BERGISCHE
UNIVERSITÄT
WUPPERTAL

# Biobjective Shape Optimization Algorithms Enhanced by Derivative Information

**Dissertation**

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften

Bergische Universität Wuppertal
Fakultät für Mathematik und Naturwissenschaften

vorgelegt von

**Onur Tanil Doganay, M.Sc.**

Wuppertal, im Oktober 2021

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-urn:nbn:de:hbz:468-20220125-105515-1
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20220125-105515-1]

DOI: 10.25926/pfta-7236
[https://doi.org/10.25926/pfta-7236]

# Acknowledgements

First of all, I want to thank my supervisors Prof. Dr. Hanno Gottschalk and Prof. Dr. Kathrin Klamroth for guiding me through this project. Without your support and ideas this work would not exist.
Next, I want to thank the whole consortium of the GIVEN project, which was supported by the federal ministry of research and education (BMBF, grant-no: 05M18PXA). In particular, I thank all my collaborators and GIVEN members of the two publications ([46, 19]) that are covered in this work. Prof. Dr. Matthias Bolten, Prof. Dr. Hanno Gottschalk, Dr. Camilla Hahn, Prof. Dr. Kathrin Klamroth, Johanna Schultes, and Dr. Michael Stiglmayr it was a pleasant experience working with you. I really learned a lot. Further, special thanks go to Dr. Camilla Hahn for supplying me with her implementation of the objective functions. You provided me with a foundation that I could built on.
I also want to thank my colleagues of both working groups that I am more or less part of. You made my time in the office worthwhile. A lot of conversations, a lot of jokes, many memorable (seminar) trips, and all in all a great working environment that I surely will miss. I hope to see you again at future christmas parties!
Last but not least, I want to thank my parents, Ayşegül and Ismail. Thank you, for always believing in me and giving me more support than one can ask for. Without your love, commitment, and patience I would not be the person that I am today, hence this work would probably not exist in this form. Thank you for everything.

# Contents

# List of Figures

# List of Tables

# Nomenclature

$\dot{x} = f(t, x)$ First order ODE

$F_Z(z) := P(Z \leq z)$ Cumulative distribution function (c.d.f.)

$(K, \Pi', \Sigma')$ Finite element

$\bar{\lambda}$ Scalarization for the biobjective descent method

$\bar{z} + \mathbb{R}^2_{\geqslant}$ The set $\{z \in \mathbb{R}^2 \,:\, z \geqslant \bar{z}\}$

$\big(Z(x)\big)_{x\in\mathcal{X}} := \big(Z(\omega, x)\big)_{x\in\mathcal{X}}$ Random field with parameter set $\mathcal{X}$

$\delta^{\max}$ Step size control in the biobjective descent and weighted sum methods

$\dot{\Sigma}_0$ Covariance vector of the known sample point and an unknown point $x^0$

$\dot{\Sigma}_Z$ Covariance matrix with gradient information incorporated

$\dot{Z}_i(x)$ $i$-th partial derivative of $Z(x)$

$(E, \mathcal{F})$ Measurable space

$\ell : V \to \mathbb{R}, \langle \ell, v \rangle$ Linear functional

$\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^\top)$ Elastic strain field

$\gamma^{\mathrm{ml}}, \gamma^{\mathrm{th}}$ Meanline and thickness B-spline coefficients

$\geqslant, \geqq, >, \leqslant, \leqq, <$ Ordering relations

$\hat{\sigma}$ Elastic stress field

$\hat{f} : \Omega \to \mathbb{R}^2$ Function describing volume forces

$\hat{g} : \partial\Omega \to \mathbb{R}^2$ Function describing surface forces

$\hat{s}^2_K(x^0)$ Mean square error of the Kriging prediction

$\hat{Z}(x^0)$ Kriging predictor

$\kappa^{\mathrm{u}}, \kappa^{\mathrm{l}}$ Curvature at the upper and lower boundaries

$K_{I_c}$ Critical K-factor

$\Lambda(\lambda, x)$ The smallest eigenvalue of the Hessian $\nabla^2_x J_\lambda(x)$

$\lambda \in (0, 1)$ Weight for the weighted sum scalarization

$\lambda_L, \mu_L$ Lamé constants

$\langle h, k \rangle_\varpi$ Modified $L^2$ scalar product, incorporating curvature

$\mathbb{C} = (\Omega \times S^{d-1} \times (0, \infty))$ Crack configuration space

$\mathcal{B}_j$ B-spline basis function

$\mathcal{O}$ Set of admissible shapes ($\Omega \in \mathcal{O}$)

$\mathcal{X} \subseteq \mathbb{R}^n$ Feasible set

$\mathcal{X}_P$ Pareto front

$\mathcal{X}_{sP}$ Set of strictly Pareto optimal solutions

$\mathcal{X}_{wP}$ Set of weakly Pareto optimal solutions

$\mu_d$ Pareto Tracer: steering direction

$\nabla_x^2 J_i$ Hessian of $J_i$ w.r.t. $x$

$\nabla_x J_i$ Gradient of $J_i$ w.r.t. $x$

$\nu$ Radon measure on the sigma algebra $mathbbA(mathbbC)$

$\nu_\mu$ Pareto Tracer: tangent vector

$\nu_P$ Poisson's ratio

$\Omega \subset \mathbb{R}^d,\ d = 2, 3$ Compact body filled with ceramic material

$\overline{\Omega}, \mathrm{cl}(\Omega)$ Closure of $\Omega$

$\partial\Omega$ Boundary of $\Omega$

$\partial\Omega_D$ Boundary of $\Omega$ with Dirichlet cond.

$\partial\Omega_{N_{fixed}}$ Part of $\partial\Omega_N$ where surface forces may act on

$\partial\Omega_{N_{free}}$ Part of $\partial\Omega_N$ which is surface force-free

$\partial\Omega_N$ Boundary of $\Omega$ with Neumann cond.

$\Phi(\cdot)$, $\phi(\cdot)$ Standard normal distribution and density function

$\Pi(u)$ Energy w.r.t. $u$

$(\Omega, \mathcal{A}, P)$ Probability space

$\psi(x^i, x^j) = \psi(x^i - x^j)$ Kriging correlation fucntion

$\mathbb{R}^2_{\geqq}$, $\mathbb{R}^2_{>}$, $\mathbb{R}^2_{\leqslant}$, $\mathbb{R}^2_{\leqq}$,$\mathbb{R}^2_{<}$ Analogous to $\mathbb{R}^2_{\geqslant}$

$\mathbb{R}^2_{\geqslant}$ The set $\{z \in \mathbb{R}^2 \,:\, z \geqslant (0,0)^\top\}$

$\rho([c,d])$ Fixes the density of cracks with radius $a$, $c \leq a \leq d$

$\rho(Z_1, Z_2)$ Correlation of $Z(x^1), Z(x^2)$

$\sigma_n$ Tensile load in normal direction

$\Sigma_Z$ Covariance matrix

$\tau = x - s$ Separation vector

$\mathtt{E}_Y$ Young's modulus

$\mathrm{div}(u)$ Divergence

$\mathrm{d}\upsilon_a$ Radon measure, occurrence of cracks of radius $a$

$\theta_h \geq 0$, $\varsigma_h \in [1,2]$ Parameters of $\psi(x^i - x^j)$

$\upsilon(A_c(\Omega, \nabla u))$ Intensity measure

$\mathrm{Var}(x^1) = \mathrm{Var}(Z(x^1)$ Variance of $Z(x^1$

$\varpi$, $\varpi = 10^{-4}$ Regularisation parameter for the curvature

$\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}}$ Meanline and thickness shape parameters

$\xi$ Element of ceramic component $\Omega$, i.e. $\xi \in \Omega$

$a \in \mathbb{R}_+$ Radius of a crack

$A_c$ Set of critical crack configurations

$B : V \times V \to \mathbb{R}$, $B(u,v)$ Bilinear form

$B_\varepsilon(x)$ Open ball centered at $x$ with radius $\varepsilon$

$C(\bar{\zeta}, \bar{\theta}, \bar{l})$ Cone

$C(x^1, x^2) = \mathrm{Cov}(Z(x^1), Z(x^2))$ Covariance of $Z(x^1), Z(x^2)$

$C^k$ Differentiability classes

$d$ ($d^{(k)}$) Descent direction (in iteration $k$)

$EI(x^0)$ Expected Improvement

$f_l(t,x)$ Approximated right hand side of the ODE

$f_Z = f_{Z_1,\ldots,Z_n}$ Probability density function (p.d.f.)

$H$ Hilbert space

$H^q(\Omega)$ Sobolev space

$J = (J_0, J_1)$ Biobjective objective function

$J_0$ Volume objective function

$J_1$ PoF/int. measure objective function

$J_\lambda = (1-\lambda)J_0 + \lambda J_1$ Weighted sum scalarization

$k_A$ Number of iterations: Armijo rule

$k_W$ Number of iterations: weighted sum descent

$K_{III}$ Stress intensity factor

$K_{II}$ Stress intensity factor

$K_I$ Stress intensity factor corresponding to compressive and tensile load

$L, L_H, L_\lambda, L_f$ Lipschitz constants

$L^2(\Omega) := L^2(\Omega; \mathbb{R})$ $L^2$ function space

$m(x) = \mathbb{E}[Z(x)]$ Expected value of $Z(x)$

$m,\ m = 5$ Weibull module

$n \in S^{d-1}$ Orientation of a crack

$n_B,\ n_B = 5$ Number of B-spline basis functions

$n_x, n_y$ Number of grid points in $\mathbb{x}$ and $\mathbb{y}$ direction

$N_{\text{LHS}}$ EGO: Number of Latin hypercube samples

$N_{\text{max.it}}$ EGO: Number of maximum iterations

$PI(x^0)$ Probability of improvement

$S^{d-1},\ d = 2, 3$ Unit sphere in $\mathbb{R}^d,\ d = 2, 3$

$S_h := S_h(\Omega, \mathbb{R}^d),\ d = 2, 3$ Discretized Sobolev space

$T_K$ Affine mapping for finite element $K$

$u : \Omega \to \mathbb{R}^d,\ d = 2, 3$ Displacement, solution of PDE

$X$ $n_x \times n_y$ grid

$x(\lambda)$ Solution of Pareto tracing by numerical integration

$x = (x_1, \ldots, x_6)$ Optimization variables for biobjective shape optimization

$x_0 = x(t_0)$ Initial condition

$x_k(\lambda)$ Approximate solution of Pareto tracing by numerical integration

$x_{0,k'';l,0.8} := x_{0,k'';l} = x_{k'';l}(0.8)$ S-shaped joint: initial solution

$x_{0,k';l,0.25} := x_{0,k';l} = x_{k';l}(0.25)$ S-shaped joint: initial solution

$x_{0,k_i;l,0.8}$ S-shaped joint: premature solutions

$x_{i,k}$ $i$-th iterate of a Runge-Kutta method/Pareto tracing by numerical integration

$Z(\omega)$ Realization of $Z$

$Z : (\Omega, \mathcal{A}, P) \to (E, \mathcal{F})$ $E$-valued *random variable*

$Z_N$ Non dominated front in the objective space

$Z_{wN}$ Set of weakly dominated points in the objective space

tr(u) Trace

uts Ultimate tensile strength

# 1 Introduction

## 1.1 Motivation

The aim of this work is to apply gradient-based optimization methods to shape optimization problems of ceramic components that consider two objective functions. In shape optimization, it is of interest to find an optimal shape, in the sense that this shape minimizes a given cost function while fulfilling constraints, see, e.g., [5, 25, 83, 145] for an introduction. Moreover, for many problems there exists an underlying partial differential equation that governs the physical behavior of the shapes, and therefore also the cost functional. Furthermore, in [28, 66, 83] the existence of optimal shapes was discussed. Shape optimization has many applications like, e.g., the assessment of the reliability of gas turbines (low cycle fatigue) [75] and ceramics [20]. Further, applying the adjoint approach to shape calculus allowed on the algorithmic side for efficient ways to calculate shape gradients, see, e.g., [32, 41, 54, 55, 100, 138, 139, 145].

As mentioned we consider ceramics, since it is a commonly used material in applications due to its advantageous qualities, such as its low density, low electric conductivity, and corrosion resistance. One of the objectives that we consider is the mechanical integrity which is one of the main objectives in mechanical engineering [13]. Traditionally, this objective is non-differentiable as it only depends on the point of maximal stress of the component. In this approach, the ultimate tensile load that the material can bear or the fatigue life of a component are described deterministically. In numerical optimization this would lead to highly unstable optimization schemes.

In [21, 20, 74, 73, 75, 134, 135, 136], alternative probabilistic approaches for mechanical integrity are proposed. These formulations overcome the problem of non-differentiablity. Moreover, note that the ground breaking work of Weibull standardized the probabilistic description of the ultimate strength of ceramics, see, e.g., [13, 24, 112, 127, 152].

In practice, in most cases there is a trade-off between the mechanical integrity of a component and its volume (cost). Usually, improving the mechanical integrity requires also a larger volume. In this work, we analyze these two conflicting goals, mechanical integrity and volume, in a biobjective optimization model. Furthermore, to apply gradient-based optimization methods, we utilize the implementation of [79], which produces shape gradients computed with the adjoint approach. We consider two dimensional ceramic components, bended beams and s-shaped joints, for our biobjective shape optimization problem, where we fix the left boundary of the components and apply tensile load on right boundary. Furthermore, methods from three classes of gradient-based biobjective optimization methods are applied to this problem.

## 1.2 Historical Background

To start the field of shape optimization a formulation of derivatives and gradients of objective functions $J$ that depend on a shape $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, was necessary to translate the optimality concepts of finite dimensional analysis, e.g., KKT conditions, to shape calculus. In 1907, Hadamard proposed a way to acquire such derivatives [78]. After a few decades, in 1975 Chenais published an existence result for optimal shapes in [28]. Furthermore, in 1979 Zolésio built on the results of Hadamard and introduced the now (in shape calculus) central 'Hadamard structure theorem' in [159]. This theorem enabled the definition of gradients and therefore also descent directions and optimality conditions for optimization methods. The summary of the historical background of shape calculus can also be found in [18].
The international society on Multi Criteria Decision Making (MCDM) states that Benjamin Franklin (1706 1790) provided the earliest known reference w.r.t. MCDM[1]. Franklin used a simple paper system for deciding important issues in which he incorporated a decision making approach which considered multiple objectives. In [52], it is stated that the book [114] that Pareto published in 1896, introducing the concept of Pareto optimality, is historically considered as the first reference considering problems with multiple conflicting objectives. Following [51], we give a short summary of the historical development of the field of multiobjective optimization since the 1950's. In 1951, Kuhn and Tucker formulated optimality conditions for nonlinear optimization problems in [99], where also multiple objectives were considered. One of the early contributions was goal programming which got its name from Charnes and Cooper in 1961 when they published [27]. Goal programming was further developed under the leadership of Lee (1972) [101] and Ignizio (1976) [87]. In the meantime, vector optimization methods to recover the set of all nondominated solutions of multiobjective optimization problems became more prominent, see, e.g., [68, 58, 17]. Furthermore, in the 1980s interactive methods to help determine final solutions of large sized nondominated sets gain in popularity, see, e.g. [15, 69, 97]. Since then, accompanied with the growth of computational power, new branches like, e.g., evolutionary multiobjective optimization arised, see, for e.g., [62, 147, 86].

## 1.3 Related Work

For the solution of shape optimization problems, there are two major approaches. Evolutionary and metaheuristic algorithms are commonly used for these problems since they do not depend on the structure of a given problem [29, 39, 40, 154]. Nonetheless, for expensive numerical simulations these methods can be inefficient. On the other hand, efficient computations of the gradient are required for gradient-based algorithms [43, 60, 156] and steepest descents with the weighted sum scalarizations of the objective functions. Therefore, they are often applied in combination with adjoint approaches.
In biobjective optimization, scalarization methods are a common choice to compute representations of the Pareto front, see, e.g., [52]. Assuming differentiability, one can utilize optimality conditions to recover additional parts of the Pareto front, see, e.g., [52, 85]. To this end, the literature provides an array of methods to compute the Pareto front, e.g., subdivi-

[1] https://www.mcdmsociety.org/content/short-mcdm-history-0

sion techniques [42, 89, 141], applying sensitivities w.r.t. the scalarization parameters [53], or continuation and predictor-corrector methods [53, 104, 103, 115, 116, 125, 132, 140]. In many cases, scalarization methods are incorporated in continuation and predictor-corrector methods, transforming the problem to a single-objective problem which depends on the scalarization parameters. Therefore, these problems can also be interpreted as parametric optimization problems. Under appropriate differentiability assumptions, where in some cases Hessian information is incorporated, predictor-corrector methods can recover parts of the Pareto front using the underlying optimality conditions. Moreover, continuation methods are capable of handling problems with constraints as well as multiobjective problems. They are also able to incorporate preference information in the exploration of the Pareto front, see, e.g., [103].
Another commonly used solution approach are surrogate based optimization methods which allow to estimate expensive objective functions using computationally cheap surrogate models. These methods are also incorporated in multiobjective optimization [123, 143]. One of the widely used surrogate based optimization methods [157], the Efficient Global Optimization (EGO) algorithm [92, 90], utilizes a Kriging surrogate model [98, 105, 130, 34, 35] and can also incorporate gradient information [111]. It was extended for the multiobjective case in [96, 36]. Furthermore, it is widely applied in aerodynamic design problems [117, 102, 11, 94, 63, 64, 133], where also affordable gradient information was incorporated into surrogate models [31, 8, 82, 157, 12].

## 1.4 Own Contribution

In [46], we published a first gradient-based biobjective approach of computing solutions for a biobjective PDE constrained shape optimization problem for the simultaneous optimization of the mechanical integrity and the cost of a ceramic component. The novelty is that in most publications in the highly developed field of numerical shape optimization neither problems with multiple objective functions, nor mechanical integrity as one of the objective functions is considered. Nevertheless, there exist some remarkable exceptions, see, e.g., [83, 6, 50, 119]. However, the cases that consider mechanical integrity as an objective function do not apply a probabilistic approach [6, 119, 50]. Building on the probabilistic formulation for ceramics under load given in [20] and a first single criteria optimization utilizing this objective in [21], we for the first time combine biobjective gradient-based optimization methods with a probabilistic assessment of mechanical integrity. My main contribution to this work was the development, implementation and numerical testing of a variant of a biobjective descent algorithm. Note that some parts of this work were published in [46]. Note that some parts of [46] are also to be published in parallel in [137], since [46] is a joint work.
Furthermore, in [19] we published a novel continuation methodin which the underlying parametric scalarizations are defined by simple weighted sums. This approach comes with the advantage that no further constraints are added to the formulation of the problem. Thus, for unconstrained biobjective optimization problems we do not require constraint handling techniques. Moreover, from our numerical experiments we obtain well distributed points on the Pareto front approximation since the numerical integration method controls the step length.
In this work, we also apply for the first time a gradient enhanced Kriging approach on

structural mechanic problems considering the mechanical integrity of ceramic components. Since the Kriging approach, with and without gradient information, is widely used in applications it provides a sufficient benchmark for the other gradient-based optimization methods of this work.

## 1.5 Structure of this Work

This work is structured as follows:
In Chapter 2, properties of ceramics are discussed, first. Then, the concept of weak solutions for *partial differential equations (PDE)* is introduced. In particular, properties of the existence of weak solutions for the linear elasticity theory, i.e., an elliptic PDE formulation that describes ceramics under tensile force, are provided. Subsequently, (Lagrangian) *finite elements* to discretize partial differential equations are reviewed. In Chapter 3, a formal introduction of the biobjective shape optimization problem, including a review of Weibull type models for the probability of failure is given. To this end, a brief overview of *biobjective optimization* and the *weighted sum scalarization* are provided, and an existence results for *Pareto optimal shapes* is stated. Chapter 4 is devoted to the discretization and parameterization of the problem, including an *adjoint approach* to compute the gradients cost efficiently. Here, the numerical implementation follows a first discretize then optimize approach. Chapter 4 also introduces the two dimensional case studies that we consider in this work. In Chapter 5, a biobjective gradient descent method and the weighted sum method, are investigated. We also provide some details on their efficient implementation. Subsequently, the approach is validated for the two dimensional case studies. We expect for the case study w.r.t. the bended beams that the solutions of the biobjective gradient descent methods are straight rods of varying thickness, since from a mechanical engineering point of view these should be the optimal forms. Subsequently, the methods are applied to the case study w.r.t. the s-shaped joints where we have no initial guess for the optimal solutions. In Chapter 6, a gradient-based biobjective continuation method, *Pareto tracing by numerical integration*, that utilizes an *ordinary differential equation (ODE)* to trace the Pareto front, is proposed. Toward this end, a brief introduction to ODEs and numerical methods to solve them, i.e., the *Runge-Kutta* methods, is given. The approach is then validated on biobjective problems for which the Pareto fronts are known. Subsequently, it is applied on the two two dimensional ceramic case studies that we investigate in this work. The third class of gradient-based optimization methods that we investigate in this work, the surrogate based method *efficient global optimization (EGO)*, where gradient information is incorporated in the surrogate model, is introduced in Chapter 7. Since, EGO utilizes a *Kriging*, i.e., a *Gaussian process*, model as a surrogate model, a brief review of *Gaussian random fields* and their analytical properties is given. For our numerical experiments, the implementation of the EGO algorithm provided by the (open source) software toolbox *Dakota* is utilized. For this purpose, an introduction to the Dakota toolbox is given and the coupling with Dakota described. Then, the EGO algorithm is applied on the two case studies. Subsequently, the numerical results w.r.t. the two case studies of Chapters 5, 6, and 7 are compared. Perspectives for future research are suggested in Chapter 8.

# 2 Ceramics: Linear Elasticity Equation and Finite Element Discretization

In this chapter, the mechanical properties of ceramic materials are discussed in Section 2.1. Then, in Section 2.2, an brief overview of some partial differential equations, i.e., elliptic boundary value problems, and their weak solutions is given. Furthermore, the *linear elasticity theory* which describes the behavior of ceramic components under physical phenomena, e.g., stress, using a PDE formulation is introduced. Subsequently, in Section 2.3 the concept of (Lagrangian) *finite element discretization* is introduced and utilized to discretize the linear elasticity equation. Note that this overview was also given in [45]. Further, note that some of these concepts are introduced for three dimensional ceramic components but are also true for two dimensional components.

## 2.1 Mechanical Properties of Ceramic Materials

Here, the main material properties of ceramic materials are discussed. This section is mainly based on [112, 76].

Ceramics are of interest in manufacturing since it possess advantageous qualities, such as its low electric conductivity, low density and corrosion resistance, which makes them a common material choice. Nevertheless, it comes with one major disadvantage, its brittleness causes the ceramics to have low stability at room temperature. Brittleness is due to the manufacturing of the ceramic materials utilizing sintering which creates small flaws in the material. Hence, it is an inescapable side effect of the manufacturing process. These flaws under high stress can induce cracks which can lead to failure of the material as plastic deformations are not able to stop stress peaks. This makes ceramic materials vulnerable to failure under tensile load. Moreover, experiments have shown that the form of the ceramic component has great influence when tensile stress is applied [112]. Hence, we are interested in the elastic behavior of ceramic materials, which is categorized as linear elastic [23]. The behavior of ceramic materials under stress, i.e., its linear elastic behavior, is among other things characterized by the Young‘s modulus $E_Y$ and Poisson's ratio $\nu_P$. In Table 2.1, an overview of some exemplary values of $E_Y$ and $\nu_P$ for some ceramic materials is given. The *ultimate tensile strength (uts)*, which is measured in megapascal (MPa), is another characteristic value of a material. It represents the maximal stress the material can endure before failing under tensile load. The transformation of a flaw in the material in an actual crack is mainly driven by the stress acting on the component, in the sense that the higher the stress, the higher the probability of failure of the component.

Stress, i.e., $\tilde{\sigma}$, in one point is described with a *stress tensor* that is given by the stresses in three planes which are in general chosen as orthogonal to the directions of the Cartesian

| Ceramic | $E_Y$ in GPa | $\nu_P$ | uts in MPa |
|---|---|---|---|
| $Al_2O_3$: dense | 410 | 0.20-0.25 | 11-276 |
| $Al_2O_3$: 95% | 320 | 0.20-0.25 | 11-276 |
| $Al_2O_3$: 88% | 250 | 0.20-0.25 | 11-276 |
| BeO | 311-340 | 0.25 | 93-140 |
| MgO | 317 | 0.17 | 96 |
| $ZrO_2$ | 160-240 | 0.22-0.30 | 123-140 |
| $B_4C$ | 450-470 | 0.17 | 155 |
| SiC | 480 | 0.16 | 41-200 |
| TiC | 460 | | 120 |
| WC | 730 | | 350 |
| AlN | 318 | 0.25 | |
| BN | 90 | | 1.6-48 |
| $Si_3N_4$ : HPSN | 320 | 0.28 | 150-375 |
| $Si_3N_4$ : RBSN | 160-200 | 0.23 | 140-170 |
| $TiB_2$ | 500-570 | 0.10 | 127 |
| $ZrB_2$ | 340 | 0.11 | 198 |
| $MoSi_2$ | 370 | | 280 |
| $Al_2TiO_5$ | 5-30 | 0.22-0.26 | |
| Mullite | 144 | 0.20 | 110 |

Table 2.1: Ceramics: values of Young's modulus $E_Y$, Poisson's ratio $\nu_P$, and uts (see, e.g., [112, 142]). See also [45].

coordinate system. We then have

$$\tilde{\sigma} = \begin{bmatrix} \tilde{\sigma}_x & \tilde{\sigma}_{xy} & \tilde{\sigma}_{xz} \\ \tilde{\sigma}_{yx} & \tilde{\sigma}_y & \tilde{\sigma}_{yz} \\ \tilde{\sigma}_{zx} & \tilde{\sigma}_{zy} & \tilde{\sigma}_z \end{bmatrix},$$

where the entries on the diagonal represent the stress acting in the normal direction of the surface it is acting on. We refer to it as *normal stress*. If an entry on the diagonal is positive we refer to it as *tensile stress*, whereas a negative entry is referred to as *compressive stress*.

## 2.2 Elliptic Boundary Value Problems

In this section, the linear elastic problem is introduced. For components $\Omega$ in $\mathbb{R}^d$, $d = 2, 3$, linear elastic problems are modeled as partial differential equations more precisely elliptic boundary value problems. In the following, a brief overview on elliptic boundary value problems and the existence of weak solutions is given. Note that this section is based on [23].

### 2.2.1 PDEs with Dirichlet and Neumann Boundary Conditions

Boundary value problems are governed by certain properties of the boundary of the considered set. Here, two important boundary conditions are introduced. To this end, the notion of *Lipschitz continuity*, see, e.g., [95], is introduced.

**Definition 2.1.** *Let $(E, d_E)$ and $(F, d_F)$ be metric spaces. We say a function $f : E \to F$ is* Lipschitz continuous *if there exists a constant $L < \infty$ such that*

$$d_F(f(x), f(y)) \leq L \cdot d_E(x, y) \text{ for all } x, y \in E.$$

*$L$ is then called the* Lipschitz constant *of $f$.*

Furthermore, we introduce the concept of *positive definiteness* for matrices.

**Definition 2.2.** *A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called positive definite if $x^\top Ax > 0$ for all $x \in \mathbb{R}^n$.*

Moreover, in the following for a domain $\Omega$ we denote the boundary of $\Omega$ as $\partial\Omega$, the interior of $\Omega$ as $\text{int}(\Omega) := \Omega \setminus \partial\Omega$, and the closure as $\bar{\Omega}$. Now, we can state following definitions.

**Definition 2.3.** *Let $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$. The boundary of $\Omega$, i.e., $\partial\Omega$, is called a* piecewise Lipschitz boundary*, if there exist open sets $U_1, U_2, ..., U_N$, with $N \in \mathbb{N}$, such that*

1. $\partial\Omega \subseteq \bigcup_{i=1}^{N} U_i$
2. *$\partial\Omega \cap U_i$ is the graph of a Lipschitz continuous function $\forall i = 1, ..., N$.*

**Definition 2.4.** *Let $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, be a domain with piecewise Lipschitz boundary $\partial\Omega$ and let*

$$Lu(\xi) := -\sum_{i,k} \partial_i(a_{ik}(\xi)\partial_k u(\xi)) + a_0(\xi)u(\xi) \tag{2.1}$$

*be a second order elliptic partial differential operator, where $a_0(\xi) \geq 0$ for $\xi \in \Omega$ and the matrix $A(\xi) := (a_{ik}(\xi))_{i,k}$ is positive definite. Then, an* elliptic (or coercive) boundary value problem with Dirichlet boundary condition *is given as*

$$\begin{aligned} Lu(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega, \\ u(\xi) &= \hat{g}(\xi) \quad \xi \in \partial\Omega, \end{aligned} \tag{2.2}$$

*with $\hat{f}$ and $\hat{g}$ arbitrary functions on $\Omega$.*

Every Dirichlet boundary value problem can be transformed to a homogeneous problem by a re-definition of $\hat{f}$ [23]. Hence, only problems with $\hat{g} = 0$ are considered in the following.
The other boundary condition that we consider is the so called *Neumann boundary condition*. Differing from the Dirichlet boundary condition it controls the values of the solution‘s derivative on $\partial\Omega$.

**Definition 2.5.**
*Let $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, be a domain with piecewise Lipschitz boundary $\partial\Omega$. Furthermore, let $Lu$ be defined as in 2.1. Then, an* elliptic boundary value problem with Neumann boundary conditions *is given as*

$$\begin{aligned} Lu(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega, \\ \sum_{i,k} n_i a_{ik} \partial_k u(\xi) &= \hat{h}(\xi) \quad \xi \in \partial\Omega, \end{aligned} \tag{2.3}$$

*where $a_{ik} = a_{ik}(\xi)$, $n := n(\xi)$ is the outward pointing normal which is defined almost everywhere on $\partial\Omega$, and $\hat{f}$ and $\hat{h}$ are arbitrary functions on $\Omega$. This can be written in a more compact way as*

$$\begin{aligned} Lu(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega, \\ n \cdot a \cdot \nabla u(\xi) &= \hat{h}(\xi) \quad \xi \in \partial\Omega. \end{aligned} \tag{2.4}$$

*Neumann boundary conditions are also called* natural boundary conditions.

In general, a problem is governed by mixed boundary conditions. Hence, the following formulation for a mixed boundary problem arises

$$\begin{aligned} Lu(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega, \\ u(\xi) &= \hat{g}(\xi) \quad \xi \in \partial\Omega_D, \\ n \cdot a \cdot \nabla u(\xi) &= \hat{h}(\xi) \quad \xi \in \partial\Omega_N, \end{aligned} \tag{2.5}$$

with $\partial\Omega_N = \partial\Omega \backslash \partial\Omega_D$ and where $\partial\Omega_D$ is the part of the boundary on which the Dirichlet conditions hold and $\partial\Omega_N$ the part on which the Neumann boundary conditions hold. The functions $\hat{f}, \hat{g}$ and $\hat{h}$ are be described later on. In a next step, the concept of weak solutions and their existence is introduced.

### 2.2.2 Weak Solutions of Elliptic PDEs

We utilize *Sobolov spaces* and the *variational form* of the boundary value problem to show the exitence of weak solutions. In this section, the assumptions required for the existence and uniqueness of weak solutions are discussed. The Dirichlet and Neumann boundary problems are considered separately, starting with the Dirichlet boundary problem. This subsection is based on [23].

#### Existence of Solutions for Dirichlet Problems

Let $\Omega$ be an open subset of $\mathbb{R}^d$, $d = 2, 3$, with piecewise Lipschitz boundary. The *Sobolev spaces* are built on the function space

$$L^2(\Omega) := L^2(\Omega; \mathbb{R}) := \left\{ f : \Omega \to \mathbb{R} \ : \ \text{f is measureable}, \ \int_\Omega |f(\xi)|^2 \, \mathrm{d}\xi < \infty \right\}.$$

On $L^2(\Omega)$ one can define the following scalar product

$$(u, v)_0 := \int_\Omega u(\xi) \cdot v(\xi) \, \mathrm{d}\xi,$$

and the corresponding norm

$$\|u\|_0 = \sqrt{(u, u)_0}.$$

Following [23], $L^2(\Omega)$ equipped with this norm is a Hilbert space, i.e., it is *complete* w.r.t. $\|\cdot\|_0$ or in other words every *Cauchy sequence* in $L^2(\Omega)$ has a limit in $L^2(\Omega)$.

**Definition 2.6.** *Let $u \in L_2(\Omega)$. We say that $u$ possesses the* weak derivative *$v = \partial^\alpha u$ in $L^2(\Omega)$ provided that $v \in L^2(\Omega)$ and*

$$(\phi, v)_0 = (-1)^{|\alpha|}(\partial^\alpha \phi, u)_0 \;\forall\; \phi \in C_0^\infty(\Omega) \tag{2.6}$$

*with $C_0^\infty(\Omega) = \{\phi \in C^\infty(\Omega) \mid \mathrm{supp}(\phi) \subset \Omega \text{ compact}\}$.*

Note that for differentiable $u$ the weak derivative is equal to the derivative.

**Definition 2.7** (Sobolev Space)**.** *Let $q \in \mathbb{N}$. Further, let $H^q(\Omega)$ be the set of all functions $u \in L^2(\Omega)$ that have weak derivatives $\partial^\alpha u \in L^2(\Omega)$ for all $|\alpha| \leq q$. We say that $H^q(\Omega)$ is a* Sobolev space*. Further, we define a scalar product on $H^q(\Omega)$ by*

$$(u, v)_q := \sum_{|\alpha| \leq q} (\partial^\alpha u, \partial^\alpha v)_0$$

*with the associated norm*

$$\|u\|_q := \sqrt{(u, u)_q} = \sqrt{\sum_{|\alpha| \leq q} \|\partial^\alpha u\|_0^2}.$$

Note that $H^q(\Omega)$ is complete w.r.t. the norm $\|\cdot\|_q$, see, e.g., [23]. The completion of $C_0^\infty(\Omega)$ regarding the Sobolev norm $\|\cdot\|_q$ is denoted by $H_0^q(\Omega)$.

**Theorem 2.8** (Characterization Theorem)**.** *Let $V$ be a linear space, and assume that*

$$B : V \times V \to \mathbb{R}$$

*is a symmetric positive bilinear form, i.e., $B(v, v) > 0$ for all $v \in V, v \neq 0$. Moreover, let*

$$\ell : V \to \mathbb{R}$$

*be a linear functional. Then,*

$$J(v) := \frac{1}{2} B(v, v) - \langle \ell, v \rangle \tag{2.7}$$

*attains its minimum over $V$ at $u$ if and only if*

$$B(u, v) = \langle \ell, v \rangle \text{ for all } v \in V. \tag{2.8}$$

*Furthermore, (2.8) has one unique solution.*

*Proof.* For $u, v \in V$ and $t \in \mathbb{R}$ we have

$$\begin{aligned} J(u+tv) &= \frac{1}{2}B(u+tv, u+tv) - \langle \ell, u+tv\rangle \\ &= \frac{1}{2}\left(B(u,u) + 2tB(u,v) + t^2 B(v,v)\right) - \langle \ell, u\rangle - t\langle \ell, v\rangle \\ &= J(u) + t(B(u,v) - \langle \ell, v\rangle) + \frac{1}{2}t^2 B(v,v). \end{aligned} \tag{2.9}$$

If $u$ satisfies condition (2.8), we can conclude with $t = 1$ that if $v \neq 0$ and as $B$ is positive definite

$$J(u+v) = J(u) + \frac{1}{2}B(v,v) > J(u). \tag{2.10}$$

Hence, $u$ is the unique minimum of $J$.
To the contrary, if the function $J$ possesses a unique minimum at $u$, then for every $v \in V$, the derivative of the function $t \mapsto J(u+tv)$ must vanish at $t = 0$. Following (2.9), the derivative has the form $B(u,v) - \langle \ell, v\rangle$, and therefore condition (2.8) follows. □

The following proposition links the boundary value problem to a variational problem.

**Proposition 2.9** (Minimal Property)**.** *Every solution of the boundary value problem*

$$\begin{aligned} Lu(\xi) = -\sum_{i,k} \partial_i(a_{ik}(\xi)\partial_k u(\xi)) + a_0(\xi)u(\xi) = \hat{f}(\xi) \quad \xi \in \Omega \\ u(\xi) = 0 \quad \xi \in \partial\Omega \end{aligned}$$

*is a solution of the variational problem*

$$\min \; J(v(\xi)) := \int_\Omega \left[\frac{1}{2}\sum_{i,k} a_{ik}(\xi)\partial_i v(\xi)\partial_k v(\xi) + \frac{1}{2}a_0(\xi)v(\xi)^2 - \hat{f}(\xi)\cdot v(\xi)\right] d\xi \tag{2.11}$$

*among all functions in $C^2(\Omega) \cap C^0(\bar{\Omega})$ with zero boundary conditions.*

In [23], Theorem 2.8 is used to prove this result. We omit most details of this proof, except that it was shown that if there exists a solution for the boundary value problem, it is the solution of equation (2.8), with

$$B(u,v) := \int_\Omega \left[\sum_{i,k} a_{ik}(\xi)\partial_i u(\xi)\partial_k v(\xi) + a_0(\xi)u(\xi)v(\xi)\right] \mathrm{d}\xi$$

and

$$\langle \ell, v\rangle := \int_\Omega \hat{f}(\xi)\cdot v(\xi)\; \mathrm{d}\xi.$$

In the following, it is shown that solving the variational problem on a suitable Hilbert space (i.e., with the right choice of topology) existence and uniqueness of the weak solution can be shown.

**Definition 2.10.** *Let $H$ be a Hilbert space. We say a bilinear form $B : H \times H \to \mathbb{R}$ is* continuous*, if there exists $C > 0$ such that*

$$|B(u, v)| \leq C\|u\| \cdot \|v\| \ \forall u, v \in H.$$

*A symmetric bilinear form $B$ is called $H$-*elliptic *or short* elliptic *or* coercive*, if for some $\alpha > 0$*

$$\alpha\|v\|^2 \leq B(v, v) \ \forall v \in H.$$

*This induces the following norm*

$$\|v\|_B := \sqrt{B(v, v)}. \tag{2.12}$$

*The norm* (2.12) *is also referred to as* energy norm*.*

We denote the space of continuous linear functions on a normed linear space $V$ by $V'$.

**Theorem 2.11** (Lax-Milgram)**.**
*Let $H$ be a Hilbert space. Further, let $V \subset H$ be a closed convex set, and let $B : H \times H \to \mathbb{R}$ be an elliptic bilinear form. Then, for every $\ell \in H'$ the variational problem*

$$\min \ J(v) := \frac{1}{2}B(v, v) - \langle \ell, v \rangle \tag{2.13}$$

*has a unique solution in $V$.*

*Proof.* The function $J$ is bounded from below, since

$$\begin{aligned} J(v) &\geq \frac{1}{2}\alpha\|v\|^2 - \|\ell\| \cdot \|v\| \\ &= \frac{1}{2\alpha}(\alpha\|v\| - \|\ell\|)^2 - \frac{\|\ell\|^2}{2\alpha} \geq -\frac{\|\ell\|^2}{2\alpha}. \end{aligned}$$

Let $c_1 := \inf\{J(v) \mid v \in V\}$. Further, let $(v_n)$ be a minimizing sequence. Then

$$\begin{aligned} \alpha\|v_n - v_m\|^2 &\leq B(v_n - v_m, v_n - v_m) \\ &= 2B(v_n, v_n) + 2B(v_m, v_m) - B(v_n + v_m, v_n + v_m) \\ &= 4J(v_n) + 4J(v_m) - 8J\left(\frac{v_n + v_m}{2}\right) \\ &\leq 4J(v_n) + 4J(v_m) - 8c_1. \end{aligned}$$

This inequality holds since $V$ is convex and therefore $\left(\frac{v_n+v_m}{2}\right) \in V$. From $J(v_n),\ J(v_m) \to c_1$ it follows that $\|v_n - v_m\| \to 0$ for $n, m \to \infty$. Hence, $(v_n)$ is a Cauchy sequence in $H$ and consequently $u = \lim\limits_{n\to\infty} v_n$ exists. Since $V$ is a closed set we have that $u \in V$. Furthermore, $J(u) = \lim\limits_{n\to\infty} J(v_n) = \inf_{v\in V} J(v)$ follows from the continuity of $J$.
In a next step, the uniqueness of the solution $u$ is shown. Assume that $u_1$ and $u_2$ are solutions of (2.13). Then, a minimizing sequence can be constructed as $u_1, u_2, u_1, u_2, \ldots$. But as already established, every minimizing sequence is a Cauchy sequence. Hence, $u_1 = u_2$.

□

A weak solution can then be defined in the following way.

**Definition 2.12.** *Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$. Let $u \in H_0^q(\Omega)$, $q > 0$. $u$ is called* weak solution *of the second order elliptic boundary value problem*

$$\begin{aligned} -\sum_{i,k} \partial_i(a_{ik}(\xi)\partial_k u(\xi)) + a_0(\xi)u(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega \\ u(\xi) &= 0 \quad \xi \in \partial\Omega \end{aligned} \tag{2.14}$$

*with homogeneous Dirichlet boundary conditions, if and only if*

$$B(u, v) = (\hat{f}, v)_0 \quad \textit{for all } v \in H_0^1(\Omega), \tag{2.15}$$

*where $B$ is a bilinear form that is given as*

$$B(u, v) := \int_\Omega \left[ \sum_{i,k} a_{ik}(\xi)\partial_i u(\xi)\partial_j v(\xi) + a_0(\xi)u(\xi)v(\xi) \right] d\xi. \tag{2.16}$$

Now, we can state the following existence result as a direct consequence of Theorem 2.13.

