimization*, Oxford, England, UK, 2006.
- [198] J. Moore and R. Chapman. Application of Particle Swarm to Multiobjective Optimization. Technical report, Department of Computer Science and Software Engineering Department, Auburn University, Auburn, AL, USA, 1999.
- [199] J.-D. Müller and P. Cusdin. On the Performance of Discrete Adjoint CFD Codes Using Automatic Differentiation. *International Journal for Numerical Methods in Fluids*, 47(8-9):939–945, 2005.
- [200] S. K. Nadarajah and A. Jameson. A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization. In *38th Aerospace Sciences Meeting and Exhibit*, AIAA 2000-0667, Reston, Virginia, 2000. American Institute of Aeronautics and Astronautics.

Bibliography

- [201] S. K. Nadarajah and A. Jameson. Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization. In *15th AIAA Computational Fluid Dynamics Conference*, AIAA 2001-2530, Reston, VA, USA, 2001. American Institute of Aeronautics and Astronautics.
- [202] U. Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2012.
- [203] M. Navarro, J. A. S. Witteveen, and J. Blom. Polynomial Chaos Expansion for General Multivariate Distributions with Correlated Variables. Technical report, CWI, Amsterdam, The Netherlands, 2014.
- [204] P. A. Newman, G. J.-W. Hou, H. E. Jones, A. C. Taylor III, and V. M. Korivi. Observations on Computational Methodologies for Use in Large-Scale, Gradient-Based, Multidisciplinary Design. Technical report, NASA Langley Research Center, Hampton, VA, USA, 1992.
- [205] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal. An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids. *Computers & Fluids*, 33(9):1131–1155, 2004.
- [206] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, New York, NY, USA, 2006.
- [207] Y.-S. Ong, P. B. Nair, and K. Y. Lum. Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, 2006.
- [208] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences Volume 153. Springer New York, New York, NY, USA, 2003.
- [209] S. Osher and J. A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [210] E. Özkaya and N. R. Gauger. Single-step One-shot Aerodynamic Shape Optimization. In *Optimal Control of Coupled Systems of Partial Differential Equations*, pages 191–204. Birkhäuser Basel, Basel, Switzerland, 2009.
- [211] E. Özkaya and N. R. Gauger. Automatic Transition from Simulation to One-Shot Shape Optimization with Navier-Stokes Equations. *GAMM-Mitteilungen*, 33(2):133–147, 2010.

- [212] E. Özkaya and N. R. Gauger. Global Aerodynamic Design Optimization via Primal-Dual Aggregation Method. In A. Dillman, G. Heller, E. Krämer, C. Wagner, C. Tropea, and J. S., editors, *New Results in Numerical and Experimental Fluid Mechanics XII, DGLR 2018*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design Volume 142, pages 48–57. Springer, 2020.
- [213] E. Özkaya, L. Kusch, T. Albring, and N. R. Gauger. Robust Aerodynamic Shape Optimization Using Adjoint Assisted Surrogate Modeling. In *EUROGEN 2017*, Extended Abstract, Madrid, Spain, 2017.
- [214] M. Padulo. *Computational Engineering Design Under Uncertainty: An Aircraft Conceptual Design Perspective*. PhD thesis, Cranfield University, Cranfield, UK, 2009.
- [215] M. Padulo, M. S. Campobasso, and M. D. Guenov. Novel Uncertainty Propagation Method for Robust Aerodynamic Design. *AIAA Journal*, 49(3):530–543, 2011.
- [216] F. Palacios, J. Alonso, K. Duraisamy, M. Colonno, J. Hicken, A. Aranake, A. Campos, S. Copeland, T. Economon, A. Lonkar, T. Lukaczyk, and T. Taylor. Stanford University Unstructured (SU2): An Open-Source Integrated Computational Environment for Multi-Physics Simulation and Design. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-0287, Grapevine, TX, USA, 2013. American Institute of Aeronautics and Astronautics.
- [217] V. Pareto. *Cours D’Economie Politique*. Rouge, Lausanne, Switzerland, 1896.
- [218] V. Pareto. *Manual of Political Economy (translated from Manuale di Economica Politica, Societa Editrice Libreria, Milano, Italy, 1906)*. Augustus M. Kelley, Publishing, New York, NY, USA, 1971.
- [219] A. Parkinson, C. Sorensen, and N. Pourhassan. A General Approach for Robust Optimal Design. *Journal of Mechanical Design*, 115(1):74–80, 1993.
- [220] J. J. Pignatiello. Strategies for Robust Multiresponse Quality Engineering. *IIE Transactions*, 25(3):5–15, 1993.
- [221] O. Pironneau. On Optimum Design in Fluid Mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.
- [222] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, UK, 1969.
- [223] M. J. D. Powell. A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In *Numerical Analysis*, volume 630, pages 144–157. Springer Berlin Heidelberg, 1978.

Bibliography

- [224] M. M. Putko, P. A. Newman, A. C. Taylor III, and L. L. Green. Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives. In *15th AIAA Computational Fluid Dynamics Conference*, AIAA 2001-2528, Anaheim, CA, USA, 2001. American Institute of Aeronautics and Astronautics.
- [225] T. Ray. Constrained Robust Optimal Design using a Multiobjective Evolutionary Algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 419–424, Honolulu, HI, USA, 2002. IEEE.
- [226] M. Renardy and R. C. Rogers. *An Introduction to Partial Differential Equations*. Texts in Applied Mathematics Volume 13. Springer-Verlag, New York, NY, USA, 2004.
- [227] J.-M. Renders and S. Flasse. Hybrid Methods Using Genetic Algorithms for Global Optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 26(2):243–258, 1996.
- [228] A. Resmini, J. Peter, and D. Lucor. Sparse Grids-Based Stochastic Approximations with Applications to Aerodynamics Sensitivity Analysis. *International Journal for Numerical Methods in Engineering*, 106(1):32–57, 2015.
- [229] N. Richert. *Erweiterung eines One-Shot-Verfahrens zur Berücksichtigung einer zusätzlichen Nebenbedingung*. Diplomarbeit, Universität Paderborn, Paderborn, Germany, 2014.
- [230] A. Ruszczyński and A. Shapiro. Optimization of Risk Measures. In G. Clafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, chapter Part I, pages 119–157. Springer London, London, UK, 2006.
- [231] M. Sagebaum, T. Albring, and N. R. Gauger. High-Performance Derivative Computations using CoDiPack. *ACM Transactions on Mathematical Software*, 45(4), 2019.
- [232] R. Sanchez, R. Palacios, T. D. Economou, H. L. Kline, J. J. Alonso, and F. Palacios. Towards a Fluid-Structure Interaction Solver for Problems with Large Deformations Within the Open-Source SU2 Suite. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2016-0205, San Diego, CA, USA, 2016. American Institute of Aeronautics and Astronautics.
- [233] S. Sankaranarayanan, R. T. Haftka, and R. K. Kapania. Truss Topology Optimization with Simultaneous Analysis and Design. *AIAA Journal*, 32(2):420–424, 1994.
- [234] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, USA, 1985. Lawrence Erlbaum Associates Inc.

- [235] C. Schillings, S. Schmidt, and V. Schulz. Efficient Shape Optimization for Certain and Uncertain Aerodynamic Design. *Computers & Fluids*, 46(1):78–87, 2011.
- [236] C. Schillings and V. Schulz. On the Influence of Robustness Measures on Shape Optimization with Stochastic Uncertainties. *Optimization and Engineering*, 16(2):347–386, 2015.
- [237] C. Schillings and C. Schwab. Sparse, Adaptive Smolyak Quadratures for Bayesian Inverse Problems. *Inverse Problems*, 29(6):065011, 2013.
- [238] S. Schmidt, C. Ilic, N. R. Gauger, and V. Schulz. Shape Gradients and Their Smoothness for Practical Aerodynamic Design Optimization. Technical Report SPP1253-10-03, DFG, 2008.
- [239] S. Schmidt, C. Ilic, V. Schulz, and N. R. Gauger. Three-Dimensional Large-Scale Aerodynamic Shape Optimization Based on Shape Calculus. *AIAA Journal*, 51(11):2615–2627, 2013.
- [240] M. Schonlau, W. J. Welch, and D. R. Jones. Global Versus Local Search in Constrained Optimization of Computer Models. In N. Flournoy, W. Rosenberger, and W. Wong, editors, *New Developments and Applications in Experimental Design*, IMS Lecture Notes - Monograph Series Volume 34, pages 11–25. Institute of Mathematical Statistics, Hayward, California, US, 1998.
- [241] V. Schulz. Solving Discretized Optimization Problems by Partially Reduced SQP Methods. *Computing and Visualization in Science*, 1(2):83–96, 1998.
- [242] V. Schulz and C. Schillings. On the Nature and Treatment of Uncertainties in Aerodynamic Design. *AIAA Journal*, 47(3):646–654, 2009.
- [243] V. Schulz and C. Schillings. Optimal Aerodynamic Design under Shape Uncertainties. Technical report, Uni Trier, Trier, Germany, 2009.
- [244] V. Schulz and C. Schillings. Problem Formulations and Treatment of Uncertainties in Aerodynamic Design. *AIAA Journal*, 47(3):646–654, 2009.
- [245] O. Schütze, A. Lara, and C. A. Coello Coello. The directed search method for unconstrained multi-objective optimization problems. *Proceedings of the EVOLVE—A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation*, 2011.
- [246] C. Schwab and R. A. Todor. Karhunen–Loève Approximation of Random Fields by Generalized Fast Multipole Methods. *Journal of Computational Physics*, 217(1):100–122, 2006.
- [247] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, volume 20, pages 151–160, New York, NY, USA, 1986. ACM Press.