**Theorem 2.13** (Existence Theorem)**.**
*Let $L$ be a second order uniformly elliptic partial differential operator. Then, the Dirichlet boundary value problem* (2.14) *has a weak solution in $H_0^q(\Omega)$. It is the minimum of the variational problem*

$$\min_{v \in H_0^q(\Omega)} \ \frac{1}{2}B(v, v) - (\hat{f}, v)_0. \tag{2.17}$$

### Existence of Solutions for Neumann Problems

To show the existence of weak solutions for Neumann boundary value problems, the concept of cones is needed. We denote by

$$C(\bar{\zeta}, \bar{\theta}, \bar{l}) := \{\xi \in \mathbb{R}^d \mid |\xi| < \bar{l}, \xi \cdot \bar{\zeta} > |\xi| \cos(\bar{\theta})\}, \quad d = 2, 3$$

a cone with height $\bar{l}$, direction $\bar{\zeta}$, and opening angle $\bar{\theta}$.

**Definition 2.14.** *Let $\hat{\Omega}$ be a bounded open set in $\mathbb{R}^d,\ d = 2, 3$. For $\bar{\theta} \in ]0, \frac{\pi}{2}[$, $\bar{l}, r > 0$, $2r \leq \bar{l}$ we say $\Omega \subset \hat{\Omega}$ satisfies the* cone property*, if for any $\xi \in \partial\Omega$ there exists a cone $C_\xi = C_\xi(\bar{\zeta}_\xi, \bar{\theta}, \bar{l})$, where $\bar{\zeta}_\xi$ is a unit vector in $\mathbb{R}^d,\ d = 2, 3$, such that*

$$\xi' + C_\xi \subset \Omega, \quad \xi' \in B_r(\xi) \cap \Omega,$$

*where $B_r(\xi)$ is the open ball in $\mathbb{R}^d,\ d = 2, 3$, with radius $r$ centered at $\xi$. Further, we denote by $\Pi(\bar{\theta}, \bar{l}, r)$ the set of all subsets $\Omega$ of $\hat{\Omega}$, which satisfy the* cone property*.*

The following well-known lemma is needed for a proof later down in this section. Note that this lemma was also used in the proof of Proposition 2.9.

**Lemma 2.15** (Green's formula)**.**
*Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. For $u, v \in C^1(\bar{\Omega})$ we have*

$$\int_\Omega v(\xi)\partial_i u(\xi)\, d\xi = -\int_\Omega u(\xi)\partial_i v(\xi)\, d\xi + \int_{\partial\Omega} v(\xi)u(\xi)n_i(\xi)\, d\xi, \tag{2.18}$$

*where $n(\xi)$ is the normal at $\xi \in \Omega$.*

For a proof we refer to, e.g., [9].

**Theorem 2.16** (Trace Theorem)**.**
*Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be bounded, and assume that $\Omega$ has piecewise Lipschitz boundary. Furthermore, assume that $\Omega$ satisfies the cone property. Then, there exists a bounded linear mapping*

$$\hat{\gamma} : H^q(\Omega) \to L^2(\partial\Omega), \quad \|\hat{\gamma}(v)\|_{0,\partial\Omega} \leq c\|v\|_{1,\Omega}, \tag{2.19}$$

*such that $\hat{\gamma}v = v_{|\partial\Omega}$ for all $v \in C^1(\bar{\Omega})$.*

For a proof of this theorem we refer to [23].

**Theorem 2.17.** *Let $\partial\Omega_D \subset \partial\Omega$ be the part of the boundary with respect to the Dirichlet boundary conditions, i.e., the part on which zero-boundary conditions hold. Furthermore, assume that the domain $\Omega$ satisfies the assumptions of the Theorem 2.16 and that $\partial\Omega_D$ has positive two-dimensional measure. Then, the variational problem*

$$\min \; J(v) := \frac{1}{2}B(v, v) - (f, v)_{0,\Omega} - (g, v)_{0,\partial\Omega}$$

*has a unique solution in $H^q(\Omega)$. The solution of the variational problem is of class $C^2(\Omega) \cap C^1(\bar{\Omega})$ if and only if there exists a classical solution of the boundary value problem*

$$\begin{aligned} Lu(\xi) &= \hat{f}(\xi) \quad \xi \in \Omega, \\ \sum_{i,k} n_i(\xi)a_{ik}(\xi)\partial_k u(\xi) &= \hat{g}(\xi) \quad \xi \in \partial\Omega, \end{aligned}$$

*in which case the two solutions are identical.*

*Proof.* Given that $B$ is a $H^q$-elliptic bilinear form, Theorem 2.11 guarantees the existence of an unique solution $u \in H^q(\Omega)$. In particular, $u$ is characterized by

$$B(u, v) = (f, v)_{0,\Omega} + (g, v)_{0,\partial\Omega} \quad \forall \in H^q(\Omega). \tag{2.20}$$

Now assume that (2.20) holds for $u \in C^2(\Omega) \cap C^1(\bar{\Omega})$. For $v \in H_0^q(\Omega)$ we have $\hat{\gamma}v = 0$ and with (2.20) we conclude that

$$B(u, v) = (f, v)_0, \quad \forall v \in H_0^q(\Omega). \tag{2.21}$$

Furthermore, it can be shown that $u$ solves the Dirichlet problem. We have

$$Lu(\xi) = \hat{f}(\xi) \quad \xi \in \Omega, \tag{2.22}$$

For $v \in H^q(\Omega)$ we have with Lemma 2.15

$$\int_\Omega v(\xi)\partial_i(a_{ik}(\xi)\partial_k u(\xi))\,\mathrm{d}\xi = -\int_\Omega \partial_i v(\xi)a_{ik}(\xi)\partial_k u(\xi)\,\mathrm{d}\xi + \int_{\partial\Omega} v(\xi)a_{ik}(\xi)\partial_k u(\xi)n_i(\xi)\,\mathrm{d}\xi. \tag{2.23}$$

Therefore,

$$\begin{aligned} B(u,v) - (f,v)_0 - (g,v)_{0,\partial\Omega} = &\int_\Omega v(\xi)[Lu(\xi) - f(\xi)]\mathrm{d}\xi \\ &+ \int_{\partial\Omega}\left[\sum_{i,k} n_i(\xi)a_{ik}(\xi)\partial_k u(\xi) - \hat{g}(\xi)\right] v(\xi)\ \mathrm{d}\xi. \end{aligned} \tag{2.24}$$

With (2.20) and (2.22) the integral w.r.t. $\partial\Omega$ in (2.24) vanishes. Further, assume that the function $v_0(\xi) = n_i(\xi)a_{ik}(\xi)\partial_k u(\xi) - \hat{g}(\xi)$ does not vanish. Then, $\int_{\partial\Omega} v_0(\xi)^2\ \mathrm{d}\xi > 0$. Given that $C^1(\bar{\Omega})$ is dense in $C^0(\bar{\Omega})$, there is a $v \in C^1(\bar{\Omega})$ such that $\int_{\partial\Omega} v_0(\xi)v(\xi)\,\mathrm{d}\xi > 0$. This contradicts the assumptions. Hence, the boundary condition has to be satisfied. On the other hand, (2.24) gives us that every solution of (2.21) satisfies (2.22). □

In a next step, we formulate a elliptic partial differential equation for ceramic materials.

### 2.2.3 Linear Elasticity Theory

In Section 2.1 the mechanical properties of ceramics were discussed. In the following, these properties are translated into mathematical terms. Following [23], ceramics behave according to the linear elasticity theory, as long as the applied stress remains below the *uts*. The variational problem w.r.t. the linear elasticity theory is given as an minimization problem of energy

$$\Pi(u) := \int_\Omega \left[\frac{1}{2}\tilde{\sigma}(\xi) : \varepsilon(u(\xi)) - \hat{f}(\xi)\cdot u(\xi)\right]\ \mathrm{d}\xi - \int_{\partial\Omega} \hat{g}(\xi)\cdot u(\xi)\ \mathrm{d}\xi,$$

where $f : \Omega \to \mathbb{R}^d$, $d = 2,3$, is the body force, $g : \Omega \times S^{d-1} \to \mathbb{R}^d$, $d = 2,3$, is the area force, and $S^{d-1}$ is the unit sphere in $\mathbb{R}^d$, $d = 2,3$. Furthermore, we define $\tilde{\sigma}(\xi) : \varepsilon(u(\xi)) := \sum_{i,k} \tilde{\sigma}_{ik}(\xi)\varepsilon_{ik}(u(\xi))$. The variables $\varepsilon, \tilde{\sigma}$ and $u$ are coupled by the following kinematic equations

$$\varepsilon_{ij}(\xi) = \frac{1}{2}\left(\frac{\partial u_i(\xi)}{\partial \xi_j} + \frac{\partial u_j(\xi)}{\partial \xi_i}\right) \tag{2.25}$$

or $\varepsilon(u(\xi)) := \nabla u(\xi)$ and the linear constitutive equations

$$\varepsilon(u(\xi)) = \frac{1+\nu_P}{\mathtt{E}_Y}\tilde{\sigma}(\xi) - \frac{\nu_p}{\mathtt{E}_Y}\ \mathrm{tr}(\tilde{\sigma}(\xi))I, \tag{2.26}$$

where $\mathrm{tr}(\cdot)$ is the trace. We therefore have for the stress

$$\tilde{\sigma}(\xi) = \frac{\mathtt{E}_Y}{1+\nu_P}\left(\varepsilon(u(\xi)) + \frac{\nu_P}{1-2\nu_P}\,\mathrm{tr}(\varepsilon(u(\xi)))I\right).$$

In some cases this equation is written component-wise, for e.g., $d = 3$,

$$\begin{bmatrix}\tilde{\sigma}_{11}\\ \tilde{\sigma}_{22}\\ \tilde{\sigma}_{33}\\ \tilde{\sigma}_{12}\\ \tilde{\sigma}_{13}\\ \tilde{\sigma}_{23}\end{bmatrix} = K\cdot\begin{bmatrix}1-\nu_P & \nu_P & \nu_P & 0 & 0 & 0\\ \nu_P & 1-\nu_P & \nu_P & 0 & 0 & 0\\ \nu_P & \nu_P & 1-\nu_P & 0 & 0 & 0\\ 0 & 0 & 0 & 1-2\nu_P & 0 & 0\\ 0 & 0 & 0 & 0 & 1-2\nu_P & 0\\ 0 & 0 & 0 & 0 & 0 & 1-2\nu_P\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\\ \varepsilon_{13}\\ \varepsilon_{23}\end{bmatrix}, \tag{2.27}$$

where $K := \frac{\mathtt{E}_Y}{(1+\nu_P)(1-2\nu_P)}$. Further, note that we omitted the dependency of $\xi$ for notational reasons. Moreover, substituting with Lamé's constants $\lambda_L = \frac{\mathtt{E}\nu_P}{(1+\nu_P)(1-2\nu_P)}$ and $\mu_L = \frac{\mathtt{E}}{2(1+\nu_P)}$ yields the common formulation for $\tilde{\sigma}$

$$\tilde{\sigma}(\xi) = \lambda_L \mathrm{tr}(\varepsilon(u(\xi)))I + \mu_L\,(\varepsilon(u(\xi)) + \varepsilon(u(\xi))^\top), \tag{2.28}$$

and we have for the energy density

$$\frac{1}{2}\tilde{\sigma}(\xi) : \varepsilon(u(\xi)) = \frac{1}{2}(\lambda_L \mathrm{tr}(\varepsilon(u(\xi)))I + 2\mu_L\,\varepsilon(u(\xi))) : \varepsilon(u(\xi)) \tag{2.29}$$

$$= \frac{\lambda_L}{2}(\mathrm{tr}(\varepsilon(u(\xi))))^2 + \mu_L\,\varepsilon(u(\xi)) : \varepsilon(u(\xi)). \tag{2.30}$$

With the *mixed method of Hellinger and Reissner* [23] one can use (2.30) to reformulate $\Pi$,

$$\Pi(u) = \int_\Omega [\mu_L\,\varepsilon(u(\xi)) : \varepsilon(u(\xi)) \tag{2.31}$$

$$+\frac{\lambda_L}{2}(\mathrm{div}\ u(\xi))^2 - \hat{f}\cdot u(\xi)\Big]\,\mathrm{d}\xi \tag{2.32}$$

$$+\int_{\partial\Omega_N} \hat{g}(\xi)\cdot u(\xi)\,\mathrm{d}\xi, \tag{2.33}$$

where $\partial\Omega$ is divided into a part where zero boundary conditions hold, $\partial\Omega_D$, and $\partial\Omega_N$. Note that $\mathrm{div}\ u(\xi)$ denotes the *divergence* of $u$ at $\xi$, i.e., the trace of the Jacobian of $u$ at $\xi$. Hence, we have the following differential equation

$$\begin{aligned} -\mathrm{div}\ \tilde{\sigma}(\xi) &= \hat{f}(\xi) && \xi\in\Omega\\ u(\xi) &= 0 && \xi\in\partial\Omega_D,\\ \tilde{\sigma}(\xi)\cdot n(\xi) &= \hat{g}(\xi) && \xi\in\partial\Omega_N, \end{aligned}$$

with $\tilde{\sigma}$ as defined in (2.28).

**Theorem 2.18** (Korn's Inequality)**.**
*Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, be an open bounded set with piecewise Lipschitz boundary. Further, assume that $\partial\Omega_D \subset \partial\Omega$ has positive two-dimensional measure. Then, there is a positive number $C = C(\Omega, \partial\Omega_D)$ such that*

$$C\|v\|_q^2 \leq \int_\Omega \varepsilon(v(\xi)) : \varepsilon(v(\xi))\ d\xi \quad \forall v \in H^q_{\partial\Omega}(\Omega).$$

*Here, $H^q_{\partial\Omega}(\Omega)$ is the closure of $\{v \in C^\infty(\Omega)^3 \mid v((\xi)) = 0,\ \xi \in \partial\Omega_D\}$ w.r.t. the $\|\cdot\|_q$-norm.*

For a proof we refer to [23]. A consequence of Theorem 2.18 is that (2.33) is elliptic. Furthermore, applying Theorem 2.11 yields the following existence result.

**Theorem 2.19.**
*Let $\Omega \subset \mathbb{R}^d,\ d = 2, 3$, be a domain with piecewise Lipschitz boundary. Further, assume that $\partial\Omega_D$ has a positive two-dimensional measure. Then, the variational problem of the linear elasticity theory has a unique solution.*

Furthermore, one can now state the variational formulation for the linear elasticity PDE

$$B(u, v) = \int_\Omega \hat{f}(\xi) \cdot v(\xi)\ \mathrm{d}\xi + \int_{\partial\Omega_N} \hat{g}(\xi) \cdot v(\xi)\ \mathrm{d}A \quad \forall v \in H^q_{\partial\Omega}(\Omega), \tag{2.34}$$

where the bilinear form $B$ on the left hand side is given by

$$B(u, v) = \int_\Omega \tilde{\sigma}(u(\xi)) : \varepsilon(v(\xi))\ \mathrm{d}\xi \tag{2.35}$$

$$= \lambda_L \int_\Omega \operatorname{div}(u(\xi))\ \operatorname{div}(v(\xi))\ \mathrm{d}\xi + 2\mu_L \int_\Omega \varepsilon(u(\xi)) : \varepsilon(v(\xi))\ \mathrm{d}\xi. \tag{2.36}$$

## 2.3 Finite Element Discretization

In this section, a brief introduction in the concept of *finite element discretization* and the *Galerkin-Method* is given. This section is based on [23]. Note that we only need the *Sobolov space* $H^1(\Omega, \mathbb{R}^d),\ d = 2, 3$, in this work.

### 2.3.1 Finite Elements

The main idea of the finite element method is to compute (2.13) not on the Sobolov space $H^1(\Omega, \mathbb{R}^d), d = 2, 3$, but instead on a discretized finite-dimensional subspace denoted as $S_h := S_h(\Omega, \mathbb{R}^d),\ d = 2, 3$. To indicate that the initial space is a Sobolev space over $\mathbb{R}^d$, $d = 2, 3$, we use $H^1_h(\Omega, \mathbb{R}^d)$ instead of $S_h$. Our choice for $S_h$ is $H^1_{\partial\Omega_D,h}(\Omega, \mathbb{R}^d)$, i.e., the subspace of $H^1_h(\Omega, \mathbb{R}^d)$ such that all $u \in H^1_h(\Omega, \mathbb{R}^d)$ are vanishing on the closure of $\partial\Omega_D$. In finite element theory the domain $\Omega$ is partitioned by, e.g., triangles and quads in the two-dimensional case, and, e.g., tetrahedrons, cubes, parallelepipeds in the three-dimensional case. On each subdomain basis functions are considered, i.e., on the finite grid that is generated by the partitioning of $\Omega$ basis functions are defined which are used

to further define weak solutions on the discretized space $S_h$. Hence, one demands certain properties of the grid.

**Definition 2.20.**

1. *A partition* $\mathcal{T} = \{K_1, K_2, ..., K_N\}$ *of* $\Omega$ *into triangles, tetrahedrons or rectangular parallelepipeds is called* admissible *if*
   a) $K_i \subseteq \Omega$ *is open for all* $i \in \{1, ..., N\}$*;*
   b) $K_i \cap K_j = \emptyset,\ i \neq j,\ \forall i, j \in \{1, ..., N\}$*;*
   c) $\bigcup_{i=1}^{N} \bar{K} = \bar{\Omega}$.
2. *We write* $\mathcal{T}_h$ *instead of* $\mathcal{T}$ *when every element of* $\mathcal{T}$ *has diameter of at most* $2h$.
3. *We say a family of partitions* $\{\mathcal{T}_h\}$ *is* shape regular *if there exists a number* $\tau > 0$ *such that every* $K$ *in* $\mathcal{T}_h$ *contains a ball of radius* $\rho_K$ *where*
   $$\rho_K \geq h_K/\tau,$$
   *where* $h_K$ *is half the diameter of* $K$.
4. *We say a family of partitions* $\{\mathcal{T}_h\}$ *is* uniform *if there is a number* $\tau > 0$ *such that every element in* $\mathcal{T}_h$ *contains a ball with radius*
   $$\rho_K \geq h/\tau,$$
   *where* $h := \max_{K \in \mathcal{T}_h} h_K$.

Now, we can state the definition of *finite elements*, compare with [23].

**Definition 2.21** (Finite Element)**.**
*A* finite element *is a triple* $(K, \Pi', \Sigma')$ *satisfying the following conditions.*

1. $K$ *is a polyhedron in* $\mathbb{R}^d$, $d = 2, 3$.
2. $\Pi' := \Pi'(K)$ *is a subspace of* $C(K)$ *with finite dimension* $s$.
3. $\Sigma' := \Sigma'(K)$ *is a set of* $s$ *linearly independent functions over* $\Pi'$. *We also call these functionals* interpolation conditions. *Further, there exists the bijective mapping* $\Sigma' = \{\varphi_1, ..., \varphi_s\} : \Pi' \to \mathbb{R}^s$, *hence each* $p \in \Pi'$ *is uniquely defined by* $p \mapsto (\varphi_1(p), ..., \varphi_s(p))$.

*The functions* $\theta^K \in \Pi'$ *are also called* local shape functions *if they form a basis of* $\Pi'$.

Hence, every $u \in \Pi'$ can also be written as $u = \sum_{i=1}^{s} u_i^K \theta_i^K$ with $u_i^K \in \mathbb{R}$. To determine an element of $\Pi'$ a *nodal basis* of $\Pi'$, i.e., $s$ interpolation points (or nodes) $X_1, ..., X_s \in K$ including at least the vertices of $K$, is needed. To this end, we use *Lagrange finite*

*elements*, since they are commonly used finite elements. Furthermore, they are governed by the following interpolation conditions $\varphi$ at the $s$ interpolation points

$$\varphi_j(\theta_i^K) = \theta_i^K(X_j) = \delta_{ij} \ \forall \ i,j \in \{1,...,s\}.$$

Now, assume that $\Omega$ is discretized via the finite element method such that $\Omega = \bigcup_{K\in\mathcal{T}_h} K$. Then, every function $u = \sum_{K\in\mathcal{T}_h} \sum_{i=1}^{s} u_i^K \theta_i^K$ defined on this set is continuous, since $\Pi'(K)$ is a subspace of $C(K)$.

**Definition 2.22.** *We say a family of finite element spaces $S_h$ of partitions $\mathcal{T}_h$ of $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, is an* affine family*, if there exists a finite element $(\hat{K}, \hat{\Pi}', \hat{\Sigma}')$, referred to as the* reference element*, which possesses the following properties. For any $K \in \mathcal{T}_h$, there is an affine mapping $T_K : \hat{K} \longrightarrow K$ such that*

1. $\hat{\Pi}' = \Pi' \circ T_K$,
2. $\hat{\theta}_j := \theta_j \circ T_K$,
3. $\hat{\varphi}_j(p \circ T_K) := \varphi_j(p)$,

*where $\hat{\theta}_j$ and $\hat{\varphi}_j$ denote for each finite element $K$ the local shape functions and interpolation conditions on the reference element $\hat{K}$.*

Therefore, this allows one to compute every required solution first on the reference element and then transform it to a solution on an element $K$.

### 2.3.2 The Galerkin-Method

In this subsection, a brief overview over the *Galerkin-method* to solve finite element problems is given. Note that we omit $\xi$ for convenience. Toward this end, consider the following variational problem

$$\min_{S_h} \ J(v) := \frac{1}{2} B(v,v) - \langle \ell, v \rangle \tag{2.37}$$

in the subspace $S_h$. From Subsection 2.2.2 it is known that $u_h$ is a solution given that

$$B(u_h, v) = \langle \ell, v \rangle \ \forall v \in S_h. \tag{2.38}$$

With the basis $\{\theta_1, ..., \theta_s\}$, i.e., set of local shape functions, of $S_h$, (2.38) is equivalent to

$$B(u_h, \theta_i) = \langle \ell, \theta_i \rangle, \ i = 1, 2, ..., s.$$

Now, suppose that $u_h \in S_h$ has the form

$$u_h = \sum_{k=1}^{s} z_k \theta_k.$$

This leads to the system of equations

$$\sum_{k=1}^{s} B(\theta_k, \theta_i) z_k = \langle \ell, \theta_i \rangle, \ i = 1, 2, ..., s,$$

which can be compactly as

$$Az = b,$$

where $A$ is positive definite, if $B$ is a $H^1$-elliptic bilinear form. Recall that for this case the existence of unique weakly solution is provided by Subsection 2.2.2. Furthermore, the Galerkin-method enables the reformulation of the problem to a system of equations, for which several numerical methods are known. The following lemma gives insight on the accuracy of the method.

**Lemma 2.23** (Céa's Lemma)**.**
*Assume that the bilinear form $B$ is $H^1$-elliptic. Furthermore, let $u$ and $u_h$ be the solution of the variational problem in $H^1$ and in $S_h \subseteq H^1$, respectively. Then there exists $C, \alpha > 0$ such that*

$$\|u - u_h\|_1 \leq \frac{C}{\alpha} \inf_{v_h \in S_h} \|u - v_h\|_1.$$

*Proof.* From the definition of $u$ and $u_h$ it follows directly that

$$\begin{aligned} B(u, v) &= \langle \ell, v \rangle \quad \forall v \in H^1 \\ B(u_h, v) &= \langle \ell, v \rangle \quad \forall v \in S_h. \end{aligned}$$

with $S_h \subseteq H^1$, it follows by subtraction that

$$B(u - u_h, v) = 0 \quad \forall v \in S_h. \tag{2.39}$$

Let $v_h \in S_h$. Moreover, with $v = v_h - u_h \in S_h$, and (2.39) it follows that $B(u - u_h, v_h - u_h) = 0$ and

$$\begin{aligned} \alpha \|u - u_h\|_1^2 &\leq B(u - u_h, u - u_h) \\ &= B(u - u_h, u - v_h) + B(u - u_h, v_h - u_h) \\ &\leq C \|u - u_h\|_1 \|u - v_h\|_1. \end{aligned}$$

Further, dividing by $\|u - u_h\|_1$ establishes the assertion. □

From Lemma 2.23 it becomes clear, that the approximation error of the solution $u_h$ depends primarily on the choice of the underlying functional space. Note that this can be controlled by the order of the polynomials and by the fineness (or coarseness) of the partitioning.

### 2.3.3 Discretization of the Linear Elasticity Equation with Finite Elements

In this subsection, the finite element method is applied on the linear elasticity equation

$$B(u,v) = \int_\Omega \hat{f}(\xi)\cdot v(\xi)\ \mathrm{d}\xi + \int_{\partial\Omega_N} \hat{g}(\xi)\cdot v(\xi)\ \mathrm{d}A \quad \forall v \in H^q_{\partial\Omega_D,h}(\Omega,\mathbb{R}^d),\ d=2,3.$$

As mentioned the function is discretized via the finite element method using Lagrange nodes. We apply the same discretization of [73]. For the computation of the integrals we apply numerical quadrature.
Let $\Omega \subset \mathbb{R}^d,\ d=2,3$, be partitioned by a finite grid $\mathcal{T}_h$ with $N$ grid points $X = \{X_i \in K \mid i = 1,...,N\}$. Further, let $N_{el}$ be the number of finite elements $\{K, \Pi'(K), \Sigma'(K)\}$ forming this grid. Moreover, for each finite element $K$ there are $n_{sh}$ local shape functions $\theta_j \in \Pi'(K),\ j=1,...,n_{sh}$ which are defined by some nodes $X_1^K,...,X_{n_{sh}}^K \in X$. Now, assume that the family of finite elements is affine. Then, there exists a reference element $\{\hat{K}, \hat{\Pi}', \hat{\Sigma}'\}$ and a bijective transformation $T_K : \hat{K} \to K$ for each finite element $K \in \mathcal{T}_h$ such that $\hat{\Pi}' = \Pi' \circ T_K$, $\hat{\theta}_j = \theta_j \circ T_K$, $j \in \{1,...,n_{sh}\}$. We have

$$T_K = T_K(\hat{\xi}, X) = \sum_{j=1}^{n_{sh}} \hat{\theta}_j(\hat{\xi}) X_j^K, \quad \hat{\xi} \in \hat{K}.$$

For the numerical quadrature we choose $q^K$ quadrature points $\hat{\xi}_l^K$ on the reference element $\hat{K}$ and weights $\hat{\omega}_l^K$ for each $K \in \mathcal{T}$. Then, the discretized bilinear form (2.36) is given as follows

$$\begin{aligned}
B(u,v) &= \lambda_L \sum_{K\in\mathcal{T}_h} \int_K \mathrm{div}(u(\xi))\,\mathrm{div}(v(\xi))\ \mathrm{d}\xi + 2\mu_L \sum_{K\in\mathcal{T}_h} \int_K \varepsilon(u(\xi)) : \varepsilon(v(\xi))\ \mathrm{d}\xi \\
&= \lambda_L \sum_{K\in\mathcal{T}_h} \int_{\hat{K}} \mathrm{div}(u(T_K(\hat{\xi})))\,\mathrm{div}(v(T_K(\hat{\xi})))\,\det(\hat{\nabla} T_K(\hat{\xi}))\ \mathrm{d}\hat{\xi} \\
&+ 2\mu_L \sum_{K\in\mathcal{T}_h} \int_{\hat{K}} \varepsilon(u(T_K(\hat{\xi}))) : \varepsilon(v(T_K(\hat{\xi})))\,\det(\hat{\nabla} T_K(\hat{\xi}))\ \mathrm{d}\hat{\xi} \\
&\approx \lambda_L \sum_{K\in\mathcal{T}_h} \sum_{l=1}^{q^K} \hat{\omega}_l^K \det(\hat{\nabla} T_K(\hat{\xi}_l^K))\,\mathrm{div}(u(T_K(\hat{\xi}_l^K)))\,\mathrm{div}(v(T_K(\hat{\xi}_l^K))) \\
&+ 2\mu_L \sum_{K\in\mathcal{T}_h} \sum_{l=1}^{q^K} \hat{\omega}_l^K \det(\hat{\nabla} T_K(\hat{\xi}_l^K)) \varepsilon(u(T_K(\hat{\xi}_l^K))) : \varepsilon(v(T_K(\hat{\xi}_l^K))).
\end{aligned}$$

With the local shape functions $\hat{\theta}_m$ on the reference element it holds that

$$u(\xi) = \sum_{m=1}^{n_{sh}} u_m \hat{\theta}_m \circ T_K^{-1}(\xi) \quad \text{for every } \xi \in K.$$

Hence,

$$\nabla u(\xi) = \sum_{m=1}^{n_{sh}} u_m \otimes (\hat{\nabla} T_K(\hat{\xi})^T)^{-1} \hat{\nabla} \theta_m(\hat{\xi}). \tag{2.40}$$

Furthermore, we have

$$\operatorname{div}(u(\xi)) = \sum_{m=1}^{n_{sh}} \operatorname{tr}\left( u_m \otimes (\hat{\nabla} T_K(\hat{\xi})^T)^{-1} \hat{\nabla} \theta_m(\hat{\xi}) \right).$$

The volume force is discretized in a similar way

$$\int_\Omega \hat{f}(\xi) \cdot (\xi) \, \mathrm{d}\xi = \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{q^K} \hat{\omega}_l^K \det\left( \hat{\nabla} T_K(\hat{\xi}_l) \right) \hat{f}(T_K(\hat{\xi}_l)) \cdot v(T_K(\hat{\xi}_l)).$$

The surface force is discretized in another way, since only the faces $F$ of the finite elements $K$ that lie on $\partial\Omega$ are considered. To this end, let $\mathcal{N}_h$ be the set of all faces $F$ of finite elements $K = K(F) \in \mathcal{T}_h$ that lie on $\partial\Omega$. Further, let $\hat{F}$ be the face on the reference element $\hat{K}$ with $T_{K(F)} : \hat{F} \to F$. Stemming from the fact that the face $F$ is of lower dimension than $K$ additional $q^F$ quadrature points $\hat{\xi}_l^F$ and weights $\hat{\omega}_l^F$ have to be considered. Furthermore, we replace the determinant of the derivative of $T_K$ with the square root of the Gram determinant $\sqrt{\det g_F(\hat{\xi}_l^F)}$, given by

$$g_F(\hat{\xi}) = \hat{\nabla}^F (T_K\big|_{\hat{F}})(\hat{\xi}) \left( \hat{\nabla}^F (T_K\big|_{\hat{F}}) \right)^T (\hat{\xi}).$$

Hence,

$$\int_{\partial\Omega} \hat{g}(\xi) \cdot v(\xi) \, \mathrm{d}A = \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{q^F} \hat{\omega}_l^F \sqrt{\det g_F(\hat{\xi}_l^F)} \, \hat{g}(T_{K(F)}(\hat{\xi}_l^F)) \cdot v(T_{K(F)}(\hat{\xi}_l^F)).$$

With the global degrees of freedom $U = (u_j)_{j \in \{1,\dots,N\}} \subseteq \mathbb{R}^d$, $d = 2, 3$, and the node coordinates $X$, with $u_j = 0$ if $X_j \in \partial\Omega_D$ this can also be expressed as

$$\begin{aligned} B(X)U &= \hat{F}(X), \\ B(X)_{(j,r),(k,s)} &= B(\theta_j e_r, \theta_k e_s), \\ \hat{F}_{(j,r)} &= \int_\Omega \hat{f}(\xi) \cdot \theta_j e_r \, \mathrm{d}\xi + \int_{\partial\Omega_N} \hat{g}(\xi) \cdot \theta_j e_r \, dA, \end{aligned} \tag{2.41}$$

where $\{e_r \mid r = 1, \dots, d\}$ is the standard basis of $\mathbb{R}^d$, $d = 2, 3$.

# 3 Biobjective Shape Optimization (of Ceramic Structures)

In this chapter, we first introduce a set of admissible (feasible) shapes and review the state equations that model the physical behavior of a shape under external forces according to the linear elasticity theory (Section 3.1), see also Chapter 2. The considered objective functions, the intensity measure modelling the mechanical integrity, and the volume of the shape, are formally introduced in Sections 3.2 and 3.3, respectively. In Section 3.4, a brief overview of biobjective optimization, which is a special case of multiobjective optimization, where only two objective functions are considered, is given and the overall problem is formulated as a biobjective optimization problem. In Section 3.5, a widely used scalarization method to solve multiobjective and biobjective optimization problems, the weighted sum scalarization, is introduced. Furthermore, the existence of Pareto-optimal solutions is shown in Section 3.6. Most of this chapter was published in [46, 19].

## 3.1 Admissible Shapes and State Equation

We follow the description from [79, 21, 80, 20] and consider a compact body (also referred to as component or shape) $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, with Lipschitz boundary that is filled with ceramic material. Furthermore, we assume that the boundary $\partial\Omega$ of $\Omega$ is subdivided into three parts with nonempty relative interior,

$$\partial\Omega = \overline{(\partial\Omega_D)} \cup \overline{(\partial\Omega_{N_{\text{fixed}}})} \cup \overline{(\partial\Omega_{N_{\text{free}}})}.$$

$\partial\Omega_D$ describes the part of the boundary where the Dirichlet boundary condition holds, $\partial\Omega_{N_{\text{fixed}}}$ the part where surface forces may act on and $\partial\Omega_{N_{\text{free}}}$ the part of the boundary that can be modified in an optimization approach. It is assumed to be force free for technical reasons [20].

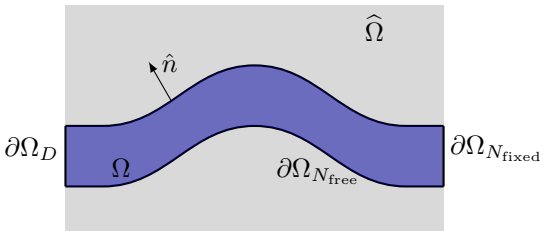



Figure 3.1: Illustration of $\Omega$ and its boundary components. See also [46, 19].

Since all feasible shapes have to coincide in $\Omega_D$ and in $\Omega_{N_{\text{fixed}}}$, it is natural to restrict the analysis to subsets of a sufficiently large bounded open set $\widehat{\Omega} \subset \mathbb{R}^d$ that satisfies

$\partial\Omega_D \subseteq \partial\widehat{\Omega}$ and $\partial\Omega_{N_{\text{fixed}}} \subseteq \partial\widehat{\Omega}$ (see Figure 3.1). We additionally assume that $\widehat{\Omega}$ satisfies the *cone property* (recall Definition 2.14) for a given angle $\bar{\theta} \in (0, \pi/2)$ and radii $\bar{r}, \bar{l} > 0$, $\bar{r} \leq \bar{l}/2$, i.e.,

$$\forall \xi \in \partial\widehat{\Omega}\ \exists \bar{\zeta}_\xi \in \mathbb{R}^d, \|\bar{\zeta}_\xi\| = 1\ :\ y + C(\bar{\zeta}_\xi, \bar{\theta}, \bar{l}) \subset \widehat{\Omega}\ \forall y \in B_{\bar{r}}(\xi) \cap \widehat{\Omega},$$

where $C(\bar{\zeta}_\xi, \bar{\theta}, \bar{l}) := \{c \in \mathbb{R}^d\ :\ \|c\| < \bar{l},\ c^\top \bar{\zeta}_\xi > \|c\| \cos(\bar{\theta})\}$ is a truncated circular cone oriented along $\bar{\zeta}_\xi$ with height $\bar{l}$ and opening angle $2\bar{\theta}$, and $B_{\bar{r}}(\xi) \subset \mathbb{R}^d$ is an open ball of radius $\bar{r}$ centered at $\xi$. Now the set of *admissible shapes* $\mathcal{O}^{\text{ad}} \subset \mathcal{P}(\mathbb{R}^d)$ can be defined as

$$\mathcal{O}^{\text{ad}} := \left\{\Omega \subseteq \widehat{\Omega} \colon \partial\Omega_D \subseteq \partial\Omega,\ \partial\Omega_{N_{\text{fixed}}} \subseteq \partial\Omega,\ \widehat{\Omega} \text{ and } \Omega \text{ satisfy the cone property}\right\}. \quad (3.1)$$

Ceramic components behave according to the linear elasticity theory [112]. The state equation can be described as an elliptic partial differential equation, see, e.g., [23]. More precisely, we get the state equation which describes the reaction of the ceramic component to external forces as a partial differential equation:

$$\begin{aligned} -\text{div}(\sigma(u(\xi))) &= \hat{f}(\xi) && \text{for } \xi \in \Omega \\ u(\xi) &= 0 && \text{for } \xi \in \partial\Omega_D \\ \sigma(u(\xi))\hat{n}(\xi) &= \hat{g}(\xi) && \text{for } \xi \in \partial\Omega_{N_{\text{fixed}}} \\ \sigma(u(\xi))\hat{n}(\xi) &= 0 && \text{for } \xi \in \partial\Omega_{N_{\text{free}}} \end{aligned} \quad (3.2)$$

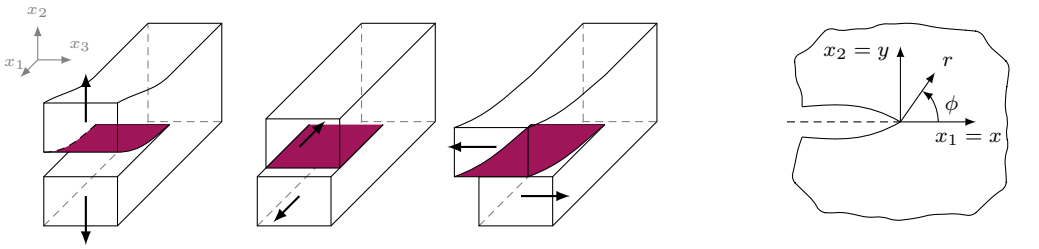
Here, $\hat{n}(\xi)$ is the outward pointing normal at $\xi \in \partial\Omega$, which is defined almost everywhere on $\partial\Omega$ given that $\partial\Omega$ is piecewise differentiable. Furthermore, let $\hat{f} \in L^2(\Omega, \mathbb{R}^d)$ be the volume forces and $\hat{g} \in L^2(\partial\Omega_{N_{\text{fixed}}}, \mathbb{R}^d)$ the forces acting on the surface $\partial\Omega_{N_{\text{fixed}}}$, e.g., the tensile load. The displacement caused by the acting forces is given by $u \in H^1(\Omega, \mathbb{R}^d)$, where $H^1(\Omega, \mathbb{R}^d)$ is the Sobolov space of $L^2(\Omega, \mathbb{R}^d)$-functions with weak derivatives in $L^2(\Omega, \mathbb{R}^{d\times d})$. The linear strain tensor $\varepsilon \in L^2(\Omega, \mathbb{R}^{d\times d})$ is given by $\varepsilon(u(\xi)) := \frac{1}{2}(\nabla u(\xi) + (\nabla u(\xi))^\top)$, where $\nabla u$ is the Jacobi matrix of $u$. It follows for the stress tensor $\sigma \in L^2(\Omega, \mathbb{R}^{d\times d})$ that $\sigma(u(\xi)) = \lambda_L \,\text{tr}(\varepsilon(u(\xi)))I + 2\mu_L \varepsilon(u(\xi))$, where $\lambda_L, \mu_L > 0$ are the Lamé constants derived from Young's modulus $E_Y$ and Poisson´s ratio $\nu_P$ as $\lambda_L = \frac{\nu_P E_Y}{(1+\nu_P)(1-2\nu_P)}$ and $\mu_L = \frac{E_Y}{2(1+\nu_P)}$. From a numerical perspective, a variational formulation of the state equation (3.2) is usually preferred, see, e.g., [79, 21, 80, 20]. This still guarantees a unique weak solution $u$, see [49]. Thus, $u$ is uniquely defined by the shape $\Omega$ [49], and we will equivalently write $\sigma(\nabla u(\xi)) := \sigma(u(\xi))$ for $\xi \in \Omega$ to highlight that $\sigma$ depends on the Jacobi matrix of $u$.

## 3.2 Probability of Failure

The primary objective function, the mechanical integrity of the ceramic component, is modelled based on the probability of failure analogous to [79, 21, 80, 20, 24, 152]. For the sake of completeness, this is briefly summarized in the following.

We want to optimize the reliability of a ceramic body $\Omega$, i.e., its survival probability, by minimizing its probability of failure under tensile load. In that sense failure means that the ceramic body breaks under the tensile load due to cracks. Such cracks occur as a result of small faults in the material caused by the sintering process. To understand the

mechanics of cracks, three types of crack opening are considered, see [76] and Figure 3.2a for an illustration. They are referred to as *Modes I, II* and *III*, respectively, and relate to different loads. Note that in the two-dimensional case, only Modes I and II can occur. We refer to [76] for a detailed introduction into this topic.




(a) Modes I, II and III (from left to right)

(b) $r$-$\phi$ coordinate system at the tip of the crack

Figure 3.2: Crack opening modes and two-dimensional model for the crack-tip field according to [79, 21, 80, 20, 76]. See also [46].