Bibliography

- [248] M. Sekishiro, G. Venter, and V. Balabanov. Combined Kriging and Gradient-Based Optimization Method. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2006-7091, Portsmouth, VA, USA, 2006. American Institute of Aeronautics and Astronautics.
- [249] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [250] P. K. Shukla and K. Deb. On Finding Multiple Pareto-Optimal Solutions using Classical and Evolutionary Generating Methods. *European Journal of Operational Research*, 181(3):1630–1652, 2007.
- [251] R. C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. Computational Science and Engineering. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014.
- [252] S. Smolyak. Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions. *Doklady Akademii Nauk SSSR*, 148(5):1042–1045, 1963.
- [253] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [254] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- [255] R. E. Steuer and E.-U. Choo. An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. *Mathematical Programming*, 26(3):326–344, 1983.
- [256] K. Stöbener, P. Klein, M. Horsch, K. Küfer, and H. Hasse. Parametrization of Two-Center Lennard-Jones Plus Point-Quadrupole Force Field Models by Multicriteria Optimization. *Fluid Phase Equilibria*, 411:33–42, 2016.
- [257] J. Su and J. E. Renaud. Automatic Differentiation in Robust Optimization. *AIAA Journal*, 35(6):1072–1079, 1997.
- [258] G. Sun, G. Li, S. Zhou, H. Li, S. Hou, and Q. Li. Crashworthiness Design of Vehicle by using Multiobjective Robust Optimization. *Structural and Multidisciplinary Optimization*, 44(1):99–110, 2011.
- [259] S. Sundaresan, K. Ishii, and D. R. Houser. A Robust Optimization Procedure with Variations On Design Variables and Constraints. *Engineering Optimization*, 24(2):101–117, 1995.
- [260] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

- [261] K. Svanberg. The Method of Moving Asymptotes – A New Method for Structural Optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.
- [262] S. Ta’asan. One Shot Methods for Optimal Control of Distributed Parameter Systems I: Finite Dimensional Control. Technical Report ICASE Report 91-2, NASA Langley Research Center, Hampton, VA, USA, 1991.
- [263] S. Ta’asan. Pseudo-Time Methods for Constrained Optimization Problems Governed by PDE. Technical report, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA, 1995.
- [264] S. Ta’asan, G. Kuruvila, and M. D. Salas. Aerodynamic Design and Optimization in One Shot. In *30th Aerospace Sciences Meeting and Exhibit*, AIAA 92-0025, Reno, NV, USA, 1992. American Institute of Aeronautics and Astronautics.
- [265] G. Taguchi. Performance Analysis Design. *International Journal of Production Research*, 16(6):521–530, 1978.
- [266] G. Taguchi. *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Asian Productivity Organization, 1989.
- [267] R. Tavakoli and H. Zhang. A Nonmonotone Spectral Projected Gradient Method for Large-Scale Topology Optimization Problems. *Numerical Algebra, Control and Optimization*, 2(2):395–412, 2012.
- [268] A. C. Taylor III, G. J.-W. Hou, and V. M. Korivi. Methodology for Calculating Aerodynamic Sensitivity Derivatives. *AIAA Journal*, 30(10):2411–2419, 1992.
- [269] J. Teich. Pareto-Front Exploration with Uncertain Objectives. In E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. W. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science 1993, pages 314–328, Zurich, Switzerland, 2001. Springer Berlin Heidelberg.
- [270] K.-L. Tsui. Robust Design Optimization for Multiple Characteristic Problems. *International Journal of Production Research*, 37(2):433–445, 1999.
- [271] J. Tu, K. K. Choi, and Y. H. Park. A New Study on Reliability-Based Design Optimization. *Journal of Mechanical Design*, 121(4):557–564, 1999.
- [272] D. Tuytens, J. Teghem, P. Fortemps, and K. V. Nieuwenhuyze. Performance of the MOSA Method for the Bicriteria Assignment Problem. *Journal of Heuristics*, 6:295–310, 2000.
- [273] S. V. Utyuzhnikov, J. Maginot, and M. Guenov. Local Pareto Approximation for Multi-Objective Optimization. *Engineering Optimization*, 40(9):821–847, 2008.

Bibliography

- [274] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary Computation and Convergence to a Pareto Front. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H. Iba, and R. Riolo, editors, *Late Breaking Papers at the Genetic Programming Conference*, pages 221–228, Stanford, California, 1998. Morgan Kaufmann.
- [275] A. Vicini and D. Quagliarella. Airfoil and Wing Design Through Hybrid Optimization Strategies. Technical report, American Institute of Aeronautics and Astronautics, Washington, DC, USA, 1998.
- [276] A. Wächter and L. T. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. Technical report, IBM, 2003.
- [277] A. Wächter and L. T. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006.
- [278] A. Walther, N. R. Gauger, L. Kusch, and N. Richert. On an Extension of One-Shot Methods to Incorporate Additional Constraints. *Optimization Methods and Software*, 31(3):494–510, 2016.
- [279] A. Walther and A. Griewank. Getting started with ADOL-C. In U. Naumann and O. Schenk, editors, *Combinatorial Scientific Computing*, pages 181–202. Chapman-Hall CRC Computational Science, 2012.
- [280] A. Walther, L. Kusch, and N. R. Gauger. New Results for the Handling of Additional Equality Constraints in One-Shot Optimization. *Vietnam Journal of Mathematics*, 46(4):825–836, 2018.
- [281] X. Wan and G. E. Karniadakis. Beyond Wiener–Askey Expansions: Handling Arbitrary PDFs. *Journal of Scientific Computing*, 27(1-3):455–464, 2006.
- [282] S. Wang. Second-Order Necessary and Sufficient Conditions in Multiobjective Programming. *Numerical Functional Analysis and Optimization*, 12(1-2):237–252, 1991.
- [283] G. Wasilkowski and H. Wozniakowski. Explicit Cost Bounds of Algorithms for Multivariate Tensor Product Problems. *Journal of Complexity*, 11(1):1–56, 1995.
- [284] M. M. Wiecek, W. Chen, and J. Zhang. Piecewise Quadratic Approximation of the Non-Dominated Set for Bi-Criteria Programs. *Journal of Multi-Criteria Decision Analysis*, 10(1):35–47, 2001.
- [285] N. Wiener. The Homogeneous Chaos. *American Journal of Mathematics*, 60(4):897, 1938.
- [286] J. A. S. Witteveen and G. Iaccarino. Simplex Elements Stochastic Collocation in Higher-Dimensional Probability Spaces. In *51st*

- AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 18th AIAA/ASME/AHS Adaptive Structures Conference 12th*, AIAA 2010-2924, Orlando, FL, USA, 2010. American Institute of Aeronautics and Astronautics.
- [287] S. Xiang and F. Bornemann. On the Convergence Rates of Gauss and Clenshaw-Curtis Quadrature for Functions of Limited Regularity. *SIAM Journal on Numerical Analysis*, 50(5):2581–2587, 2012.
- [288] Xiaolin Hu, Zhangean Huang, and Zhongfan Wang. Hybridization of the Multi-Objective Evolutionary Algorithms and the Gradient-Based Algorithms. In *The 2003 Congress on Evolutionary Computation*, pages 870–877, Canberra, ACT, Australia, 2003. IEEE.
- [289] D. Xiu. Efficient Collocational Approach for Parametric Uncertainty Analysis. *Communications in Computational Physics*, 2(2):293–309, 2007.
- [290] D. Xiu. Fast Numerical Methods for Stochastic Computations: A review. *Communications in Computational Physics*, 5(2-4):242–272, 2009.
- [291] D. Xiu and G. Em Karniadakis. Modeling Uncertainty in Steady State Diffusion Problems via Generalized Polynomial Chaos. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4927–4948, 2002.
- [292] D. Xiu and G. E. Karniadakis. The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.
- [293] D. Xiu, D. Lucor, C.-H. Su, and G. E. Karniadakis. Stochastic Modeling of Flow-Structure Interactions Using Generalized Polynomial Chaos. *Journal of Fluids Engineering*, 124(1):51, 2002.
- [294] S. Y. Yang, L.-J. Park, C. H. Park, and J. W. Ra. A Hybrid Algorithm using Genetic Algorithm and Gradient-Based Algorithm for Iterative Microwave Inverse Scattering. In *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, pages 450–455, Perth, WA, Australia, 1995. IEEE.
- [295] J. Yen, J. C. Liao, B. Lee, and D. Randolph. A Hybrid Approach to Modeling Metabolic Systems Using a Genetic Algorithm and Simplex Method. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 28(2):173–191, 1998.
- [296] P.-L. Yu. A Class of Solutions for Group Decision Problems. *Management Science*, 19(8):936–946, 1973.
- [297] L. Zadeh. Optimality and Non-Scalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, AC-8(1):59–60, 1963.

Bibliography

- [298] T. A. Zang, M. J. Hensch, M. W. Hilburger, S. P. Kenny, J. M. Luckring, P. Maghami, S. L. Padula, and W. J. Stroud. Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles. Technical report, National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, US, 2002.
- [299] M. Zeleny. Compromise Programming. In J. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 262–301. University of South Carolina Press, Columbia, SC, USA, 1973.
- [300] E. Zitzler and L. Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, 1998.
- [301] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

A. Appendix

A.1. Appendix for Chapter 2

A.1.1. Variant of the Equality Constraint Method

One may formulate a variant of the equality constraint method, where the constraint values depend on the objective function to be minimized. The new minimization problem of the j -th step with $j \in 1, \dots, n_s$ and $s \in \{1, \dots, k\}$ is then given by

$$\min_{\mathbf{x} \in \mathcal{D}} f_s(\mathbf{x}) \quad (\text{A.1})$$

$$\text{s.t. } f_i(\mathbf{x}) = f_{i,s}^{(j)} \quad \forall i \in \{1, \dots, k\} \text{ with } i \neq s. \quad (\text{A.2})$$

The updated constraints in the j -th iteration in step $s + 1$ of the method are chosen from the set $\prod_{i \in \{1, \dots, k\}, i \neq s} \mathcal{S}_{i,s}$, where

$$\mathcal{S}_{i,s} = \{f_{i,s}^{(j)} \mid f_{i,s}^{(j)} = f_{i,s}^{(0)} - m \Delta f_{i,s} \text{ for } m = 0, \dots, n_u\}$$

is the set of all possible constraint values $f_{i,s}^{(j)}$ for the i -th objective function for $i = 1, \dots, k$ with $i \neq s$. The starting value is chosen as

$$f_{i,s}^{(0)} = f_i(x_s^*) \quad \forall i \in \{1, \dots, k\} \text{ with } i \neq s.$$

The step size is different for each objective function s to be minimized in step $s + 1$, i.e.,

$$\Delta f_{i,s} = \frac{1}{n_u + 1} (f_i(x_s^*) - f_s^*) \quad \forall i \in \{1, \dots, k\} \text{ with } i \neq s.$$

Again, for $k = 2$ the distance for the j -th iteration is given as $\Delta f_{i,s}^{(j)} = \Delta f_{i,s}$. An optimization problem with $k > 2$ implies that all possible combinations of constraint values for f_i have to be applied.

A.1.2. Variants of the Epsilon-Constraint Method

There exist some variants of the epsilon-constraint method. In the following, two of them are shortly introduced.

Modified Epsilon-Constraint Method A variant of the epsilon-constraint method (see e.g. [41]), sometimes referred to as a hybridization of the weighted sum method and the

A. Appendix

epsilon-constraint method, is to solve the modified constrained optimization problems

$$\begin{aligned} \min \sum_{i=1}^k w_i f_i(\mathbf{x}) \\ \text{s.t. } f_l(\mathbf{x}) \leq f_l^{(j)} \text{ for } l = 1, \dots, k. \end{aligned} \quad (\text{A.3})$$

It can be shown that for $w_i \geq 0$ for $i = 1, \dots, k$ a feasible solution of the problem (A.3) is an optimal solution if and only if it is a Pareto optimal solution (see e.g. [190, p.96]). However, like in the weighted sum method, a goal-oriented scanning is not possible. The reason for this is that the choice of suitable weights w_i and bounds $f_l^{(j)}$ to find a representative distribution of Pareto optimal points is not straightforward.

Improved Epsilon-Constraint Method In [55], the authors propose to use a vector of slack variables $\mathbf{s}^+ \in \mathbb{R}^{k-1}$ (see also Section 2.3.2.4) for reformulating the inequality constraints and to include them in the objective function. This is done by formulating the modified optimization problem

$$\begin{aligned} \min f_i(\mathbf{x}) - \sum_{l=1, l \neq i}^k \lambda_l s_l^+ \\ \text{s.t. } f_l(\mathbf{x}) + s_l^+ \leq f_l^{(j)} \text{ for } l = 1, \dots, k \text{ with } l \neq i, \quad \mathbf{s}^+ \geq \mathbf{0}. \end{aligned} \quad (\text{A.4})$$

This ensures that a feasible solution is Pareto optimal for $\boldsymbol{\lambda} > \mathbf{0}$ and weakly Pareto optimal for $\boldsymbol{\lambda} \geq \mathbf{0}$. Moreover, proper Pareto optimality is given for $\boldsymbol{\lambda} > \mathbf{0}$ and $\mathbf{s}^+ > \mathbf{0}$.¹ Furthermore, a relaxation of constraints using a vector of surplus variables $\mathbf{s}^- \in \mathbb{R}^{k-1}$ is proposed by the authors, which results in the problem

$$\begin{aligned} \min f_i(\mathbf{x}) + \sum_{l=1, l \neq j}^k \mu_l s_l^- \\ \text{s.t. } f_l(\mathbf{x}) - s_l^- \leq f_l^{(j)} \text{ for } l = 1, \dots, k \text{ with } l \neq i, \quad \mathbf{s}^- \geq \mathbf{0}. \end{aligned} \quad (\text{A.5})$$

The strategy allows the constraints to be violated by making them more flexible and improves the behavior of the numerical optimization algorithm. This is especially important when the upper bounds are too restricting. The combination of both strategies mentioned above is used in the so-called *improved epsilon-constraint method* formulated in [55] as

$$\begin{aligned} \min f_i(\mathbf{x}) + \sum_{l=1, l \neq j}^k (-\lambda_l s_l^+ + \mu_l s_l^-) \\ \text{s.t. } f_l(\mathbf{x}) + s_l^+ - s_l^- \leq f_l^{(j)} \text{ for } l = 1, \dots, k \text{ with } l \neq i, \quad \mathbf{s}^+, \mathbf{s}^- \geq \mathbf{0}. \end{aligned} \quad (\text{A.6})$$

¹Proper Pareto optimality was not introduced in this thesis since it is not pursued in the context of this research. A solution can be described as *proper Pareto optimal* if for each objective function i , there is at least one objective function j such that a finite improvement of the objective function i is only possible when worsening the objective function j . This is expressed in terms of positive and bounded trade-off rates, which can also be identified as the multipliers λ of Theorem 6.

For $\mu - \lambda \geq 0$ and $\mu, \lambda > 0$ the solution of the improved epsilon-constraint method is Pareto optimal.

As it is not of interest in the present work to find proper Pareto optimal solutions, and one assumes the absence of weakly Pareto optimal solution, the modified strategy is not pursued. However, in interior-point methods, e.g., constraint handling with the help of slack variables is implied, and the objective function can be easily adjusted. This type of constraint handling can also be implemented in one-shot methods for bound constrained optimization problems, which is presented in Chapter 3.

A.1.3. Multi-Objective Shape Optimization of 3D U-Bend

In this section, the equality constraint method with a hybrid strategy is applied to another type of problem in engineering design. One considers the internal flow through a three-dimensional U-shaped bend. The inlet flow has a Reynolds number of 10.000, and the flow is modeled in SU2 using the incompressible RANS equation with a $k - \omega$ SST turbulence model [185].

A typical objective function for optimizing pipe flows is the pressure drop or pressure loss, defined as the difference in the total pressure p_t , i.e., the sum of static pressure p_s and the dynamic pressure, given by

$$p_t = p_s + \frac{\rho}{2} \|\mathbf{v}\|^2$$

at the inlet and outlet of the pipe. Here, ρ is the density and \mathbf{v} the velocity of the fluid. The loss of pressure is caused by friction effects occurring in the pipe flow due to pipe walls and high velocities. The total pressure at the inlet can be used as an objective function for optimization, as the predefined outlet pressure is set as a boundary condition.