The stresses and strains close to a crack are represented by the *crack-tip field* which depends on the respective crack opening modes. It is described locally by a two-dimensional model, see Figure 3.2b for an illustration. With $K_{\mathrm{I}}, K_{\mathrm{II}}$ and $K_{\mathrm{III}}$ being the *stress-intensity factors* (also called *K-factors*) corresponding to Modes I, II, and III, respectively, one can describe the crack-tip field $\sigma$ locally according to linear fracture mechanics as

$$\sigma(\xi) = \sigma(r,\phi) = \frac{1}{\sqrt{2\pi r}}\Big\{K_{\mathrm{I}}\tilde{\sigma}^{\mathrm{I}}(\phi) + K_{\mathrm{II}}\tilde{\sigma}^{\mathrm{II}}(\phi) + K_{\mathrm{III}}\tilde{\sigma}^{\mathrm{III}}(\phi)\Big\} + R(r,\phi). \tag{3.3}$$

Here, $r$ is the distance to the crack tip, and $\phi$ the angle w.r.t. the $x_1$-axis (aligned with the crack plane), see Figure 3.2b. The functions $\tilde{\sigma}^{\mathrm{I,II,III}}(\phi)$ are known functions of the angle $\phi$, see again [76], and $R(r,\phi)$ is a regular function of the considered position in $x \in \Omega$ that is independent of the crack. Note that in the two-dimensional case, Mode III is omitted from (3.3) since it does not exist. Moreover, experimental evidence has shown that Mode I, which relates to tensile and compressive load, is the most relevant for the failure of ceramic structures [24], see [76] for approaches for multi-mode failure. We will thus focus on $K_{\mathrm{I}}$ in the following as the driving parameter for crack development under tensile load.

In order to evaluate $K_{\mathrm{I}}$ analogous to [20], we adopt the concept of equivalent circular discs to represent different crack shapes and crack sizes, and hence assume that the cracks are *penny shaped*. Then, a particular crack can be identified by its configuration

$$(\xi, a, n) \in \mathcal{C} := \Omega \times (0,\infty) \times S^{d-1},$$

where $\xi \in \Omega$ is its location, $a \in (0,\infty)$ its radius, and $n \in S^{d-1}$ its orientation ($S^{d-1}$ denotes the unit sphere in $\mathbb{R}^d$). $\mathcal{C}$ is called the *crack configuration space*. Given a crack $(\xi, a, n) \in \mathcal{C}$, $K_{\mathrm{I}}$ can be computed as a function of the radius $a$ and of the tensile load

$\sigma_n(\nabla u(\xi))$ in the normal direction $n$ of the stress plane at the crack location $\xi$ as

$$K_{\mathrm{I}} = K_{\mathrm{I}}(a, \sigma_n(\nabla u(\xi))) = \frac{2}{\pi}\sigma_n(\nabla u(\xi))\sqrt{\pi a}, \tag{3.4}$$

see, e.g., Table 4.1 in [76]. Following [20] we set

$$\sigma_n(\nabla u(\xi)) := \max\{n^\top \sigma(\nabla u(\xi))\, n\,,\, 0\}.$$

Note that negative values of $\sigma_n(\xi)$ correspond to compressive loads which can be ignored in the analysis of crack development, see Figure 3.2a above.

A crack $(\xi, a, n) \in \mathcal{C}$ becomes critical, i.e., a fracture occurs and the material fails, if $K_{\mathrm{I}}$ exceeds a material-specific critical value $K_{\mathrm{I}c}$ (the *ultimate tensile strength* of the material). Note that (3.4) implies that all cracks with radius

$$a > a_c := \frac{\pi}{4}\left(\frac{K_{\mathrm{I}c}}{\sigma_n(\nabla u(\xi))}\right)^2 \tag{3.5}$$

are critical. We denote the set of *critical configurations* by

$$A_c := A_c(\Omega, \nabla u) = \{(\xi, a, n) \in \mathcal{C} \ :\ K_{\mathrm{I}}(a, \sigma_n(\nabla u(\xi))) > K_{\mathrm{I}c}\}$$

and want to minimize the probability of finding a crack with configuration in $A_c$.

Following [79, 21, 80, 20], we assume that the parameters $(\xi, a, n)$ are random (i.e., they are not deterministically given by the sintering process), that the cracks are statistically homogeneously distributed in $\Omega$, and that their orientations are isotropic. Let $A \subseteq \mathcal{C}$ be a measurable subset of the configuration space. Then, under quite general assumptions the random number $N(A)$ of cracks in $A$ is Poisson distributed (see [93, 151]), and hence $N(A)$ is a Poisson point process. It follows that $P(N(A) = k) = e^{-\upsilon(A)}\frac{\upsilon(A)^k}{k!} \sim Po(\upsilon(A))$, where $\upsilon$ is the (Radon) *intensity measure* of the process. Recall that a component fails if $N(A_c) > 0$. Given a displacement field $u \in H^1(\Omega, \mathbb{R}^p)$, we can now write the survival probability of the component $\Omega$ as

$$p_s(\Omega|\nabla u) = P(N(A_c(\Omega, \nabla u)) = 0) = \exp\{-\upsilon(A_c(\Omega, \nabla u))\}.$$

Hence, to maximize the survival probability of a component $\Omega$ we need to minimize the intensity measure $\upsilon$. Since only cracks $(\xi, a, n)$ with radius $a > a_c$ need to be considered (c.f. (3.5) above), [79, 21, 80, 20] determine the intensity measure as

$$\upsilon(A_c(\Omega, \nabla u)) = \frac{\Gamma(\frac{d}{2})}{2\pi^{\frac{d}{2}}} \int_{\Omega} \int_{S^{d-1}} \int_{a_c}^{\infty} \mathrm{d}\upsilon_a(a)\, \mathrm{d}n\, \mathrm{d}\xi$$

with $\mathrm{d}\xi$ the Lebesgue measure on $\mathbb{R}^d$, $\mathrm{d}n$ the surface measure on $S^{d-1}$, and $\mathrm{d}\upsilon_a(a) = c \cdot a^{-\tilde{m}}\mathrm{d}a$ being a positive Radon measure modelling the occurrence of cracks of radius $a$ in $\Omega$ ($c > 0$ and $\tilde{m} \geq \frac{3}{2}$ are positive constants). Note that for $d = 3$ the $\Gamma$-function takes the value $\Gamma(\frac{3}{2}) = \frac{\sqrt{\pi}}{2}$ and for $d = 2$ we obtain $\Gamma(1) = 1$. With $m := 2(\tilde{m} - 1) \geq 1$ and

using again (3.5) the inner integral can be evaluated, yielding

$$v(A_c(\Omega, \nabla u)) = \frac{\Gamma(\frac{d}{2})}{2\pi^{\frac{d}{2}}} \int_{\Omega} \int_{S^{d-1}} \left( \frac{\sigma_n(\nabla u(\xi))}{\sigma_0} \right)^m \mathrm{d}n \, \mathrm{d}\xi, \tag{3.6}$$

where $\sigma_0$ is an appropriately chosen positive constant. As highlighted in [79, 21, 80, 20], this is in accordance with the statistical model introduced by Weibull [152]. In this context, the parameter $m$ is referred to as *Weibull module* and typically assumes values between 5 and 30.
Summarizing the discussion above, we define our primary objective function $f_1 : \mathcal{O}^{\mathrm{ad}} \to \mathbb{R}$ as

$$J_1(\Omega) := v(A_c(\Omega, \nabla u)) \tag{3.7}$$

and refer to it as *intensity measure*, modelling the probability of failure (PoF) of the component $\Omega$. Recall that $u(\Omega)$ is uniquely defined by $\Omega$ and thus $J_1(\Omega)$ is completely defined by the shape $\Omega$ (given fixed boundary conditions $\hat{f}, \hat{g}$).
Furthermore, for a theoretical analysis w.r.t. continuous dependency of optimal shapes on preference parameters in the context of shape optimization we refer to [72].

## 3.3 Material Consumption

Improving the intensity measure $J_1$ of a ceramic component (and hence its PoF) usually comes at the price of an increased material consumption, which is directly correlated with the cost of the component. In order to avoid excessively expensive solutions, classical approaches thus set a predetermined bound on the allowable volume of the shape $\Omega$ (see, e.g., [79, 21, 80, 20]). We follow a more general approach in this manuscript and interpret the volume (and hence the cost) of the component as an equitable second objective function. This facilitates, in particular, the analysis of the trade-off between these two criteria and supports the engineer in finding a preferable design. We thus define $J_0 : \mathcal{O}^{\mathrm{ad}} \to \mathbb{R}$ as the *volume* of a shape $\Omega \in \mathcal{O}^{\mathrm{ad}}$ given by

$$J_0(\Omega) := \int_{\Omega} \mathrm{d}\xi. \tag{3.8}$$

## 3.4 Biobjective Optimization

When multiple conflicting goals are relevant in an optimization problem, a common approach is to use a weighted sum of the individual objectives as an overall objective function and then resort to classical optimization algorithms. The advantages and also the shortcomings of this so-called *weighted sum scalarization* are discussed in the following section, see also [52]. Particularly when choosing fixed weights, this method is of limited applicability. While fixed weights may represent the preferences of one decision maker, another decision maker may have other preferences, i.e., other weights. Moreover, the objective ranges and the scales of the objectives may be very different or even incomparable, which

generally leads to numerical difficulties.
Another common approach to handle multiple conflicting goals is to select one "most important" objective function to minimize, e.g., the probability of failure, and set upper bounds on the acceptable objective function values of the other objective functions. In our case this would imply a constraint on the allowable material consumption, see, e.g., [79, 21, 80, 20]. This approach is referred to as *$\varepsilon$-constraint scalarization*, see again [52] for a general discussion of this topic. In addition to the numerical difficulties that may arise from adding potentially complicating constraints to the problem formulation, this approach has similar drawbacks as the weighted sum scalarization: The selection of meaningful upper bound values may be difficult, and trade-off information is ignored.
A more general approach is to formulate a multiobjective optimization problem, and hence to compute a set of relevant solution alternatives rather than one single "optimal" solution. By providing a set of solution alternatives the decision maker can not only choose a solution that aligns the most with his preferences, but he can also inspect the trade-off between alternative solutions and can adjust his preferences accordingly. A decision maker may, for example, prefer reliability over volume, but looking into the trade-off between solution alternatives there may be a solution that is some small percentage worse w.r.t. the reliability while it is a lot better regarding the volume. This may lead to a re-evaluation of the decision maker's preferences.
Next, a general definition of multiobjective optimization problems is given, while for the rest of this chapter biobjective optimization problems, which are multiobjective problems with two objective functions, are investigated. Without loss of generality only minimization problems are considered. Most of this section is based on [52]. Note that [109] also provides a suitable introduction to multiobjective optimization.

**Definition 3.1.** *Let $f = (f_1, \dots, f_p) : \mathcal{X} \to \mathbb{R}^p$, $p \geq 2$, be an objective function vector, where $\mathcal{X} \subseteq \mathbb{R}^n$ is called the* feasible set *and $\mathbb{R}^p$ the* objective space*. A* multiobjective optimization problem *is of the form*

$$\begin{aligned} &\min\ f(x) = (f_1(x), \dots, f_p(x)) \\ &s.t.\ x \in \mathcal{X}. \end{aligned} \tag{3.9}$$

In this work we introduce a rather unconventional notation for biobjective optimization problems, since this notation is more convenient for the method introduced in Chapter 6. The objective functions $f, f_1$ and $f_2$ are replaced by $J, J_0$ and $J_1$, respectively.

**Definition 3.2.** *Let $J = (J_0, J_1) : \mathcal{X} \to \mathbb{R}^2$, be an objective function vector, where $\mathcal{X} \subseteq \mathbb{R}^n$ is called the* feasible set *and $\mathbb{R}^2$ the* objective space*. A* biobjective optimization problem *is then of the form*

$$\begin{aligned} &\min\ J(x) = (J_0(x), J_1(x)) \\ &s.t.\ x \in \mathcal{X}. \end{aligned} \tag{3.10}$$

Therefore, with our two objective functions "intensity measure" ($J_1$, modeling the PoF) and "volume" ($J_0$), the following *biobjective shape optimization problem* arises:

$$\begin{aligned} &\min_{\Omega \in \mathcal{O}^{\mathrm{ad}}}\ J(\Omega) := (J_0(\Omega), J_1(\Omega)) \\ &\text{s.t. } u \in H^1(\Omega, \mathbb{R}^d) \text{ solves the state equation (3.2)}, \end{aligned} \tag{3.11}$$

where $J_1$ and $J_0$ are defined according to Sections 3.2 and 3.3 above. Note that only $J_1$ depends on the displacement field $u(\Omega)$. We call $J = (J_0, J_1) : \mathcal{O}^{\text{ad}} \longrightarrow \mathbb{R}^2$ the *biobjective function vector* and $\mathbb{R}^2$ the *objective space*. Later on, discretizations of admissible shapes are used for the numerical studies. To this end, a mapping $\mathcal{X} \to \mathcal{O}^{\text{ad}}$, $\mathcal{X} \subset \mathbb{R}^n$, that parameterizes the admissible shapes is described in Section 4.3. Thus, the problem (3.11) with the feasible set $\mathcal{O}^{\text{ad}}$ can be transformed as in (3.10) to a problem with feasible set $\mathcal{X} \subset \mathbb{R}^n$.
Let $Z := J(\mathcal{X}) \subset \mathbb{R}^2$ denote the set of all *feasible outcome vectors* in the objective space, i.e., the set of all outcome vectors that are images of admissible shapes $x \in \mathcal{X}$. Further, we assume that $J \in C^2$. In the following, we denote the gradient of $J_i$ at $x$ as $\nabla_x J_i(x)$, $i = 0, 1$. In contrast to single objective optimization, we have to define optimality in the presence of two objectives, since there is no natural order on $\mathbb{R}^2$. For two shapes $x_1, x_2 \in \mathcal{X}$, let $z^1 = J(x_1)$ and $z^2 = J(x_2)$ be the respective outcome vectors in $Z$. We write

$$\begin{aligned} z^1 \leqq z^2 &\iff z_j^1 \leq z_j^2,\ j = 0, 1 \\ z^1 \leqslant z^2 &\iff z^1 \leqq z^2 \text{ and } z^1 \neq z^2 \\ z^1 < z^2 &\iff z_j^1 < z_j^2,\ j = 0, 1. \end{aligned}$$

Note that $z^1 \leqslant z^2$ implies that $z_j^1 \leq z_j^2$ for $j = 0, 1$ with at least one strict inequality. We use the notation

$$\mathbb{R}^2_{\geqslant} := \{z \in \mathbb{R}^2 \,:\, z \geqslant (0,0)^\top\} \quad \text{and} \quad \bar{z} + \mathbb{R}^2_{\geqslant} := \{z \in \mathbb{R}^2 \,:\, z \geqslant \bar{z}\} \quad \text{for } \bar{z} \in \mathbb{R}^2.$$

The notations $\mathbb{R}^2_{\geqq}$, $\mathbb{R}^2_{>}$, $\mathbb{R}^2_{\leqslant}$, $\mathbb{R}^2_{\leqq}$ and $\mathbb{R}^2_{<}$ are used accordingly. We have the following Pareto optimality definitions w.r.t. $z^1$ and $z^2$.

**Definition 3.3** (Pareto optimality)**.**

*(i) We say that* $z^1$ dominates $z^2$ *if and only if* $z^1 \leqslant z^2$*, i.e., if and only if* $z^1 \in z^2 + \mathbb{R}^2_{\leqslant}$*.*

*(ii) An outcome vector* $\bar{z} \in Z$ *is called* nondominated *if there is no other outcome vector* $z \in Z$ *such that* $z \leqslant \bar{z}$*.*

*(iii) An admissible shape* $x_P \in \mathcal{X}$ *is called* Pareto-optimal *or* efficient*, if there is no other admissible shape* $x \in \mathcal{X}$ *such that* $J(x) \leqslant J(x_P)$*. The set of all Pareto-optimal shapes is called the* Pareto front *and denoted by* $\mathcal{X}_P$*. Similarly, the set of all nondominated outcome vectors* $Z_N := f(\mathcal{X}_P)$ *is referred to as the* nondominated front *in the objective space.*

*(iv) An admissible shape* $x_{\ell P} \in \mathcal{X}$ *is called* locally Pareto-optimal *or* locally efficient*, if there is a neighborhood* $\mathcal{N} \subseteq \mathcal{X}$ *of* $x_{\ell P}$ *such that there is no other admissible shape* $x \in \mathcal{N}$ *with* $f(x) \leqslant f(x_{\ell P})$*.*

*(v) An admissible shape* $x_{wP} \in \mathcal{X}$ *is called* weakly Pareto-optimal *or* weakly efficient*, if there is no other admissible shape* $x \in \mathcal{X}$ *such that* $J(x) < J(x_{wP})$*. The set of all weakly Pareto-optimal shapes is denoted by* $\mathcal{X}_{wP}$*. Similarly, the set of all weakly nondominated outcome vectors is denoted by* $Z_{wN}$*.*

*(vi) An admissible shape $x_{sP} \in \mathcal{X}$ is called* strictly Pareto-optimal *or* strictly efficient*, if there is no other admissible shape $x \in \mathcal{X}$ with $x \neq x_{sP}$ such that $J(x) \leqq J(x_{sP})$. The set of all strictly Pareto-optimal shapes is denoted by $\mathcal{X}_{sP}$.*

With this definition we have

$$Z_N \subseteq Z_{wN}$$

and

$$\mathcal{X}_{sP} \subseteq \mathcal{X}_P \subseteq \mathcal{X}_{wP}.$$

In biobjective optimization strictly efficient solutions correspond to unique optimal solutions in single objective optimization. We are mainly interested in Pareto-optimal shapes since these are precisely those shapes that can not be improved in one objective without deterioration in the other objective. As in single-objective optimization, one often has to resort to local minima if the underlying optimization problem is nonconvex (and difficult). Since derivative information is available, necessary optimality conditions can be formulated that generalize the concept of critical points from single-objective optimization. Toward this end, we omit the constraints implied by the parametric representation of admissible shapes to keep the exposition simple. All constraints will be handled implicitly in the numerical tests described in Chapters 5, 6 and 7. Assuming that both objective functions are continuously differentiable, a necessary condition for a solution $x \in \mathcal{X}$ to be locally Pareto-optimal, and a relaxation of this condition, can be formulated [60].

**Definition 3.4** (Pareto Critical)**.**

*(i) A* biobjective descent direction *$d \in \mathbb{R}^n$ at $x$, is a search direction such that we have $\nabla_x J_i(x)^\top d < 0$ for $i \in \{0, 1\}$. We say $x$ is* Pareto critical*, if*

$$\left\{ d \in \mathbb{R}^n \,|\, \nabla_x J_i(x)^\top d < 0,\, i = 0, 1 \right\} = \emptyset, \tag{3.12}$$

*i.e., there does not exist a direction $d \in \mathcal{X}$ that is a descent direction for both objectives*

*(ii) For $\varepsilon > 0$, $x$ is called* $\varepsilon$-Pareto critical*, if*

$$\left\{ d \in \mathbb{R}^n \,|\, \nabla_x J_i(x)^\top d \leq -\varepsilon \|d\| \wedge \nabla_x J_j(x)^\top d < 0,\ i, j \in \{0, 1\},\, i \neq j \right\} = \emptyset.$$

In this work, we aim at the efficient computation of Pareto critical shapes that, ideally, approximate the Pareto front. Since derivative information can be obtained for both objective functions, we select solution methods that efficiently utilize this information and that can be adopted such that a meaningful representation of a Pareto critical front is obtained. In the following, when we use the term *approximation of the Pareto front*, we imply a set of $\varepsilon$-Pareto critical points that cover a sufficient range of the (local) Pareto front. Next, some properties of the set of nondominated objective vectors $Z_N$ are stated.

### 3.4.1 Nondominated Set

Let $Z \subset \mathbb{R}^2$ and $Z_N$ the corresponding set of all nondominated outcome vectors. It is $Z_n \subseteq Z$. This subsection is based on [52].

**Proposition 3.5.** $Z_N = \left(Z + \mathbb{R}^2_{\leqq}\right)_N$.

*Proof.* For the trivial case $Z = \emptyset$ it directly follows that $Z + \mathbb{R}^2_{\leqq} = \emptyset$, and therefore both of their nondominated sets are empty, too.
Let now $Z \neq \emptyset$. First, we assume that $z \in (Z + \mathbb{R}^2_{\leqq})_N$, but $z \notin Z_N$. In that case there exist two possibilities. If $z \notin Z$ there exists a $z' \in Z$ and $0 \neq d \in \mathbb{R}^2_{\leqq}$ such that $z = z' + d$. Since $z' = z' + 0 \in Z + \mathbb{R}^2_{\leqq}$ we get $z \notin (Z + \mathbb{R}^2_{\leqq})_N$, which is a contradiction. If $z \in Z$ there exists a $z' \in Z$ with $z' \leq z$. We then have with $d = z - z' \in \mathbb{R}^2_{\leqq} \setminus \{0\}$ that $z = z' + d$. This implies that $z \notin (Z + \mathbb{R}^2_{\leqq})_N$, which is again a contradiction. Therefore in either case $z \in Z_N$.
Second, assume $z \in Z_N$, but $z \notin (Z + \mathbb{R}^2_{\leqq})_N$. Then, there exists a $z' \in Z + \mathbb{R}^2_{\leqq}$ with $z - z' = d' \in \mathbb{R}^2_{\leqq} \setminus \{0\}$. In other words, $z' = z'' + d''$ with $z'' \in Z, d'' \in \mathbb{R}^2_{\leqq} \setminus \{0\}2$, and we therefore have $z = z' + d' = z'' + (d' + d'') = z'' + d$ with $d = d' + d'' \in \mathbb{R}^2_{\leqq} \setminus \{0\}$. This implies $z \notin Z_N$, which contradicts the assumption. Hence, $z \in (Z + \mathbb{R}^2_{\leqq})_N$. □

In Figure 3.3 an illustration of Proposition 3.5 is given.

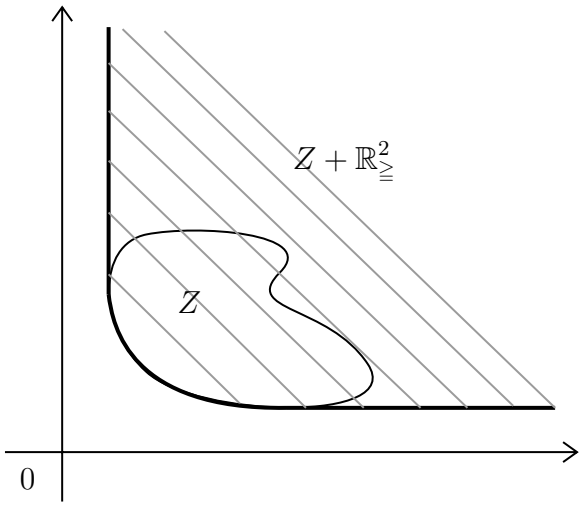



Figure 3.3: The nondominated points of $Z$ coincide with the nondominated points of $Z + \mathbb{R}^2_{\leqq}$.

Another result for $Z_N$ is that efficient points are located on the boundary of $Z$, that we denote as $\partial Z$. The interior of $Z$ is then given as $\text{int}(Z) := Z \setminus \partial Z$.

**Proposition 3.6.** $Z_N \subseteq \partial Z$

*Proof.* Let $z \in Z_N$ and assume that $z \notin \partial Z$. Hence, $z \in \text{int}(Z)$ and there exists an $\varepsilon$-neighbourhood $B_\varepsilon(z)$ of $z$. Here, $B_\varepsilon(z)$ is an open ball centered at $z$ with radius $\varepsilon$. Now choose $z' \in B_\varepsilon(z)$ such that $z = z' + d$ and $d \in \mathbb{R}^2_{\leqq}$, $d \neq 0$. We then have $z' \leq z$ and therefore $z \notin Z_N$, contradicting the assumption. □

## 3.5 Weighted Sum Scalarization

In this section, the *weighted sum method*, or *weighted sum scalarization*, which is a scalarization method that in order to solve multiobjective problems transforms them into single

objective problems, is introduced. This section is based on [52, 38].

For a multiobjective problem

$$\min_{x\in\mathcal{X}} f(x) = (f_1(x), \dots, f_p(x))$$

the *weighted sum scalarization* is given as

$$\min_{x\in\mathcal{X}} \sum_{k=1}^{p} \lambda_k f_k(x), \tag{3.13}$$

with $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p_{\geqq}$, where $\mathbb{R}^p_{\geqq}$ is defined analogously to $\mathbb{R}^2_{\geqq}$. Consequently, for our biobjective problem

$$\min_{x\in\mathcal{X}} J(x) = (J_0(x), J_1(x))$$

the *weighted sum scalarization* is given as

$$\min_{x\in\mathcal{X}} \lambda_0 J_0(x) + \lambda_1 J_1(x) \tag{3.14}$$

with $\lambda = (\lambda_0, \lambda_1) \in \mathbb{R}^2_{\geqq}$. Let $Z = J(\mathcal{X}) \in \mathbb{R}^2$. For a fixed $\lambda \in \mathbb{R}^2_{\geqq}$ we denote by

$$\mathcal{S}(\lambda, Z) := \left\{ \hat{z} = (\hat{z}_0, \hat{z}_1) \in Z \ : \ \langle \lambda, \hat{z} \rangle = \min_{z\in Z} \langle \lambda, z \rangle \right\} \tag{3.15}$$

the set of optimal points of $Z$ with respect to $\lambda$. In Figure 3.4, an example of an optimal set $\mathcal{S}(\lambda, Z)$ is given that consists of two points $z^1$ and $z^2$. These nondominated points are at the intersection points of a level set $\{z \in \mathbb{R}^2 : \langle \lambda, z \rangle = \hat{c}\}$. Considering the family of lines $(\{z \in \mathbb{R}^2 : \langle \lambda, z \rangle = c\})_c$, $\hat{c}$ is chosen in a way that it is the smallest value of $c$ such that the intersection of the corresponding line and $Z$ is nonempty. To find $\hat{c}$ graphically, we start with a sufficiently large value of $c$ and translate the line in parallel toward the origin as far as possible while assuring that the intersection of the line and $Z$ is still nonempty. An analytical approach translates to finding elements of $\mathcal{S}(\lambda, Z)$. For this purpose, we write $\mathrm{ri}(Z)$ for the relative interior of $Z$, i.e., $\mathrm{ri}(Z) = \{z \in Z \ : \ \exists\, \varepsilon > 0,\ B_\varepsilon(z) \cap \mathrm{aff}(Z) \subseteq Z\}$, where $\mathrm{aff}(Z) = \{\sum_{i=1}^k \alpha_i z^i \ : \ k > 0, z^i \in Z, \alpha_i \in \mathbb{R}, \sum_{i=1}^k \alpha_i = 1\}$ is the *affine hull* of $Z$. Note that following the definition of nondominated points we only consider nonnegative weights $\lambda \in \mathbb{R}^2$. A further distinction between nonnegative and positive weights is made. Toward this end, we define the following sets

$$\begin{aligned} \mathcal{S}(Z) &:= \bigcup_{\lambda\in\mathbb{R}^2_{>}} \mathcal{S}(\lambda, Z) = \bigcup_{\{\lambda>0 \ : \ \lambda_0+\lambda_1=1\}} \mathcal{S}(\lambda, Z) \\ \mathcal{S}_0(Z) &:= \bigcup_{\lambda\in\mathbb{R}^2_{\geqq}} \mathcal{S}(\lambda, Z) = \bigcup_{\{\lambda\geqq 0 \ : \ \lambda_0+\lambda_1=1\}} \mathcal{S}(\lambda, Z). \end{aligned} \tag{3.16}$$

Note that the assumption $\lambda_0 + \lambda_1 = 1$ just normalizes the weights and does not change $\mathcal{S}(\lambda, Z)$. In the following, we assume that the weights are normalized. For convenience

we introduce following notation

$$\begin{aligned} \Lambda &:= \left\{\lambda \in \mathbb{R}^2_{>} \,:\, \lambda_0 + \lambda_1 = 1\right\}, \\ \Lambda_0 &:= \mathrm{ri}(\Lambda) = \left\{\lambda \in \mathbb{R}^2_{\geqq} \,:\, \lambda_0 + \lambda_1 = 1\right\}. \end{aligned} \tag{3.17}$$

Since for $\lambda = 0$ we have $\mathcal{S}(0, Z) = Z$, we exclude this case. Further, from the definition it directly follows that

$$\mathcal{S}(Z) \subseteq \mathcal{S}_0(Z). \tag{3.18}$$

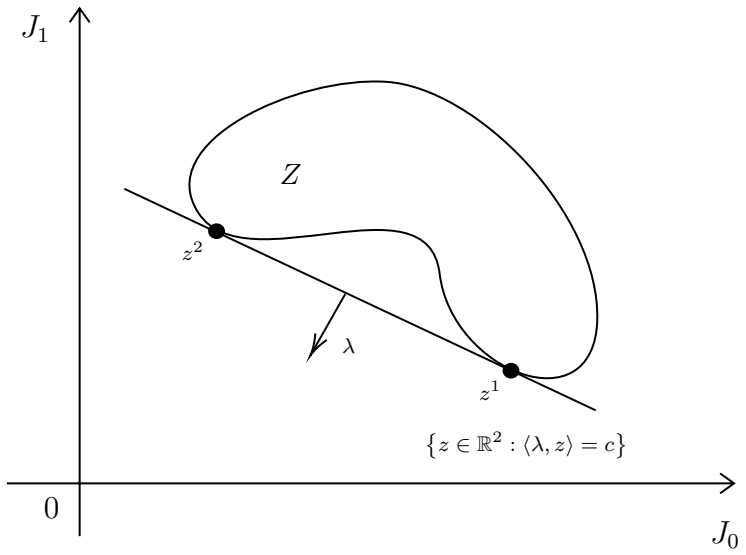



Figure 3.4: Exemplary optimal set $\mathcal{S}(\lambda, Z)$ containing the points $z^1$ and $z^2$.

For many of the following results of this section some convexity assumptions are needed. Since, following Proposition 3.5 the nondominated points are located "south west" of $Z$ [52], we define $\mathbb{R}^2_{\geqq}$-convexity.

**Definition 3.7.** *A set $Z \subset \mathbb{R}^2$ is called $\mathbb{R}^2_{\geqq}$-convex if $Z + \mathbb{R}^2_{\geqq}$ is convex.*

Note that every convex set $Z$ is $\mathbb{R}^2_{\geqq}$-convex. The set $Z$ depicted in Figure 3.4 is neither convex nor $\mathbb{R}^2_{\geqq}$-convex. The set $Z$ illustrated in Figure 3.3 is nonconvex but $\mathbb{R}^2_{\geqq}$-convex. For convex sets it is known that nonintersecting convex sets can be separated by a hyperplane.

**Theorem 3.8.** *Let $Z_1, Z_2 \subset \mathbb{R}^2$ be nonempty convex sets. There exists some $z' \in \mathbb{R}^2$ such that*

$$\begin{aligned} \inf_{z \in Z_1} \langle z, z'\rangle &\geq \sup_{z \in Z_2} \langle z, z'\rangle \\ \textit{and} \quad \sup_{z \in Z_1} \langle z, z'\rangle &> \inf_{z \in Z_2} \langle z, z'\rangle \end{aligned} \tag{3.19}$$

*if and only if $ri(Z_1) \cap ri(Z_2) = \emptyset$. $Z_1$ and $Z_2$ are then called properly separated by a hyperplane with normal $z'$.*

For a proof we refer to [126]. In the following, a theorem that describes the relation between weighted sum solutions and (weakly) efficient solutions of a biobjective optimization problem is given.

**Theorem 3.9.** *For any $Z \subset \mathbb{R}^2$ we have $\mathcal{S}_0(Z) \subseteq Z_{wN}$.*

*Proof.* Let $\lambda \in \mathbb{R}^2_{\leqq}$ and $\hat{z} \in \mathcal{S}(\lambda, Z)$. Then,

$$\langle \lambda, \hat{z} \rangle \leq \langle \lambda, z \rangle \quad \text{for all } z \in Z.$$

Assume that $\hat{z} \notin Z_{wN}$. Then, there exists a $z' \in Z$ with $z' < \hat{z}$ and therefore

$$\langle \lambda, z' \rangle < \langle \lambda, \hat{z} \rangle,$$

since at least one of the weights $\lambda_0$ and $\lambda_1$ has to be positive. This contradicts the assumption. □

For $\mathbb{R}^2_{\leqq}$-convex sets the converse inclusion of this theorem can be shown.

**Theorem 3.10.** *If $Z \subset \mathbb{R}^2$ is $\mathbb{R}^2_{\leqq}$-convex, then $\mathcal{S}_0(Z) = Z_{wN}$*

*Proof.* Following Theorem 3.9 we only have to show that $Z_{wN} \subseteq \mathcal{S}_0(Z)$. Replacing $\mathbb{R}^2_{\leqq}$ with $\mathbb{R}^2_{>}$ in the proof of Proposition 3.5 one can show that $Z_{wN} \subseteq (Z + \mathbb{R}^2_{>})_{wN}$. Therefore, for $\hat{z} \in Z_{wN}$ we have

$$(Z_{wN} + \mathbb{R}^2_{>} - \hat{z}) \cap (-\mathbb{R}^2_{>}) = \emptyset.$$

In other words, the intersection of the relative interior of these two convex sets is empty. Theorem 3.8 guarantees the existance of some $\lambda \in \mathbb{R}^2 \setminus \{0\}$ such that

$$\langle \lambda, z + d + \hat{z} \rangle \geq 0 \geq \langle \lambda, -d' \rangle \tag{3.20}$$

for all $z \in Z$ and $d, d' \in \mathbb{R}^2_{>}$.
Since $\langle \lambda, -d' \rangle \leq 0$ for all $d' \in \mathbb{R}^2_{>}$ we choose $d' = e_k + \varepsilon e$, where $\varepsilon > 0$ arbitrarily small, $e_k$ is the $k$-th unit vector, i.e., $e_1 = (1, 0)$ and $e_2 = (0, 1)$, and $e = (1, 1) \in \mathbb{R}^2$ a vector of all ones, to show that $\lambda_0, \lambda_1 \geq 0$. Further, choosing $d = \varepsilon e$ in $\langle \lambda, z + d + \hat{z} \rangle \geq 0$ implies

$$\langle \lambda, z \rangle + \varepsilon \langle \lambda, e \rangle \geq \langle \lambda, \hat{z} \rangle \tag{3.21}$$

for all $z \in Z$ and therefore

$$\langle \lambda, z \rangle > \langle \lambda, \hat{z} \rangle. \tag{3.22}$$

Hence, $\lambda \in \mathbb{R}^2_{\leqq}$ and $\hat{z} \in \mathcal{S}(\lambda, Z) \subseteq \mathcal{S}(Z)$. □

Next we relate the weighted sum scalarization sets of optimal points $\mathcal{S}_0(Z)$ and $\mathcal{S}(Z)$ to $Z_N$.

**Theorem 3.11.** *Let $Z \in \mathbb{R}^2$. Then, $\mathcal{S}(Z) \subseteq Z_N$.*

*Proof.* Let $\hat{z} \in \mathcal{S}(Z)$. Then, there exists some $\lambda \in \mathbb{R}^2_{>}$ satisfying $\langle \lambda, \hat{z} \rangle \geq \langle \lambda, z \rangle$ for all $z \in Z$. Assume that $\hat{z} \notin Z_N$. Therefore there exists a $z' \in Z$ with $z' \leq \hat{z}$. Furthermore, componentwise multiplication with the weights gives $\lambda_k z'_k \leq \lambda_k \hat{z}_k$, $k = 0, 1$, and a strict inequality for at least one $k$. This strict inequality and the fact that $\lambda \in \mathbb{R}^2_{>}$ implies that $\lambda_0 z'_0 + \lambda_1 z_1 < \lambda_0 \hat{z}'_0 + \lambda_1 \hat{z}_1$, which contradicts $\hat{z} \in \mathcal{S}(Z)$. □

**Corollary 3.12.** *Let $Z \in \mathbb{R}^2$. If $Z$ is $\mathbb{R}^2_{\leqq}$-convex, then $Z_N \subseteq \mathcal{S}_0(Z)$.*

*Proof.* This follows immediately from Theorem 3.11, since $Z_N \subseteq Z_{wN} = \mathcal{S}_0(Z)$. □

Furthermore, one can extend this theorem for the case of unique elements of $\mathcal{S}(\lambda, Z)$.

**Proposition 3.13.** *If $\{\hat{z}\} = \mathcal{S}(\lambda, Z)$ for some $\lambda \in \mathbb{R}^2_{\geqslant}$ then $\hat{z} \in Z_N$.*

*Proof.* Suppose $\hat{z} \notin Z_N$. Then, there exists $z' \in Z$, such that $z' \leq \hat{z}$. Furthermore, componentwise multiplication with the weights gives $\lambda_k z'_k \leq \lambda_k \hat{z}_k$, $k = 0, 1$. There are two possibilities, since $\lambda \in \mathbb{R}^2_{\geqslant}$. First, if $\langle \lambda, z' \rangle < \langle \lambda, \hat{z} \rangle$, $\hat{z} \notin \mathcal{S}(\lambda, Z)$ follows, which is a contradiction. Second, if $\langle \lambda, z' \rangle = \langle \lambda, \hat{z} \rangle$, we have $z' \in \mathcal{S}(\lambda, Z)$. Which is a contradiction, since $\hat{z}$ is the unique element of $\mathcal{S}(\lambda, Z)$. Hence, $\hat{z} \in Z_N$. □

Summarizing the results of Theorem 3.11 and Corollary 3.12 we have the following implications

$$\mathcal{S}(Z) \subseteq Z_N; \quad \mathcal{S}_0(Z) \subseteq Z_{wN} \tag{3.23}$$

in general and

$$\mathcal{S}(Z) \subseteq Z_N \subseteq \mathcal{S}_0(Z) = Z_{wN} \tag{3.24}$$

for $\mathbb{R}^2_{\leqq}$-convex sets. In the following, theorem all the results of this chapter are summarized.

**Proposition 3.14.** *Let $\hat{x}$ be an optimal solution of the weighted sum problem*

$$\min_{x \in \mathcal{X}} \lambda_0 J_0(x) + \lambda_1 J_1(x) \tag{3.25}$$

*with $\lambda = (\lambda_0, \lambda_1) \in \mathbb{R}^2_{\leqq}$. Then, the following statements hold.*

1. *If $\lambda \in \mathbb{R}^2_{\leqq}$, then $\hat{x} \in \mathcal{X}_{wP}$.*
2. *If $\lambda \in \mathbb{R}^2_{>}$, then $\hat{x} \in \mathcal{X}_P$.*
3. *If $\lambda \in \mathbb{R}^2_{\geqslant}$, and $\hat{x}$ is the unique solution of (3.25) then $\hat{x} \in \mathcal{X}_{sP}$.*

*Proof.* This follows directly from Theorem 3.9, Theorem 3.11 and Proposition 3.13. □

For convex problems we thus have.

**Proposition 3.15.** *Let $\mathcal{X}$ be a convex set, and let $J_0, J_1 : \mathcal{X} \to \mathbb{R}^2$ be convex functions. If $\hat{x} \in X_{wP}$ there exists a weighting vector $\lambda \in \mathbb{R}^2_{\geqslant}$ such that $\hat{x}$ is an optimal solution of (3.25).*

*Proof.* This follows from Theorem 3.10. □

Therefore, in this case every solution of the weighted sum scalarization (3.14) is Pareto-optimal for (3.11). Nevertheless, for the numerical experiments of this work we consider weights $\lambda \in \mathbb{R}^2_{>}$. Thus, following (3.23) it is not guaranteed that we can recover all $\hat{x} \in X_{wP}$ even if $\mathcal{X}$ is a convex set, and $J_0, J_1 : \mathcal{X} \to \mathbb{R}^2$ are convex functions. Furthermore, a disadvantage of the weighted sum method is, however, that only solutions that map to the convex hull $\operatorname{conv}(Z) = \{\sum_{i=1}^k \alpha_i z^i \,:\, z^i \in Z, k > 0, \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1\}$ of the image

set $Z = f(\mathcal{X})$ in the objective space can be found, and thus relevant compromise solutions in nonconvex areas of the nondominated front may be missed. Moreover, [38] showed at simple biobjective test instances that evenly distributed weights do in general not lead to well distributed outcome vectors in the objective space. This is particularly problematic if the considered objective function values are of largely different magnitude, which is the case here. In order to obtain solutions that are consistent with the preferences expressed by $\lambda$, we thus normalize the objective functions by using appropriate scaling factors $\kappa_0, \kappa_1 > 0$, and replace $J_0$ and $J_1$ in (3.14) by $\kappa_0 J_0$ and $\kappa_1 J_1$, respectively. Despite the difficulties mentioned above, the weighted sum method is usually well-suited to efficiently compute at least a rough approximation of the Pareto front. Note that from here on for simplicity we still refer to the objectives as $J_0$ and $J_1$, while assuming that they are approximately scaled.

### 3.5.1 First and Second-Order Optimality Conditions

Recall, that we normalize the weights, i.e., $\lambda_0 + \lambda_1 = 1$, in the discussion in the previous section. To normalize the weights of the weighted sum scalarization of a biobjective optimization problem it is sufficient to choose one $\lambda \in [0,1]$ and set $\lambda_0 = 1 - \lambda$ and $\lambda_1 = \lambda$. Hence, one can reformulate the weighted sum sclarization (3.14) to

$$\min_{x \in \mathcal{X}} J_\lambda(x) := (1-\lambda) J_0(x) + \lambda J_1(x) \tag{3.26}$$

with $\lambda \in [0,1]$. Note that with this notation $J_\lambda$ is equal to $J_0$ for $\lambda = 0$ and $J_1$ for $\lambda = 1$, respectively. Moreover, if $J_0, J_1$ (and therefore $J_\lambda$) are two times differentiable one can formulate the following first and second-order optimality conditions.

**Definition 3.16.** *Let $\hat{x} \in \mathcal{X}$ and let $J_0, J_1 : \mathcal{X} \to \mathbb{R}$ be differentiable. Further, let $J_\lambda = (1-\lambda)J_0 + \lambda J_1$ be the weighted sum scalarization for some weight $\lambda \in [0,1]$.*

*(i) We say $\hat{x}$ is* (locally) optimal *with respect to $\lambda$, if $J_\lambda(\hat{x}) \leq J_\lambda(x')$ for $x' \in \mathcal{X}$ ($x' \in \mathcal{U} \subset \mathcal{X}$, with $\mathcal{U} = B_\varepsilon(x)$).*

*(ii) We say $\hat{x}$ is* critical *for $J_\lambda$, or* $\lambda$-critical*, if*

$$\nabla_x J_\lambda(\hat{x}) = (1-\lambda)\nabla_x J_0(\hat{x}) + \lambda \nabla_x J_1(\hat{x}) = 0. \tag{3.27}$$

*(iii) For $\varepsilon > 0$, $\hat{x}$ is called* $\varepsilon$-critical with respect to $J_\lambda$, *if*

$$\|\nabla_x J_\lambda(\hat{x})\| \leq \varepsilon. \tag{3.28}$$

**Lemma 3.17.** *If for some $\varepsilon > 0$ and $\lambda \in (0,1)$ the point $\hat{x} \in \mathcal{X} \subset \mathbb{R}^n$ is* $\varepsilon$-critical with respect to $J_\lambda$, *i.e., $\|\nabla_x J_\lambda(\hat{x})\| \leq \varepsilon$, it follows directly that $\hat{x}$ is also $\varepsilon'$-Pareto critical with $\varepsilon' = \frac{\varepsilon}{\min\{\lambda,(1-\lambda)\}}$.*

*Proof.* Assume the contrary, i.e., that there exists a search direction $d \in \mathbb{R}^n$ such that

$$\nabla_x J_i(\hat{x})^\top d \leq -\varepsilon' \|d\| \quad \text{and} \quad \nabla_x J_j(\hat{x})^\top d < 0, \quad i,j \in \{0,1\},\ i \neq j,$$

where we choose $j = 0$ and $i = 1$ without loss of generality. Since $\hat{x}$ is $\varepsilon$-critical with respect to $J_\lambda$ we have

$$\left\|(1-\lambda)\nabla_x J_0(\hat{x})^\top d + \lambda \nabla_x J_1(\hat{x})^\top d\right\| = \left\|\nabla_x J_\lambda(\hat{x})^\top d\right\| \leq \|\nabla_x J_\lambda(\hat{x})\|\|d\| \leq \varepsilon\|d\|. \quad (3.29)$$

Furthermore, multiplying both sides of (3.29) with $1/\min\{\lambda, (1-\lambda)\} > 0$ yields

$$\left\|\frac{(1-\lambda)}{\min\{\lambda,(1-\lambda)\}}\nabla_x J_0(x)^\top d + \frac{\lambda}{\min\{\lambda,(1-\lambda)\}}\nabla_x J_1(x)^\top d\right\| \leq \frac{\varepsilon\|d\|}{\min\{\lambda,(1-\lambda)\}} = \varepsilon'\|d\|. \quad (3.30)$$

Thus, contradicting

$$\left\|\underbrace{\frac{(1-\lambda)}{\min\{\lambda,(1-\lambda)\}}\nabla_x J_0(x)^\top d}_{<0} + \underbrace{\frac{\lambda}{\min\{\lambda,(1-\lambda)\}}\nabla_x J_1(x)^\top d}_{\leq -\varepsilon'\|d\|}\right\| > \varepsilon'\|d\|.$$

Hence, the assertion follows. □

Next, recall the concept of *positive definiteness*.

**Definition 3.18.** *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called positive definite if $x^\top Ax > 0$ for all $x \in \mathcal{X} \subset \mathbb{R}^n$.*

With this property second-order optimality conditions can be stated.

**Definition 3.19.** *Let $\hat{x} \in \mathcal{X}$ and let $J_0, J_1 : \mathcal{X} \to \mathbb{R}$ be two times differentiable. Further, let $J_\lambda = (1-\lambda)J_0 + \lambda J_1$ be the weighted sum scalarization for some weight $\lambda \in [0,1]$ and let $\nabla_x^2 J_\lambda(\hat{x})$ be the Hessian of $J_\lambda$ at $\hat{x}$. Then, we say that $\hat{x}$ satisfies* the second-order optimality conditions for $J_\lambda(\hat{x})$ strictly *if $\hat{x}$ is $\lambda$-critical and $\nabla_x^2 J_\lambda$ is strictly positive definite.*

Note that if $\hat{x}$ satisfies strict second-order $J_\lambda$-optimality, it is locally $J_\lambda$-optimal, i.e., $\hat{x}$ is a local Pareto-optimal point.

## 3.6 Existence of Pareto-optimal Shapes

In order to prove the existence of Pareto-optimal shapes, we consider the weighted sum scalarization (3.26) of problem (3.11), since following Section 3.5 every optimal solution of problem (3.26) is Pareto-optimal for problem (3.11). Moreover, following [20] and as a continuation of Section 3.2, some assumptions for the crack size measure $\mathrm{d}\upsilon_a$ have to be made to show the existence of Pareto-optimal shapes.

**Definition 3.20.** *A crack size measure $d\upsilon_a$ has the* non decreasing hazard property*, if and only if the function $\Upsilon : (0,1) \to \mathbb{R}$, which is defined as*

$$\Upsilon(k) := d\upsilon_a\left(\left]\frac{1}{k^2}, \infty\right[\right), \quad (3.31)$$

*is convex in $k$.*

**Theorem 3.21.** *If the crack size measure $d\upsilon_a$ has the non decreasing stress hazard property then the set $\mathcal{O}_P^{ad}$ is non-empty.*

*Proof.* Suppose that $\lambda \in [0,1]$ is chosen arbitrarily, but fixed. Then, the weighted sum objective can be evaluated as

$$\begin{aligned} J_\lambda(\Omega) &= \lambda \left( \frac{\Gamma(\frac{p}{2})}{2\pi^{\frac{p}{2}}} \int\limits_\Omega \int\limits_{S^{p-1}} \left( \frac{\sigma_n(\nabla u(\xi))}{\sigma_0} \right)^m \mathrm{d}n \, \mathrm{d}\xi \right) + (1-\lambda) \int\limits_\Omega \mathrm{d}\xi \\ &= \lambda \frac{\Gamma(\frac{p}{2})}{2\pi^{\frac{p}{2}}} \int\limits_\Omega \int\limits_{S^{p-1}} \left( \frac{\sigma_n(\nabla u(\xi))}{\sigma_0} \right)^m \mathrm{d}n \; + \; \underbrace{\frac{2\pi^{\frac{p}{2}}(1-\lambda)}{\Gamma(\frac{p}{2})\lambda}}_{\text{constant}} \mathrm{d}\xi. \end{aligned}$$

Thus, the incorporation of the volume $J_0$ into the scalarized objective function corresponds to the addition of a constant term in the shape integral of $J_1$. This does not affect the convergence analysis of [20], which is based on convexity of the integrand in $\nabla u$, see [28, 66]. We can conclude that the weighted sum scalarization has an optimal solution for every $\lambda \in [0,1]$. Since every such solution is Pareto-optimal for (3.11), the result follows. □

# 4 Discretization of the Objective Functionals and the Numerical Test Cases

To actually compute locally Pareto optimal shapes, we adopt the finite element discretization implemented in [79] for two-dimensional instances (i.e., $d = 2$). In this implementation, the shapes $\Omega \in \mathcal{O}^{\mathrm{ad}}$, the state equation (3.2), the objective functions $J_0$ and $J_1$ and their gradients are discretized. Standard Lagrangian finite elements are used for the discretization of the state equation (3.2), and all integrals are calculated using numerical quadrature. The discretized shape gradients are obtained by an adjoint approach to reduce computational costs.
In the following, based on [79, 21, 80] we give a brief overview over the above mentioned discretization in Section 4.1 and the adjoint approach in Section 4.2. Further, the geometry definition and finite element mesh utilized in this work is introduced in Section 4.3. Subsequently, in Section 4.4 the two ceramic test cases that are considered in this work are presented. Some parts of this chapter, mainly Sections 4.3 and 4.4 are already published in [46].

## 4.1 Discretization of the Objective Functionals

Now, that we have stated our biobjective optimization problem (3.11), following [79, 21, 80] we discretize the objective functions for the two-dimensional case via the finite element method. To this end, we utilize Lagrangian nodes as described in Chapter 2. Recall that we have

- $N_G$ grid points $X = \{X_1, ..., X_{N_G}\}$,
- $N_{el}$ Lagrange finite elements $\{K, \Pi'(K), \Sigma'(K)\}$,
- $n_{sh}$ local shape functions $\theta_k^K \in \Pi'(K)$ defined by local nodes $X_1^K, ..., X_{n_{sh}}^K \in K$,
- a reference element $\{\hat{K}, \hat{\Pi}', \hat{\Sigma}'\}$ with
- a bijective transformation $T_K : \hat{K} \to K$ for each $K$ and
- quadrature points $\hat{\xi}_l^K$ and weights $\hat{\omega}_l$.

Further, recall that the inner integral of (3.6), i.e., the PoF objective function, is given as

$$I(u) := \int_{S^1} \left( \left( n^\top \sigma(\nabla u(\xi)) n \right)^+ \right)^m \mathrm{d}n.$$

Then, a transformation via polar coordinates yields

$$I(u) = \int_0^{2\pi} \left( \left( \cos^2(\varphi)\sigma_{11} + 2\cos(\varphi)\sin(\varphi)\sigma_{12} + \sin^2(\varphi)\sigma_{22} \right)^+ \right)^m \mathrm{d}\varphi,$$

for which in [79, 21, 80] it is shown that the discretized objective functional is of the form

$$\begin{aligned}
J(\Omega, u) &= \sum_{K \in \mathcal{T}_h} \int_K \hat{\psi}\left(\sigma\left(\xi\right), \varphi\right) \mathrm{d}\xi \\
&= \sum_{K \in \mathcal{T}_h} \int_{\hat{K}} \hat{\psi}\left(\sigma\left(T_K\left(\hat{\xi}\right)\right), \varphi\right) \det\ \left(\hat{\nabla} T_K\left(\hat{\xi}\right)\right) \mathrm{d}\hat{\xi} \\
&\approx \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{q^K} \hat{\omega}_l^K \frac{2\pi}{n} \left( \left( \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{11}^+ \right)^m \right. \\
&+ \sum_{i=1}^{n-1} \left( \left( \cos^2\left(\frac{i2\pi}{n}\right) \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{11} \right. \right. \\
&+\ 2\cos\left(\frac{i2\pi}{n}\right) \sin\left(\frac{i2\pi}{n}\right) \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{12} \\
&+\ \left. \left. \sin^2\left(\frac{i2\pi}{n}\right) \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{22} \right)^+ \right)^m \Bigg) \cdot \det\ \left(\hat{\nabla} T_K\left(\hat{\xi}_l^K\right)\right),
\end{aligned}$$

where $\hat{\psi}(\sigma, \phi)$ is obtained with the trapezoidal rule for $n$ interpolation points. For further details we refer to [79, 21, 80]. The much simpler volume objective function is discretized in an analogous way.

## 4.2 Adjoint Equation

In this section, a cost-efficient way to calculate discretized gradients, i.e., the *adjoint approach*, is introduced. Note that this section is based on the corresponding section in [45]. For our gradient based optimization methods we need the derivative of $J(X, U)$ with respect to $X$ which is given as

$$\frac{\mathrm{d}J(X, U(X))}{\mathrm{d}X} = \frac{\partial J(X, U(X))}{\partial X} + \frac{\partial J(X, U(X))}{\partial U} \frac{\partial U(X)}{\partial X}. \tag{4.1}$$

Computationally a cost problem arises when computing the gradients straightforward, since the computation of $\frac{\partial U(X)}{\partial X}$ is very expensive. Therefore, one want to compute the gradients without using this term by applying the discrete adjoint approach. First, con-

sider the derivative of the state equation

$$\begin{aligned} \frac{\partial B(X)}{\partial X}U(X) + B(X)\frac{\partial U(X)}{\partial X} &= \frac{\partial \hat{F}(X)}{\partial X} \\ \Leftrightarrow \quad \frac{\partial U(X)}{\partial X} &= B(X)^{-1}\left[\frac{\partial \hat{F}(X)}{\partial X} - \frac{\partial B(X)}{\partial X}U(X)\right]. \end{aligned} \tag{4.2}$$

Then by substituting $\frac{\partial U(X)}{\partial X}$ in (4.1) with (4.2) we obtain

$$\frac{dJ(X,U(X))}{dX} = \frac{\partial J(X,U(X))}{\partial X} + \frac{\partial J(X,U(X))}{\partial U}B(X)^{-1}\left[\frac{\partial \hat{F}(X)}{\partial X} - \frac{\partial B(X)}{\partial X}U(X)\right].$$

Further, let

$$\hat{\Lambda} := \frac{\partial J(X,U(X))}{\partial U}B(X)^{-1}.$$

Furthermore, with the fact that $B$ is symmetric we can deduce the *adjoint equation*

$$B^T(X)\,\hat{\Lambda} = \frac{\partial J(X,U(X))}{\partial U},$$

which gives us the *adjoint state method*:

If $\hat{X}$ solves the minimization problem $\min J(X,U(X))$ *s.t.* $B(X)\,U(X) = \hat{F}(X)$, then it also is a solution of the following system of equations

$$\begin{aligned} B^T(X)\hat{\Lambda} &= \frac{\partial J(X,U(X))}{\partial U} \\ B(X)\,U(X) &= \hat{F}(X) \\ \frac{\partial J(X,U(X))}{\partial X} + \hat{\Lambda}\left[\frac{\partial \hat{F}(X)}{\partial X} - \frac{\partial B(X)}{\partial X}U(X)\right] &= 0. \end{aligned} \tag{4.3}$$

### 4.2.1 Derivative of the Objective Functional

In this subsection, we explain the discretization of the adjoint equation (4.3). Following [79, 21, 80], we discretize the derivatives of $J$ with respect to $U$ and $X$, and refer to [73, 129] for the other derivatives. In contrast to [73, 129], we only consider local derivatives instead of global derivatives. To this end, recall that to compute discretized objective function values of $J$ a sum over all objective values on finite elements $K \in \mathcal{T}_h$ is taken into account. In a local approach the computation of the local derivatives of $J^{\text{loc}}$ with respect to $U^{\text{loc}}$ and $X^{\text{loc}}$ takes place on every $K$ separately and are then assembled into global derivatives. Thus, for every $K \in \mathcal{T}_h$ we only consider the local properties like the shape functions $\vartheta_{K,k}$ and degrees of freedom. For any $K \in \mathcal{T}_h$ with $\omega_l := \hat{\omega}_l^K \cdot \det \hat{\nabla} T_K(\hat{\xi_l^K})$, for $l = 1, ..., q^K$,

we then have

$$
\begin{aligned}
J^{\mathrm{loc}}(X, U(X)) = \sum_{l=1}^{q^K} \omega_l \frac{2\pi}{n} \Bigg( \Bigg( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \Bigg)^m \\
+ \sum_{i=1}^{n-1} \Bigg( \Bigg( \cos^2 \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11} \\
+ 2 \cos \left( \frac{i2\pi}{n} \right) \sin \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{12} \\
+ \sin^2 \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{22} \Bigg)^{+} \Bigg)^m \Bigg),
\end{aligned}
$$

For notational simplicity we set

$$
\begin{aligned}
T_i^{(n)} := \cos^2 \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l \right) \right)_{11} \\
+ 2 \cos \left( \frac{i2\pi}{n} \right) \sin \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l \right) \right)_{12} \\
+ \sin^2 \left( \frac{i2\pi}{n} \right) \sigma \left( T_K \left( \hat{\xi}_l \right) \right)_{22}.
\end{aligned} \tag{4.4}
$$

### Derivative with Respect to $U^{\mathrm{loc}}$

Let $K \in \mathcal{T}_h$. Further, let $j = 1, ..., n_{sh}$ be the index of corresponding shape functions and $k = 1, 2$ the dimension. Then, the local derivative with respect to the local degrees of freedom $U$ is given as

$$
\begin{aligned}
\frac{\partial J^{\mathrm{loc}}(X, U(X))}{\partial U_{jk}^{\mathrm{loc}}} &= \frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \sum_{l=1}^{q^K} \omega_l \frac{2\pi}{n} \left( \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m + \sum_{i=1}^{n-1} \left( \left( T_i^{(n)} (\hat{\xi}_l^K) \right)^{+} \right)^m \right) \\
&= \sum_{l=1}^{q^K} \omega_l \frac{2\pi}{n} \left( \frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m \right. \\
&\quad \left. + \sum_{i=1}^{n-1} \frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \left( \left( T_i^{(n)} (\hat{\xi}_l^K) \right)^{+} \right)^m \right).
\end{aligned}
$$

In a next step, we calculate the derivatives of $(\sigma(T_K(\hat{\xi}_l^K))_{11}^{+})^m$. We get

$$
\begin{aligned}
\frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m = & \mathbb{1}_{\{\sigma(T_K(\hat{\xi}_l^K))_{11} > 0\}} (\hat{\xi}_l^K) \frac{\partial \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}}{\partial U_{jk}^{\mathrm{loc}}} \\
& \times m \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11} \right)^{m-1}.
\end{aligned}
$$

To calculate $\frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \sigma(T_K(\hat{\xi}_l^K))_{11}$ a "bottom-up" approach is followed, i.e., first the smallest

required entity to calculate the next smallest entity is calculated and so forth. Hence, we begin by calculating the derivative of $\hat{\nabla}u(\hat{\xi}_l^K) := (\nabla u \circ T_K)(\hat{\xi}_l^K)$.

$$\begin{aligned}\frac{\partial \hat{\nabla} u(\hat{\xi}_l^K)_{i\ell}}{\partial U_{jk}^{\mathrm{loc}}} &= \sum_{r=1}^{n_{sh}} \sum_{s=1}^{2} \frac{\partial U_{rs}^{\mathrm{loc}}}{\partial U_{jk}^{\mathrm{loc}}} \left( \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right)^{-1} \right)_{s\ell} \hat{\nabla}_s \hat{\theta}_r(\hat{\xi}_l^K) \\ &= \sum_{r=1}^{n_{sh}} \sum_{s=1}^{2} \delta_{rj}\delta_{ik} \left( \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right)^{-1} \right)_{s\ell} \hat{\nabla}_s \hat{\theta}_r(\hat{\xi}_l^K) \\ &= \delta_{ik} \sum_{s=1}^{2} \left( \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right)^{-1} \right)_{s\ell} \hat{\nabla}_s \hat{\theta}_r(\hat{\xi}_l^K).\end{aligned}$$

With (2.27) and Lamé's constants we then have

$$\sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{i\ell} = \mu_{\mathrm{L}} \left( \hat{\nabla} u(\hat{\xi}_l^K)_{i\ell} + \hat{\nabla} u(\hat{\xi}_l^K)_{\ell i} \right) + \lambda_{\mathrm{L}} \delta_{i\ell} \sum_{r=1}^{2} \hat{\nabla} u(\hat{\xi}_l^K)_{rr}.$$

Thus, the derivative is then

$$\frac{\partial (\sigma \circ T_K)_{i\ell}}{\partial U_{jk}^{\mathrm{loc}}} = \mu_{\mathrm{L}} \left( \frac{\partial \hat{\nabla} u_{i\ell}}{\partial U_{jk}^{\mathrm{loc}}} + \frac{\partial \hat{\nabla} u_{\ell i}}{\partial U_{jk}^{\mathrm{loc}}} \right) + \lambda_{\mathrm{L}} \delta_{i\ell} \sum_{r=1}^{2} \frac{\partial \hat{\nabla} u_{rr}}{\partial U_{jk}^{\mathrm{loc}}}.$$

In a next step $\frac{\partial}{U_{jk}^{\mathrm{loc}}}((T_i^{(n)}(\hat{\xi}_l^K))^+)^m$ is calculated. We obtain

$$\frac{\partial}{\partial U_{jk}^{\mathrm{loc}}} \left( \left( T_i^{(n)}(\hat{\xi}_l^K) \right)^+ \right)^m = \mathbb{1}_{\{T_i^{(n)} > 0\}}(\hat{\xi}) \frac{\partial T_i^{(n)}(\hat{\xi}_l^K)}{\partial U_{jk}^{\mathrm{loc}}} m \left( T_i^{(n)}(\hat{\xi}_l^K) \right)^{m-1}.$$

For the derivative of $T_i^{(n)}$, we then have

$$\begin{aligned}\frac{\partial T_i^{(n)}(\hat{\xi}_l^K)}{\partial U_{jk}^{\mathrm{loc}}} &= \cos^2\left(\frac{i2\pi}{n}\right) \frac{\partial \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{11}}{\partial U_{jk}^{\mathrm{loc}}} \\ &\quad + 2\cos\left(\frac{i2\pi}{n}\right) \sin\left(\frac{i2\pi}{n}\right) \frac{\partial \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{12}}{\partial U_{jk}^{\mathrm{loc}}} \\ &\quad + \sin^2\left(\frac{i2\pi}{n}\right) \frac{\partial \sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{22}}{\partial U_{jk}^{\mathrm{loc}}}.\end{aligned}$$

### Derivative with Respect to $X^{\text{loc}}$

Let $K \in \mathcal{T}_h$. The derivative with respect to a local $X$ is given as

$$\begin{aligned}\frac{\partial J^{\text{loc}}(X,U(X))}{\partial X_{jk}^{loc}} &= \frac{\partial}{\partial X_{jk}^{loc}} \sum_{l=1}^{q^K} \omega_l \frac{2\pi}{n} \left( \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m + \sum_{i=1}^{n-1} \left( \left( T_i^{(n)}(\hat{\xi}_l^K) \right)^{+} \right)^m \right) \\ &= \sum_{l=1}^{q^K} \left[ \frac{\partial \omega_l}{\partial X_{jk}^{loc}} \cdot \frac{2\pi}{n} \left( \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m + \sum_{i=1}^{n-1} \left( \left( T_i^{(n)}(\hat{\xi}_l^K) \right)^{+} \right)^m \right) \right. \\ &\quad \left. + \omega_l \frac{2\pi}{n} \left( \frac{\partial}{\partial X_{jk}^{loc}} \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m + \sum_{i=1}^{n-1} \frac{\partial}{\partial X_{jk}^{loc}} \left( \left( T_i^{(n)}(\hat{\xi}_l^K) \right)^{+} \right)^m \right) \right].\end{aligned}$$

Next, the remaining derivatives are computed following our "bottom-up" approach. Starting with

$$\frac{\partial \omega_l}{\partial X_{jk}^{\text{loc}}} = \frac{\partial}{\partial X_{jk}^{\text{loc}}} \left( \hat{\omega}_l \det(\hat{\nabla} T_K(\hat{\xi}_l^K)) \right) = \hat{\omega}_l \frac{\partial}{\partial X_{jk}^{\text{loc}}} \left( \det(\hat{\nabla} T_K(\hat{\xi}_l^K)) \right).$$

The derivative of $\hat{\nabla} T_K(\hat{\xi})$ was already calculated, it is

$$\frac{\partial \hat{\nabla} T_K(\hat{\xi}_l^K)_{i\ell}}{\partial X_{jk}^{\text{loc}}} = \sum_{r=1}^{n_{sh}} \frac{\partial X_{ri}^{\text{loc}}}{\partial X_{jk}^{\text{loc}}} \frac{\partial \hat{\theta}_r(\hat{\xi}_l^K)}{\partial \hat{X}_\ell} = \sum_{r=1}^{n_{sh}} \delta_{rj}\delta_{ik} \frac{\partial \hat{\theta}_r(\hat{\xi}_l^K)}{\partial \hat{X}_\ell} = \delta_{ik} \frac{\partial \hat{\theta}_j(\hat{\xi}_l^K)}{\partial \hat{X}_\ell}.$$

Hence, utilizing the formula for derivatives of determinants $\frac{\partial \det(A)}{\partial x} = \det(A)\text{tr}\left(A^{-1}\frac{\partial A}{\partial x}\right)$ we obtain

$$\frac{\partial}{\partial X_{jk}^{\text{loc}}} \left( \det \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right) \right) = \det \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right) \text{tr} \left( \left( \hat{\nabla} T_K(\hat{\xi}_l^K) \right)^{-1} \frac{\partial \hat{\nabla} T_K(\hat{\xi}_l^K)}{\partial X_{jk}^{\text{loc}}} \right).$$

To calculate the derivative of $\sigma$

$$\frac{\partial}{X_{jk}^{\text{loc}}} \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}^{+} \right)^m = \mathbb{1}_{\{\sigma_{11}>0\}}(\hat{\xi}_l^K) \frac{\partial \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11}}{X_{jk}^{\text{loc}}} m \left( \sigma \left( T_K \left( \hat{\xi}_l^K \right) \right)_{11} \right)^{m-1},$$

the "bottom-up" approach is followed and we obtain, similar to the derivative with respect to $U$,

$$\frac{\partial (\sigma \circ T_K)_{i\ell}}{\partial X_{jk}^{\text{loc}}} = \mu_{\text{L}} \left( \frac{\partial \hat{\nabla} u_{i\ell}}{\partial X_{jk}^{\text{loc}}} + \frac{\partial \hat{\nabla} u_{\ell i}}{\partial X_{jk}^{\text{loc}}} \right) + \lambda_{\text{L}} \delta_{i\ell} \sum_{r=1}^{2} \frac{\partial \hat{\nabla} u_{rr}}{\partial X_{jk}^{\text{loc}}}.$$

For further details we refer to [79, 21, 80]. The remaining derivative of $\left(\left(T_i^{(n)}(\hat{\xi}_l^K)\right)^+\right)^m$ is

$$\frac{\partial}{\partial X_{jk}^{\mathrm{loc}}}\left(\left(T_i^{(n)}(\hat{\xi}_l^K)\right)^+\right)^m = \mathbb{1}_{\{T_i^{(n)}>0\}}(\hat{\xi})\frac{\partial T_i^{(n)}(\hat{\xi}_l^K)}{\partial X_{jk}^{\mathrm{loc}}} m\left(T_i^{(n)}(\hat{\xi}_l^K)\right)^{m-1}, \tag{4.5}$$

with

$$\begin{aligned}\frac{\partial T_i^{(n)}(\hat{\xi}_l^K)}{\partial X_{jk}^{\mathrm{loc}}} &= \cos^2\left(\frac{i2\pi}{n}\right)\frac{\partial\sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{11}}{\partial X_{jk}^{\mathrm{loc}}} \\ &\quad + 2\cos\left(\frac{i2\pi}{n}\right)\sin\left(\frac{i2\pi}{n}\right)\frac{\partial\sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{12}}{\partial X_{jk}^{\mathrm{loc}}} \\ &\quad + \sin^2\left(\frac{i2\pi}{n}\right)\frac{\partial\sigma\left(T_K\left(\hat{\xi}_l^K\right)\right)_{22}}{\partial X_{jk}^{\mathrm{loc}}}.\end{aligned} \tag{4.6}$$

In the following, the "bottom-up" approach is further explained. All components of the local derivatives are calculated in separate functions, that have no interdependence of functions of the same level. In Figure 4.1, the hierarchy of this principle is exemplary illustrated for the partial derivatives with respect to $U$. As one can observe there exists a strict order for the sub-function use for the derivative computation and each sub-function can be incorporated in various other applications.

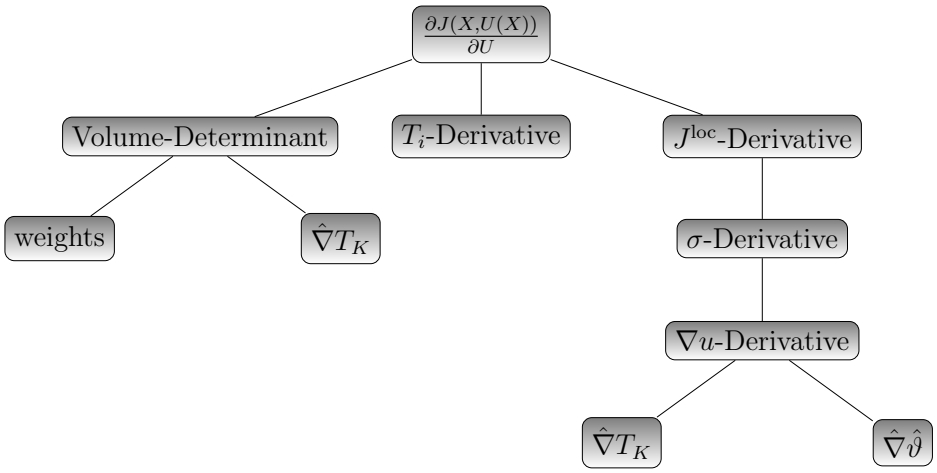



Figure 4.1: Scheme of dependence of the subfunctions in the "bottom-up" approach.

## 4.3 Geometry Definition and Finite Element Mesh

In this section, the definition of geometry and the finite element mesh representing the ceramic components is described. The two-dimensional shapes $\Omega \in \mathcal{O}^{\mathrm{ad}} \subset \mathcal{P}(\mathbb{R}^2)$ are discretized by an $n_x \times n_y$ mesh $X := X^\Omega = (X_{ij}^\Omega)_{n_x \times n_y}$ (we write $X_{ij} := X_{ij}^\Omega \in \mathbb{R}^2$ for short) using triangles, with $n_x, n_y \in \mathbb{N}$ being the number of grid points in x and y direction

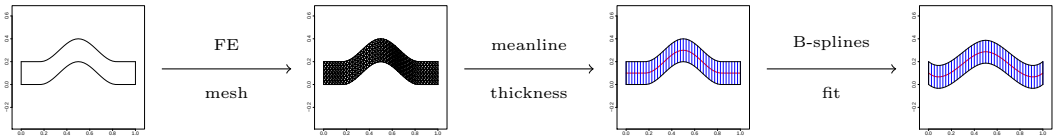



Figure 4.2: Transformation: mesh $X \to$ meanline/thickness $\varrho \to$ B-spline fit $\gamma$. See also [46].

of $\xi$, respectively. Given a shape $\Omega \in \mathcal{O}^{\mathrm{ad}}$ and its discretization $X$, the objective function values $J_0(X)$ and $J_1(X)$ as well as their gradients $\nabla J_0(X)$ and $\nabla J_1(X)$ are computed using the implementation of [79, 21, 80].

For the optimization process we fix the $\mathbb{x}$-component of all grid points to equidistant values $\xi_1^{\mathbb{x}}, \ldots, \xi_{n_x}^{\mathbb{x}}$, and we only consider the $\mathbb{y}$-components of those grid points that define the boundary of the shape to avoid deformation of the inner mesh structure. Note that this reformulation reduces the number of optimization variables from $2n_xn_y$ to $2n_x$. As a consequence, feasible shapes can alternatively be represented by a *shape parameter* $\varrho$ containing, for every relevant $\xi^{\mathbb{x}}$-coordinate, the $\xi^{\mathbb{y}}$-coordinate of the *meanline* $\varrho_i^{\mathrm{ml}} \in \mathbb{R}$ of the shape, and the *thickness* $\varrho_i^{\mathrm{th}} \in \mathbb{R}_>$ of the shape, $i = 1, \ldots, n_x$. Given a feasible shape represented by $\varrho := (\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}}) \in \mathbb{R}^{2n_x}$ with $\varrho^{\mathrm{th}} \in \mathbb{R}_>^{n_x}$, an associated mesh representation $X$ can be obtained using

$$X_{i,j} := \left( \xi_i^{\mathbb{x}} \ , \ \varrho_i^{\mathrm{ml}} + \frac{\varrho_i^{\mathrm{th}}}{n_y - 1} \left( j - \frac{n_y + 1}{2} \right) \right) \in \mathbb{R}^2, \qquad i = 1, \ldots, n_x, \ j = 1, \ldots, n_y. \tag{4.7}$$

To further reduce the computational burden and to obtain smoother shapes, the shape parameters $\varrho^{\mathrm{ml}} \in \mathbb{R}^{n_x}$ and $\varrho^{\mathrm{th}} \in \mathbb{R}_>^{n_x}$ are modelled using *B-splines*. Let $n_B \in \mathbb{N}$, with $n_B < n_x$, be the number of B-spline basis functions, and let $\{\mathcal{B}_j : \mathbb{R} \to \mathbb{R}_\geq, \ j = 1, \ldots, n_B\}$ be a B-spline basis (see, e.g., [120]). Feasible shapes are then represented by B-spline coefficients $\gamma := (\gamma^{\mathrm{ml}}, \gamma^{\mathrm{th}}) \in \mathbb{R}^{2n_B}$. The corresponding meanline and thickness values can be computed using the auxiliary functions

$$\hat{\varrho}^{\mathrm{ml}}(x) := \sum_{j=1}^{n_B} \gamma_j^{\mathrm{ml}} \, \mathcal{B}_j(x) \quad \text{and} \quad \hat{\varrho}^{\mathrm{th}}(x) := \sum_{j=1}^{n_B} \gamma_j^{\mathrm{th}} \, \mathcal{B}_j(x), \qquad x \in \mathbb{R}.$$

These auxiliary meanline and thickness functions are then evaluated at the fixed $x$-coordinates of the gridpoints which yields

$$\varrho_i^{\mathrm{ml}} := \hat{\varrho}^{\mathrm{ml}}(x_i) \quad \text{and} \quad \varrho_i^{\mathrm{th}} := \hat{\varrho}^{\mathrm{th}}(x_i), \qquad i = 1, \ldots, n_x. \tag{4.8}$$

Using the B-spline coefficients $\gamma = (\gamma^{\mathrm{ml}}, \gamma^{\mathrm{th}}) \in \mathbb{R}^{2n_B}$ as optimization variables yields a further reduction of the number of variables to $2n_B$. Moreover, the B-spline representation leads to an implicit regularization and smoothing of the represented shapes. In the following, we denote the set of *feasible shape parametrizations* by $\Gamma \subseteq \{(\gamma^{\mathrm{ml}}, \gamma^{\mathrm{th}}) \in \mathbb{R}^{2n_B}\}$. The transformation from the mesh to the B-spline fit is visualized in Figure 4.2.

To evaluate the objective functions $J_j(\gamma)$ and their gradients $\nabla J_j(\gamma) = \partial J_j / \partial \gamma$, $j = 0, 1$, w.r.t. the new parametrization of shapes based on B-spline parameters $\gamma$, while

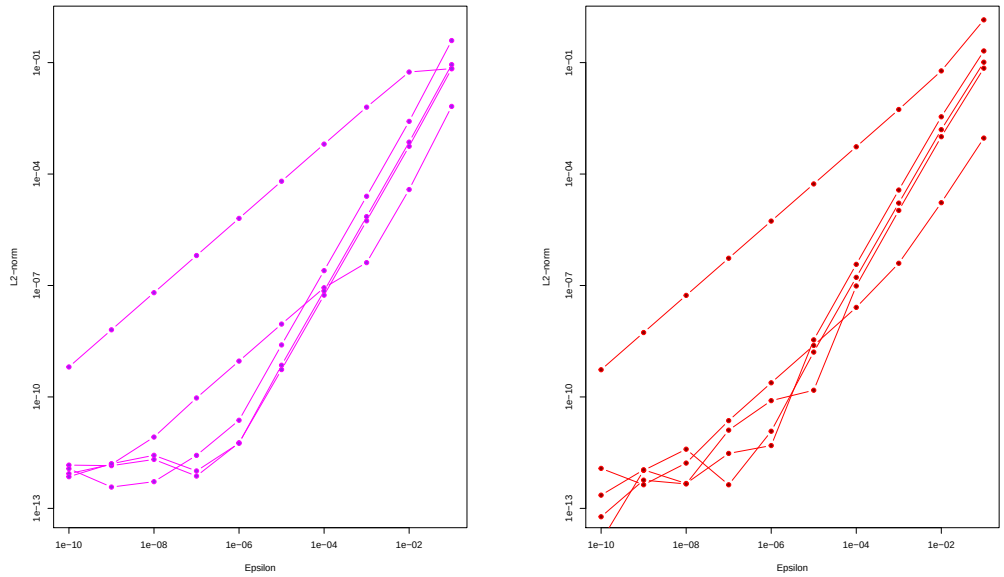
still using the implementation of [79, 21, 80], we compute an associated grid $X$ using a two step transformation. First, the fixed B-spline basis is utilized to construct the auxiliary functions of meanline and thickness, the evaluation of which (via (4.8)) then generates the shape parameters for the next step of the grid computation (4.7). While the resulting objective function values can be used immediately in the optimization process, the gradients computed w.r.t. the grid $X$ need to be translated to the space of B-spline coefficients, i.e.,

$$\frac{\partial J_j}{\partial \gamma^{\mathrm{ml}}} = \frac{\partial J_j}{\partial X}\,\frac{\partial X}{\partial \varrho^{\mathrm{ml}}}\,\frac{\partial \varrho^{\mathrm{ml}}}{\partial \gamma^{\mathrm{ml}}} \quad \text{and} \quad \frac{\partial J_j}{\partial \gamma^{\mathrm{th}}} = \frac{\partial J_j}{\partial X}\,\frac{\partial X}{\partial \varrho^{\mathrm{th}}}\,\frac{\partial \varrho^{\mathrm{th}}}{\partial \gamma^{\mathrm{th}}}, \qquad j = 0, 1. \tag{4.9}$$

The numerical computation of gradients of $J_j$, $j = 0, 1$, w.r.t. a B-spline representation $\gamma$ of a feasible shape $\Omega$ is thus based on a two-step projection of $\gamma$ onto the original grid $X$. The thus computed gradients of $J_1$ (the intensity measure) were validated, using finite differences, at the sample shape shown in Figure 4.4a. The validation is based on a grid $(X_{ij})_{41\times 7}$, i.e., $n_x = 41$ and $n_y = 7$. Consequently, for the corresponding meanline and thickness representation we have $\varrho = (\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}}) \in \mathbb{R}^{82}$, where $\varrho^{\mathrm{th}} \in \mathbb{R}^{41}_{>}$. Moreover, we used a B-spline basis with five basis functions, i.e., $n_B = 5$ and $\gamma = (\gamma^{\mathrm{ml}}, \gamma^{\mathrm{th}}) \in \mathbb{R}^{10}$. We computed all ten partial derivatives w.r.t. $\gamma$ via the respective transformations to the grid representation and compared them with finite differences. The results of this comparison, i.e., the absolute values of the differences between computed derivatives and finite differences, are shown in Figure 4.3a and 4.3b for the meanline and thickness parameters, respectively. The figures indicate in all cases that, when the finite differences are evaluated for decreasing values of the increment $\varepsilon$, then they correspond well to the computed gradients.