As a second objective function the flow uniformity index $\gamma \in [0, 1]$ can be of interest. It measures the average deviation from a mean velocity in a given surface with area A and is described by the formula

$$\gamma = 1 - \frac{1}{2\bar{v}A} \sum_{i=1}^n |v_i - \bar{v}|A_i,$$

where v_i is the local element velocity normal to the surface, \bar{v} is the mean normal velocity and A_i is the local surface area.

The convergence history of the primal and adjoint pressure residuals measured in the L_2 -norm are shown in Figure A.1. It can be observed that the contraction rates of the primal and adjoint iteration are the same. The runtime factor of the adjoint iteration in comparison to the primal iteration for this particular application is in average 1.3 and can be obtained for two different mesh resolutions.

The multi-objective optimization was carried out considering the pressure drop in the bend and the flow uniformity index at the outlet. A free form deformation box was used to define 30 design variables describing the displacement of control points in in-stream direction with prescribed lower and upper bounds.

A. Appendix

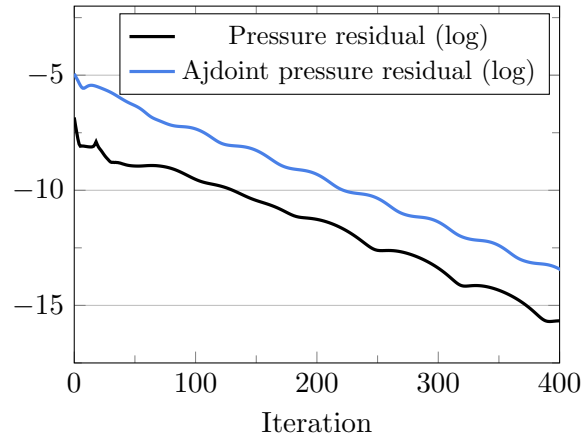


Figure A.1.: Convergence history of the pressure and the adjoint pressure using a logarithmic scale.

The solutions found with the ε -constraint method are shown in Figure A.2 in objective space. A total number of eleven Pareto optimal points were found using the hybrid strategy with IPOPT as a gradient-based optimizer. It can be seen that some parts of the Pareto optimal front are non-convex, and a weighted sum method would not have been able to find such solutions.

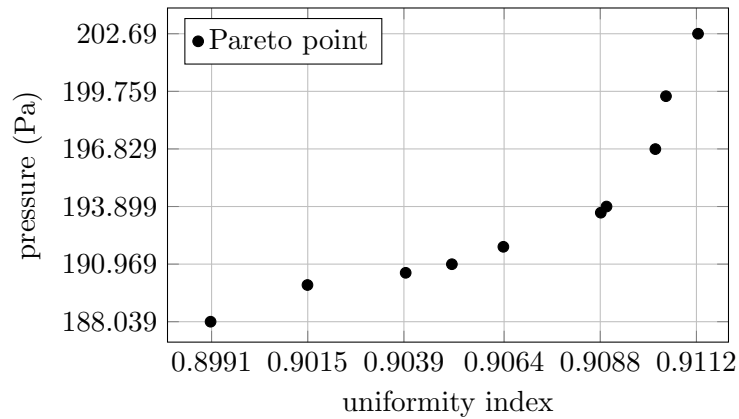


Figure A.2.: Pareto optimal solutions in objective space (inlet pressure and uniformity index). The dots denote the Pareto optimal solutions.

A.2. Appendix for Chapter 3

A.2.1. Adjoint Approach: Duality Viewpoint

The duality viewpoint shows that the adjoint approach can be applied when one wants to evaluate a quantity (\mathbf{v}, ϕ) and ϕ should satisfy the primal problem

$$D\phi = \mathbf{w}, \quad (\text{A.7})$$

where D is, for example, a differential operator or a matrix in the discretized form. The expression (\cdot, \cdot) is an inner product in the continuous setting or a scalar product in the discrete setting. The adjoint operator D^* is defined such that for the inner product the following holds:

$$(\psi, D\phi) = (D^*\psi, \phi) \quad \forall \phi, \psi. \quad (\text{A.8})$$

The dual variable ψ satisfying the dual problem

$$D^*\psi = \mathbf{v} \quad (\text{A.9})$$

can be used to evaluate (\mathbf{v}, ϕ) , as

$$(\mathbf{v}, \phi) = (D^*\psi, \phi) = (\psi, D\phi) = (\psi, \mathbf{w}). \quad (\text{A.10})$$

Now equation (3.14) can also be derived in a discrete dual way by identifying the different expressions $D = c_{\mathbf{y}}$, $\phi = \partial \mathbf{y} / \partial \mathbf{u}$, $\mathbf{w} = -c_{\mathbf{u}}$, $\mathbf{v} = -\nabla_{\mathbf{y}} f$, $D^* = c_{\mathbf{y}}^\top$ and $\psi = \bar{\mathbf{y}}$, as one can substitute according to (A.10) the expression $\left(\frac{\partial \mathbf{y}}{\partial \mathbf{u}}\right)^\top \nabla_{\mathbf{y}} f$ in (3.10) by $c_{\mathbf{y}}^\top \bar{\mathbf{y}}$.

A.2.2. Inexact rSQP Approach for One-Shot Optimization

This section shortly describes a version of the one-shot approach that can be interpreted as an inexact reduced sequential programming (rSQP) approach.

Hazra and Schulz introduced the approach in [113], where it was applied to a boundary control problem. In [116] and [110] it was applied to an aerodynamic shape optimization problem. Theoretical results on convergence properties can be obtained for quadratic objective functions with linear equality constraints [134]. Thus, it can be presumed that convergence is given locally in the vicinity of the optimal solution. A combination with multigrid strategies is proposed in [112].

The simultaneous optimization procedure of the presented one-shot strategy is a continuous reduced SQP method applied to the continuous version of the constrained optimization problem (3.1). In comparison to a SQP method, the reduced SQP method uses an approximation of the Hessian such that it is projected on the kernel of the first-order linearized state equation (see e.g. [241]). This means that all steps of the optimization scheme are located in the tangent space of the current state equation c . After having formulated a reduced SQP approach, the operators $c_{\mathbf{y}}$ and $c_{\mathbf{y}}^\top$ are discretized resulting

A. Appendix

in an inexact reduced SQP approach. The resulting inexact reduced SQP step is

$$\begin{pmatrix} 0 & 0 & A_a(\mathbf{y}_k, \mathbf{u}_k) \\ 0 & S & c_u(\mathbf{y}_k, \mathbf{u}_k)^\top \\ A_f(\mathbf{y}_k, \mathbf{u}_k) & c_u(\mathbf{y}_k, \mathbf{u}_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta \mathbf{y}_k \\ \Delta \mathbf{u}_k \\ \Delta \bar{\mathbf{y}}_k \end{pmatrix} = \begin{pmatrix} -\nabla_{\mathbf{y}} L(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k) \\ -\nabla_{\mathbf{u}} L(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k) \\ -\nabla_{\bar{\mathbf{y}}} L(\mathbf{y}_k, \bar{\mathbf{y}}_k, \mathbf{u}_k) \end{pmatrix} \quad (\text{A.11})$$

with $\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta \mathbf{y}_k$, $\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta \mathbf{u}_k$ and $\bar{\mathbf{y}}_{k+1} = \bar{\mathbf{y}}_k + \Delta \bar{\mathbf{y}}_k$. $A_f \approx c_y$ and $A_a \approx c_y^\top$ are the matrices of the discretized flow equations and the discretized adjoint equation. Similarly to the reduced gradient one can derive a reduced Hessian S .

The so-called reduced Hessian is approximated with a BFGS-Update or by an analytically found pseudo-differential operator [113]. One can show that the method can be interpreted as a simultaneous pseudo-time stepping method.

The idea of pseudo-time stepping is to transform the equation

$$F : X \rightarrow Y, \quad F(\mathbf{x}) = 0$$

with the Banach spaces X and Y into the ordinary differential equation $\dot{\mathbf{x}} = -F(\mathbf{x})$ with a given initial condition. It is iterated numerically in the fashion of an ordinary time integration scheme to reach the stationary point satisfying $F(\mathbf{x}) = 0$. One identifies $\dot{\mathbf{x}}$ as the substantial derivative in time. A stationary point can be reached if the right-hand side $-F$ has damping properties.