## 4.4 Test Cases

In this section, two test cases are introduced, which then are used in the numerical experiments of Chapters 5, 6 and 7. We consider 2D ceramic shapes made out of beryllium oxide (BeO) under tensile load. Therefore, we set Young's modulus to $\mathtt{E}_Y = 320\,\mathrm{GPa}$ (see, e.g., [112]), Poisson's ratio to $\nu_P = 0.25$, and the ultimate tensile strength to 140 MPa, according to [142]. Weibull's modulus is set to $m = 5$, which is on the lower bound for industrial ceramics where $m$ is between 5 and 30 depending on the production process [110]. All considered shapes have a fixed length of 1.0 m and a fixed height of 0.2 m on the left and right boundaries. The shapes are fixed on the left boundary, where Dirichlet boundary conditions hold ($\partial\Omega_D$), and on the right boundary, where surface forces may act on and Neumann boundary conditions hold ($\partial\Omega_{N_{\mathrm{fixed}}}$). The upper and lower boundaries are assumed to be force free ($\partial\Omega_{N_{\mathrm{free}}}$). They can be modified within the optimization process. We set $\tilde{f} = 0$ neglecting the gravity forces and $\tilde{g} = 10^7\,\mathrm{Pa}$, representing tensile load. Note that, in order to be consistent with 3D models, we define the force density w.r.t. $\mathrm{Pa} = \mathrm{N/m^2}$ (and not w.r.t. N/m). This is motivated by assuming a constant width of the 2D component of 1 unit (i.e., 1m). Then plane stresses and plane strains are assumed by neglecting Poisson effects in the third dimension. For the biobjective gradient descents performed in Chapter 5, both starting solution are chosen without involving a decision maker. In the first case, we choose an obviously not efficient solution for a horizontal




(a) Validation of $\partial J_1/\partial \gamma_i^{\mathrm{ml}}$

(b) Validation of $\partial J_1/\partial \gamma_i^{\mathrm{th}}$

Figure 4.3: Validation of gradients computed according to (4.9) using finite differences. On the $x$-axis: increment $\varepsilon$ used for the finite difference evaluation; on the $y$-axis: absolute deviation between $\partial J_1/\partial \gamma_i^{\mathrm{ml,th}}$ computed according to (4.9) and the corresponding finite difference, $i = 1, \ldots, 5$, for meanline (left) and thickness (right). See also [46].

load transfer to see how the biobjective gradient descent algorithms works. The second test case simulates a shifted load transfer. We thus take an *a posteriori* approach on decision making with regard to design or cost preferences. We are aware that using only one starting design for each test case may bias the solutions of both optimization methods. Nevertheless, numerical experiments with moderately modified initial designs yielded comparable solutions, indicating that in these special cases there is not much to gain by varying the starting designs. In Chapter 6, a continuation method is applied on Pareto critical solutions of both test cases that are computed in Chapter 5. Furthermore, in Chapter 7 surrogate based optimization is applied and to that end explicit design spaces for both test cases are chosen. The shapes are discretized by a $41 \times 7$ grid (i.e., $n_x = 41$ and $n_y = 7$) using triangles as detailed in Section 4.3. The B-spline representation is based on $n_B = 5$ basis functions. Thus, we have in total ten B-spline coefficients. Since the left and right boundary are fixed and we only modify the upper and lower boundary of the components, we have to fix the first and last B-spline coefficients for both, the auxiliary meanline and thickness functions. All in all, we have now six control variables. For the rest of this work we denote the six control variables $\gamma_2^{\mathrm{ml}}, \gamma_3^{\mathrm{ml}}, \gamma_4^{\mathrm{ml}}, \gamma_2^{\mathrm{th}}, \gamma_3^{\mathrm{th}}$ and $\gamma_4^{\mathrm{th}}$ ( recall that $\gamma_1^{\mathrm{ml}}, \gamma_5^{\mathrm{ml}}$ and $\gamma_1^{\mathrm{th}}, \gamma_5^{\mathrm{th}}$ are fixed) as

$$x = (x_1, \ldots, x_6) = (\gamma_2^{\mathrm{ml}}, \gamma_3^{\mathrm{ml}}, \gamma_4^{\mathrm{ml}}, \gamma_2^{\mathrm{th}}, \gamma_3^{\mathrm{th}}, \gamma_4^{\mathrm{th}}) \in \mathcal{X} \subset \mathbb{R}^6. \tag{4.10}$$

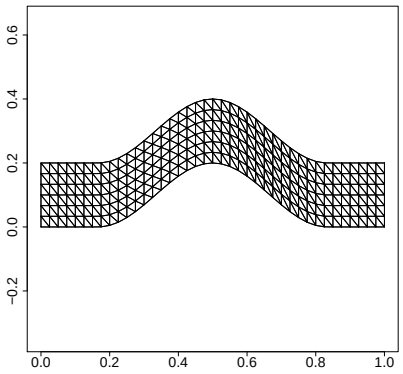
Most implementations are realized in R version 3.5.0, where the B-spline implementation of [124] is utilized. Furthermore, we use the adjoint finite element code of [79, 21, 80] as a subroutine. Only in Chapter 7 additional external code from an optimization toolbox, i.e., *Dakota*, is utilized. The details are explained in Chapter 7.

### 4.4.1 Test Case 1: A Straight Joint

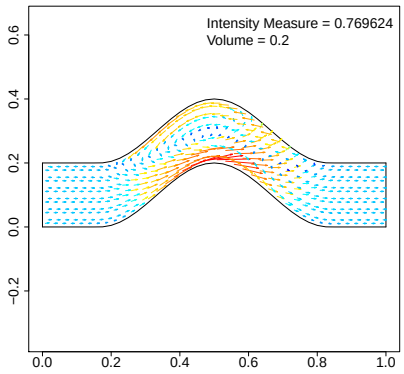
In the first test case, a straight joint is sought that is fixed at the left side, while the tensile load acts on the right side. This is a particularly simple situation where the straight rod connecting from the left to the right (Figure 4.4d) can be expected to be optimal, with varying thickness depending on the trade-off between the intensity measure ($J_1$) and the volume ($J_0$). The biobjective gradient descent algorithms of Chapter 5 are challenged by providing a bended beam as a starting shape, which is clearly far from being optimal.
The starting shape, is shown in Figure 4.4a, together with the $41 \times 7$ tetrahedral discretization $X$. Its objective values are $J_1(X^{(1)}) = 0.769624$ (intensity measure) and $J_0(X^{(1)}) = 0.2$ (volume), respectively. The relatively high value for the intensity measure $J_1$ can be explained by the relatively high stresses that are illustrated in Figure 4.4b. Figure 4.4c shows that the B-spline representation based on only five basis functions leads to a rather inaccurate representation, particularly at the left and right boundary. This could be improved by fixing the slopes at the left and right boundary, however, at the price of a significantly reduced design space. Indeed, a majority of the Pareto critical shapes computed during our numerical tests do not have zero slopes at the left and right boundary, particularly in the case of the S-shaped joint considered in Section 4.4.2 below. Note that the smoothing induced by the B-spline representation in this case already leads to dominating objective values of $J_1(\gamma^{(1)}) = 0.453867$ and $J_0(\gamma^{(1)}) = 0.2$.

### 4.4.2 Test Case 2: An s-Shaped Joint

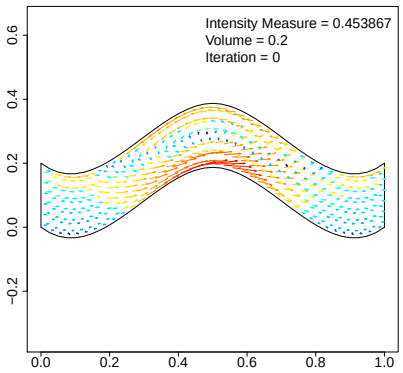
A more complex situation is obtained when the left and right boundaries are not fixed at the same height, i.e., when an S-shaped joint is to be designed. In our tests, we fix the right boundary about 0.27 m lower than the left boundary. The starting shape for the biobjective gradient descent algorithms of Chapter 5 and its $41 \times 7$ tetrahedral discretization $X$, that is used for all optimization runs, is shown in Figure 4.5a. Figure 4.5b highlights the stresses that are particularly strong toward the left boundary. The respective objective values are $J_1(X^{(1)}) = 1.520058$ (intensity measure) and $J_0(X^{(1)}) = 0.2$ (volume), respectively. As can be expected, the intensity measure (and hence also the PoF) is considerably higher than in the case of the straight joint discussed in Section 4.4.1. Despite the significant smoothing induced by the B-spline representation of the initial shape shown in Figure 4.5c, it has an even higher value of the intensity measure of $J_1(\gamma^{(1)}) = 1.910532$ (and hence a higher PoF value), while $J_0(\gamma^{(1)}) = 0.2$ remains constant.



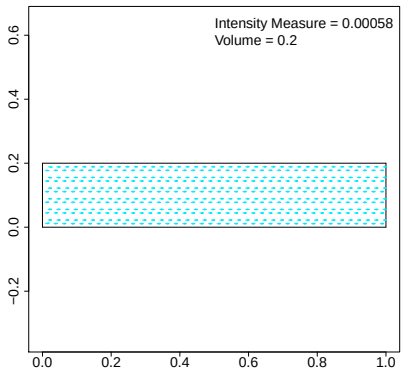

(a) Tetrahedral mesh $X$




(b) Objective values and stresses

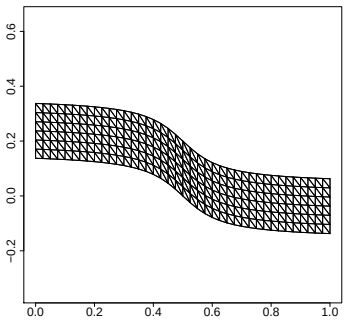



(c) Approximation with B-splines

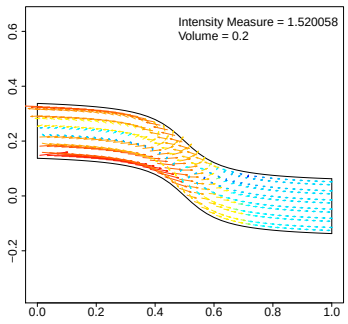



(d) Expected result: Straight rod

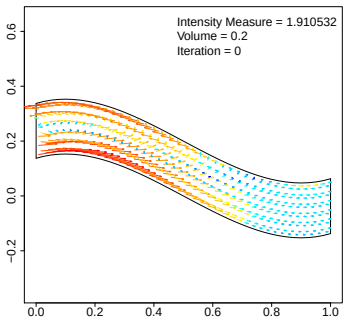
Figure 4.4: Test case 1 - straight joint: starting solution and expected solution [46].




(a) Tetrahedral mesh $X$




(b) Objective values and stresses




(c) Approximation with B-splines

Figure 4.5: Test case 2 - s-shaped joint: starting solution [46].

# 5 Gradient Based Biobjective Shape Optimization to Improve Reliability and Cost of Ceramic Components

In this chapter, the first gradient based optimization methods of this work are applied to the shape optimization problem (3.11), i.e., the test cases described in Section 4.4. We consider a biobjective steepest descent algorithm and a steepest descent approach with the weighted sum scalarization for varying weights $\lambda \in (0, 1)$. Most of this chapter was already published in [46].

The optimization of the design of mechanical structures is a central task in mechanical engineering. If the material for a component is chosen and the use cases are defined, implying in particular the mechanical loads, then the central task of engineering design is to define the shape of the component. Among all possible choices, those shapes are preferred that guarantee the desired functionality at minimal cost. The functionality, however, is only guaranteed if the mechanical integrity of the component is preserved. The fundamental design requirements of functional integrity and cost are almost always in conflict, which makes mechanical engineering an optimization problem with at least two objective functions to consider.

Mathematically, the task of choosing the shape of a structure is formulated by the theory of shape optimization, see, e.g., [5, 25, 83, 145] for an introduction. We thus consider admissible shapes $\Omega \subseteq \mathbb{R}^d$, $d = 2, 3$, along with an objective function $J(\Omega)$ which returns lower values for better designs. The task then is to find an admissible shape $\Omega^* \in \arg\min J(\Omega)$. The existence of optimal shapes has been studied in [28, 66, 83] – for the specific objective function $J$ of compliance see [5]. On the algorithmic side, the adjoint approach to shape calculus has led to efficient strategies to calculate shape gradients, see, e.g., [32, 41, 54, 55, 100, 138, 139, 145]. While theory and numerical algorithms of shape calculus are highly developed mathematically, most publications in the field neither deal with multiobjective optimization problems, nor directly consider mechanical integrity as one of the objective functions, see [83, 6, 50, 119] for some remarkable exceptions.

In mechanical engineering, mechanical integrity is one of the central objectives, see, e.g., [13]. However, if objectives like the ultimate load that the structure can bear or the fatigue life of a component are formulated deterministically, then the objective function is in general non-differentiable as it only depends on the point of maximal stress. Thus, this would lead to highly instable numerical optimization schemes. At the same time, as material properties are subject to considerable scatter, a deterministic approach is not realistic. To overcome these two shortcomings, an alternative probabilistic approach to mechanical integrity has been proposed in [21, 20, 74, 73, 75, 134, 135, 136] (see also Section 3.2), which has a smoothing effect on the singularities that are typical for deterministic models. Note that the probabilistic description of the ultimate strength of ceramics has become

a standard in material engineering since the ground breaking work of Weibull, see, e.g., [13, 24, 112, 127, 152]. In practice, there usually is a trade-off between the mechanical integrity of a structure and its volume (cost), since an improved mechanical integrity usually comes at the cost of a larger volume. Instead of presetting a fixed bound on the allowable volume, the trade-off between these two conflicting goals can be analyzed in a biobjective model. Other relevant objective functions may be, for example, the minimal buckling load of a structure or its minimal natural frequency, see, for example, [83]. In Chapter 3, a general introduction into the field of biobjective optimization is given. In the context of shape optimization problems, two major solution approaches can be distinguished: Metaheuristic and, in particular, evolutionary algorithms are widely applicable solution paradigms that do not utilize the particular structure of a given problem [29, 39, 40, 154]. However, in combination with expensive numerical simulations such approaches tend to be inefficient. On the other hand, gradient-based algorithms [43, 60, 156] require efficient gradient computations and are often applied in the context of adjoint approaches and using weighted sum scalarizations of the objective functions. See [122] for a comparison. In this work, we consider biobjective PDE constrained shape optimization problem (3.11) for the simultaneous optimization of the mechanical integrity and the cost of a ceramic component. As we have seen, most papers in shape optimization of structures do not deal with mechanical integrity as an objective, but use elastic compliance [5], or in the case that they consider mechanical integrity, do not use a probabilistic approach [6, 119, 50]. In [75, 134, 136, 135], probabilistic models for metals under low cycle fatigue are introduced and in [73, 74] some numerical studies are made. Further, a probabilistic formulation for ceramics under load is provided in [20] and a first single criteria optimization with this objective while using a volume constraint is done in [21]. Multicriteria optimization including mechanical integrity is widely considered, see, e.g., [29] for a recent example. However, these works neither consider probabilistic effects nor use gradients. This work for the first time combines biobjective gradient based optimization methods with a probabilistic assessment of mechanical integrity.
This chapter is structured as follows. In Section 5.1, a biobjective descent algorithm and the gradient descent with the weighted sum scalarization are introduced. In Section 5.2, the numerical implementation is described and subsequently the numerical results are presented in Section 5.3.

## 5.1 Biobjective Gradient Descent Methods

Recall that we aim at the efficient computation of Pareto critical shapes while incorporating the available gradient information. Two fundamental approaches in this category, a parametrized weighted sum method and a biobjective descent algorithm are chosen and explained in Sections 5.1.1 and 5.1.2, respectively. Their performance in the context of 2D shape optimization problems is compared in Section 5.3.

### 5.1.1 Weighted Sum Method

Maybe the easiest way to compute a representation of the Pareto front is to iteratively solve weighted sum scalarizations (3.26) with varying weights. Recall that the weighted

sum scalarization of problem (3.11) can be restated as

$$\min_{x\in\mathcal{X}} \; J_\lambda(x) := (1-\lambda)J_0(x) + \lambda J_1(x), \tag{5.1}$$

where $\lambda \in (0,1)$ is the *weight* specifying the relative importance of $J_1$ and $J_0$, respectively. For this purpose, problem (3.14) is solved iteratively for varying weights (in our case, we choose $\lambda \in \{0.2, 0.25, 0.3, \ldots, 0.9\}$ since numerical experiments showed that this yields meaningful trade-offs). Each single objective optimization problem (3.14) is then individually solved using a classical gradient descent algorithm with stepsizes determined according to the Armijo rule, see, for example, [14].

**Algorithm 1:** Parametric weighted sum algorithm using gradient descent

**Data:** Choose $\beta \in (0,1)$, $x^{(1)} \in \mathcal{X}$, weights $\lambda_1, \ldots, \lambda_J \in (0,1)$, and $\varepsilon > 0$.
**Result:** Set of approximations of Pareto critical solutions $\tilde{x}_1, \ldots, \tilde{x}_J$.
**for** $j = 1$ **to** $J$ **do**
  Set $\lambda = \lambda_j$, set $k := 1$, and set $d^{(0)} := -\nabla_x J_\lambda(x^{(1)})$ and $h_0 := 1$;
  **while** $\|h_{k-1}\, d^{(k-1)}\| > \varepsilon$ **do**
    Compute a search direction $d^{(k)} = -\nabla_x J_\lambda(x^{(k)})$ ;
    Compute a step length $h_k \in (0,1]$ as

$$\max\Big\{h = \frac{1}{2^\ell} \;:\; \ell \in \mathbb{N}_0,\; J_\lambda(x^{(k)} + h d^{(k)}) \leq J_\lambda(x^{(k)}) + \beta\, h \nabla_x J_\lambda(x^{(k)})^\top d^{(k)}\Big\};$$

    $x^{(k+1)} := x^{(k)} + h_k\, d^{(k)}$ and $k := k+1$;
  **end**
  $\tilde{x}_j := x^{(k)}$
**end**

Under appropriate assumptions, the gradient descent algorithm in the inner loop of Algorithm 1 converges to a critical point of (3.14), see, e.g., [14].

In our implementation, the inner loop is also terminated when a prespecified maximum number of iterations is reached. However, in this case there is no guarantee that the final iterate is close to a Pareto critical solution.

Note that a critical point of the weighted sum scalarization (3.14) is necessarily Pareto critical for the biobjective shape optimization problem (3.11), while the converse is not true in general. This has some correspondence to the fact that global optimal solutions of a weighted sum scalarization (3.14) are always Pareto optimal, while nonconvex problems may have Pareto optimal solutions that are not optimal for any weighted sum scalarization (3.14), see, Chapter 3.

Note also that the search direction $d^{(k)} = -\nabla_x J_\lambda(x^{(k)})$ does not necessarily satisfy $\nabla_x J_j(x^{(k)})^\top d^{(k)} < 0$, $j = 0, 1$, in all iterations. In other words, one objective function may deteriorate during the optimization process if only the other objective function compensates for this.

### 5.1.2 Biobjective Descent Algorithm

Different from the weighted sum method described above and in Chapter 3, biobjective descent algorithms – as a natural generalization of single-objective gradient descent algorithms – are potentially capable of finding every Pareto optimal solution, if only the starting solution is chosen appropriately. While this is a rather theoretical advantage, biobjective descent algorithms are indeed highly efficient in finding (or approximating) one Pareto critical solution without the necessity to specify preferences. However, if a representation of the complete Pareto front is sought, they need to be combined with other search strategies. We adopt the multiobjective descent algorithm proposed in [60] (see also [61]) for the biobjective optimization problem (3.11). Similar approaches have been suggested in [43, 44, 70]. This section is mostly based on [60].

Recall, that we call $\hat{x} \in \mathcal{X}$ Pareto critical if and only if

$$\left\{ d \in \mathbb{R}^n \mid \nabla_x J_i(\hat{x})^\top d < 0,\, i = 0, 1 \right\} = \emptyset.$$

Biobjective descent algorithms iteratively improve both objective functions simultaneously. This is based on the observation that, if a solution $x \in \mathcal{X}$ is not Pareto critical according to (3.12), then there exists a direction $d \in \mathbb{R}^n$ which is a descent direction for both objectives. Now, suppose that $x \in \mathcal{X}$ is not Pareto critical. Then, there exists a biobjective descent direction $d \in \mathbb{R}^n$ with $\nabla_x J_i(x)^\top d < 0, \;\; i = 0, 1$. Further, let

$$J^x(d) := \max(\nabla_x J_0(x)^\top d, \nabla_x J_1(x)^\top d). \tag{5.2}$$

$J^x$ is convex and positive homogeneous in $d$ [60].

Then according to [60] a *direction of steepest biobjective descent* $d \in \mathbb{R}^n$ can be defined as a direction solving the auxiliary optimization problem

$$\begin{aligned} \min \quad & J^x(d) + \frac{1}{2}\|d\|^2 \\ \text{s.t.} \quad & d \in \mathbb{R}^n. \end{aligned} \tag{5.3}$$

Which can be reformulated to get rid of the non differentiabilities

$$\begin{aligned} \min_{\alpha \in \mathbb{R}, d \in \mathbb{R}^n} \quad & \alpha + \frac{1}{2}\|d\|^2 \\ \text{s.t.} \quad & \nabla_x J_j(x^{(k)})^\top d \leq \alpha, \;\; j = 0, 1. \end{aligned} \tag{5.4}$$

Problem (5.4) is a convex quadratic optimization problem with linear inequality constraints. Note that the term $\frac{1}{2}\|d\|^2$ in the objective function ensures that the problem is bounded, and that the solution $\alpha = 0$, $d = 0$ is always feasible. Note also that the optimal value $\alpha^*$ is negative if and only if $d^* \neq 0$, i.e., if a direction of steepest biobjective descent exists. The following definition and lemma are needed for a proof of another important lemma.

**Definition 5.1.** *Let $(\mathcal{X}, \|\cdot\|)$ be a real normed space, $C \subset \mathcal{X} \subset \mathbb{R}^n$ a convex subset and $c > 0$ a positive constant. A function $h : C \to \mathbb{R}$ is called strongly convex with modulus $c$*

*if, and only if for all* $x, y \in C$ *and* $a \in (0,1)$

$$h(ax + (1-a)y) \leq ah(x) + (1-a)h(y) - ca(1-a)\|x-y\|^2.$$

**Lemma 5.2.** *Let* $(\mathcal{X}, \|\cdot\|)$ *be a real normed space,* $C \subset \mathcal{X} \subset \mathbb{R}^n$ *a convex subset and* $c > 0$ *a positive constant. Further, let* $h : C \to \mathbb{R}$*. Then, the following two properties are equivalent*

*(i)* $h$ *is strongly convex with modulus* $c$*,*

*(ii)* $h - c\|\cdot\|^2$ *is convex.*

For a proof of this lemma we refer to, e.g. [113]. We can now prove the following lemma from [60].

**Lemma 5.3.** *Let* $x \in \mathcal{X}$*. Further, let* $d(x) := \arg\min(J^x(d) + \frac{1}{2}\|d\|^2)$ *be the solution and* $\alpha(x) := \min_d(J^x(d) + \frac{1}{2}\|d\|^2)$ *the optimal value of* (5.3)*. Then*

1. *If* $x$ *is Pareto critical, then* $d(x) = 0 \in \mathbb{R}^n$ *and* $\alpha(x) = 0$*.*
2. *If* $x$ *is not Pareto critical, then* $\alpha(x) < 0$ *and*

$$\begin{aligned} &J^x(d(x)) < -\frac{1}{2}\|d(x)\|^2 < 0, \\ &\nabla_x J_i(x)^\top d(x) \leq J^x(d(x)),\ i = 0, 1. \end{aligned}$$

3. $x \mapsto d(x)$ *and* $x \mapsto \alpha(x)$ *are continuous.*

*Proof.* 1. If $x$ is Pareto critical, then $\{d \in \mathbb{R}^n \,|\, \nabla_x J_i(\hat{x})^\top d < 0,\ i = 0,1\} = \emptyset$ and therefore $J^x(d) \geq 0\ \forall d \in \mathbb{R}^n$. Since $J^x(0) = 0$, the conclusion follows.

2. If $x$ is not Pareto critical, then there exists a $d \in \mathbb{R}^n$ such that $J^x(d) < 0$. With

$$\hat{t} := -\frac{J^x(d)}{\|d\|^2} > 0, \quad \hat{d} := \hat{t}d,$$

and since $J^x$ is positive homogeneous we have

$$\begin{aligned} J^x(\hat{d}) - \frac{1}{2}\|\hat{d}\|^2 &= \hat{t}J^x(d) + \frac{1}{2}\hat{t}^2\|d\|^2 \\ &= -\frac{1}{2}\frac{J^x(d)^2}{\|d\|^2} < 0. \end{aligned}$$

Thus, $\alpha(x) < 0$ and both inequalities follow directly.

3. Let $x_0 \in \mathcal{X}$ and $\varepsilon > 0$. Set

$$L_\varepsilon := \{d \in \mathbb{R}^n \mid \|d(x_0) - d\| = \varepsilon\}.$$

Let $d(x_0)$ be the optimal solution of (5.3) for $x = x_0$. Recall that $J^x$ is convex. Thus, the objective function of (5.3) is convex, and as a conclusion of Lemma 5.2 also strongly convex with modulus $\frac{1}{2}$. Hence, we have

$$J^{x_0}(d) + \frac{1}{2}\|d\|^2 \geq J^{x_0}(d(x_0)) + \frac{1}{2}\|d(x_0)\|^2 + \frac{1}{2}\varepsilon^2 \ \forall d \in L_\varepsilon.$$

Since the mapping $(x, d) \mapsto J^x(d)$ is continuous and the set $L_\varepsilon$ is compact, it follows with the inequality from above that there exists a $\delta > 0$ such that, if $\|x - x_0\| \leq \delta$, then

$$J^x(d) + \frac{1}{2}\|d\|^2 > J^x(d(x_0)) + \frac{1}{2}\|d(x_0)\|^2 \ \forall d \in L_\varepsilon.$$

Take now $x \in \mathcal{X}$ such that $\|x - x_0\| \leq \delta$. Since $d \mapsto J^x(d) + \frac{1}{2}\|d\|^2$ is convex, one can conclude from the above inequality that $d(x)$, the minimal value of the objective function $J^x(\cdot) + \frac{1}{2}\|\cdot\|^2$ is not in the region $\|d(x_0) - d\| \geq \varepsilon$, hence $\|d(x_0) - d(x)\| < \varepsilon$. Continuity of $\alpha(x)$ follows directly.

□

See also [48]. Following [60], we say for points $x$ that are $\varepsilon$-Pareto critical, with $\varepsilon > 0$, that $d = d(x)$, as the corresponding solution of (5.3) w.r.t. $x$, is an *approximate solution* of (5.3). Then we can state the following lemma from [60].

**Lemma 5.4.** *Suppose that for some $\varepsilon > 0$ $x$ is $\varepsilon$-Pareto critical and that $d$ is a corresponding approximate solution of* (5.3)*. Then*

$$\|d\| \leq 2\|\nabla_x J(x)\|_{\infty,2}.$$

*Here, $\|A\|_{\infty,2}$ with $A \in \mathbb{R}^{m\times n}$ is given as*

$$\|A\|_{\infty,2} := \max_{i=1,\ldots,m} \left( \sum_{j=1}^{n} A_{i,j}^2 \right)^{\frac{1}{2}}.$$

Note that following [60], $\|\cdot\|_{\infty,2}$ is a norm on $\mathbb{R}^{m\times n}$. When a direction of steepest biobjective descent $d \neq 0$ is found, then we move from $x$ into the direction $d$ to a new point $x := x + hd$. The step length $h > 0$ is computed using an Armijo-like rule. Toward this end, let $\beta \in (0, 1)$ be a prespecified constant. Then, a step length $h$ is accepted if it guarantees a sufficient biobjective descent in the sense that

$$J_j(x + h\,d) \leq J_j(x) + \beta\, h\, \nabla_x J_j(x)^\top\, d,\ j = 0, 1. \tag{5.5}$$

In order to compute an acceptable step length $h$, we iteratively test the values $(\frac{1}{2})^\ell$, $\ell = 0, 1, 2, \ldots$ until condition (5.5) is satisfied. The finiteness of this procedure is given in the following lemma .

**Lemma 5.5.** *Let $J$ be differentiable and let $d \in \mathbb{R}^n$ with $\nabla_x J(x)^\top d < 0$. Then, there*

*exists* $\varepsilon := \varepsilon(x, d, \beta) > 0$ *such that*

$$J(x + hd) < J(x) + \beta h \nabla_x J(x)^\top d \ \forall h \in ]0, \varepsilon].$$

*Proof.* Since $J$ is differentiable we have for $h \in \mathbb{R} \setminus \{0\}$

$$J(x + h) = J(x) + \nabla_x J(x)^\top h + R(h)$$

with

$$\lim_{h \to 0} \frac{|R_i(h)|}{\|h\|} = 0, \ i = 0, 1.$$

Set $a := \max_i(\nabla_x J_i(x)^\top d)$. It follows directly that $a < 0$ and $d \neq 0$. Since $\beta < 1$, there is some $\varepsilon > 0$ such that

$$\begin{aligned} 0 < h \leq \varepsilon \Rightarrow \frac{|R_i(hd)|}{\|hd\|} &< \frac{(1 - \beta)|a|}{\|d\|}, \ i = 0, 1 \\ \Rightarrow |R_i(hd)| &< t(1 - \beta)|a|, \ i = 0, 1. \end{aligned}$$

Since $|a| = -a = \min_i(-\nabla_x J_i(x)^\top d)$ it follows that

$$R(hd) < -h(1 - \beta)\nabla_x J(x) v$$

Thus, we have for $0 < h \leq \varepsilon$

$$\begin{aligned} J(x + hd) &= J(x) + h\nabla_x J(x)^\top d + R(hd) \\ &< J(x) + h\nabla_x J(x)^\top d - h(1 - \beta)\nabla_x J(x)^\top d \\ &= J(x) + h\beta\nabla_x J(x)^\top d. \end{aligned}$$

□

The overall method is summarized in Algorithm 2. Let for iteration $k$, $x^{(k)} \in \mathcal{X}$ be the iterate, $d^{(k)} \in \mathbb{R}^n$ be the search direction, and $h_k \in (0, 1]$ the step length.
Next, following [60] we show that if $J_1$ and $J_0$ are continuously differentiable and $\varepsilon = 0$ then Algorithm 2 converges to a Pareto critical solution.

**Theorem 5.6.** *Let $(x^{(k)})_k$ be an infinite sequence generated by Algorithm 2. Then, every accumulation point of the sequence $(x^{(k)})_k$ is a Pareto critical point. Further, if the objective function $J$ has bounded level sets, i.e., the set $\{x \in \mathcal{X} \mid J(x) \leq J(x^{(1)})\}$ is bounded, then the sequence $(x^{(k)})_k$ stays bounded and has at least one accumulation point.*

*Proof.* Let $y$ be an accumulation point of the sequence $(x^{(k)})_k$. further, let $d(y)$ be the solution and $\alpha(y)$ be the optimum value of 5.3 at $y$, i.e.

$$d(y) := \arg\min \left( (J^y(d) + \frac{1}{2}\|d\|^2 \right), \quad \alpha(y) := \min_d \left( (J^y(d) + \frac{1}{2}\|d\|^2 \right),$$

**Algorithm 2:** Biobjective descent algorithm according to [60]

**Data:** Choose $\beta \in (0,1)$, $x^{(1)} \in \mathcal{X}$ and $\varepsilon > 0$, set $k := 1$.
**Result:** Approximation of a Pareto critical solution $\tilde{x} := x^{(k)}$.
Compute $d^{(0)} := d^{(1)}$ as a solution of (5.4) and set $h_0 := 1$;
**while** $\|h_{k-1}\, d^{(k-1)}\| > \varepsilon$ **do**
  Compute $d^{(k)}$ as a solution of (5.4);
  Compute a step length $h_k \in (0,1]$ as

$$\max\Big\{h = \frac{1}{2^\ell} \;:\; \ell \in \mathbb{N}_0,\; J_j(x^{(k)} + h d^{(k)}) \leq J_j(x^{(k)}) + \beta h \nabla_x J_j(x^{(k)})^\top d^{(k)},\; j = 0, 1\Big\};$$

  $x^{(k+1)} := x^{(k)} + h_k\, d^{(k)}$ and $k := k + 1$;
**end**

where $J^y(d) := \max_i(\nabla_x J_i(y) d)$. Following Lemma 5.3, it is sufficient to show that $\alpha(y) = 0$.

From Algorithm 2 we can directly conclude that the sequence $(J(x^{(k)}))_k$ is componentwise strictly decreasing and

$$\lim_{k\to\infty} J(x^{(k)}) = J(y).$$

Thus,

$$\lim_{k\to\infty} \|J(x^{(k)}) - J(x^{(k+1)})\| = 0.$$

But

$$J(x^{(k)}) - J(x^{(k+1)}) \geq -h_k \beta \nabla_x J(x^{(k)}) d^{(k)} \geq 0,$$

and therefore

$$\lim_{k\to\infty} -h_k \beta \nabla_x J(x^{(k)}) d^{(k)} = 0. \tag{5.6}$$

We observe that $h_k \in (0,1]$ for all $k$. Select a subsequence $(x^{(k_u)})_u$ of $(x^{(k)})_k$ converging to $y$. There are two cases to cosnider

$$\limsup_{u\to\infty} h_{k_u} > 0 \quad \text{and} \quad \limsup_{u\to\infty} h_{k_u} = 0.$$

**First case** ($\limsup_{u\to\infty} h_{k_u} > 0$)**:** Here, a subsequence $(x^{(k_l)})_l$ of $(x^{(k_u)})_u$ exists that converges to $y$ and satisfies

$$\lim_{l\to\infty} h_{k_l} > 0.$$

With (5.6) it follows that

$$\lim_{l\to\infty} \nabla_x J(x^{(k_l)})^\top d^{(k_l)} = 0,$$

and thus

$$\lim_{l\to\infty} \alpha(x^{(k_l)}) = 0.$$

Since the mapping $x \mapsto \alpha(x)$ is continuous, it follows that $\alpha(y) = 0$. Thus, $y$ is Pareto critical.

**Second case** ($\limsup_{u\to\infty} h_{k_u} = 0$)**:** Lemma 5.4 assures that the sequence $(v^{(k_u)})_u$ is bounded. Therefore, we can select a subsequence $(x^{(k_r)})_r$ of $(x^{(k_u)})_u$ such that the sequence $(d^{(k_r)})_r$ converges to some $\bar{d}$. Recall that for all $r$ we have

$$\max_{i=0,1} \left(\nabla_x J_i(x^{(k_r)})^\top d^{(k_r)}\right) \leq \hat{\epsilon}\alpha(x^{(k_r)}) < 0, \quad \text{with } \hat{\epsilon} \in (0, 1].$$

With $r \to \infty$ we obtain

$$\frac{1}{\hat{\epsilon}} \max_{i=0,1} \left(\nabla_x J_i(y)^\top \bar{d}\right) \leq \alpha(y) \leq 0. \tag{5.7}$$

Take some $p \in \mathbb{N}$. For $r$ sufficiently large,

$$h_{k_r} < \frac{1}{2^p},$$

thus the Armijo-like rule (5.5) does not hold for $h = \frac{1}{2}^p$, i.e.

$$J\left(x^{(k_r)} + \left(\frac{1}{2}\right)^p d^{(k_r)}\right) \not\leq J(x^{(k_r)}) + \beta\left(\frac{1}{2}\right)^p \nabla_x J(x^{(k_r)})^\top d^{(k_r)}.$$

In a next step take a subsequence $(x^{(k_s)})_s$ of $(x^{(k_r)})_r$ such that $(d^{(k_s)})_s$ converges to $\bar{d}$. Then, we have

$$J\left(x^{(k_s)} + \left(\frac{1}{2}\right)^p d^{(k_s)}\right) \not\leq J(x^{(k_s)}) + \beta\left(\frac{1}{2}\right)^p \nabla_x J(x^{(k_s)})^\top d^{(k_s)}.$$

Then, passing onto the limit $s \to \infty$ yields

$$J_j\left(y + \left(\frac{1}{2}\right)^p \bar{d}\right) \geq J_j(y) + \beta\left(\frac{1}{2}\right)^p \nabla_x J_j(y)\hat{v}$$

for at least one $j \in \{0, 1\}$. This inequality holds for any $p \in \mathbb{N}$.
Thus, from Lemma 5.5 it follows that

$$\max_{i=0,1} \left(\nabla_x J_i(y)^\top \bar{d}\right) \geq 0,$$

implying together with (5.7) that $\alpha(y) = 0$. Thus, we can also conclude that $y$ is Pareto critical. □

A natural stopping condition for practical implementations of Algorithm 2, motivated by

(3.12), is that $\|h_k d^{(k)}\| \leq \varepsilon$, with $\varepsilon > 0$ a prespecified small constant.
In practice, we also terminate the algorithm when a prespecified maximum number of iterations is reached. In this case, the final solution has to be used with caution since the optimization procedure has generally not yet converged.
The choice of the search direction using problem (5.4) together with condition (5.5) implies that the iterates of Algorithm 2 satisfy $J(x^{(k+1)}) < J(x^{(k)})$ for all $k = 1, 2, \ldots$. In other words, the objective vector $J(x^{(k+1)})$ in iteration $k+1$ is bounded above by the objective vector $J(x^{(k)})$ of the previous iteration $k$, i.e., $J(x^{(k+1)}) \in J(x^{(k)}) - \mathbb{R}^2_>$.
Several alternative Pareto critical solutions (and hence trade-off information between them) can be obtained, for example, by varying the starting solution. We follow a different approach in our implementation that is somewhat similar to the weighted sum method, and that is based on the observation that the optimal solution of problem (5.4) (i.e., the direction of steepest biobjective descent) depends on the scaling of the objective functions $J_1$ and $J_0$. Thus, Algorithm 2 is executed repeatedly, using different scalings of the objective functions. In our implementation, we use a scaling parameter $s := \bar{\lambda} r^{\max} > 0$ and replace $J_0$ by $sJ_0$ in the optimization process, where the parameter $r^{\max} > 0$ is chosen as the largest ratio between partial derivatives of $J_1$ and $J_0$, evaluated at the starting solution $x^{(1)}$. Note that the latter aims at the constraints in problem (5.4) in the sense that they should be comparable, i.e., both objective functions should equally contribute to active constraints and thus influence the choice of the search direction. By varying the parameter $\bar{\lambda} \in \{0.5, 0.6, \ldots, 2\}$, we implicitly control the run of the gradient descent algorithm and thus obtain different solutions starting from the same initial shape. Note that the volume of the solutions can be expected to increase with larger values of $\bar{\lambda}$.
Note also that the resulting parametric version of Algorithm 2 is fundamentally different from the weighted sum method in Algorithm 1 in the way the search directions are chosen and in the way the iterates converge to a Pareto critical solution.
Note that the biobjective descent algorithm is introduced as a multiobjective descent algorithm in [60]. Therefore, it can handle multiple objectives without a modification. However, in practice the initial scaling of the objectives is difficult, and the probability of getting stuck in local minima generally increases with the number of objectives. The weighted sum method also naturally extends to more than two objectives, but the computational cost to explore the weight space $\{\lambda \in [0,1]^q \colon \sum \lambda_i = 1\}$ with $q > 2$ may become prohibitive. This may be, for example, handled by adaptive strategies for weight selections that aim at approximations of the Pareto set, see, e.g., [131].

## 5.2 Numerical Implementation

In this section an alternative scalar product for the computation of shape gradients is described and a control of step sizes that was utilized for the biobjective gradient descents is introduced.

### 5.2.1 Scalar Products and Gradients in Shape Optimization

The performance of Algorithms 1 and 2 depends largely on the choice of the search direction, which is computed based on the discretized gradients $\nabla_x J_j(x)$, $j = 0, 1$. Michor and Mumford [108] showed that (continuous) shape gradients calculated with respect to

the ordinary $L^2$-scalar product lead to an ill defined notion of the distance of two shapes, as the infimum over all deformation path lengths is zero. They suggest a modified scalar product given by

$$\langle h, k\rangle_{\varpi} = \int_{\partial\Omega} \langle h, k\rangle_{\mathbb{R}^2} \left(1 + \varpi\kappa^2\right) \mathrm{d}A \tag{5.8}$$

and show that this indeed leads to a well defined Riemannian metric on the shape space. Here, $h, k$ are two vector fields in normal direction to the boundary of $\partial\Omega$, $\mathrm{d}A$ is the induced surface measure, $\kappa$ is the scalar curvature of the surface, and $\varpi > 0$ is a regularization parameter. In practice, this corresponds to a transformation of function values on $\partial\Omega$ that, given some function $g : \partial\Omega \to \mathbb{R}^2$, can be described by $g_\varpi(x) = \frac{g(x)}{1+\varpi\kappa^2(x)}$ for $x \in \partial\Omega$.

Despite discretizing the space of shapes, we also discretize this definition of the gradient in order to obtain stability in the limit of small finite element mesh size and a high number of spline basis elements. We adopt a discretized version of this concept in the numerical implementation of shape gradients for both objectives $J_j$, $j = 0, 1$. More precisely, a discretized scalar curvature $\kappa$ is computed at grid points on the boundary $\partial\Omega$, which is represented by a polygonal approximation induced by the shape parameters $(\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}}) \in \mathbb{R}^{2n_x}$, $\varrho^{\mathrm{th}} \in \mathbb{R}^{n_x}_{>}$. Since the upper and lower boundary of the shape $\Omega$ may have a different curvature at the same $x$-coordinate value $x_i$ ($i \in \{1, \ldots, n_x\}$), we have to compute the curvature for upper and lower boundary points separately. For the upper boundary, this is realized by comparing the normals $n_i^{\mathrm{u}}$ and $n_{i+1}^{\mathrm{u}}$ on two consecutive facets of length $l_i^{\mathrm{u}}$ and $l_{i+1}^{\mathrm{u}}$, respectively. Similarly, for the lower boundary we use $n_i^{\mathrm{l}}$, $n_{i+1}^{\mathrm{l}}$ and $l_i^{\mathrm{l}}$, $l_{i+1}^{\mathrm{l}}$, and obtain

$$\begin{aligned} \kappa_i^{\mathrm{u}} &:= \kappa^{\mathrm{u}}(x_i) = \frac{2\|n_i^{\mathrm{u}} - n_{i+1}^{\mathrm{u}}\|_2}{l_i^{\mathrm{u}} + l_{i+1}^{\mathrm{u}}}, \\ \kappa_i^{\mathrm{l}} &:= \kappa^{\mathrm{l}}(x_i) = \frac{2\|n_i^{\mathrm{l}} - n_{i+1}^{\mathrm{l}}\|_2}{l_i^{\mathrm{l}} + l_{i+1}^{\mathrm{l}}}, \end{aligned} \qquad i = 1, \ldots, n_x - 1. \tag{5.9}$$

The upper and lower boundaries of the shape $\Omega$ are reconstructed from the meanline and thickness representation using the linear transformation $\varrho_i^{\mathrm{u}} = \varrho_i^{\mathrm{ml}} + \frac{1}{2}\varrho_i^{\mathrm{th}}$ and $\varrho_i^{\mathrm{l}} = \varrho_i^{\mathrm{ml}} - \frac{1}{2}\varrho_i^{\mathrm{th}}$, $i = 1, \ldots, n_x$. In other words, $(\varrho^{\mathrm{u}}, \varrho^{\mathrm{l}}) \in \mathbb{R}^{2n_x}$ is obtained from $(\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}}) \in \mathbb{R}^{2n_x}$, $\varrho^{\mathrm{th}} \in \mathbb{R}^{n_x}_{>}$, as $(\varrho^{\mathrm{u}}, \varrho^{\mathrm{l}}) = M\,(\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}})$, using an appropriate transformation matrix $M \in \mathbb{R}^{2n_x \times 2n_x}$. This leads to a discretized representation of the respective boundaries by points $(x_i, \varrho_i^{\mathrm{u}})$ (upper boundary) and $(x_i, \varrho_i^{\mathrm{l}})$ (lower boundary), from which the $\kappa$ values can be computed according to (5.9).