Note that the reduced SQP approach matches the explicit Euler method for the preconditioned system

$$\begin{pmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{u}} \\ \dot{\bar{\mathbf{y}}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & A_a(\mathbf{y}_k, \mathbf{u}_k) \\ 0 & S & c_u(\mathbf{y}_k, \mathbf{u}_k)^\top \\ A_f(\mathbf{y}_k, \mathbf{u}_k) & c_u(\mathbf{y}_k, \mathbf{u}_k) & 0 \end{pmatrix}^{-1} \begin{pmatrix} -\nabla_{\mathbf{y}} L \\ -\nabla_{\mathbf{u}} L \\ -c \end{pmatrix} \quad (\text{A.12})$$

with a preconditioner obtained from the reduced SQP method.

A.2.3. One-Shot Approach for Topology Optimization of Nonlinear Structures

So far, the one-shot approach has been mainly used for shape optimization problems. The main bottleneck for shape optimization is the need to perform a mesh deformation or even a re-meshing in each optimization step, which might become an overhead in the one-shot framework. For topology optimization, this problem is not apparent, which can be a high potential for one-shot optimization methods. A popular nested approach for topology optimization is the method of moving asymptotes, introduced in [261]. Simultaneous analysis and design has been performed in the context of topology optimization of truss structures [233], and interior-point multi-grid methods have been applied for the topology optimization of linear elastic materials [175]. In the following, first results of using the single-step one-shot strategy for the topology optimization of nonlinear elastic materials are presented, and challenges are formulated.

The Topology Optimization Problem The structural analysis for the nonlinear material under large displacements is based on the principle of virtual work. The resulting weak form of the equilibrium equation in the current configuration with volume V is given by

$$c(\mathbf{u}) := \int_V \boldsymbol{\sigma} : \delta \mathbf{e} \, dV - \left(\int_V \mathbf{f} \cdot \delta \mathbf{u} \, dV + \int_{\delta V} \mathbf{t} \cdot \delta \mathbf{u} \, dA \right) = 0 \quad (\text{A.13})$$

with the Cauchy stress tensor $\boldsymbol{\sigma}$, the rate of deformation tensor $\delta \mathbf{e}$, the virtual displacement field $\delta \mathbf{u}$, the boundary element dA , the volume force \mathbf{f} and the surface traction \mathbf{t} . The nonlinear hyper-elastic material is modeled as a Neo-Hookean solid. Equation (A.13) is discretized using a finite element approach and solved iteratively with a Newton-Raphson scheme leading to the fix-point formulation $\mathbf{u} = G(\mathbf{u})$. Note that it is made use of the common notation of topology optimization, i.e. \mathbf{u} is the vector of state variables (displacements) and in the following $\boldsymbol{\rho}$ denotes the vector of design variables (densities).

For topology optimization one may consider the mean compliance as an objective function to maximize the stiffness of the structure. Using the SIMP (Solid Isotropic Material with Penalization, [16]) approach, the density ρ_e serves as a design variable and is used in each element e to model areas containing material ($\rho_e = 1$) and void areas ($\rho_e = 0$). The Young's modulus E_e of an element is given by $E_e(\rho_e) = E_{\min} + \rho_e^p (E - E_{\min})$, where $p > 1$ penalizes densities between 0 and 1 and $E_{\min} > 0$ is a very small value representing the void regions in the stiffness matrix. An additional constraint is imposed on the volume of the resulting structure by prescribing a volume fraction $f_v > 0$. The resulting optimization problem to be solved reads

$$\begin{aligned} \min_{\mathbf{u}, \boldsymbol{\rho} \in \mathcal{D}} \quad & c(\mathbf{u}, \boldsymbol{\rho}) := \int_V \mathbf{f} \cdot \mathbf{u} \, dV \\ \text{s.t.} \quad & G(\mathbf{u}, \boldsymbol{\rho}) = \mathbf{u}, \\ & V(\boldsymbol{\rho})/V_0 = f_v, \quad 0 \leq \rho_e \leq 1. \end{aligned} \quad (\text{A.14})$$

Challenges for One-Shot Optimization There are several challenges for one-shot optimization inherent to topology optimization, among them the following:

- The constraints for the density have to be fulfilled at least for the optimal design. As in the main part of the thesis, a projection step can be applied separately from the updating scheme. In the following, the projection method proposed by Tavakoli and Zhang [267] can be used for fulfilling the volume constraint and the box constraints. The projection of $\bar{\boldsymbol{\rho}}$ is made by finding the unique minimizer of

$$\mathcal{L}(\mathbf{z}, \mu) := \frac{1}{2} \|\mathbf{z}\|_2^2 - \bar{\boldsymbol{\rho}}^\top \mathbf{z} + \frac{1}{2} \|\bar{\boldsymbol{\rho}}\|_2^2 + \mu \left(\mathbf{1}^\top \mathbf{z} - f_v \cdot V_0 \right),$$

with $0 \leq z \leq 1$.

- The number of design variables is large, which makes the approximation of the design space preconditioner difficult. Additionally, the preconditioner of the one-shot method itself or an additional preconditioning step has to serve as a filter to

A. Appendix

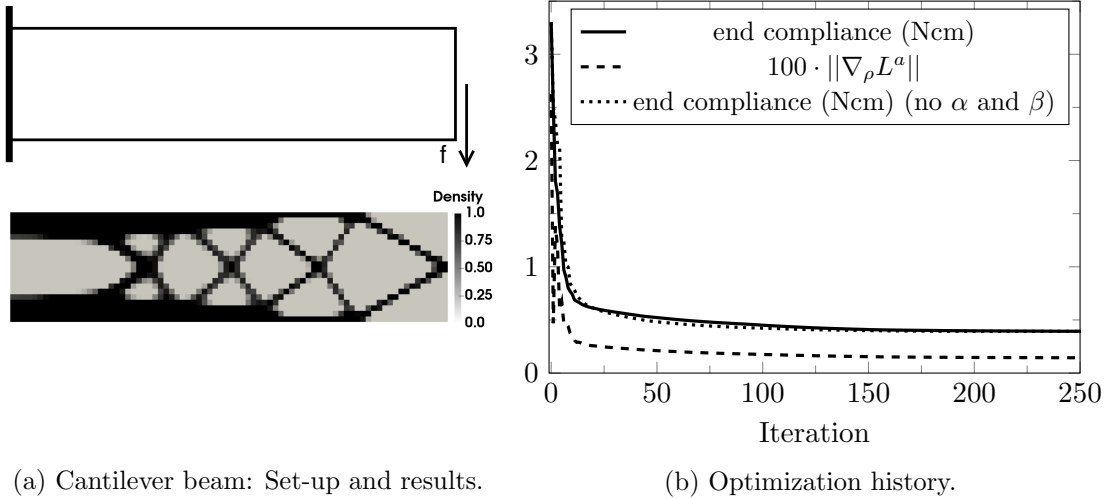


Figure A.3.: Results of one-shot topology optimization after 250 iterations (a). The set-up of the tip-loaded cantilever beam is shown in the upper part. The optimization history for the minimization of end-compliance is shown (b).

ensure mesh-independency and prevent checkerboard patterns. A popular preconditioner used for topology optimization is the Helmholtz filter [160]. Let \mathbf{d} present the unfiltered design update, then the filtered update $\tilde{\mathbf{d}}$ can be found by solving the Helmholtz-type PDE $-r^2 \nabla^2 \tilde{\mathbf{d}} + \tilde{\mathbf{d}} = \mathbf{d}$ with homogeneous Neumann boundary conditions, where r is the filter radius. When used in the one-shot method, the idea is to apply the filtering to the reduced gradient of the doubly augmented Lagrangian, such that $\mathbf{d} = -\nabla_{\rho} L^{\alpha}$.

- For the structural analysis of nonlinear materials undergoing large displacement, it can be advantageous to apply the load in an incremental fashion making the underlying problem unsteady. This is not pursued in the following since it requires the reformulation of the one-shot strategy.
- As it is the case in a nested approach, the found local minimum depends highly on the chosen starting value. This issue will not be treated in the following, but in future work, strategies like the continuation method have to be adjusted to the one-shot framework.

Numerical Example Results are shown for the minimization of end compliance for a tip-loaded cantilever beam in a two-dimensional setting (see Figure A.3a, top) with a load of 100N, assuming plane stress. The material is modeled with a Young's modulus of 2GPa and a Poisson ratio of 0.4. The solution procedure for the structural analysis is provided in the open-source framework SU2 (see [232]). The finite element analysis is performed using 4-node elements. The domain of 100 cm \times 25 cm is discretized using 80 by 20 finite elements.