Now (5.8) can be applied to the gradients of $J_j$ w.r.t. $(\varrho^{\mathrm{u}}, \varrho^{\mathrm{l}})$, $j = 0, 1$, by multiplying the respective partial derivatives by

$$d_{\varpi,i}^{\mathrm{u}} := \frac{1}{1 + \varpi\,(\kappa_i^{\mathrm{u}})^2} \quad \text{and} \quad d_{\varpi,i}^{\mathrm{l}} := \frac{1}{1 + \varpi\,(\kappa_i^{\mathrm{l}})^2}, \quad i = 1, \ldots, n_x.$$

Since we actually need the gradients of $J_j$ w.r.t. $\varrho = (\varrho^{\mathrm{ml}}, \varrho^{\mathrm{th}})$, $j = 0, 1$, we additionally have to consider the linear tranformation $M$. Let $D_\varpi = (d_{\varpi,ij})_{2n_x \times 2n_x} \in \mathbb{R}^{2n_x \times 2n_x}$ be a

diagonal matrix with diagonal elements given by

$$d_{\varpi,ii} := d^{\mathrm{u}}_{\varpi,i}, \quad i = 1, \dots, n_x \quad \text{and} \quad d_{\varpi,ii} := d^{\mathrm{l}}_{\varpi,i-n_x}, \quad i = n_x+1, \dots, 2n_x,$$

and set $\bar{D}_\varpi := M^{-1} D_\varpi M$. Then, we obtain the curvature adapted B-spline gradients as

$$\left(\frac{\partial J_j}{\partial x}\right)_\varpi = \bar{D}_\varpi \left(\frac{\partial J_j}{\partial X}\frac{\partial X}{\partial \varrho}\right)\frac{\partial \varrho}{\partial x}, \qquad j = 0, 1. \tag{5.10}$$

Note that for $\varpi = 0$ the matrix $\bar{D}_0$ is the identity matrix, and hence the $L^2$-gradient of $J_j$ w.r.t. $x$, $j = 0, 1$, is recovered in this case, c.f. (4.9).

### 5.2.2 Control of Step Sizes

Large mesh deformations may cause numerical difficulties and thus have to be avoided. We thus limit the step size during the optimization procedure. Recall that the representation of feasible shapes, using meanline and thickness values $(\varrho_i^{\mathrm{ml}}, \varrho_i^{\mathrm{th}})$ at fixed $x_i$ coordinates, $i = 1, \dots, n_x$, implies that grid points can only move vertically. A natural choice for a maximum admissible step in one iteration of the optimization process is thus determined by the thickness of the shape, divided by the number $n_y$ of gridpoints in $y$-direction. Since in our case studies the shapes are fixed at the left boundary (i.e., at $x = x_1$) and hence their thickness is constant at $x_1$, we set

$$\delta^{\max} := 0.8\,\frac{\varrho_1^{\mathrm{th},(1)}}{n_y}$$

i.e., to 80% of the vertical distance between grid points on the left boundary of the initial shape. For a given search direction $d^{(k)} = (d^{\mathrm{ml},(k)}, d^{\mathrm{th},(k)}) \in \mathbb{R}^{2n_B}$ in iteration $k$ of the optimization algorithms, we check whether $\max_{i=1,\dots,2n_B} |d_i^{(k)}| \le \delta^{\max}$. Otherwise, $d^{(k)}$ is scaled by a factor $\delta^{\max}/\max_{i=1,\dots,2n_B} |d_i^{(k)}|$. Then, the step length $t \le 1$ is computed according to the Armijo rule as indicated in Algorithms 1 and 2.
While $\delta^{\max}$ is derived from the mesh $X^{(1)}$, it still is a meaningful upper bound for a step $d^{(k)}$ in the B-spline representation. Indeed, if $\{\mathcal{B}_j,\, j = 1, \dots, n_B\}$ is a B-spline basis and $x^{(k)} = (x^{\mathrm{ml},(k)}, x^{\mathrm{th},(k)}) \in x$ is the current iterate, then the B-spline basis properties $\sum_{j=1}^{n_B} \mathcal{B}_j(x) = 1$ and $\mathcal{B}_j(x) \ge 0$, $j = 1, \dots, n_B$ (see, e.g., [120]) imply that, for all $i = 1, \dots, n_x$,

$$\begin{aligned}\left|\varrho_i^{\mathrm{ml},(k+1)} - \varrho_i^{\mathrm{ml},(k)}\right| &= \left|\sum_{j=1}^{n_B}(x_j^{\mathrm{ml},(k)} + d_j^{\mathrm{ml},(k)})\mathcal{B}_j(x_i) - \sum_{j=1}^{n_B} x_j^{\mathrm{ml},(k)}\mathcal{B}_j(x_i)\right| \\ &\le \sum_{j=1}^{n_B} |d_j^{\mathrm{ml},(k)}|\,|\mathcal{B}_j(x_i)| \le \max_{j=1,\dots,n_B} |d_j^{\mathrm{ml},(k)}| \sum_{j=1}^{n_B} |\mathcal{B}_j(x_i)| = \max_{j=1,\dots,n_B} |d_j^{\mathrm{ml},(k)}|.\end{aligned}$$

An analogous bound holds for the corresponding thickness parameters. Note that the above inequalities do in general not guarantee that *all* grid points of the corresponding mesh $X^{(k)}$ move by at most 80%, since this also depends on the current shape and the mutual movement of meanline and thickness values. In some situations it may thus be

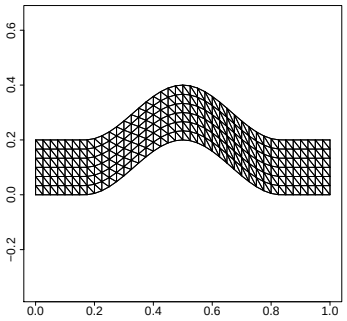
necessary to adapt this bound to a smaller value. However, this never occured in our numerical tests.

## 5.3 Numerical Results

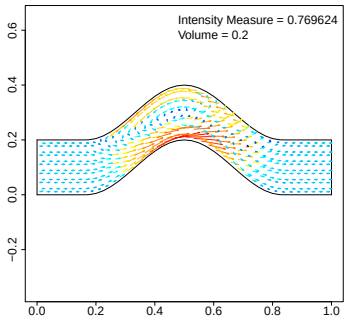
In this section, the biobjective descent methods proposed in Section 5.1.2 are applied on the two test cases described in Section 4.4. To that end, the curvature regularization parameter is set to $\varpi = 10^{-4}$, see Section 5.2.1. During the optimization process, we monitor the Euclidean norm of the update of the design variables in every iteration and stop when it is lower than $10^{-4}$. The implementation is realized in R version 3.5.0 and uses the adjoint finite element code of [79, 21, 80] as a subroutine.

### 5.3.1 A Straight Joint

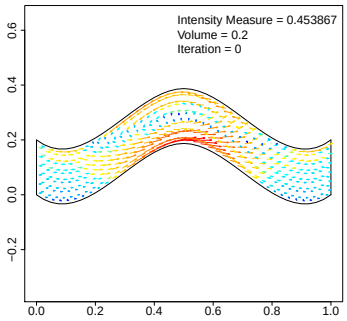
Among all shapes with a fixed volume of $J_0(X) = 0.2$, the straight rod shown in Figure 5.1d can be expected to have the mimimum possible intensity measure $J_1$. Indeed, the straight rod shown in Figure 5.1d achieves an objective value of $J_1(X) = 0.00058$. Figures 5.1e and 5.1f show the results of the weighted sum method (Algorithm 1) with weight $\lambda = 0.8$ and of the biobjective descent algorithm (Algorithm 2) with scaling parameter $\bar{\lambda} = 1.8$. Both methods show a rather quick convergence (with the expected advantage for the biobjective descent algorithm) to solutions that are close to optimal. However, the solution of the biobjective descent algorithm seems to be a local solution with slightly higher stresses (and thus slightly higher objective value for $J_1$).
Figure 5.2 shows iteration histories of exemplary runs of the weighted sum method (Algorithm 1) and of the biobjective descent algorithm (Algorithm 2), respectively. It nicely illustrates that, in contrast to the biobjective descent algorithm, the weighted sum method permits iterations where one objective function deteriorates while the weighted sum objective is still decreasing. This may, in certain situations, help to overcome local Pareto critical solutions. On the other hand, the weighted sum method may get stuck in local minima as well. Indeed, independent of the chosen weight, the histories of the weighted sum method have a similar structure: First, mainly the intensity measure (representing the PoF) is improved (since in early stages of the algorithm the gradient of $J_1$ is considerably larger than the gradient of $J_0$). Only at later stages of the algorithm, the volume is varied to a larger extent, depending on the given weight.
Note also that the final solution obtained with the biobjective descent algorithm largely depends on the starting solution, since the objective values can never deteriorate during the optimization process. Thus, when the starting solution has a volume of $J_0(X) = 0.2$, then all Pareto critical shapes that can be computed with the biobjective descent algorithm have a volume of at most 0.2, irrespective of the scaling.
Three shapes with progressively reduced volume (and hence lower cost) are shown in Figures 5.1g to 5.1i. As was to be expected, a lower cost comes at the price of a higher intensity measure (and hence higher PoF). A comparison between Figures 5.1h and 5.1g suggests that also for the low volume solutions, the weighted sum solutions slightly outperform the biobjective descent solutions.
Figure 5.3 summarizes the results of several optimization runs with varying weights (Algorithm 1) and varying scalings (Algorithm 2), respectively. The same starting solution was



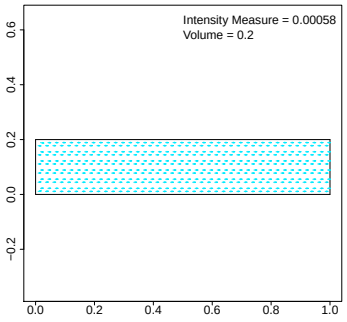
(a) Starting shape: Tetrahedral mesh $X$



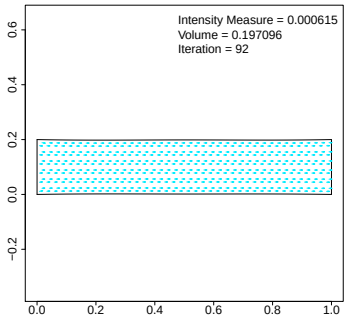

(b) Starting shape: Objective values and stresses



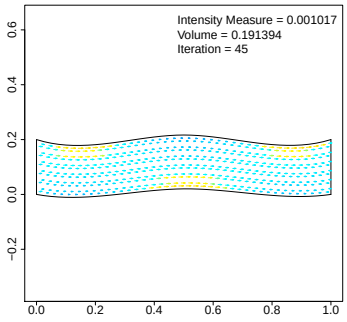

(c) Starting shape: Approximation with B-splines



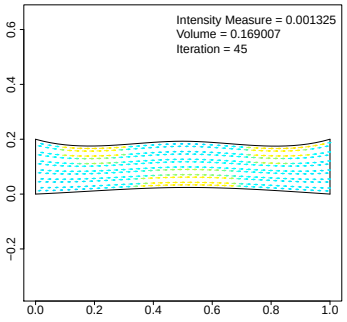

(d) Expected result: Straight rod



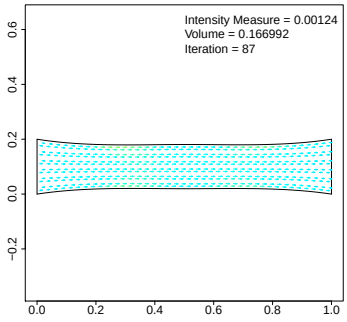

(e) Weighted sum, $\lambda = 0.8$



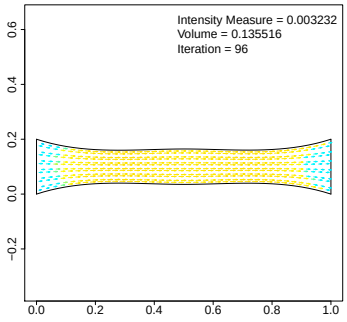

(f) MO descent, $\bar{\lambda} = 1.8$




(g) MO descent, $\bar{\lambda} = 0.5$




(h) Weighted sum, $\lambda = 0.6$




(i) Weighted sum, $\lambda = 0.3$

Figure 5.1: Straight joint: Starting solution (row 1), straight rod solutions (row 2), and low volume solutions (row 3). See also [46].

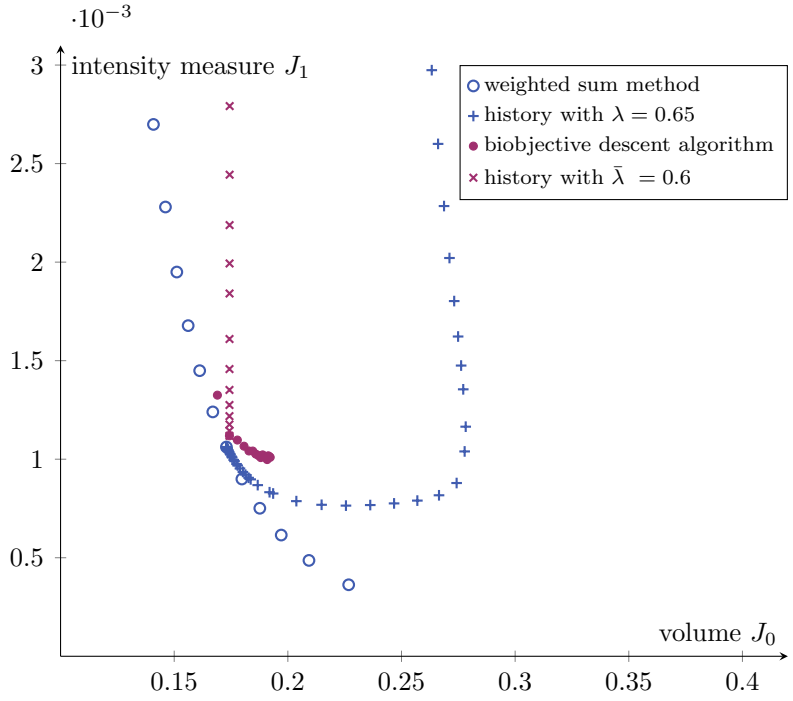



Figure 5.2: Iteration histories of an exemplary run of the weighted sum method (Algorithm 1) and of the biobjective descent algorithm (Algorithm 2). Note that both algorithms use the same starting solution. See also [46].

used in all cases, see Figure 5.1c. While the solution quality of the weighted sum method and of the biobjective descent algorithm is comparable, a clear advantage of the weighted sum method seems to be that it is not so much constrained by the (performance of the) starting solution. Indeed, the weighted sum solutions shown in Figure 5.3 span a large range of alternative objective values in the objective space and thus provide the decision maker with meaningful trade-off information and a variety of solution alternatives.

Both algorithms need in general one gradient computation and $k_A$ function evaluations per iteration, where $k_A$ is the number of iterations in the Armijo rule to calculate a step length. Additionally, the biobjective descent needs one gradient evaluation, whereas the weighted sum requires one objective function evaluation to determine an initial scaling of the objectives. In this test case the mean number of iterations for the weighted sum method was 94 with around 3.8 Armijo iterations on average. The biobjective descent needed 46 iterations with 1.7 Armijo iterations on average. On this rather coarse grid ($41 \times 7$) a function evaluation takes about 1.2 seconds and a gradient evaluation around 15.48 seconds, a finer grid would extend the run time significantly. Note that the underlying simulation code for the function evaluation and gradient computation is not optimized w.r.t. runtime. Summing up, an optimization run with the weighted sum method for this test case on a $41 \times 7$ grid took about 31.4 minutes on average. The biobjective descent algorithm took about 14.9 minutes on average. All algorithms are implemented in R version 3.5.0, and the numerical tests run on a PC with Intel Core i7-8700 CPU @ 3.20 GHz, 31.2 GB RAM.

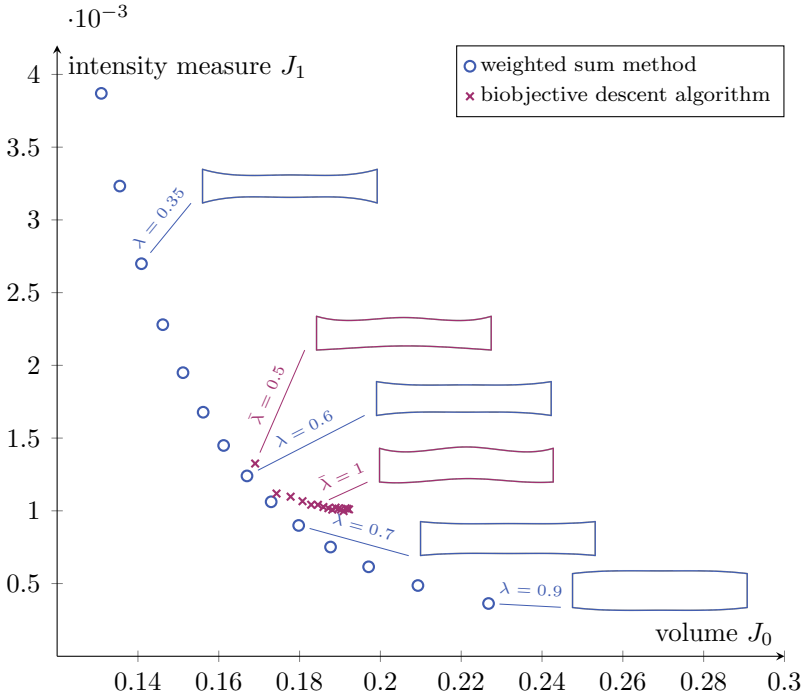
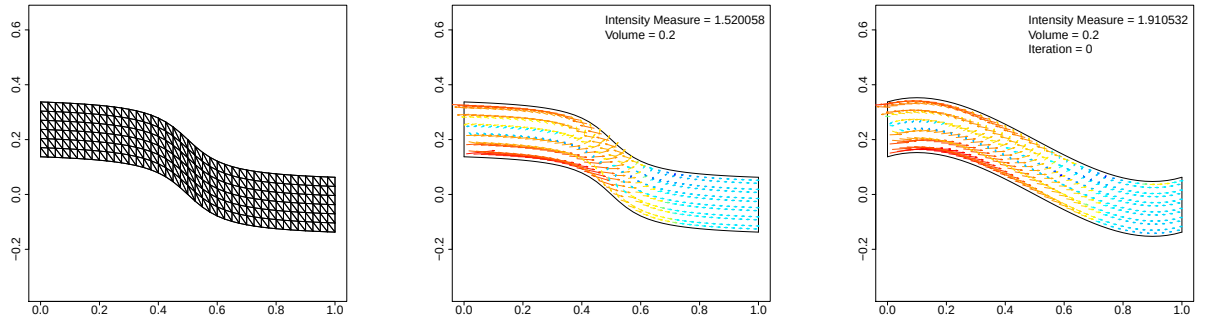



Figure 5.3: Approximated nondominated front for the straight joint. The associated Pareto critical shapes are shown for selected weightings/scalings. See also [46].

### 5.3.2 An S-Shaped Joint

We observe that the resulting shapes resemble the profile of a whale. If we consider 1st principal stress of the stress tensor on the grid points of the initial shape resulting from tensile load, see Figure 5.4c, we observe an anti clockwise eddy in the left part of the joint. The hunch close to the left boundary of the optimized shapes gives room for the occurring stresses and therefore improves the intensity measure and, likewise, the PoF.
Note that, different from the case of the straight rod, we have no prior knowledge on the Pareto optimal shapes. For the solutions shown in Figures 5.4d and 5.4e, we can only guarantee that they are (approximately) Pareto critical, i.e., the respective optimization runs terminated due to the criticality test. Figure 5.4f shows a shape with a significantly higher volume of $J_0(X) = 0.225906$, and with a largely improved intensity measure of $J_1(X) = 0.196791$. This shape was obtained with the weighted sum method with weight $\lambda = 0.85$ after 150 iterations. In this case, the algorithm terminated since it reached the maximum number of iterations and not due to convergence. We observed that all optimization runs of the weighted sum method with $\lambda \geq 0.85$ were not converging in this setting. Thus in these cases it is not guaranteed, that the resulting solutions are Pareto critical. Note that, given a starting solution with a volume of 0.2, this shape is not attainable with the biobjective descent algorithm.
However, there is no guarantee that the computed shapes are Pareto optimal. For example, the shape shown in Figure 5.4e obtained with the weighted sum method with weight $\lambda = 0.8$ achieves objective values of $J_1(X) = 0.293853$ and $J_0(X) = 0.188445$, and hence slightly dominates the shape shown in Figure 5.4d obtained with the biobjective descent algorithm with scaling parameter $\bar{\lambda} = 1.1$ that has objective values $J_1(X) = 0.300996$ and



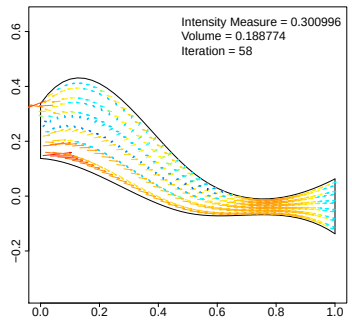
(a) Starting shape: Tetrahedral mesh $X$



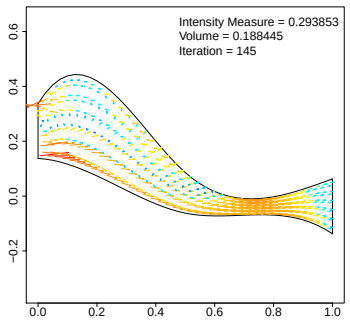

(b) Starting shape: Objective values and stresses



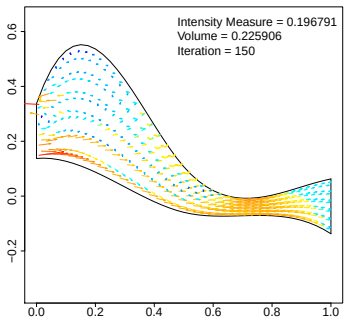

(c) Starting shape: Approximation with B-splines




(d) MO descent, $\bar{\lambda} = 1.1$




(e) Weighted sum, $\lambda = 0.8$




(f) Weighted sum, $\lambda = 0.85$, not converged

Figure 5.4: S-shaped joint: Starting solution (row 1), two exemplary Pareto critical solutions (5.4d and 5.4e) , and a not converged solution of the weighted sum method (5.4f). See also [46].

$J_0(X) = 0.188774$.

Figure 5.5 summarizes the results of several optimization runs of both Algorithms 1 and 2 in the objective space. Note that not all solutions of the weighted sum method lie on the convex hull of the computed points (and are thus not globally optimal for a weighted sum scalarization). In some cases, the biobjective descent algorithm also computes dominated points, while in other cases it found solutions that lie even below the convex hull of the weighted sum solutions (see, e.g., the result for $\bar{\lambda} = 0.5$ in Figure 5.5).

A larger range of alternative objective vectors is, as in the case of the straight rod, obtained with the weighted sum method. A cross-test between the two methods, where the final solution of Algorithm 1 was used as starting solution for Algorithm 2, confirms that local Pareto critical solutions were found for $\lambda \leq 0.8$.

Compared to test case 1, the optimization runs for test case 2 needed in general more iterations. The mean number of iterations for the weighted sum method in test case 2 was 107 with around 5.3 Armijo iterations on average. The biobjective descent needed 74 iterations with 3.9 Armijo iterations on average. Thus, the weighted sum method needed about 39.06 minutes on average and the biobjective descent algorithm took about 26.96

minutes on average.

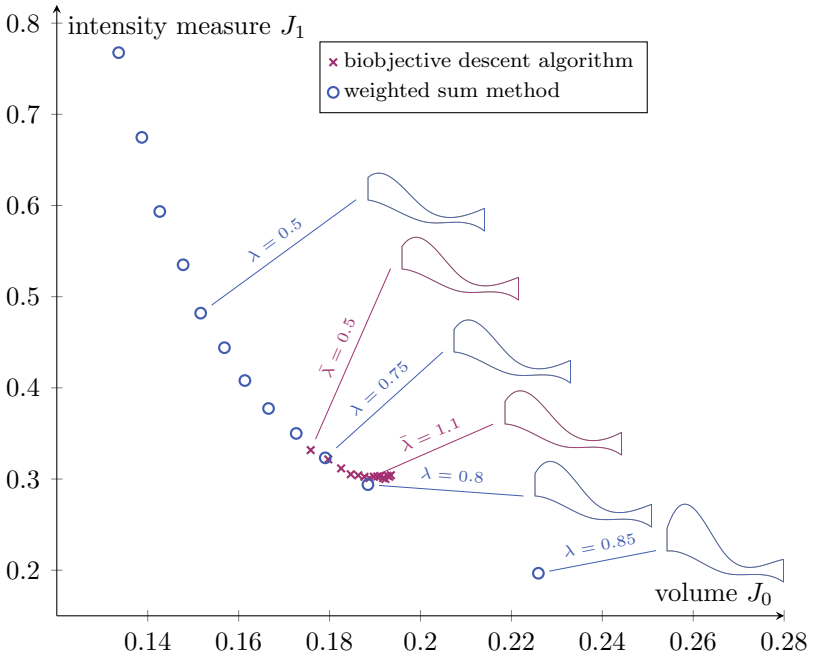



Figure 5.5: Outcome vectors for the S-shaped joint. The associated Pareto critical shapes are shown for selected weightings / scalings. Compare with [46].

# 6 Pareto Tracing by Numerical Integration

Most of this chapter was published in [19]. Here, we additionally give a brief overview of standard ODE theory and provide more detail than in [19].
The aim of this chapter is to introduce a reliable and efficient method to approximate the Pareto front of convex and sufficiently smooth unconstrained biobjective optimization problems. Based on the optimality conditions of the *weighted sum scalarization*, see also Section 3.5, (parts of) the Pareto front can be described as a parametric curve, parameterized by the scalarization parameter $\lambda$, i.e., the weight in the weighted sum scalarization. Differentiating it w.r.t. the parameter yields an (explicit) ordinary differential equation (ODE) that, given an arbitrary initial solution on the Pareto front, enables one to trace the Pareto front by numerical integration. We call this novel approach *Pareto tracing by numerical integration*. While the developed methods are tailored for convex problems, we show that they are more generally applicable and that they can be adapted to handle nonconvex problems and to approximate convex parts of non connected Pareto fronts.
To compute representations and approximations of the Pareto front, scalarization methods are a common choice, see, e.g., [52]. Under differentiability assumptions, optimality conditions, for example, the classical KKT-conditions, can be incorporated to obtain further parts of the Pareto front, see, for example, [52, 85]. Following the literature, one can then apply sensitivities w.r.t. the scalarization parameters [53], subdivision techniques [42, 89, 141], or continuation and predictor-corrector methods [53, 104, 103, 115, 116, 125, 132, 140] to recover the Pareto front. Continuation and predictor-corrector methods usually are based on scalarizations, yielding corresponding single-objective problems depending on one or several scalarization parameters, e.g., the weights $\lambda$ in the case of weighted sum scalarizations. Hence, they can be interpreted as parametric optimization problems, which under appropriate differentiability assumptions can then be associated to the single-objective case for predictor-corrector methods, see, e.g., [7, 77]. Under appropriate differentiability assumptions, combinations of predictor steps (that are often derived from dual information from the previous iteration) and corrector steps can then be used to recover the manifolds that are induced by the optimality conditions for the respective parametric optimization problems. Continuation methods can handle constrained problems as well as problems with more than two objective functions, and they can be adapted in order to incorporate preference information in the exploration of the Pareto front, see, for example, [103].
An alternative perspective on parametric scalarizations in the biobjective case is suggested in [121]: When interpreting the scalarization parameter as an independent variable, then the parametric optimization problem induces a system of ordinary differential equations that can be solved by numerical integration methods. In the article [121] the normalized normal constraint (NNC) scalarization method of [107] and its utopia line are utilized to

formulate an ODE to trace the Pareto front, which is then, given an initial value, solved with standard integrators. Since the NNC scalarization induces additional constraints to the parameterized problem, an active set method is employed. This may lead to the generation of locally Pareto-optimal outcome vectors which are filtered in a post processing step.

In this work, we follow a similar strategy, however, we use simple weighted sums to define the underlying parametric scalarizations. The main advantage of this approach is that no additional constraints are added to the problem formulation, and hence, at least for unconstrained biobjective optimization problems, no constraint handling techniques are required. Nevertheless our numerical results show that we obtain well distributed points on the Pareto front approximation since the step length is controlled by the numerical integration method.

This chapter is structured as follows. First, a brief introduction in the existence and uniqueness theory of solutions of a system of first-order *ordinary differential equations* (ODEs) is given in Section 6.1. Then, the *Pareto tracing by numerical integration* method, suggested in [19], is explained in detail in Section 6.2. Under appropriate differentiability assumptions and a given initial Pareto-optimal solution of a weighted sum scalarization this method uses an explicit first-order ODE to compute further Pareto-optimal solutions. Toward this end, assuming local Lipschitz continuity of the Hessians of both objective functions, the existence and continuity of the solutions of the ODE are established in Section 6.2.1. Since the subject of this work are biobjective shape optimization problems, i.e., complex real-world applications with only approximated solutions, the results are then extended to the case that initial solutions are $\varepsilon$-Pareto critical, with $\varepsilon > 0$ (Section 6.2.2). In a next step, in Section 6.3 the application of the well-established Runge-Kutta methods is suggested, local and global error estimates are provided and the Pareto tracing by numerical integration algorithm is stated for this case. Further, in Section 6.4, a brief overview of the multiobjective predictor-corrector method *Pareto Tracer* is given, see also [103]. Subsequently, in Section 6.5, the Pareto tracing by numerical integration approach is first validated and tested against Pareto Tracer on quadratic test problems (Section 6.5.1) and a variant of the biobjective test problem ZDT3 (Section 6.5.2), see also [158], for which the Pareto fronts are known. In a next step, it is applied on our biobjective shape optimization problem with the two test cases described in Sections 4.4.1 and 4.4.2.

## 6.1 A Brief Overview of First-Order Ordinary Differential Equations

In this section a brief summary of some of the important results of the existence and uniqueness theory of solutions of *first-order ordinary differential equations* is given. Subsequently these results will be extended to *systems of first-order ordinary differential equations*. For a more detailed discussion of this field we refer to [4], on which this section is based.

### 6.1.1 First-Order Ordinary Differential Equations

First, a definition of ordinary differential equations of order $n$ is given.

**Definition 6.1** (Ordinary Differential Equation (ODE))**.** *An ordinary differential equation is a relation that contains one real independent variable $t \in \mathbb{R}$ and the real dependent variable $x = x(t)$ with some of its derivatives $(\dot{x}, \ddot{x}, \ldots, x^{(n)})$. In general, an ODE can be written in* implicit *form*

$$F\left(t, x, \dot{x}, \ddot{x}, \ldots, x^{(n)}\right) = 0, \tag{6.1}$$

*where $F$ is a known function of $n+2$ variables. Then, the* order of an ODE *is given by the order of the highest derivative, $x^{(n)}$, in the equation.*

Then, a first-order ordinary differential equation is defined as follows.

**Definition 6.2** (first-order Ordinary Differential Equation)**.** *The* explicit differential equation of the first-order *is given by*

$$\dot{x} = f(t, x) \tag{6.2}$$

*where the real function $x = x(t)$ is unknown and $f(t, x)$ is a given function of two real variables, defined on some domain $D \subset \mathbb{R}^2$.*

**Definition 6.3.** *Let $I \subset \mathbb{R}$ be an interval. A function $x : I \to \mathbb{R}$ is said to be a* (particular) solution *of* (6.2) *in $I$, if for any $t \in I$ $x(t)$ is differentiable, $\dot{x}(t) = f(t, x(t))$ holds for all $t \in I$, and $(t, x(t)) \in D$ for all $t \in I$. The family of all particular solutions of* (6.2) *is denoted as the* general solution *of* (6.2)*.*

In practice it is desirable that for a given $t_0 \in I$ the solution of the ODE (6.2) satisfies an additional initial condition $x_0 = x(t_0)$, called *initial boundary condition.* The first-order ODE (6.2) with an additional initial boundary condition $x_0 = x(t_0)$ is of the form

$$\begin{cases} \dot{x} = f(t, x), \\ x(t_0) = x_0, \end{cases} \tag{6.3}$$

and is called *initial value problem* (IVP). In the following, the existence and uniqueness of solutions for the IVP (6.3) is discussed. Toward this end, we assume from here on that $f(t, x)$ is continuous in a domain $D$ that contains the point $(t_0, x_0)$.

**Definition 6.4.** *Let $I \subset \mathbb{R}$ be an interval containing $t_0$. A* solution *of the initial value problem* (6.3) *on the interval $I$ is a function $x : I \to \mathbb{R}$ satisfying*

*(i)* $x(t_0) = x_0$,

*(ii)* $x(t)$ *is differentiable for all* $t \in I$,

*(iii)* $(t, x(t)) \in D$ *for all* $t \in I$,

*(iv)* $\dot{x} = f(t, x)$ *for all* $t \in I$.

One can prove the uniqueness of solutions for functions $f$ that are *Lipschitz* continuous.

**Definition 6.5.** *A function $f : D \to \mathbb{R}$ is called* Lipschitz continuous *in $x$ if there exists a* Lipschitz constant $L > 0$*, such that*

$$|f(t, x'(t)) - f(t, x''(t))| \leq L\, |x'(t) - x''(t)| \quad \text{for all } (t, x'(t)), (t, x''(t)) \in D. \tag{6.4}$$

*A function $f : D \to \mathbb{R}$ is* locally Lipschitz continuous *in $x$ if for any $(t, x(t)) \in D$ there are $\delta, \Delta > 0$ such that the rectangle*

$$B = [t - \Delta, t + \Delta] \times [x(t) - \delta, x(t) + \delta] \subset D$$

*and there exists a* Lipschitz constant $L := L(t, x(t)) > 0$*, such that*

$$|f(t, x'(t)) - f(t, x''(t))| \leq L\, |x'(t) - x''(t)| \quad \text{for all } (t, x'(t)), (t, x''(t)) \in B. \tag{6.5}$$

If the partial derivative $\nabla_x f(t, x(t))$ of $f(t, x(t))$ exists and possesses some additional properties, one can draw some conclusions about the Lipschitz continuity of $f$.

**Lemma 6.6.**

*(i) If $\nabla_x f(t, x(t))$ exists and is bounded in a rectangle $B \subset \mathbb{R}^2$ then $f(t, x(t))$ is Lipschitz for all $x \in B$.*

*(ii) If $\nabla_x f(t, x(t))$ exists and is continuous in an open set $\Gamma \subset \mathbb{R}^2$ then $f(t, x(t))$ is Lipschitz for all $x \in \Gamma$.*

*Proof.*

(i) Let $(t, x'), (t, x'') \in B$ and assume without loss of generality that $x' < x''$. Since $B$ is a rectangle, the connecting interval between these points is also in $B$. Applying the mean value theorem yields

$$f(t, x'') - f(t, x') = \nabla_x f(t, \xi)(x'' - x'),$$

for some $\xi \in [x', x'']$. By assertion we also have

$$|f\,(t, x'') - f\,(t, x')| = |\nabla_x f\,(t, \xi)\,(x'' - x')| \leq \sup_B |\nabla_x f\,(t, \xi)|\, |x'' - x'|\,, \tag{6.6}$$

hence $f$ is Lipschitz with $L := \sup_B |\nabla_x f(t, \xi)| < \infty$.

(ii) Let $(t_0, x_0) \in \Gamma$ and choose $\delta, \Delta > 0$ sufficiently small, such that the bounded closed set $B = [t - \Delta, t + \Delta] \times [x(t) - \delta, x(t) + \delta]$ is in $\Gamma$, this is always possible since $\Gamma$ is open. Now (ii) follows with (i).

□

In the theory of ordinary differential equations it is practical to reformulate the IVP (6.3) as an integral equation.

**Theorem 6.7.** *Let $f : D \to \mathbb{R}$ be continuous. A function $x$ is a solution of the IVP* (6.3)*, if and only if $x$ is also a solution of*

$$x(t) = x_0(t) + \int_{t_0}^{t} f(y, x(y))\, dy. \tag{6.7}$$

*Proof.* For any solution $x(t)$ of (6.3) it is $\dot{x} = f(t, x)$. Integrating both sides yields

$$x(t) - x(t_0) = \int_{t_0}^{t} f(y, x(y))\, dy.$$

On the other hand, for every solution $x(t)$ of (6.7) it is $x(t_0) = x_0$ and differentiating (6.7) yields $\dot{x} = f(t, x)$. □

The well-known *Gronwall-type integral inequalities* are useful to formulate error estimations. Note that in the following, the absolute values of the integrals have to be considered, since $t \in [t_0 - \Delta, t_0 + \Delta]$, $\Delta > 0$, can be smaller than $t_0$.

**Theorem 6.8.** *Let $u(t), p(t)$ and $q(t)$ be non negative continuous functions on $[t_0 - \Delta, t_0 + \Delta]$, for $\Delta > 0$, and*

$$u(t) \le p(t) + \left| \int_{t_0}^{t} q(y)\, u(y)\, dy \right| \quad \text{for all } t \in [t_0 - \Delta, t_0 + \Delta]. \tag{6.8}$$

*Then the following inequality holds*

$$u(t) \le p(t) + \left| \int_{t_0}^{t} q(y)\, u(y)\, \exp\left( \left| \int_{y}^{t} q(s)\, ds \right| \right) dy \right| \quad \text{for all } t \in [t_0 - \Delta, t_0 + \Delta]. \tag{6.9}$$

*Proof.* For the proof of (6.9) consider $t \in [t_0, t_0 + \Delta]$, where the case $t \in [t_0 - \Delta, t_0]$ is analogous. First define

$$r(t) = \int_{t_0}^{t} q(t)u(t)\, dt$$

such that $r(t_0) = 0$ and

$$\dot{r}(t) = q(t)u(t).$$

With the hypothesis (6.8), $u(t) \le p(t) + r(t)$, it follows that

$$\dot{r}(t) \le p(t)q(t) + q(t)r(t),$$

which multiplied by $\exp\left(-\int_{t_0}^{t} q(s)\, ds\right)$ yields

$$\frac{d}{dt}\left( \exp\left( -\int_{t_0}^{t} q(s)\, ds \right) r(t) \right) \le p(t)q(t) \exp\left( -\int_{t_0}^{t} q(s)\, ds \right).$$

Furthermore, we obtain by integration

$$r(t) \le \int_{t_0}^{t} p(y)q(y) \exp\left( \int_{y}^{t} q(s)\, ds \right) dy$$

and consequently (6.9) follows from $u(t) \le p(t) + r(t)$. □

**Corollary 6.9.** *If in Theorem 6.8 the function $p(t) \equiv 0$, then $u(t) \equiv 0$.*

**Corollary 6.10.** *If the functions $u(t)$ and $p(t)$ of Theorem 6.8 are of the form $u(t) = c_0 + c_1|t - t_0|$ and $p(t) = c_2$, where $c_0, c_1, c_2 > 0$, then*

$$u(t) \leq \Big(c_0 + \frac{c_1}{c_2}\Big) \exp\Big(c_2|t - t_0|\Big) - \frac{c_1}{c_2} \quad \textit{for all } t \in [t_0 - \Delta, t_0 + \Delta]. \tag{6.10}$$

For a proof we refer to [4]. In the following, some fundamental definitions and theorems from real analysis are needed and are stated without proof.