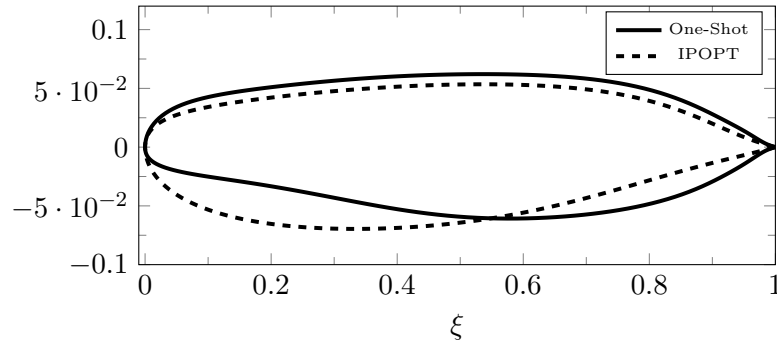


Figure A.4.: Optimal design for lift constraint of about 0.4 obtained by one-shot method and IPOPT (dashed line).

It is made use of the volume projection method with a volume fraction of 0.4. The factors for the doubly augmented Lagrangian are $\alpha = 2$ and $\beta = 10^{-4}$, and the penalty factor of the SIMP method is $p = 3$. Using the familiar one-shot preconditioner, one obtains a design with unintended grey areas, which represent densities between 0 and 1.

Instead, a Helmholtz-type filtering method with a filter radius of $r = 2$ is used based on the same values for α and β . A backtracking line search using the reduced gradient of the doubly augmented Lagrangian is employed. The convergence history is shown in Figure A.3b and the resulting optimized design is given in Figure A.3a (bottom). The obtained design is a typical end-compliance design for a small load. The primal solution converges in 25 iterations. The used method needs around 200 outer iterations and 200 inner iterations to converge. As the material does not exhibit a highly nonlinear behavior under the given load, the method will also converge for $\alpha = 0$ and $\beta = 0$. The corresponding convergence history presented by the dotted line is very similar, but the method will need an additional number of around 150 inner iterations. This shows that the use of the doubly augmented Lagrangian speeds up the convergence of the optimization. In future work, it needs to be analyzed if the methodology is successful for highly nonlinear problems.

A.2.4. Comparison of Results for Multi-Objective Optimization

When comparing the designs for high lift constraints obtained from the one-shot approach with the designs obtained with IPOPT as a gradient-based strategy, a significant difference can be observed in Figure A.4. The interior-point method tends to find designs with a round lower surface. In contrast, the one-shot strategy will find designs similar to the designs found when optimizing for robustness under geometrical uncertainties. The objective function values are similar. For the result of the one-shot method one gets a drag coefficient of $c_d = 0.001691$ and a lift coefficient of $c_l = 0.4020$. The values obtained by the hybrid strategy with IPOPT are $c_d = 0.001682$ and $c_l = 0.3913$.

A.2.5. Limited-Memory BFGS method for Bound Constraints

The limited-memory BFGS method was proposed in [28] and is presented in Algorithm 9 for bound constraints (cp. [142]). The function BFGSRec denotes the call to the recursive BFGS update.

Algorithm 9: Limited-memory BFGS with bound constraints.

Global variables:

$x_0, \mu_1, \mu_2, l_{\max}, \mathcal{S} = \{1, \dots, n\}$ (input)

l , stored values $\mathbf{r}_i, \mathbf{s}_i$ for $i = l - l_{\max}, \dots, l - 1$

Function Main():

$l \leftarrow 0, k \leftarrow 0$

while $\|\mathbf{x}_k - P(\mathbf{x}_k - \nabla f(\mathbf{x}_k))\| \geq \mu_1 + \mu_2 \|\mathbf{x}_0 - P(\mathbf{x}_0 - \nabla f(\mathbf{x}_0))\|$ **do**

$\tilde{\mathbf{d}}_k = -\nabla f(\mathbf{x}_k)$

Call BFGSRec($l, \tilde{\mathbf{d}}, P_{\mathcal{I}^\varepsilon(\mathbf{x}_k)}$)

$\mathbf{d}_k = -P_{\mathcal{A}^\varepsilon(\mathbf{x}_k)} \nabla f(\mathbf{x}_k) + \tilde{\mathbf{d}}_k$

Find γ such that Equation (3.85) holds.

$\mathbf{x}_{k+1} = P(\mathbf{x}_k + \gamma \mathbf{d}_k)$

$\mathbf{r}_l \leftarrow P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})}(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))$

$\mathbf{s}_l \leftarrow P_{\mathcal{I}^\varepsilon(\mathbf{x}_{k+1})}(\mathbf{x}_{k+1} - \mathbf{x}_k)$

if $\mathbf{r}_l^\top \mathbf{s}_l > 0$ **then**

$l \leftarrow l + 1$

else

$l \leftarrow 0$

end if

$k \leftarrow k + 1$

end while

Function BFGSRec($j, \mathbf{d}, P_{\mathcal{I}}$):

$\mathbf{d} \leftarrow P_{\mathcal{I}} \mathbf{d}$

if $j == 0$ or $j == l - l_{\max}$ **then**

return $\mathbf{d} \leftarrow I \mathbf{d}$

end if

$\theta = (\mathbf{s}_{j-1}^\top \mathbf{d}) / (\mathbf{r}_{j-1}^\top \mathbf{s}_{j-1})$

$\mathbf{d} \leftarrow \mathbf{d} - \theta \mathbf{r}_{j-1}$

Call BFGSRec($j - 1, \mathbf{d}, P_{\mathcal{S}}$)

$\mathbf{d} \leftarrow \mathbf{d} + \theta \mathbf{s}_{j-1} - (\mathbf{r}_{j-1}^\top \mathbf{d}) / (\mathbf{r}_{j-1}^\top \mathbf{s}_{j-1}) \mathbf{s}_{j-1}$

$\mathbf{d} \leftarrow P_{\mathcal{I}} \mathbf{d}$

A.3. Appendix for Chapter 4

A.3.1. Approximating the Signed Distance Function

The loss function in multi-objective space is defined based on a signed distance function. It is not always feasible to calculate the distance function exactly since it has to be done for every sample in objective space. This can, for example, be circumvented by numerically approximating a level set function representing the signed distance function using a *level set method* [209]. The representation of the deterministic Pareto optimal front Γ_0 is defined as the zero level set of the level set function $\varphi(\mathbf{F}, t)$ at time $t = 0$, i.e.,

$$\Gamma_0 := \{\mathbf{F} \in \mathbb{R}^k \mid \varphi(\mathbf{F}, 0) = 0\}. \quad (\text{A.15})$$

Here, t is a pseudo-time that is introduced to describe the propagation of the zero level set in objective space, such that

$$\Gamma_0(t) := \{\mathbf{F} \in \mathbb{R}^k \mid \varphi(\mathbf{F}, t) = 0\}. \quad (\text{A.16})$$

One may identify the level set function φ at $t = 0$ as a representation of the signed distance function δ . The transport equation

$$\partial_t \varphi(\mathbf{F}, t) + \text{sign}(\varphi(\mathbf{F}, 0)) \|\nabla_{\mathbf{F}} \varphi(\mathbf{F}, t)\| = 0 \quad (\text{A.17})$$

can be used for propagating the zero level set function at any pseudo-time t . Given an initial level set function at $t = 0$ that satisfies (A.15), equation (A.17) can be solved numerically on a Cartesian grid in objective space for different time steps. The pseudo-time t indicates the location of the zero level set of the level set function in objective space and thus can be used to measure the distance to the zero level set at time $t = 0$ for a given point in objective space using interpolation between two time steps (see [208, p.65]). As a result, the level set method can be used to approximately calculate the signed distance to the deterministic Pareto optimal front. If it is too expensive to precompute the solution on a Cartesian grid in objective space, equation (A.17) can be solved for a fixed number of sample points in objective space.