**Definition 6.11.** *We say a sequence of functions* $\{x_q(t)\}$ converge uniformly *to a function $x(t)$ in an interval $I \subset \mathbb{R}$ if for every $\varepsilon > 0$ there exists a $n_0 \in \mathbb{N}$ such that for $q \geq n_0$, $|x_q(t) - x(t)| \leq \varepsilon$ for all $t \in I$*

**Theorem 6.12.** *Let $\{x_q(t)\}$ be a sequence of continuous functions with $\lim_{q\to\infty} x_q(t) = x(t)$ uniformly in $I \subset \mathbb{R}$. Then, $x(t)$ is continuous in $I$.*

**Theorem 6.13** (Lebesgue's Dominated Convergence Theorem)**.** *Let $\{x_q(t)\}$ be a sequence of functions with $\lim_{q\to\infty} x_q(t) = x(t)$ uniformly in $I \subset \mathbb{R}$ and let $f(t, x)$ be a continuous function in $D$, such that $(t, x_q(t)) \in D$ for all $q$ and $t \in I$. Then*

$$\lim_{q\to\infty} \int_I f(t, x_q(t))\, dt = \int_I \lim_{q\to\infty} f(t, x_q(t))\, dt = \int_I f(t, x(t))\, dt. \tag{6.11}$$

**Theorem 6.14** (Weierstrass' M-Test)**.** *Let $\{x_q(t)\}$ be a sequence of functions. Furthermore, let $|x_q(t)| \leq M_q$ for all $t \in I$ with $\sum_{q=0}^{\infty} M_q < \infty$. Then, $\sum_{q=0}^{\infty} x_q(t)$ converges uniformly in $I$ to a unique function $x(t)$.*

**Theorem 6.15** (Implicit Function Theorem)**.** *Let $f(t, x)$ be defined in $D = I \times \mathbb{R}$, continuous in $t$ and differentiable in $x$. Let further $0 < m \leq f(t, x) \leq M < \infty$ for all $(t, x) \in D$. Then, there exists a unique continuously differentiable solution $x(t)$ in $I$ of the equation $f(t, x) = 0$.*

In the following, to prove the existence of solutions, the integral equation (6.7) will be solved with the Picard method, see, e.g., [4]. To this end, we assume a continuous function $x_0(t)$, where $x_0(t) \equiv x_0$ is a common choice, as the initial approximation of the desired solution $x(t)$ of (6.3), and define the iterate $x_1(t)$ as

$$x_1(t) = x_0(t) + \int_{t_0}^{t} f(y, x_0(y))\, dy. \tag{6.12}$$

Substituting $x_1(t)$ with $x_0(t)$ on the right hand side of (6.12) yields the next iterate $x_2(t)$. Therefore, the $q-$th approximation $x_q(t)$ is obtained from $x_{q-1}(t)$ through

$$x_q(t) = x_0(t) + \int_{t_0}^{t} f(y, x_{q-1}(y))\, dy, \quad q = 1, 2, 3, \ldots. \tag{6.13}$$

Following Theorem 6.13, if the sequence of functions $\{x_q(t)\}$ converges uniformly to a continuous function $x(t)$ in some interval $I$ that contains $t_0$ and $(t, x_q(t)) \in D$ for all $t \in I$,

we have

$$x(t) = \lim_{q\to\infty} x_q(t) = x_0(t) + \lim_{q\to\infty} \int_{t_0}^{t} f(y, x_{q-1}(y))\, dy = x_0(t) + \int_{t_0}^{t} f(y, x(y))\, dy. \quad (6.14)$$

Let $\{x_q(t)\}$ be the sequence obtained by the approximation method (6.13). The following famous theorems, see, e.g., [4, Theorem 8.1 and 8.2], ensure the uniform convergence of $\{x_q(t)\}$ to the unique solution $x(t)$ of (6.3) under sufficient conditions.

**Theorem 6.16** (Local Existence Theorem)**.** *Let the following conditions be satisfied*

*(i)* $f(t,x)$ *is continuous in the closed rectangle* $S = \{(t,x) \in D \,|\, |t_0 - t| \le \Delta,\ |x_0 - x| \le \delta\}$*, and therefore there exists a* $M > 0$ *such that* $|f(t,x)| \le M$ *for all* $(t,x) \in S$*,*

*(ii)* $f(t,x)$ *is locally Lipschitz in* $S$ *with Lipschitz constant* $L$*,*

*(iii)* $x_0(t)$ *is continuous in* $[t_0 - \Delta, t_0 + \Delta]$*, and* $|x_0(t) - x_0| \le \delta$*.*

*Then the sequence of functions* $\{x_q(t)\}$*, generated with* (6.13)*, converges uniformly to the unique solution* $x(t)$ *of the initial value problem* (6.3)*. The validity of this solution is ensured on the interval* $I_h := [t_0 - h, t_0 + h]$*, where* $h = \min(\Delta, \delta/M)$*. Further, the following error estimate holds for all* $t \in I_h$*:*

$$|x(t) - x_q(t)| \le e^{Lh} \max_{t\in I_h} |x_1(t) - x_0(t)| \, \min\left(1, \frac{(Lh)^q}{q!}\right), \quad q = 1, 2, 3, \ldots. \quad (6.15)$$

*Proof.* First, we show that the iterates $\{x_q(t)\}$ generated via (6.13) are continuous in $I_h$ and for each $t \in I_h$ we have that $(t, x(t)) \in S$. We use the following inductive argument. Since $x_0(t)$ is continuous in $[t_0 - \Delta, t_0 + \Delta]$, further the function $F_0(t) = f(t, x_0(t))$ is continuous in $I_h$ and therefore $x_1(t)$ is also continuous in $I_h$. Moreover following inequality holds

$$|x_1(t) - x_0(t)| \le \left| \int_{t_0}^{t} |f(y, x_0(y))| \, \mathrm{d}y \right| \le M|t_0 - t| \le Mh \le \delta.$$

Let us assume that the assertion is true for $x_{q-1}(t)$, $q \ge 2$, then it is sufficient to prove that it is also true for $x_q(t)$. Toward this end, the function $F_{q-1}(t) = f(t, x_{q-1}(t))$ is continuous in $I_h$, since by assumption $x_{q-1}(t)$ is continuous in $I_h$, and we have

$$|x_q(t) - x_0(t)| \le \left| \int_{t_0}^{t} |f(y, x_{q-1}(y))| \, \mathrm{d}y \right| \le M|t_0 - t| \le Mh \le \delta.$$

In a next step, it is shown that $\{x_q(t)\}$ converges uniformly in $I_h$. For this, since the functions $x_0(t)$ and $x_1(t)$ are continuous in $I_h$, it follows that there exists a constant $N > 0$ such that $|x_0(t) - x_1(t)| \le N$. Furthermore, for all $t \in I_h$ the inequality

$$|x_q(t) - x_{q-1}(t)| \le N \frac{(L|t_0 - t|)^{q-1}}{(q-1)!}, \quad q = 1, 2, \ldots, \quad (6.16)$$

holds, given the following inductive argument. It is obvious that (6.16) holds for $q = 1$, further if it is also true for $q = w \geq 1$, then we have with assertion (ii) and (6.13)

$$\begin{aligned}|x_{w+1}(t) - x_w(t)| &\leq \left| \int_{t_0}^{t} |f(y, x_w(y)) - f(y, x_{w-1}(y))|\, dy \right| \\ &\leq L \left| \int_{t_0}^{t} |x_w(y) - x_{w-1}(y)|\, dy \right| \\ &\leq L \left| \int_{t_0}^{t} N \frac{(L|t_0 - y|)^{w-1}}{(w-1)!}\, dy \right| = N \frac{(L|t_0 - t|)^w}{(w)!}.\end{aligned}$$

Hence, the inequality (6.16) holds for all $q$.
Now, with Theorem 6.14 and the fact that

$$N \sum_{q=1}^{\infty} \frac{(L|t_0 - t|)^{q-1}}{(q-1)!} \leq N \sum_{q=1}^{\infty} \frac{(Lh)^q}{(q)!} = N e^{Lh} < \infty,$$

it follows that the series

$$x_0(t) + \sum_{q=1}^{\infty} (x_q(t) - x_{q-1}(t))$$

converges absolutely and uniformly in $I_h$. Therefore, its partial sums $\{x_q(t)\}$ converge to a continuous function in $I_h$, i.e., $\lim_{q\to\infty} x_q(t) = x(t)$, which is a solution of (6.7).
To show the uniqueness of $x(t)$, we assume that (6.7) has another solution $x'(t)$ in $I_h$ such that $(t, x'(t)) \in S$ for all $t \in I_h$. Then, it follows with (ii) that

$$|x(t) - x'(t)| \leq \left| \int_{t_0}^{t} |f(y, x(y)) - f(y, x'(y))|\, dy \right| \leq L \left| \int_{t_0}^{t} |x(y) - x'(y)|\, dy \right|.$$

However, applying Corollary 6.9 implies that $|x(t) - x'(t)| = 0$ for all $t \in I_h$, and consequently $x(t) = x'(t)$ for all $t \in I_h$.

The error bound (6.15) is then established as follows. For $r > q$, the inequality (6.16) gives

$$\begin{aligned}|x_r(t) - x_q(t)| &\leq \sum_{w=q}^{r-1} |x_{w+1}(t) - x_w(t)| \leq \sum_{w=q}^{r-1} N \frac{(L|t_0 - t|)^w}{w!} \\ &\leq N \sum_{w=q}^{r-1} \frac{(Lh)^w}{w!} = N(Lh)^q \sum_{w=q}^{r-q-1} \frac{(Lh)^w}{(q+w)!}.\end{aligned} \tag{6.17}$$

Nevertheless, as $1/(q+w)! \leq 1/(q!\, w!)$ we have

$$|x_r(t) - x_q(t)| \leq N \frac{(Lh)^q}{q!} \sum_{w=q}^{r-q-1} \frac{(Lh)^w}{(w)!} \leq N \frac{(Lh)^q}{q!} e^{Lh}$$

and with the limit $r \to \infty$, we get

$$|x(t) - x_q(t)| \leq N \frac{(Lh)^q}{q!} e^{Lh}. \tag{6.18}$$

From inequality (6.17), we also obtain

$$|x_r(t) - x_q(t)| \leq N \sum_{w=q}^{r-1} \frac{(Lh)^w}{w!} \leq N e^{Lh}$$

and as $r \to \infty$, we get

$$|x(t) - x_q(t)| \leq N e^{Lh}. \tag{6.19}$$

The error bound (6.15) follows then from (6.18) and (6.19). □

Theorem 6.16 guarantees the local existence of a solution, the global existence is covered with the following theorem.

**Theorem 6.17** (Global Existence Theorem)**.** *Let the following conditions be satisfied:*

*(i)* $f(t, x)$ *is continuous in the strip* $T = \{(t, x) \in D \,|\, |t_0 - t| \leq \Delta, |x| \leq \infty\}$,

*(ii)* $f(t, x)$ *is locally Lipschitz in* $T$ *with Lipschitz constant* $L$,

*(iii)* $x_0(t)$ *is continuous in* $[t_0 - \Delta, t_0 + \Delta]$.

*Then the sequence of functions* $\{x_q(t)\}$*, generated with* (6.13)*, converges uniformly to the unique solution* $x(t)$ *of the initial value problem* (6.3)*. The validity of this solution is ensured on the whole interval* $[t_0 - \Delta, t_0 + \Delta]$.

*Proof.* Since $x_0(t)$ is continuous, each $x_q(t)$ exists and satisfies $|x_q(t)| < \infty$. By replacing $h$ with $\Delta$ in the proof of Theorem 6.16, further recalling that $f(t, x(t))$ is Lipschitz in $T$, the uniform convergence of $\{x_q(t)\}$ to $x(t)$ in $[t_0 - \Delta, t_0 + \Delta]$ is established. □

In most cases an initial value problem (6.3) describes a model based on physical data, which in some cases might not be measured accurately. Therefore, there may be some underlying errors in the function $f(t, x(t))$ as well as the initial condition $(t_0, x_0)$. Note that this can be by choice in order to simplify the given model. Thus, the question arises how the solution of (6.3) behaves when $f(t, x(t))$ and $(t_0, x_0)$ are altered. The following theorem, see, e.g., [4], answers this question.

**Theorem 6.18.** *Let* $(t_0, x_0), (t_1, x_1) \in D$ *be initial conditions of the IVP* (6.3) *and let* $f(t, x(t))$ *and* $g(t, x(t))$ *be functions in* $D$*. Here* $(t_1, x_1)$ *and* $g(t, x(t))$ *are slightly altered. Furthermore, let following conditions be satisfied:*

*(i)* $f(t, x)$ *is continuous and bounded by* $M$ *in* $D$,

*(ii)* $f(t, x)$ *is locally Lipschitz in* $D$ *with Lipschitz constant* $L$,

*(iii)* $g(t, x)$ *is continuous and bounded by* $\hat{M}$ *in* $D$,

*(iv) the solutions $x(t)$ and $y(t)$ of (6.3) and*

$$\dot{y}(t) = f(t, x(t)) + g(t, x(t)), \quad y(t_1) = x_1$$

*exist in an interval $I$ that contains $t_0$ and $t_1$.*

*Then the following inequality*

$$|x(t) - y(t)| \leq \left(|x_0 - x_1| + (M + \hat{M})|t_0 - t_1| + \frac{1}{L}\hat{M}\right) \times \exp(L|t_0 - t|) - \frac{1}{L}\hat{M} \tag{6.20}$$

*holds for all $t \in I$.*

*Proof.* With Theorem 6.7 it follows for all $t \in I$ that

$$\begin{aligned} y(t) &= x_1 + \int_{t_1}^{t} [f(s, y(s)) + g(s, y(s))]\, ds \\ &= x_1 + \int_{t_0}^{t} f(s, y(s))\, ds + \int_{t_1}^{t_0} f(s, y(s))\, ds + \int_{t_1}^{t} g(s, y(s))\, ds \end{aligned} \tag{6.21}$$

and consequently

$$\begin{aligned} x(t) - y(t) = x_0 - x_1 &+ \int_{t_0}^{t} [f(s, x(s)) - f(s, y(s))]\, ds \\ &+ \int_{t_0}^{t_1} f(s, x(s))\, ds - \int_{t_1}^{t} g(s, x(s))\, ds. \end{aligned} \tag{6.22}$$

Using the assumptions and taking absolute values of (6.22), we find

$$\begin{aligned} |x(t) - y(t)| \leq |x_0 - x_1| &+ (M + \hat{M})|t_1 - t_0| + \hat{M}|t - t_0| \\ &+ L \left| \int_{t_0}^{t} |x(s) - y(s)|\, ds \right|. \end{aligned} \tag{6.23}$$

Comparing inequality (6.23) with the inequality considered in Corollary 6.10 yields that they are the same with $c_0 = |x_0 - x_1| + (M + \hat{M})|t_1 - t_0|, c_1 = \hat{M}, c_2 = L$ and $u(t) = |x(t) - y(t)|$, and consequently (6.20) follows. □

Hence, according to inequality (6.20) reasonable small changes in $(t_0, x_0)$ and $f(t, x(t))$ lead to small differences between the solutions $x(t)$ and $y(t)$ in an interval $I$.

### 6.1.2 Systems of First-Order Ordinary Differential Equations

Until now, the existence and uniqueness of solutions of initial value problems with a scalar initial value were discussed. Following [4] these results are extended to a *system of*

*first-order ODEs.* For this purpose, we consider a system of first-order ODEs of the form

$$\begin{aligned} \dot{x}_1(t) &= f_1(t, x_1(t), \dots, x_n(t)) \\ \dot{x}_2(t) &= f_2(t, x_1(t), \dots, x_n(t)) \\ &\vdots \\ \dot{x}_n(t) &= f_n(t, x_1(t), \dots, x_n(t)) \end{aligned} \tag{6.24}$$

From here on it is assumed that the functions $f_1, \dots, f_n$ are continuous in some domain $E \subset \mathbb{R}^{n+1}$.

**Definition 6.19.** *A* solution *of* (6.24) *in an interval $I$ are $n$ functions $x_1, \dots, x_n$ such that*

*(i)* $\dot{x}_1(t), \dots, \dot{x}_n(t)$ *exist for all* $t \in I$,

*(ii)* $(t, x_1, \dots, x_n) \in E$ *for all* $t \in I$,

*(iii)* $\dot{x}_i(t) = f_i(t, x_1(t), \dots, x_n(t))$ *for all* $t \in I$.

As in the scalar case, one can specify initial conditions to the problem (6.24) which are of the form

$$x_i(t_0) = x_0^i \quad \text{for all } i = 1, \dots, n, \tag{6.25}$$

where $t_0 \in I$ is known and $x_0^1, \dots, x_0^n$ are given values such that $(t_0, x_0^1, \dots, x_0^n) \in E$. These conditions combined with the system of ODEs (6.24) form an initial value problem, which we write in a compact vector notation by setting

$$x(t) = (x_1(t), \dots, x_n(t)) \quad \text{and} \quad f(t, x(t)) = (f_1(t, x(t)), \dots, f_n(t, x(t))$$

and establishing that the differentiation and integration operators act componentwise, i.e., $\dot{x}(t) = (\dot{x}_1(t), \dots, \dot{x}_n(t))$ and $\int x(t)\,dt = (\int x_1(t)\,dt, \dots, \int x_n(t)\,dt)$.
Additionally we say the function $f(t, x(t))$ is continuous in $E$, if all of its components are continuous in $E$. The initial value problem is then of the form

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \tag{6.26}$$

which is analogous to the scalar formulation (6.3).

**Definition 6.20.** *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. The function $f(t, x(t))$ is called* Lipschitz continuous *in $x$ in $E$ if there exists a* Lipschitz constant *$L > 0$, such that*

$$\|f(t, x'(t)) - f(t, x''(t))\| \leq L \, \|x'(t) - x''(t)\| \quad \text{for all } (t, x'(t)), (t, x''(t)) \in E. \tag{6.27}$$

*The function $f(t, x(t))$ is called* locally Lipschitz continuous *in $x$ in $E$ if for any $(t, x(t)) \in E$ there exists $\delta, \Delta > 0$ such that*

$$G = [t - \Delta, t + \Delta] \times \bar{B}_\delta(x(t)) \subset E,$$

*and there exists a* Lipschitz constant $L := L(t, x(t)) > 0$, *such that*

$$\|f(t, x'(t)) - f(t, x''(t))\| \leq L \, \|x'(t) - x''(t)\| \quad \textit{for all } (t, x'(t)), (t, x''(t)) \in G. \tag{6.28}$$

If the function $f(t, x(t))$ is continuous in $E$, then solving (6.26) is equivalent to solving the integral equation

$$x(t) = x_0(t) + \int_{t_0}^{t} f(y, x(y)) \, dy. \tag{6.29}$$

This follows from the same arguments as in Theorem 6.7.
As in the scalar case, the Picard method of approximations comes handy to find a solution of (6.29). Let a continuous function $x_0(t)$, where $x_0(t) \equiv x_0$ is a common choice, be the assumed initial approximation of the solution, and define the iterates $x_q(t)$ as

$$x_q(t) = x_0(t) + \int_{t_0}^{t} f(y, x_{q-1}(y)) \, dy. \tag{6.30}$$

It is clear that as before, if in some interval $I$ that contains $t_0$ the sequence of functions $\{x_q(t)\}$ converges uniformly to a continuous function $x(t)$ and $(t, x(t)) \in E$ for all $t \in I$, then $x(t)$ solves (6.29) and consequently (6.26).
Finally, the existence theorems for the initial value problem (6.3) Theorem 6.16 and Theorem 6.17 can easily be extended to systems of ordinary differential equations. The following existence theorems for systems of ODEs are stated without proof since the proofs are similar to the scalar case, for further detail we refer to [4].

**Theorem 6.21** (Local Existence Theorem)**.** *Let the following conditions be satisfied:*

*(i)* $f(t, x)$ *is continuous in* $S = \{(t, x) \in D \,|\, |t_0 - t| \leq \Delta, \, \|x_0 - x\| \leq \delta\}$, *and therefore there exists a* $M > 0$ *such that* $\|f(t, x)\| \leq M$ *for all* $(t, x) \in S$,

*(ii)* $f(t, x)$ *is locally Lipschitz in* $S$ *with Lipschitz constant* $L$,

*(iii)* $x_0(t)$ *is continuous in* $[t_0 - \Delta, t_0 + \Delta]$, *and* $\|x_0 - x_0(t)\| \leq \delta$.

*Then the sequence of functions* $\{x_q(t)\}$, *generated with* (6.30), *converges uniformly to the unique solution* $x(t)$ *of the initial value problem* (6.26). *The validity of this solution is ensured on the interval* $I_h := [t_0 - h, t_0 + h]$, *where* $h = \min(\Delta, \delta/M)$. *Further, the following error estimate holds for all* $t \in I_h$

$$\|x(t) - x_q(t)\| \leq N \, e^{Lh} \min\left(1, \frac{(Lh)^k}{k!}\right), \quad q = 1, 2, 3, \ldots, \tag{6.31}$$

*where* $\|x_1(t) - x_0(t)\| \leq N$.

**Theorem 6.22** (Global Existence Theorem)**.** *Let the following conditions be satisfied:*

*(i)* $f(t, x)$ *is continuous in* $T = \{(t, x) \in D \,|\, |t_0 - t| \leq \Delta, \, \|x\| \leq \infty\}$,

*(ii)* $f(t, x)$ *is locally Lipschitz in* $T$ *with Lipschitz constant* $L$,

*(iii)* $x_0(t)$ *is continuous in* $[t_0 - \Delta, t_0 + \Delta]$.

*Then the sequence of functions* $\{x_q(t)\}$*, generated with* (6.30)*, converges uniformly to the unique solution* $x(t)$ *of the initial value problem* (6.26)*. The validity of this solution is ensured on the whole interval* $[t_0 - \Delta, t_0 + \Delta]$*.*

## 6.2 Pareto Tracing Using ODEs

As already mentioned, the *Pareto tracing by numerical integration* method was published in [19]. Here we provide more details than in [19].
Let $J = (J_0, J_1) : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^2$ be a biobjective objective function. Further, let $J_\lambda = (1-\lambda)J_0 + \lambda J_1$ be the weighted sum scalarization for some weight $\lambda \in (0,1)$, see Section 3.5. For this section, we assume that $J$ is two times differentiable, i.e., $J \in C^2$, and therefore $J_\lambda \in C^2$ for all $\lambda \in (0,1)$. In the following, we denote the gradient of $J_\lambda$ and $J_i$ at $x$ as $\nabla_x J_\lambda(x)$ and $\nabla_x J_i(x)$, $i = 0, 1$, and the Hessian matrix of $J_\lambda$ and $J_i$ at $x$ as $\nabla_x^2 J_\lambda(x)$ and $\nabla_x^2 J_i(x)$, $i = 0, 1$, respectively. Furthermore, recall the optimality conditions from Section 3.5.1 as they are the base for the following discussion.

### 6.2.1 Implicit and Explicit ODEs for Local Pareto Optimality

From here on, we assume that $\nabla_x^2 J_i$ is locally Lipschitz with Lipschitz constant $L_H(x, \delta)$ on the ball $B_\delta(x)$ with radius $\delta > 0$ centered at $x$, i.e.,

$$\|\nabla_x^2 J_i(x) - \nabla_x^2 J_i(x')\| \leq L_H(x, \delta)\|x - x'\|, \quad i = 0, 1,$$

where $\|A\|$ is the spectral norm, i.e., the square root of the maximum eigenvalue of $A^\top A,\ A \in \mathbb{R}^{n \times n}$.
Let us first assume that on some interval $\lambda \in [\lambda_l, \lambda_u] \subseteq (0,1)$ we have attained $J_\lambda$ critical points $x(\lambda)$. Further, suppose that $x(\lambda)$ is differentiable w.r.t. $\lambda$. Differentiating the first-order optimality conditions $\nabla_x J_\lambda(x(\lambda)) = 0$ with respect to $\lambda$ yields

$$\nabla_x^2 J_\lambda(x(\lambda))\, \dot{x}(\lambda) = \nabla_x J_0(x(\lambda)) - \nabla_x J_1(x(\lambda)). \tag{6.32}$$

Note that the differentiability of $x(\lambda)$ is assured by the Implicit Function Theorem (6.15). This implicit ODE can be rearranged to a explicit ODE $\dot{x}(\lambda) = f(\lambda, x(\lambda))$, if $x(\lambda)$ additionally satisfies the second-order optimality conditions with respect to $J_\lambda$ strictly. Hence, $f$ is then given by

$$\dot{x}(\lambda) = \left(\nabla_x^2 J_\lambda(x(\lambda))\right)^{-1} \left(\nabla_x J_0(x(\lambda)) - \nabla_x J_1(x(\lambda))\right) = f(\lambda, x(\lambda)). \tag{6.33}$$

Now, let us on the contrary assume that we have attained a point $x_0$ which fulfills the strict second-order optimality condition for $J_{\lambda_0}$, i.e., for some $\lambda_0 \in (0,1)$. If the right hand side of (6.33) is locally Lipschitz in $x$ on some open neighborhood $\mathcal{U}$ of $x_0$ with a Lipschitz constant $L_f$ that is uniform in $\lambda$ on some interval $[\lambda_l, \lambda_u] \subseteq (0,1)$, the conditions of the local existence theorem Theorem 6.16 are satisfied.

**Lemma 6.23.** *Let* $A_1,\ A_2 \in \mathbb{R}^{n \times n}$ *be two strictly positive definite matrices with smallest*

*eigenvalue not smaller than some $\varepsilon > 0$. Then, we have*

$$\|A_1^{-1} - A_2^{-1}\| \leq \frac{1}{\varepsilon^2}\|A_1 - A_2\|. \tag{6.34}$$

*Proof.* Let $A^\kappa = \kappa A_1 + (1-\kappa)A_2$ for $\kappa \in (0,1)$. Then, $A^\kappa$ is also positive definite with smallest eigenvalue not smaller than $\varepsilon$. The assertion follows directly from the sub multiplicativity of the spectral norm:

$$\begin{aligned}
\|A_1^{-1} - A_2^{-1}\| &= \left\| \int_0^1 \frac{d}{d\kappa}(A^\kappa)^{-1}\,\mathrm{d}\kappa \right\| \\
&= \left\| \int_0^1 (A^\kappa)^{-1}\frac{d}{d\kappa}A^\kappa(A^\kappa)^{-1}\,\mathrm{d}\kappa \right\| \\
&= \left\| \int_0^1 (A^\kappa)^{-1}(A_1 - A_2)(A^\kappa)^{-1}\,\mathrm{d}\kappa \right\| \\
&\leq \int_0^1 \left\|(A^\kappa)^{-1}(A_1 - A_2)(A^\kappa)^{-1}\right\|\,\mathrm{d}\kappa \\
&\leq \frac{1}{\varepsilon^2}\|A_1 - A_2\|.
\end{aligned} \tag{6.35}$$

□

Thus, we can establish the following bound for the Lipschitz constant $L_f$ of the right hand side of (6.33).

**Lemma 6.24** (c.f. [19], Lemma 1)**.** *Let $\lambda \in (0,1)$ and for a solution $x$ that fulfills the second-order optimality conditions w.r.t. $J_\lambda$, let $\Lambda(\lambda, x)$ be the smallest eigenvalue of the Hessian $\nabla_x^2 J_\lambda(x)$:*

*(i) $\Lambda(\lambda, x)$ is locally Lipschitz in $x$ with Lipschitz constant $L_H(x,\delta)$ on $B_\delta(x)$.*

*(ii) $\Lambda(\lambda, x)$ is Lipschitz in $\lambda$ on $(0,1)$ with Lipschitz constant $L_\lambda(x) = \|\nabla_x^2 J_0(x)\| + \|\nabla_x^2 J_1(x)\|$.*

*(iii) Let $1 > \varrho > 0$, then for $\lambda' \in (0,1)$ and $x' \in B_\delta(x)$ such that*

$$L_H(x,\delta)\|x - x'\| + L_\lambda(x)|\lambda - \lambda'| \leq (1-\varrho)\Lambda(\lambda, x),$$

*it is $\Lambda(\lambda', x') \geq \varrho\Lambda(\lambda, x)$.*

*(iv) Let the interval $[\lambda_l, \lambda_u]$ containing $\lambda$ and $B_\delta(x)$ be given such that $L_H(x,\delta)\delta + L_\lambda(x)\max\{\lambda_u - \lambda, \lambda - \lambda_l\} \leq (1-\varrho)\Lambda(\lambda, x)$. This can always be achieved, as $L_H(x,\delta)$ is monotonically decreasing in $\delta$. Then, $f(\lambda', x')$ is uniformly (in $\lambda'$) Lipschitz in $x'$ on $[\lambda_l, \lambda_u] \times B_\delta(x)$ and its Lipschitz constant is bounded by*

$$L_f(x, \delta, \varrho) = 2\left(\left(\frac{1}{\varrho\,\Lambda(\lambda, x)}\right)C_2(x,\delta) + \left(\frac{1}{\varrho\,\Lambda(\lambda, x)}\right)^2 L_H(x,\delta)C_1(x,\delta)\right),$$

*where $C_2(x,\delta) = \max\limits_{i\in\{0,1\}} \sup\limits_{x'\in B_\delta(x)} \|\nabla_x^2 J_i(x')\|$ and $C_1(x,\delta) = \max\limits_{i\in\{0,1\}} \sup\limits_{x'\in B_\delta(x)} \|\nabla_x J_i(x')\|$.*

*Proof.*

(i) Let $x', x'' \in B_\delta(x)$. Without loss of generality we assume that $\Lambda(\lambda, x') \geq \Lambda(\lambda, x'')$. Then

$$\begin{aligned}
0 &< \Lambda(\lambda, x') - \Lambda(\lambda, x'') \\
&= \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \nabla_x^2 J_\lambda(x') v - \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \nabla_x^2 J_\lambda(x'') v \\
&\leq \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \left(\nabla_x^2 J_\lambda(x') - \nabla_x^2 J_\lambda(x'')\right) v \\
&\leq \sup_{v\in\mathbb{R}^n:\|v\|=1} v^\top \left(\nabla_x^2 J_\lambda(x') - \nabla_x^2 J_\lambda(x'')\right) v \\
&= \left\|\nabla_x^2 J_\lambda(x') - \nabla_x^2 J_\lambda(x'')\right\| \\
&\leq L_H(x,\delta)\ \|x' - x''\|\,.
\end{aligned}$$

(ii) Let $\lambda', \lambda'' \in (0,1)$ with $\Lambda(\lambda', x) \geq \Lambda(\lambda'', x)$ similar to (i), then

$$\begin{aligned}
0 &< \Lambda(\lambda', x) - \Lambda(\lambda'', x) \\
&= \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \nabla_x^2 J_{\lambda'}(x) v - \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \nabla_x^2 J_{\lambda''}(x) v) \\
&\leq \inf_{v\in\mathbb{R}^n:\|v\|=1} v^\top \left(\nabla_x^2 J_{\lambda'}(x) - \nabla_x^2 J_{\lambda''}(x)\right) v \\
&\leq \sup_{v\in\mathbb{R}^n:\|v\|=1} v^\top \left(\nabla_x^2 J_{\lambda'}(x) - \nabla_x^2 J_{\lambda''}(x)\right) v \\
&= \left\|\nabla_x^2 J_{\lambda'}(x) - \nabla_x^2 J_{\lambda''}(x)\right\| \\
&= \left\|(\lambda'' - \lambda')\nabla_x^2 J_0(x) + (\lambda' - \lambda'')\nabla_x^2 J_1(x)\right\| \\
&\leq \left(\|\nabla_x^2 J_0(x)\| + \|\nabla_x^2 J_1(x)\|\right) |\lambda' - \lambda''|\,.
\end{aligned}$$

(iii) For $x' \in B_\delta(x)$, (iii) now follows from (i) and (ii) by

$$\begin{aligned}
\Lambda(\lambda', x') &= \Lambda(\lambda, x) + (\Lambda(\lambda', x) - \Lambda(\lambda, x)) + (\Lambda(\lambda', x') - \Lambda(\lambda', x)) \\
&\geq \Lambda(\lambda, x) - L_H(x,\delta)\,\|x - x'\| - L_\lambda(x)\,|\lambda - \lambda'| \\
&\geq \varrho\Lambda(\lambda, x).
\end{aligned}$$

(iv) Hence, with (iii) and (6.34) with $\varepsilon = \rho\Lambda(\lambda, x)$ it follows for $x', x'' \in B_\delta(x)$,

$\lambda' \in [\lambda_l, \lambda_u]$ and $K := \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} \left(\nabla_x J_0(x'') - \nabla_x J_1(x'')\right)$ that

$$
\begin{aligned}
0 &\leq \|f(\lambda', x') - f(\lambda', x'')\| \\
&= \Big\| \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} \left(\nabla_x J_0(x') - \nabla_x J_1(x')\right) - K \\
&\quad - \left(\nabla_x^2 J_{\lambda'}(x'')\right)^{-1} \left(\nabla_x J_0(x'') - \nabla_x J_1(x'')\right) + K \Big\| \\
&\leq \Big\| \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} \left(\nabla_x J_0(x') - \nabla_x J_0(x'') - \nabla_x J_1(x') + \nabla_x J_1(x'')\right) \Big\| \\
&\quad + \Big\| \left( \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} - \left(\nabla_x^2 J_{\lambda'}(x'')\right)^{-1} \right) \left(\nabla_x J_0(x'') - \nabla_x J_1(x'')\right) \Big\| \\
&\leq \left\| \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} \left( \int_{x''}^{x'} \nabla_x^2 J_0(y)\, dy - \int_{x''}^{x'} \nabla_x^2 J_1(y)\, dy \right) \right\| \\
&\quad + \left\| \left( \left(\nabla_x^2 J_{\lambda'}(x')\right)^{-1} - \left(\nabla_x^2 J_{\lambda'}(x'')\right)^{-1} \right) \right\| \times \|(\nabla_x J_0(x'') - \nabla_x J_1(x''))\| \\
&\leq \frac{1}{\varrho \Lambda(\lambda, x)} \left( \sup_{x^* \in B_\delta(x)} \|\nabla_x^2 J_0(x^*)\| + \sup_{x^* \in B_\delta(x)} \|\nabla_x^2 J_1(x^*)\| \right) \|x' - x''\| \\
&\quad + \left( \frac{1}{\varrho \Lambda(\lambda, x)} \right)^2 \left\| \nabla_x^2 J_{\lambda'}(x') - \nabla_x^2 J_{\lambda'}(x'') \right\| \\
&\quad \times \sup_{x^* \in B_\delta(x)} \|\nabla_x J_0(x^*) - \nabla_x J_1(x^*)\| \\
&\leq 2 \left( \frac{1}{\varrho \Lambda(\lambda, x)} C_2(x, \delta) + \left( \frac{1}{\varrho \Lambda(\lambda, x)} \right)^2 L_H(x, \delta) C_1(x, \delta) \right) \|x' - x''\|.
\end{aligned}
$$

□

With these results we can establish the existence of solutions of (6.33) with the following theorem:

**Theorem 6.25** (c.f. [19], Theorem 2)**.** *Let $x_0 \in \mathbb{R}^n$ and $\lambda_0 \in (0,1)$ given such that $x_0$ fulfills the strict second-order optimality conditions with respect to $J_{\lambda_0}$, see Definition 3.19. Further, using the notation of Lemma 6.24, let $\Delta, \delta > 0$ and $1 > \varrho > 0$ such that $L_H(x_0, \delta)\delta + L_\lambda(x_0)\Delta \leq (1-\varrho)\Lambda(\lambda_0, x_0)$. Let $\tilde{C}(x_0, \delta, \Delta) = \sup_{\substack{\lambda \in [\lambda_0 - \Delta, \lambda_0 + \Delta] \\ x \in B_\delta(x_0)}} \|f(\lambda, x)\|$. Let also $\Delta' = \min\{\Delta, \delta / \tilde{C}(x_0, \delta, \Delta)\}$ and $\lambda_l = \lambda_0 - \Delta'$, $\lambda_u = \lambda_0 - \Delta'$. Then,*

- *(i) the solution $x(\lambda)$ of (6.33) with initial value $x(\lambda_0) = x_0$ at $\lambda_0$ exists and is unique locally on the interval $[\lambda_l, \lambda_u]$. Further, $x(\lambda)$ is continuously differentiable on $[\lambda_l, \lambda_u]$;*
- *(ii) one can extend $x(\lambda)$ to a solution of (6.33) to a maximal time interval $(\lambda_l', \lambda_u') \subseteq (0,1)$ containing $\lambda_0$ such that $\Lambda(\lambda, x(\lambda)) > 0$ for $\lambda \in (\lambda_l', \lambda_u')$ and either $\lambda_l = 0$ ($\lambda_u = 1$) or $\Lambda(\lambda, x(\lambda))$ has accumulation point 0 as $\lambda \searrow \lambda_l'$ ($\lambda \nearrow \lambda_u'$);*
- *(iii) $x(\lambda)$ satisfies the strict second-order optimality conditions with respect to $J_\lambda$ and thus is locally $J_\lambda$ optimal and locally Pareto-optimal with respect to $J$ on the interval $(\lambda_l', \lambda_u')$.*

*Proof.*

(i) We have shown that the conditions of Theorem 6.21 are fulfilled by Lemma 6.24 (iv). Hence, the assertion follows.

(ii) Since we have $\Lambda(\lambda_u, x(\lambda_u)) \geq \rho\Lambda(\lambda_0, x(\lambda_0))$, the statement of (i) can be iterated by replacing $\lambda_0$ with $\lambda_l = \lambda_0^{(1)}$ and $x_0$ with $x(\lambda_l)$. This can be done, until either $\lambda_0^{(n)}$ is reaching one or $\Lambda(\lambda_0^{(n)}, x(\lambda_0^{(n)}))$ is approaching 0. Now one can define the minimal lower boundary as $\lambda_l' = \lim_{n\to\infty} \lambda_0^{(n)}$. For the maximal upper bound $\lambda_u'$ one can argue analogously.

(iii) We recall that for any $\lambda \in (\lambda_l', \lambda_u')$, (6.33) implies (6.32), i.e., $\nabla_x J_\lambda(x(\lambda)) = 0$, and thus

$$\nabla_x J_\lambda(x(\lambda)) = \nabla_x J_{\lambda_0}(x_0) + \int_{\lambda_0}^{\lambda} \frac{d}{d\tau} \nabla_x J^\tau(x(\tau))\, \mathrm{d}\tau = 0, \tag{6.36}$$

therefore, $x(\lambda)$ is $J_\lambda$ critical and consequently Pareto critical. Further, since in (ii), it was established that $\Lambda(\lambda, x) > 0$ holds for $\lambda \in (\lambda_l', \lambda_u')$, it follows that $\nabla_x^2 J_\lambda(x(\lambda))$ is strictly positive definite. Hence, $x(\lambda)$ satisfies strict second-order optimality for $J_\lambda$ and is locally Pareto-optimal.

□

**Remark 6.26.** *(i) All results of this section consider all $x \in \mathbb{R}^n$ as feasible solutions simplifying the notation, i.e., an unrestricted domain is considered. Nevertheless, obvious adaptations enable one to extend the results of this section to the general case, where $J_i(x)$, $i \in \{0, 1\}$ is only defined on an open subset of $\mathbb{R}^n$. Toward this end, one has to choose the constants $\delta > 0$ of the local constructions in Lemma 6.24 and Theorem 6.25 smaller than the distance to the boundary of the feasible domain and adjust the maximal intervals of existence $(\lambda_l', \lambda_u')$.*
*(ii) If the two objective functions are equal, i.e., if $J_1 = J_0$, and therefore $J_\lambda = J_1 = J_0$ for all $\lambda \in (0, 1)$, then any existing optimal solution of $J_1$ is Pareto-optimal. In this case, the nondominated set consists of exactly one outcome vector and consequently* (6.33) *becomes $\dot{x}(\lambda) = 0$, coinciding with the fact that there only exists one unique outcome vector.*

Since estimating the quality of the numerical approximation of $x(\lambda)$ often relies on the regularity of $x(\lambda)$, we recall the following well-known result on the regularity of solutions to ODEs, see also [19].

**Lemma 6.27** (c.f. [19], Lemma 4)**.** *Assume that $J_i$, $i \in \{0, 1\}$, is $p+2$ times differentiable with locally bounded $p+2$nd derivative, $p \in \mathbb{N}_0$. Let $[\lambda_l, \lambda_u] \subset (\lambda_l', \lambda_u')$ be a closed interval in the maximal interval from Theorem 6.25(ii). Then $x(\lambda)$ is $p+1$ times differentiable with bounded $p+1$st derivative on $[\lambda_l, \lambda_u]$.*

*Proof.* First we note that inverting an invertible matrix $A$ as an operation is $C^\infty$ on a neighborhood of $A$. Next, by redefining the right hand side of (6.33) as $f(\lambda, x) = f^{(0)}(\lambda, x)$, it becomes obvious that $f^{(0)}$ is $p$ times differentiable in $\lambda$ and $x$. For $l = 1, \ldots, p$, we recursively define $f^{(l)}(\lambda, x) = \frac{\partial}{\partial\lambda} f^{(l-1)}(\lambda, x) + \nabla_x f^{(l-1)}(\lambda, x)^\top f(\lambda, x)$, where $f^{(l)}$ is $p-l$ times differentiable in $x$ and $\lambda$ and locally bounded, where $p = l$. Now, differentiating $x^{(l)}(\lambda) = \left(\frac{d}{d\lambda}\right)^l x(\lambda) = f^{(l-1)}(\lambda, x(\lambda))$ with respect to $\lambda$, yields that $x^{(l+1)}(\lambda) = f^{(l)}(\lambda, x(\lambda))$ for $l = 0, \ldots, p$ exists and is bounded on $[\lambda_l, \lambda_u]$ if $l = p$. □

### 6.2.2 Approximately Pareto Critical Initial Conditions and Numerical Stability

Until now we have assumed that the initial value $x_0$ is the (local) optimum to the single-criteria optimization problem given by the objective function $J_{\lambda_0}$, i.e., $x_0$ satisfies the strict second-order optimality conditions for $J_{\lambda_0}$. In general, we do not know $x_0$, especially when dealing with applications like the shape optimization problem (3.11) that we investigate in this work. However, one can use approximations of $x_0$, which can be obtained by, e.g., the gradient descend methods proposed in Chapter 5, as initial values. In this subsection, we define approximates of $x_0$ that for example can be obtained by a gradient descent method or Newton-type method applied to $J_{\lambda_0}$.

**Definition 6.28.** *Let $x_{0,k}$ be the iterates of some optimization algorithm started sufficiently close to $x_0$. Further assume that under these circumstances the optimization problem is convex and convergence is guaranteed, i.e., $x_{0,k} \to x_0$, $k \to \infty$.*

We may further assume that $x_{0,k}$ is sufficiently close to $x_0$ such that also $\nabla_x^2 J_{\lambda_0}(x_{0,k})$ is strictly positive definite. The terminal output $x_{0,k}$ of the optimization algorithm is $\varepsilon$-$J_{\lambda_0}$ critical, assuming a gradient based stopping criterion was utilized, e.g., $\|\nabla_x J_{\lambda_0}(x_{0,k})\| \leq \varepsilon$ for some $\varepsilon > 0$.