When using the above strategy, it is not guaranteed that the level set function remains to be a signed distance function for every pseudo-time t . This is even the case if the initial condition is already the signed distance function to be approximated, or the zero level set has the properties of a signed distance function. Thus, depending on the initial condition and the solution procedure, the problem based on (A.17) can be ill-conditioned. Therefore an alternative strategy is to prescribe the properties of a signed distance function for φ , namely that

$$\|\nabla_{\mathbf{F}} \delta(\mathbf{F}, \Gamma_0)\| = 1 \quad (\text{A.18})$$

has to hold almost everywhere in objective space. Equation (A.18) is a special case of the *Eikonal equation*. If (A.18) has to hold for the level set function φ depending on

A. Appendix

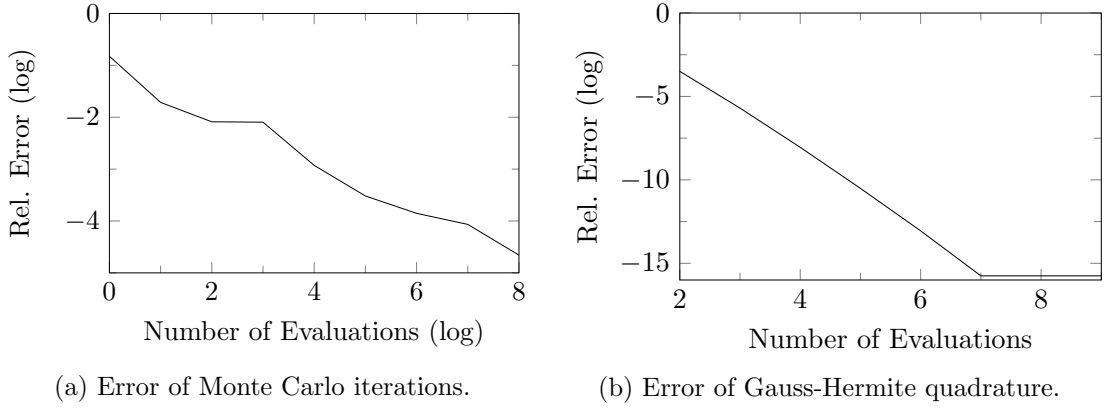


Figure A.5.: Logarithm of the relative error of the expected value for the Monte Carlo approach (a) and non-intrusive polynomial chaos using Gauss-Hermite quadrature (b).

pseudo-time t , the idea is to find a steady-state solution such that the condition holds for $t \geq t^*$, where t^* is fixed. Note that in this case the pseudo-time t is introduced for the level set function to achieve convergence to a steady-state. The steady-state solution can, for example, be found by solving the modified transport equation

$$\partial_t \varphi(\mathbf{F}, t) + \text{sign}(\varphi(\mathbf{F}, 0))(\|\nabla_{\mathbf{F}} \varphi(\mathbf{F}, t)\| - 1) = 0 \quad (\text{A.19})$$

with the prescribed condition (A.15) and thus can be directly used to approximate the signed distance function δ . However, depending on the smoothness of φ and the initial condition, the zero level set may be moved when solving (A.19). Different strategies were proposed for tackling this problem. Different numerical solution procedures like finding the viscosity solution may help to increase stability. The *fast marching method* [249] uses a different strategy based on the Eikonal equation to find the signed distance function. It is, for example, used in the field of computer vision. Note that the approaches explained above are sometimes combined [260] to increase the overall accuracy by solving (A.19) to reinitialize the solution of (A.17) either globally or locally in a neighborhood of the zero level set after a certain number of pseudo-time steps.

A.3.2. Monte Carlo Methods versus Non-Intrusive Polynomial Chaos

To compare the performance of the Monte Carlo method and the non-intrusive polynomial chaos approach, the test function $f(x) = \exp(-x)$ is considered. It is assumed that x is normally distributed with mean $\mu = 0.5$ and standard deviation $\sigma = 0.25$. The expected value of $f(x)$ is calculated using Monte Carlo integration and non-intrusive polynomial chaos using Gauss-Hermite quadrature. The exact value is given by $E(f(x)) = \exp(-\mu + 0.5\sigma^2)$.

The relative error of the expected value for Monte Carlo integration is plotted in a logarithmic plot in A.5a. The values are averaged over 5 runs. The slow convergence with

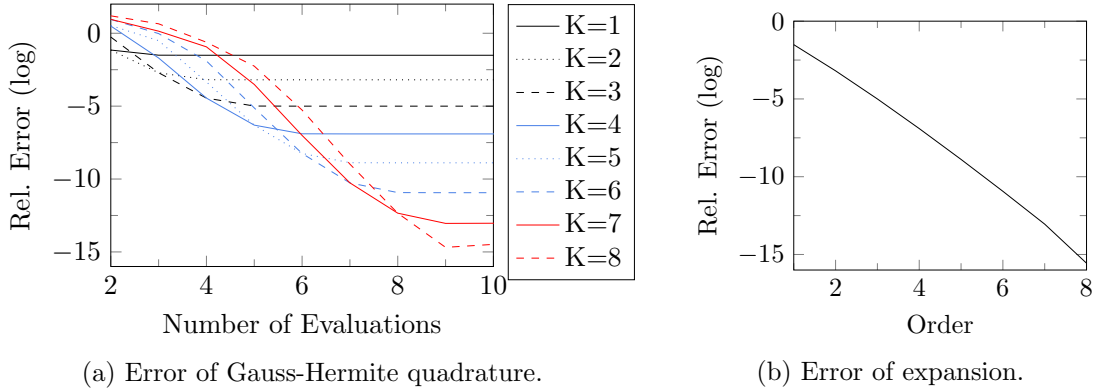


Figure A.6.: Logarithm of the relative error of the variance for Gauss-Hermite quadrature with different maximum expansion orders K (a) and for the non-intrusive polynomial chaos expansion with negligible quadrature error.

a convergence rate of the order $1/\sqrt{M}$ can be observed for Monte Carlo with M evaluations of the objective function. The logarithm of the relative error for Gauss-Hermite quadrature is plotted over the number of quadrature points in A.5b. Since the underlying function is smooth, fast convergence is observed for Gauss-Hermite quadrature.

For approximating the variance of f with non-intrusive polynomial chaos, the maximum order of polynomials K in the expansion is of importance. The logarithm of the relative error of the variance is plotted against the number of quadrature points in A.6a for different maximum orders of polynomials K . As expected, a higher order of chaos requires a higher number of quadrature points. The plateaus in convergence occur when the quadrature error is reduced up to machine accuracy, and only the truncation error of the expansion is present. This case is considered in Figure A.6b, where the relative truncation error is plotted over the maximum order of the expansion.

A.3.3. Additional Results for Operational Uncertainties

The present section shows additional interesting results for aerodynamic multi-objective robust design under the consideration of uncertainties.

First, it may be observed that the optimization based on expected values will not necessarily improve the amount of variations. This is especially the case for designs with a high lift coefficient. This can be observed in Figure A.7a, where vertical and horizontal lines indicate the standard deviations for the drag coefficient and the lift coefficient of the optimal designs and the robust optimal designs.

When optimizing the nominal objective functions while constraining the variance as proposed in problem (4.57), a similar effect can be observed. Figure A.7b shows the results for constraining the variance in such a way that $var(\tilde{c}_d) + var(\tilde{c}_l) \leq 0.001$, where \tilde{c}_d and \tilde{c}_l are scaled with respect to the nominal values. It can be seen that, despite the introduction of scaling, constraining the variance results in designs for high lift values that are far away from the Pareto optimal front.

A. Appendix

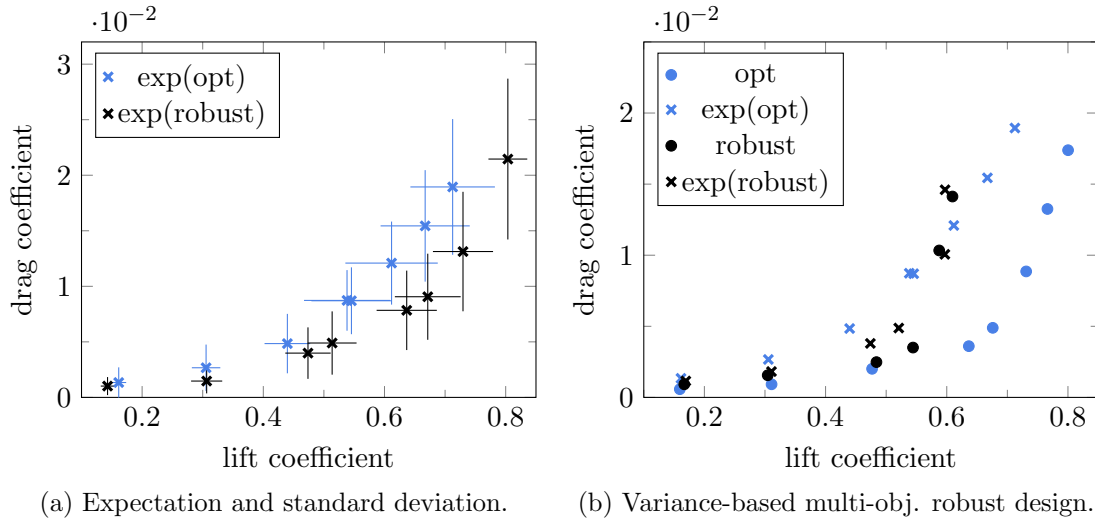


Figure A.7.: Expectation and standard deviation of individual objective functions (b) for optimal designs (blue) and robust optimal designs (black). The standard deviations are indicated by the vertical and horizontal lines. Robust optimal designs (b) found when constraining the L_1 -norm of the vector of variances (black points) and corresponding expected values (black crosses).

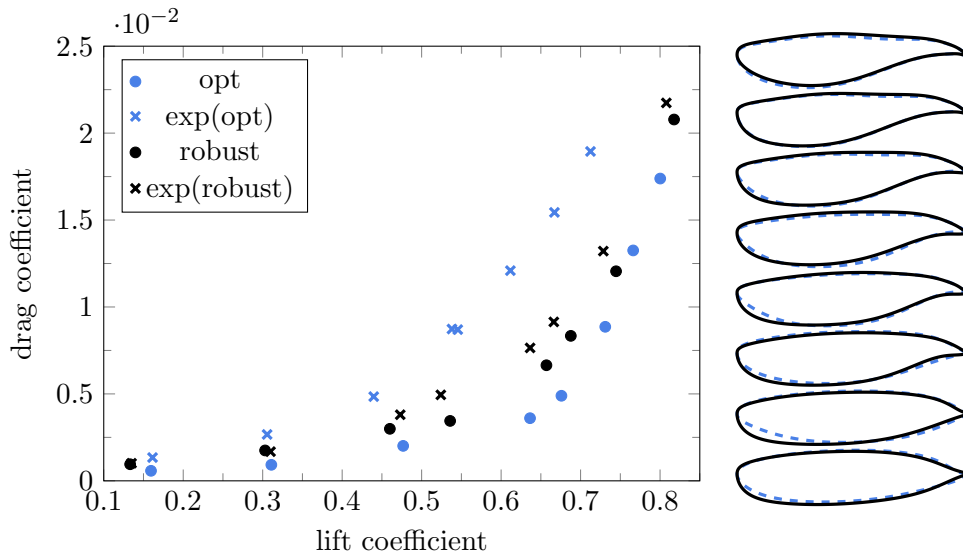


Figure A.8.: Robust Pareto optimal points for uncertainties in the Mach number and the angle of attack (black points). The crosses indicate the corresponding expected values and the designs are shown on the right side.

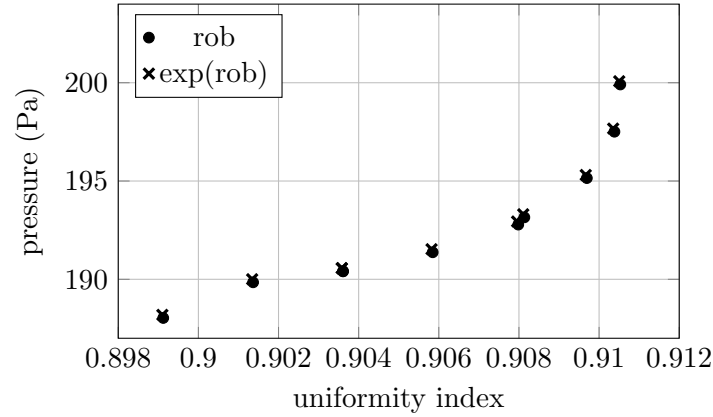


Figure A.9.: Robust Pareto optimal solutions in objective space. The dots denote designs evaluated at the nominal point and the crosses correspond to the expected values.

It can be concluded that the choice of an appropriate upper bound for the variance is not intuitive, and might be altered for specific regions in objective space. A similar but weaker effect can be seen for the direct approach based on estimated expected losses. When constraining losses with respect to the deterministic Pareto optimal front, such a problem is not observed. Concluding the numerical analysis of the variance of objective functions, it can be stated that an optimization based on losses in multi-objective space will not necessarily improve the variance. The obtained values for the variances of the individual objective functions are similar to the values of the expectation-based approach.

Finally, the angle of attack α can be considered as an additional uncertainty, which is normally distributed with a mean of 1.25 and a standard deviation of 0.01. The uncertainties in the Mach number and the angle attack are assumed to be independent. For the computation of the two-dimensional random integral, it is made use of a full tensor grid with 16 Gauss-Hermite quadrature points. Note that the minimum value of the moment coefficient will be obtained for the smallest outcomes of the Mach number and the angle of attack. Thus, the corresponding constraint can be prescribed in the discretized semi-infinite robust design formulation. The resulting robust Pareto optimal points are marked in black in A.8. The black crosses indicate the respective expected values. The resulting designs, shown on the right side, are very similar to the solutions found when only considering an uncertainty in the Mach number.

A.3.4. Multi-Objective Robust Design of 3D U-Bend

The expectation-based formulation for multi-objective robust design is applied to the three-dimensional U-shaped bend by assuming uncertain inflow conditions. Different numerical experiments are conducted, including an uncertain flow speed $\|\mathbf{v}\|$, which is normally distributed with $\mu = 0.4$ and $\sigma = 0.01$, and an uncertain direction of the input

A. Appendix

flow given by $(\sqrt{1 - \omega_1^2 - \omega_2^2}, \omega_1, \omega_2)$ with $\omega_1, \omega_2 \sim N(0, 0.0001)$. Figure A.9 shows the results for an uncertain flow speed. For all numerical experiments the resulting robust optimal designs are very close to the optimal designs obtained for a deterministic optimization, or coincide with the optimal designs. Thus, for the problem under consideration, the mean performance is comparable to the performance obtained when not considering uncertainties.

Curriculum Vitae

Personal Details

Name Lisa Kusch

Education

1995–1999 Gemeinschaftsgrundschule Mutter-Teresa (elementary school),
Düren, Germany
1999–2008 Städtisches Gymnasium am Wirteltor (secondary school),
Düren, Germany
2008–2012 B.Sc. studies (Computational Engineering Science),
RWTH Aachen, Aachen, Germany
2012–2013 M.Sc. studies (Computational Engineering Science),
RWTH Aachen, Aachen, Germany
since 2014 Ph.D. studies,
TU Kaiserslautern, Kaiserslautern, Germany

Work Experience

2009–2012 Student assistant, Mathematics Division of CCES,
RWTH Aachen, Aachen, Germany
2012–2013 Student employee, Computational Fluid Dynamics,
Ricardo Deutschland GmbH, Aachen, Germany
since 2014 Research assistant, Chair for Scientific Computing,
TU Kaiserslautern, Kaiserslautern, Germany

Lebenslauf

Persönliche Details

Name Lisa Kusch

Ausbildung

1995–1999 Gemeinschaftsgrundschule Mutter-Teresa,
Düren, Deutschland

1999–2008 Städtisches Gymnasium am Wirteltor,
Düren, Deutschland

2008–2012 Studiengang mit Abschluss B.Sc. (Computational Engineering Science),
RWTH Aachen, Aachen, Deutschland

2012–2013 Studiengang mit Abschluss M.Sc. (Computational Engineering Science),
RWTH Aachen, Aachen, Deutschland

since 2014 Promotionsstudium,
TU Kaiserslautern, Kaiserslautern, Deutschland

Arbeitserfahrung

2009–2012 Studentische Hilfskraft, Mathematikabteilung des CCES,
RWTH Aachen, Aachen, Deutschland

2012–2013 Werkstudentin, Computational Fluid Dynamics,
Ricardo Deutschland GmbH, Aachen, Deutschland

since 2014 Wissenschaftliche Mitarbeiterin, AG Scientific Computing,
TU Kaiserslautern, Kaiserslautern, Deutschland

List of Publications

- [Kusch et al.,2018] L. Kusch, T. Albring, A. Walther, and N. R. Gauger. A One-Shot Optimization Framework with Additional Equality Constraints Applied to Multi-Objective Aerodynamic Shape Optimization. *Optimization Methods and Software*, 33(4-6):694–707, 2018.
- [Kusch et al.,2016] L. Kusch and N. R. Gauger. The One-Shot Method in SU2. *Proceedings in Applied Mathematics and Mechanics*, 16(1):699–700, 2016.
- [Kusch et al.,2019] L. Kusch and N. R. Gauger. Robust Airfoil Design in the Context of Multi-objective Optimization. *Computational Methods in Applied Sciences*, 48:391–403, 2019.
- [Kusch et al.,2020] L. Kusch and N. R. Gauger. Robustness Measures for Multi-Objective Robust Design. *Computational Methods in Applied Sciences (Accepted for Publication)*, 2020.
- [Kusch et al.,2014] L. Kusch, N. R. Gauger, and M. Spiller. Efficient Calculation of Pareto-Optimal Points for Shape Optimization. In *Full Paper Compilation: Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems - EURO-GEN 2013, ISBN 978-84-617-2141-2*, Universidad de Las Palmas de Gran Canaria, Spain, 2014.
- [Walther et al.,2016] A. Walther, N. R. Gauger, L. Kusch, and N. Richert. On an Extension of One-Shot Methods to Incorporate Additional Constraints. *Optimization Methods and Software*, 31(3):494–510, 2016.
- [Walther et al.,2018] A. Walther, L. Kusch, and N. R. Gauger. New Results for the Handling of Additional Equality Constraints in One-Shot Optimization. *Vietnam Journal of Mathematics*, 46(4):825–836, 2018.