**Definition 6.29.** *Let $k > 0$ and let $x_{0,k}$ be the $k$-th iterate of some optimization algorithm as described as in Definition 6.28. Let further $\nabla_x^2 J_{\lambda_0}(x_{0,k})$ be strictly positive definite. Then, starting the ODE* (6.33) *in the approximative initial value $x_{0,k}$ yields the following ODE*

$$\dot{x}_k(\lambda) = f(\lambda, x_k(\lambda)), \tag{6.37}$$

*where $f$ is defined as in* (6.33).

In the following, an error bound for $x_k(\lambda)$ when starting the ODE (6.33) in a $\varepsilon$-$J_{\lambda_0}$ critical initial solution $x_{0,k}$ is given.
Additionally to the approximations of initial solution $x_{0,k}$, many applications further rely on numerical approximations of the function $f(\lambda, x(\lambda))$.

**Definition 6.30.** *Let $f_l(\lambda, x(\lambda))$ be an approximation of the function $f(\lambda, x(\lambda))$ that has limited accuracy, such that the numerical error is controlled by the parameter $l$ in the sense that $\varepsilon_l(\mathcal{C}, \lambda) = \sup_{x \in \mathcal{C}} \|f(\lambda, x) - f_l(\lambda, x)\| \to 0$ if $l \to \infty$ and $\mathcal{C} \subseteq \mathbb{R}^n$ is compact.*

The following proposition provides estimates to control the effect of the error caused by the error in the initial condition $x_0 - x_{0,k}$ on the solution of (6.33) as well as the numerical error in $f - f_l$. Furthermore, using $x_{0,k}$ as the initial value of ODE (6.33) provides $\varepsilon$-critical solutions $x_k(\lambda)$ with respect to $J_\lambda$ for $\lambda$ in some interval containing $\lambda_0$.

**Proposition 6.31** (c.f. [19], Proposition 5)**.** *Let $x_0$ fulfill the strict second-order optimality condition with respect to $J_{\lambda_0}$ and $x_{0,k} \to x_0$ as $k \to \infty$:*

*(i) Let $\varepsilon > 0$. For $k$ sufficiently large, solutions $x_k(\lambda)$ to* (6.33) *started with initial condition $x_k(\lambda_0) = x_{0,k}$ at $\lambda_0$ exist on some maximal intervals $(\lambda'_{l,k}, \lambda'_{u,k}) \subset (0, 1)$ and $x_k(\lambda)$ is $J_\lambda$ $\varepsilon$-critical, hence, $\varepsilon' = \frac{\varepsilon}{\min\{\lambda, (1-\lambda)\}}$-Pareto critical for $\lambda \in (\lambda'_{l,k}, \lambda'_{u,k})$.*

*(ii)* *Let $I = [\lambda_l, \lambda_u] \subset (\lambda_l', \lambda_u')$ a compact subinterval of the maximal interval from (i), where it is established that $x(\lambda)$ exist. Let $\Lambda'(I) = \inf_{\lambda\in I} \Lambda(\lambda, x(\lambda))$, $L_H(I,\delta) = \sup_{\lambda\in I} L_H(x(\lambda),\delta)$, and $C_i(\delta, I) = \sup_{\lambda\in I} C_i(x(\lambda),\delta)$, $i \in \{1,2\}$, where $C_i(x(\lambda),\delta)$, $i \in \{1,2\}$, are defined as in Lemma 6.24(iv). Further, let $0 < \delta < \Lambda'(I)/L_H(I,\delta)$, which is always possible, since $L_H(I,\delta)$ is finite and monotonically increasing in $\delta$. Additionally we set*

$$L_f(\delta, I) = 2\left(\frac{C_2(\delta,I)}{\Lambda'(I) - \delta L_H(I,\delta)} + \left(\frac{1}{\Lambda'(I) - \delta L_H(I,\delta)}\right)^2 L_H(I,\delta) C_1(\delta, I)\right).$$

*Then, for $k$ sufficiently large, $x_k(\lambda)$ exists for $\lambda \in [\lambda_l, \lambda_u]$ and*

$$\|x(\cdot) - x_k(\cdot)\|_{C(I,\mathbb{R}^n)} \leq \|x_0 - x_{0,k}\|\, e^{L_f(\delta,I)\max\{\lambda_0-\lambda_l, \lambda_u-\lambda_0\}}, \tag{6.38}$$

*where $\|\cdot\|_{C(I,\mathbb{R}^n)}$ is the maximum norm on $I$. Therefore, $x_k(\lambda)$ converges with the same rate to the locally $J_\lambda$ and locally Pareto optimal point $x(\lambda)$ as $x_{0,k}$ converges to the locally $J_{\lambda_0}$ optimal and locally Pareto optimal point $x_0$.*

*(iii)* *Additionally, let $f_l$ be a locally Lipschitz function such that $f - f_l \to 0$, as $l \to \infty$ uniformly on compact sets. Let further $\delta$ as in (ii) and $k, l$ sufficiently large. Then, the solution $x_{k;l}(\lambda)$ of $\dot{x}_{k;l}(\lambda) = f_l(\lambda, x_{k;l}(\lambda))$ with initial value $x_{k;l}(\lambda_0) = x_{0,k}$ at $\lambda_0$ exists on $I$, and the following estimate holds*

$$\begin{aligned}\|x(\cdot) - x_{k;l}(\cdot)\|_{C(I,\mathbb{R}^n)} \leq{}& \|x_0 - x_{0,k}\|\, e^{L_f(\delta,I)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}} \\ &+ \frac{1}{L_f(\delta,I)}\left(e^{L_f(\delta,I)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}} - 1\right)\|f_l - f\|_{C(\overline{\mathcal{U}(I,\delta)},\mathbb{R}^n)},\end{aligned} \tag{6.39}$$

*where $\mathcal{U}(I,\delta) = \bigcup_{\lambda\in I} B_\delta(x(\lambda))$ and $\|\cdot\|_{C(\overline{\mathcal{U}(I,\delta)},\mathbb{R}^n)}$ denotes the maximum norm on $\overline{\mathcal{U}(I,\delta)}$.*

*Proof.*

(i) For sufficiently large $k$ for which $\delta = \|x_{0,k} - x_0\|$ satisfies $\delta L_H(x,\delta) < \Lambda(\lambda_0, x)$ and $\Lambda(\lambda_0, x_{0,k}) > 0$, by repeating the proof of Theorem 6.25 (ii) one obtains that $x_k(\lambda)$ exists for some maximal interval $(\lambda_{l,k}', \lambda_{u,k}')$. Furthermore, if $k$ is sufficiently large $x_{0,k}$ is $\varepsilon$-critical for $J_{\lambda_0}$, since $J_{\lambda_0}(x)$ is continuous in $x$. Furthermore, integrating as in (6.36) yields

$$\nabla_x J_\lambda(x_k(\lambda)) = \nabla_x J_{\lambda_0}(x_{0,k}).$$

Hence, for sufficiently large $k$ the solution $x_k(\lambda)$ is then $\varepsilon$-critical for $J_\lambda$ for $\lambda \in I$. The $\varepsilon'$-Pareto criticality of $x_k(\lambda)$ is then established as in Remark 3.17.

(ii) Let now $I = [\lambda_l, \lambda_u] \subseteq (\lambda_l', \lambda_u')$ be some closed interval and let $\delta > 0$ be sufficiently small such that $0 < \delta < \Lambda'(I)/L_H(I,\delta)$. Further, let $k$ be sufficiently large such that $\|x_0 - x_{0,k}\| < \delta e^{-L_f(\delta,I)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}}$, which implies $x_{0,k} \in B_\delta(x_0) \subseteq \mathcal{U}(I,\delta) = \bigcup_{\lambda\in I} B_\delta(x(\lambda))$. Further, $L_H(I,\delta)$ gives an upper bound for the uniform Lipschitz constant of $f$ on $\mathcal{U}(I,\delta)$ by Lemma 6.24 (iv) with the choice $\varrho = 1 - \delta L_H(I,\delta)/\Lambda'(I)$. Thus, $x_k(\lambda)$ exists on some interval $I_n = [\lambda_{l,k}, \lambda_{u.k}] \subseteq I$ that contains $\lambda_0$. Now,

applying Theorem 6.18 gives the following estimate on the continuous dependence on the initial condition

$$\|x(\lambda) - x_k(\lambda)\| \leq \|x_0 - x_{0,k}\| \, e^{L_f(I,\delta)|\lambda-\lambda_0|} \leq \|x_0 - x_{0,k}\| \, e^{L_f(I,\delta)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}} < \delta, \tag{6.40}$$

for $\lambda \in I_n$. Hence, $x_k(\lambda) \in \mathcal{U}(I,\delta)$ and one can extend $x_k(\cdot)$ beyond $I_n$. Applying the above estimate repeatedly yields that $I = [\lambda_l, \lambda_u]$ is contained in the maximal interval of existence $I_n' = (\lambda_{l,n}', \lambda_{u,n}')$ for $x_k(\cdot)$ since for $\lambda \in I$, $x_k(\lambda)$ stays in $\mathcal{U}(I,\delta)$, and the inequality (6.40) holds for all $\lambda \in I$, proving the proposition's second assertion.

(iii) This case is also covered by Theorem 6.18 by essentially the same arguments as in (ii). Toward this end, let, as in (ii), $I = [\lambda_l, \lambda_u] \subseteq (\lambda_l', \lambda_u')$ be some closed interval and let $\delta > 0$ be sufficiently small such that $0 < \delta < \Lambda'(I)/L_H(I,\delta)$. Further, let $k, l$ be sufficiently large such that $\|x_0 - x_{0,k;l}\| < \delta e^{-L_f(\delta,I)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}}$, which implies $x_{0,k;l} \in B_\delta(x_0) \subseteq \mathcal{U}(I,\delta) = \bigcup_{\lambda\in I} B_\delta(x(\lambda))$. We also have $L_H(I,\delta)$ as an upper bound for the uniform Lipschitz constant of $f$ on $\mathcal{U}(I,\delta)$, and consequently $x_{k;l}(\lambda)$ exists on some interval $I_n = [\lambda_{l,k}, \lambda_{u.k}] \subseteq I$ that contains $\lambda_0$. Now, with $g = f_l - f$, i.e., $g$ is bounded by $\|f_l - f\|_{C(\overline{\mathcal{U}(I,\delta)},\mathbb{R}^n)}$, Theorem 6.18 yields the following estimate

$$\begin{aligned}
\|x(\lambda) - x_{k;l}(\lambda)\| &\leq \|x_0 - x_{0,k}\| \, e^{L_f(I,\delta)|\lambda-\lambda_0|} \\
&\quad + \frac{1}{L_f(\delta,I)} \left(e^{L_f(\delta,I)|\lambda-\lambda_0|} - 1\right) \|f_l - f\|_{C(\overline{\mathcal{U}(I,\delta)},\mathbb{R}^n)} \\
&\leq \|x_0 - x_{0,k}\| \, e^{L_f(I,\delta)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}} \\
&\quad + \frac{1}{L_f(\delta,I)} \left(e^{L_f(\delta,I)\max\{\lambda_0-\lambda_l,\lambda_u-\lambda_0\}} - 1\right) \|f_l - f\|_{C(\overline{\mathcal{U}(I,\delta)},\mathbb{R}^n)} \\
&< \delta
\end{aligned} \tag{6.41}$$

for $\lambda \in I_n$. Furthermore, with the same arguments as in (ii) the validity of this estimate can also be extended to the whole interval $I$.

□

## 6.3 Pareto Front Tracing by Numerical Integration

Until now only the existence and the uniqueness of solutions of ODEs were discussed. In the following, the *Runge-Kutta* methods, see, e.g., [81], to solve IVPs (6.3) are introduced and based on them an algorithm to compute a solution for (6.33) is formulated.
Now recall from (6.37), that to compute an approximation $x_k$ of the Pareto front, the following ODE has to be solved:

$$\dot{x}_k(\lambda) = f(\lambda, x_k(\lambda)).$$

Next, an initial value $x_{0,k} = x(\lambda_0)$ at $\lambda_0$ is needed for a numerical integration approach to approximate (6.37). An obvious choice for $\lambda_0$ would be $\lambda_0 = 0$ or $\lambda_0 = 1$, following the ($\varepsilon$-)Pareto critical points forward or backward, respectively. One could also start at

a compromise solution w.r.t., e.g., $\lambda_0 = 0.5$. Note that the problem might not be well-posed for $\lambda_0 = 0$ or $\lambda_0 = 1$, in which case a starting point w.r.t. a compromise solution is chosen. Solving two independent initial value problems separately, one can then recover the Pareto critical points in two directions at the same time. In the following, we denote the $i$th iterate of a numerical integration method by $x_{i,k}$.
The explicit Euler method is the most straightforward method for solving (6.37) numerically. It approximates the derivative of $x_k$ by

$$\dot{x}_k(\lambda) \approx \frac{x_k(\lambda + h) - x_k(\lambda)}{h},$$

which leads to

$$x_k(\lambda + h) \approx x_{1,k} := x_k(\lambda) + h \left(\nabla_x^2 J_\lambda(x_k(\lambda))\right)^{-1} \left(\nabla_x J_0(x_k(\lambda)) - \nabla_x J_1(x_k(\lambda))\right),$$

where $h > 0$ denotes the step size of the method. The global error of the Euler method is then given by $Ch$, where $C > 0$ is depending on the problem [26]. For problems of higher order, Runge-Kutta methods [10, 128, 19] can be used.

**Definition 6.32** (Explicit Runge-Kutta method)**.**
*Let $s \in \mathbb{N}$, $h > 0$, and let $a_{2,1}, a_{3,1}, a_{3,2}, \ldots, a_{s,1}, a_{s,2}, \ldots, a_{s,s-1}, b_1, \ldots, b_s, c_2, \ldots, c_s \in \mathbb{R}$. Then, the method*

$$\begin{aligned}
k_1 &= f(\lambda_0, x_{0,k}) \\
k_2 &= f(\lambda_0 + c_2 h, x_{0,k} + h a_{2,1} k_1) \\
k_3 &= f(\lambda_0 + c_3 h, x_{0,k} + h(a_{3,1} k_1 + a_{3,2} k_2)) \\
&\vdots \\
k_s &= f(\lambda_0 + c_s h, x_{0,k} + h(a_{s,1} k_1 + \cdots + a_{s,s-1} k_{s-1})) \\
x_{1,k} &= x_{0,k} + h(b_1 k_1 + \cdots + b_s k_s)
\end{aligned} \tag{6.42}$$

*is called an* s-stage explicit Runge-Kutta method *for* (6.37)*.*

**Definition 6.33** (c.f. [81], Definition II.1.2)**.** *A Runge-Kutta method* (6.42) *is of* order *$p$ if for sufficiently smooth problems* (6.32)*, there exists a constant $K > 0$ that does not depend on $h$ such that*

$$\|x_k(\lambda_0 + h) - x_{1,k}\| \leq K h^{p+1}.$$

The method described in (6.42) can also be symbolized in a more compact way with the so called *Butcher tableau*, see, e.g., [81, 26].

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s,1} & a_{s,2} & \ldots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \ldots & b_{s-1} & b_s
\end{array} \tag{6.43}$$

Furthermore, the following error estimate holds true.

**Theorem 6.34** (c.f. [81], Theorem II.3.1)**.** *If the Runge-Kutta method* (6.42) *has order* $p$ *and if* $f(\lambda, x_k(\lambda))$ *is* $p$ *times continuously differentiable, then we have the following estimate for the local error of* (6.42)

$$\|x_k(\lambda_0+h)-x_{1,k}\| \leq h^{p+1}\left(\frac{1}{(p+1)!}\max_{t\in(0,1)}\|x_k^{(p+1)}(\lambda_0+th)\| + \frac{1}{p!}\sum_{i=1}^{s}|b_i|\max_{t\in(0,1)}\|k_i^{(p)}(th)\|\right).$$

*Proof.* We write

$$x_k(\lambda_0+h) - x_{1,k} = x_k(\lambda_0+h) - x_{0,k} - h\sum_{i=1}^{s} b_i k_i(h). \tag{6.44}$$

Further, with $0 < \theta, \theta_i < 1$, $i = 1, \ldots, s$, we have the following Taylor expansions

$$\begin{aligned} x_k(\lambda_0+h) &= x_{0,k} + \dot{x}_k(\lambda_0)h + \ddot{x}_k(\lambda_0)\frac{h^2}{2!} + \cdots + x_k^{(p+1)}(\lambda_0+\theta h)\frac{h^{p+1}}{(p+1)!} \\ k_i(h) &= k_i(0) + \dot{k}_i(0)h + \cdots + k_i^{(p)}(\theta_i h)\frac{h^p}{p!} \quad i = 1, \ldots, s, \end{aligned}$$

where the formula is valid componentwise (for possibly distinct $\theta$'s). The assertion then follows from the order conditions.

□

Hence, a Runge-Kutta method of order $p$ can be applied on $f$, as given in (6.33), if it is continuously differentiable $p$ times. To this end, the objective functions $J_0$ and $J_1$ have to be $(p+2)$ times continuously differentiable. Since Theorem 6.34 holds for each step $j$ of the Runge-Kutta method, we can formulate the following estimates for $j = 1, \ldots, N$, where $N$ is the number of integration points, and using $x_{j-1,k}$ as the initial value in step $j$

$$\|e_j\| := \|x_k(\lambda_0 + jh) - x_{j,k}\| \leq Ch^{p+1}. \tag{6.45}$$

To establish a global error, i.e., the error after several steps in the approximation, estimation the "fundamental lemma" theorem is needed and is stated without proof.

**Theorem 6.35** (The "fundamental lemma", c.f. [81], Theorem I.10.2)**.** *Let* $x(\lambda)$ *be a solution of an IVP* (6.3) *and let* $y(\lambda)$ *be a approximate solution. If for some* $\rho > 0$ *and* $\varepsilon > 0$

*(i)* $\|x(\lambda_0) - y(\lambda_0)\| \leq \rho$,

*(ii)* $\|\dot{y}(\lambda) - f(\lambda, y(\lambda))\| \leq \varepsilon$,

*(iii)* $\|f(\lambda, x(\lambda)) - f(\lambda, y(\lambda))\| \leq L_{f_x}\|x(\lambda) - y(\lambda)\|$,

*then, for* $\lambda \geq \lambda_0$*, we have the following error estimate*

$$\|x(\lambda) - y(\lambda)\| \leq \rho e^{L_{f_x}|\lambda-\lambda_0|} + \frac{\varepsilon}{L_{f_x}}\Big(e^{L_{f_x}|\lambda-\lambda_0|} - 1\Big). \tag{6.46}$$

Furthermore, the following theorem from [81] is also useful for a global estimation.

**Theorem 6.36** (c.f. [81], Theorem I.10.6)**.** *Let $x(\lambda)$ be a solution of an IVP* (6.3) *and let $y(\lambda)$ be a approximate solution. Suppose that*

$$\|\nabla f(\lambda, \nu)\| \leq L_{f_x} \quad \textit{for} \quad \nu \in [x(\lambda), y(\lambda)],$$

*and*

$$\|x(\lambda_0) - y(\lambda_0)\| \leq \rho, \quad \|\dot{y}(\lambda) - f(\lambda, y(\lambda))\| \leq \delta(\lambda).$$

*Then for $\lambda > \lambda_0$ we have*

$$\|x(\lambda) - y(\lambda)\| \leq e^{L_{f_x}|\lambda - \lambda_0|} \left( \rho + \int_{\lambda_0}^{\lambda} e^{-L_{f_x}|s - \lambda_0|} \delta(s)\, ds \right). \tag{6.47}$$

The global error can then be estimated by an *error transport* along $N - j$ steps of the numerical scheme [81]. The following theorem gives an estimate on the global error of a Runge-Kutta method of order $p$.

**Theorem 6.37** (c.f. [81], Theorem II.3.4)**.** *Let $\mathcal{U}$ be a neighborhood of $\{(\lambda, x_k(\lambda)) | \lambda \in I\}$, where $x_k(\lambda)$ is the exact solution of* (6.37) *and $I$ as defined in the previous sections. Suppose that in $\mathcal{U}$*

$$\|\nabla f\| \leq L_{f_x}$$

*and that the local error estimates* (6.45) *hold in $\mathcal{U}$. Then, the global error*

$$E = x_k(\lambda_u) - x_{N,k}$$

*can be estimated as*

$$\|E\| \leq h^p \frac{C}{L_{f_x}} \left( e^{L_{f_x}|I|} - 1 \right),$$

*for sufficiently small $h$ such that the solution remains in $\mathcal{U}$.*

*Proof.* Inserting $\rho_j = \|e_j\|$, $j = 1, \ldots, N$, in Theorem 6.35 with $\varepsilon = 0$ and Theorem 6.36 with $\delta = 0$, respectively, we obtain the following estimate

$$\|E_j\| \leq \exp(L_{f_x}|\lambda_u - \lambda_{j,k}|)\, \|e_j\|, \quad j = 1, \ldots, N, \tag{6.48}$$

where $E_j = x_k(\lambda_u) - x_{j,k}$, $j = 1, \ldots, N$. Then, this together with (6.45) is inserted in

$$\|E\| \leq \sum_{j=1}^{N} \|E_j\|.$$

We then have

$$\|E\| \leq \sum_{j=1}^{N} \exp(L_{f_x}|\lambda_u - \lambda_{j,k}|)\, C h^{p+1},$$

and consequently

$$\|E\| \leq h^p C \underbrace{\left( \sum_{j=1}^{N} h \exp(L_{f_x} |\lambda_u - \lambda_{j,k}|) \right)}_{(*)}.$$

The expression (*) can then be bounded by

$$\int_I \exp(L_{f_x} |\lambda_u - y|)\, dy,$$

proving the assertion. □

For the numerical experiments in Section 6.5 we apply 2nd-order and 4th-order Runge-Kutta methods to solve (6.37) in two different settings. To formulate the explicit methods one needs to specify the parameters $s \in \mathbb{N}$ and $a_{2,1}, a_{3,1}, a_{3,2}, \ldots, a_{s,1}, a_{s,2}, \ldots, a_{s,s-1} \in \mathbb{R}$, $b_1, \ldots, b_s \in \mathbb{R}$, and $c_2, \ldots, c_s \in \mathbb{R}$. The Butcher tableau (6.43) for the 2nd-order and 4th-order Runge-Kutta methods that we choose is then given as follows:

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 = \frac{1}{2} & a_{2,1} = \frac{1}{2} & & & \\ c_3 = \frac{1}{2} & a_{3,1} = 0 & a_{3,2} = \frac{1}{2} & & \\ c_4 = 1 & a_{4,1} = 0 & a_{4,2} = 0 & a_{4,3} = 1 & \\ \hline & b_1 = 1/6 & b_2 = 1/3 & b_3 = 1/3 & b_4 = 1/6 \end{array} \tag{6.49}$$

Thus, the 2nd-order and 4th-order Runge-Kutta methods can then be formulated as follows:

**Definition 6.38.**

*(i) The 2nd-order Runge-Kutta method, i.e., order $p = 2$, is given by*

$$x_{1,k} = x_{0,k} + h f(0 + \frac{h}{2}, x_{0,k} + \frac{h}{2} f(0, x_{0,k})).$$

*(ii) The classical 4th-order Runge-Kutta method (RK4-method), i.e., order $p = 4$, is given by*

$$\begin{aligned} k_1 &= f(\lambda_0, x_{0,k}) \\ k_2 &= f(\lambda_0 + \frac{h}{2}, x_{0,k} + \frac{k_1}{2}) \\ k_3 &= f(\lambda_0 + \frac{h}{2}, x_{0,k} + \frac{k_2}{2}) \\ k_4 &= f(\lambda_0 + h, x_{0,k} + k_3) \\ x_{1,k} &= x_{0,k} + h \left( \frac{1}{6} k_1 + \frac{2}{6} k_2 + \frac{2}{6} k_3 + \frac{1}{6} k_4 \right). \end{aligned}$$

Now the following algorithm to recover the Pareto front numerically using an explicit Runge-Kutta scheme for a given initial point $\lambda_0$ can be stated.

**Algorithm 3:** Pareto front tracing

**Input :** initial value $(\lambda_0, x_k(\lambda_0))$, number of points to integrate $N$, number of steps $s$ and parameters $a_{i,\ell}, b_i, c_i$, $i = 1, \ldots, s$, $\ell = 1, \ldots, i-1$ of the chosen explicit Runge-Kutta method

**Output:** approximations to points on the Pareto front $(\lambda_j, x_k(\lambda_j))$, $j = 1, \ldots, N$

$\mathrm{h} = (\lambda_u - \lambda_0)/N$

**for** $j = 1, \ldots, N$ **do**

  **for** $i = 1, \ldots, s$ **do**

    $k_i = f(\lambda_0 + jh + \sum_{\ell=2}^{i} c_\ell h, x_{j-1,k} + h \sum_{\ell=1}^{i-1} a_{i,\ell} k_i)$

  **end**

  $x_{j,k} = x_{j-1,k} + h \sum_{\ell=1}^{s} b_\ell k_\ell$

**end**

**Remark 6.39.** *The presented algorithm uses only integration forward in $\lambda$, as in traditional time integration. By taking $-f(\lambda, x(\lambda))$ as the right hand sight and by transforming the ordered set of negative $\lambda$-directions*

$$\{\lambda_{-q}, \ldots, \lambda_{-2}, \lambda_{-1} \mid \lambda_j < \lambda_0,\ \lambda_j < \lambda_{j+1},\ j = -q, -q+1, \ldots, -1\}, \quad \textit{for some } q > 0,$$

*into $\bar{\lambda}_j = \lambda_0 - \lambda_j$, $j = -q, -q+1, \ldots, -1$, in a next step reverting the ordering of these $\bar{\lambda}_j$, and re-transforming these $\bar{\lambda}_j$ during the evaluation of the right hand side of the ODE, one can also trace the Pareto front backward by going from $\lambda_0$ up to $\lambda_l$.*

## 6.4 A Related Method: Pareto Tracer

In this section we give a brief overview of *Pareto Tracer* (PT) a predictor-corrector method based on the Karush-Kuhn-Tucker (KKT) condition for multiobjective optimization problems introduced in [103], where for our purposes we only need the variant for unconstrained biobjective problems. Note that in [103] also a strategy for constrained problems is provided.

### 6.4.1 Predictor

Given an unconstrained biobjective optimization problem

$$\min_{x \in \mathbb{R}^n} J(x) = (J_0(x), J_1(x)), \tag{6.50}$$

where $J : \mathbb{R}^n \to \mathbb{R}^2$ one can formulate the (local) optimality condition, i.e., the Karush-Kuhn-Tucker (KKT) condition, as:

There exist Lagrange multipliers $\alpha \in \mathbb{R}^2$ such that

$$\begin{aligned} \alpha_0 \nabla J_0(x) + \alpha_1 \nabla J_1(x) &= 0, \\ \alpha_i &\geq 0, \quad i = 0, 1, \\ \alpha_0 + \alpha_1 &= 1. \end{aligned} \tag{6.51}$$

A point $x \in R^n$ satisfying (6.51) is called a *Karush-Kuhn-Tucker (KKT)* point. Next, predictor corrector (PC) methods are briefly introduced and then applied on the KKT conditions (6.51), see also [103]. Consider the following equation

$$G(x) = 0, \tag{6.52}$$

where $G : \mathbb{R}^{q+1} \to \mathbb{R}^q$ is sufficiently smooth. Assume that we have a solution $\bar{x}$ of (6.52) with $\mathrm{rk}(\nabla_x G(\bar{x})) = \mathrm{rk}(G'(\bar{x})) = q$, where $\mathrm{rk}(G'(\bar{x}))$ is the rank of $G'(\bar{x})$. The implicit function theorem then implies that there exists a value $\varepsilon > 0$ and a curve $c : (-\varepsilon, \varepsilon) \to \mathbb{R}^{q+1}$ such that $c(0) = \bar{x}$ and

$$G(c(t)) = 0, \quad \forall t \in (-\varepsilon, \varepsilon). \tag{6.53}$$

Further differentiating (6.53) yields

$$G'(c(t)) \cdot c'(t) = 0. \tag{6.54}$$

Therefore, computing kernel vectors for $G'(x)$ leads to tangent vectors $c'(t)$. To compute these a $QR$ factorization of $G'(x)^\top$ is used. Since for $G'(x)^\top = QR$, where $Q$ is an orthogonal matrix and $R$ an upper right triangular matrix the last column of $Q$ yields such a kernel vector, see also [103]. Moving in the direction of a given tangent vector then leads to a predictor point $p$. Now applying a corrector step with Algorithm 4 brings one back on the curve $c$. A PC method for unconstrained multiobjective optimization problems was introduced in [85] which considers for biobjective optimization problems

$$\hat{J}(x, \alpha) = \begin{pmatrix} \alpha_0 \nabla J_0(x) + \alpha_1 \nabla J_1(x) \\ \alpha_0 + \alpha_1 - 1 \end{pmatrix} = 0, \tag{6.55}$$

where $\alpha_i \geq 0,\ i = 0, 1$. The zero set of $\hat{J}$ contains all KKT points of problem (6.55) motivating the continuation along $\hat{J}^{-1}(0)$. The kernel vectors of $\hat{J}'$ are then computed via a $QR$ factorization of $\hat{J}'^\top$. Following [85], from this factorization one obtains an orthonormal basis of the linearized solution set in $(x, \alpha)-$space. Pareto Tracer computes such vectors in a way that allows the two spaces to be separated yielding tangent vectors to the Pareto set [103]. We only give a brief overview, for further details we refer to [103]. Let $x \in \mathbb{R}^n$ be a KKT point of (6.50) and $\alpha \in \mathbb{R}^2$ its corresponding Lagrange multipliers with $\alpha_i \geq 0,\ i = 0, 1,\ \alpha_0 + \alpha_1 = 1$ and

$$\alpha_0 \nabla J_0(x) + \alpha_1 \nabla J_1(x) = 0.$$

Further, let $\nu \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^2$ such that

$$\hat{J}'(x,\alpha)\begin{pmatrix}\nu\\ \mu\end{pmatrix} = \begin{pmatrix}\alpha_0\nabla J_0(x) + \alpha_1 \nabla J_1(x) & \nabla J_0(x) & \nabla J_1(x)\\ 0 & 1 & 1\end{pmatrix}\begin{pmatrix}\nu\\ \mu\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}. \tag{6.56}$$

Then a tangent vector can be obtained by

$$\nu_\mu = -W_\alpha^{-1}\nabla f(x)^\top \mu, \tag{6.57}$$

where we assume that the $n \times n$ matrix

$$W_\alpha := \alpha_0 \nabla^2 J_0(x) + \alpha_1 \nabla^2 J_1(x) \in \mathbb{R}^{n\times n} \tag{6.58}$$

is regular. It is desired to steer the search into a direction in the objective space such that the direction is orthogonal to $\alpha$ [103]. Following [103] for biobjective problems there are only two choices for the direction $\mu$

$$\mu^{(1)} = (-1,1)^\top \quad \text{and} \quad \mu^{(2)} = (1,-1)^\top. \tag{6.59}$$

To obtain a predictor $p := x + h\nu_\mu$ at a given KKT point $x$ a step length $h$ is computed as

$$h = \frac{h_{PT}}{\|\nabla f(x)\nu_\mu\|}, \tag{6.60}$$

where $h_{PT} \approx \|f(x_i) - f(x_{i+1})\|$ is a user specified value that corresponds to the Euclidean distance of two consecutive solutions $x_i$ and $x_{i+1}$ on the Pareto front. After computing a predictor point $p$ a modification of Algorithm 2 where Hessian information is incorporated in the computation of a search direction is applied as a corrector step.

### 6.4.2 Corrector

Following [59], we modify the biobjective descent algorithm (Algorithm 2) from Subsection 5.1.2 by incorporating Hessian information, if available. In [59], a Newton's method for multiobjective optimization problems was introduced which we formulate for the biobjective case. In [59], a modification of the quadratic problem to compute a search direction for the biobjective gradient descent (5.4) is considered and modified by adding the Hessians to the problem. Thus, one obtains the following quadratic problem to compute a Newton search direction for biobjective optimization problems

$$\begin{aligned}\min_{\rho\in\mathbb{R}, d\in\mathcal{X}} \quad & \rho\\ \text{s.t.} \quad & \nabla J_j(x^{(k)})^\top d + \frac{1}{2} d^\top \nabla^2 J_j(x^{(k)})\, d \le \rho,\ j = 0,1.\end{aligned} \tag{6.61}$$

In [59], it was established that for strictly convex objective functions with Lipschitz continuous second derivatives a descent with Newton search directions computed via (6.61) combined with step lengths computed with the Armijo-like rule (5.5) converges quadratically to a KKT point. Thus, one can state the following algorithm as a modification of Algorithm 2.
Similar to Algorithm 2 the stopping condition is chosen as $\|h_k d^{(k)}\| \le \varepsilon$, with $\varepsilon > 0$

**Algorithm 4:** Newton's method for biobjective optimization methods according to [59]

**Data:** Choose $\beta \in (0,1)$, $x^{(1)} \in \mathcal{X}$ and $\varepsilon > 0$, set $k := 1$.
**Result:** Approximation of a Pareto critical solution $\tilde{x} := x^{(k)}$.
Compute $d^{(0)} := d^{(1)}$ as a solution of (6.61) and set $h_0 := 1$;
**while** $\|h_{k-1}\, d^{(k-1)}\| > \varepsilon$ **do**
  Compute $d^{(k)}$ as a solution of (6.61);
  Compute a step length $h_k \in (0,1]$ as

$$\max\left\{h = \frac{1}{2^\ell} \;:\; \ell \in \mathbb{N}_0,\; J_j(x^{(k)} + h d^{(k)}) \leq J_j(x^{(k)}) + \beta h \nabla J_j(x^{(k)})^\top d^{(k)},\; j = 0,1\right\};$$

  $x^{(k+1)} := x^{(k)} + h_k\, d^{(k)}$ and $k := k + 1$;
**end**

a prespecified small constant. Furthermore, one can observe for Algorithm 4 the same limitations w.r.t. the iterates as for Algorithm 2, i.e., for the objective vector $J(x^{(k+1)})$ in iteration $k+1$ we have $J(x^{(k+1)}) \in J(x^{(k)}) - \mathbb{R}^2_>$.

### 6.4.3 PC Method

The predictor-corrector method Pareto Tracer for biobjective optimization problems can now be stated as the following algorithm.

**Algorithm 5:** Pareto Tracer[103]

**Data:** KKT point $x^{(0)}$ of (6.50) with associated convex weight, directions $\mu_1, \ldots, \mu_s \in \mathbb{R}^2$ and $\tau > 0$.
**Result:** KKT points $x^{(i)}$, $i = 1, \ldots, s$, in proximity of $x^0$.
**for** $i = 1, \ldots, s$ **do**
  Compute $\nu_i := \nu_{\mu_i}$ as in (6.57);
  Compute $h_i$ as in (6.60);
  Compute $p_i := x^{(0)} + h_i \nu_i$;
**end**
**for** $i = 1, \ldots, s$ **do**
  Compute a KKT point $x^{(i)}$ and associated weight via Algorithm 4 starting with $p_i$;
**end**

## 6.5 Numerical Results

The Pareto tracing by numerical integration algorithm is tested on a simple biobjective convex quadratic optimization problem (Section 6.5.1), and on a variant of the biobjective test problem ZDT3 (see, e.g., [158]) (Section 6.5.2) as well as on our biobjective shape optimization problem (3.11). All numerical experiments are realized in R (version 3.5).

The resulting ODEs are solved with the implementations of the 2nd-order and 4th-order Runge-Kutta method of the R package "deSolve", version 1.27.1, see also [144]. Furthermore, in Section 6.5.1 and Section 6.5.2, the results of the presented approach are compared with the predictor-corrector method *Pareto Tracer* (Section 6.4), where the MATLAB toolbox provided in [103] is used for the numerical experiments. This method iteratively computes points on the Pareto front that have a Euclidean distance of $h_{PT}$, i.e., the step length $h_{PT}$ determines the distance between points on the Pareto front.

### 6.5.1 Pareto Tracing by Numerical Integration for Biobjective Convex Quadratic Optimization

We consider an unconstrained and strictly convex biobjective optimization problem with two quadratic objective functions $J_i : \mathbb{R}^n \to \mathbb{R}$, $i = 0, 1$, given by

$$J_0(x) = \frac{1}{2}(x - \chi_0)^\top Q_0(x - \chi_0), \qquad J_1(x) = \frac{1}{2}(x - \chi_1)^\top Q_1(x - \chi_1)$$

with positive definite matrices $Q_0, Q_1 \in \mathbb{R}^{n\times n}$ and arbitrary but fixed vectors $\chi_0, \chi_1 \in \mathbb{R}^n$. The biobjective optimization problem is then of the form

$$\begin{array}{ll} \min & J(x) = (J_0(x), J_1(x)) \\ \text{s.t.} & x \in \mathbb{R}^n. \end{array} \tag{6.62}$$

Quadratic problems, as a class of problems, are particular useful to assess the quality of approximated Pareto fronts, since for problems of this class an analytic description of the Pareto-optimal set exists, see, e.g., [149]. For the sake of completeness, the derivation is provided below. Since $J_i$, $i = 0, 1$ are strictly convex, every Pareto-optimal solution can be obtained as the unique optimal solution $x(\lambda)$ of a weighted sum scalarization

$$\begin{aligned} \min J_\lambda(x) :=& (1-\lambda)J_0(x) + \lambda J_1(x) \\ =& \frac{(1-\lambda)}{2}(x - \chi_0)^\top Q_0(x - \chi_0) + \frac{\lambda}{2}(x - \chi_1)^\top Q_1(x - \chi_1) \end{aligned} \tag{6.63}$$

with $\lambda \in (0, 1)$. Note that this condition is only necessary and sufficient for strictly convex problems, but not in general. Since in this case the Hessian $\nabla_x^2 J_\lambda(x(\lambda)) = \lambda Q_1 + (1-\lambda)Q_0$ is positive definite for all $\lambda \in (0, 1)$ (regardless of $x(\lambda)$), the second-order optimality condition is strictly satisfied by every such solution $x(\lambda)$, i.e., every $x(\lambda)$ is Pareto-optimal for all $\lambda \in (0, 1)$. Thus $x(\lambda)$ is optimal for (6.63) if and only if $\nabla_x J_\lambda(x) = 0$, where

$$\begin{aligned} \nabla_x J_\lambda(x(\lambda)) &= (1-\lambda)\nabla_x J_0(x(\lambda)) + \lambda \nabla_x J_1(x(\lambda)) \\ &= (1-\lambda)Q_0(x(\lambda) - \chi_0) + \lambda Q_1(x(\lambda) - \chi_1) \\ &= [\lambda Q_1 + (1-\lambda)Q_0]\, x(\lambda) - [(1-\lambda)Q_0\chi_0 + \lambda Q_1\chi_1]. \end{aligned} \tag{6.64}$$

Furthermore, from the optimality condition one can derive that

$$\begin{aligned} \nabla_x J(\lambda)(x(\lambda)) &= 0 \\ \Leftrightarrow \qquad x(\lambda) &= [\lambda Q_1 + (1-\lambda)Q_0]^{-1}((1-\lambda)Q_0\chi_0 + \lambda Q_1\chi_1). \end{aligned} \tag{6.65}$$

Hence, the Pareto set is contained in a parameterized curve that is completely described by the function $x : (0,1) \to \mathbb{R}^n$ given by $x(\lambda) = [\lambda Q_1 + (1-\lambda)Q_0]^{-1}((1-\lambda)Q_0\chi_0 + \lambda Q_1\chi_1)$ for $\lambda \in (0,1)$. The two limiting points are obtained for $\lambda = 0$ and $\lambda = 1$, respectively, as the unique minima of the individual objective functions:

$$\begin{aligned} \lambda = 0: &\quad x(0) = \chi_0 \\ \lambda = 1: &\quad x(1) = \chi_1. \end{aligned} \tag{6.66}$$

We can conclude that $x(\lambda)$ solves $\nabla_x J_\lambda(x(\lambda)) = 0$ for all $\lambda \in (0,1)$. Next as described in Sections 6.2 and 6.3 above the first-order optimality conditions are differentiated w.r.t. $\lambda$ to state the implicit ODE (6.32) on which the Pareto tracing by numerical integration algorithm is based on. Thus, yielding

$$\frac{d}{d\lambda}\nabla_x J_\lambda(x(\lambda)) = 0 \quad \Leftrightarrow \quad [(1-\lambda)Q_0 + \lambda Q_1]\,\dot{x}(\lambda) - Q_0(x(\lambda) - \chi_0) + Q_1(x(\lambda) - \chi_1) = 0$$

for $\lambda \in (0,1)$. Since the Hessian $\nabla_x^2 J_\lambda(x(\lambda))$ is positive definite for all $\lambda \in (0,1)$, one can rearrange this to a standard ODE (6.33) as

$$\dot{x}(\lambda) = [(1-\lambda)Q_0 + \lambda Q_1]^{-1}(Q_0(x(\lambda) - \chi_0) - Q_1(x(\lambda) - \chi_1)) = f(\lambda, x(\lambda)), \tag{6.67}$$

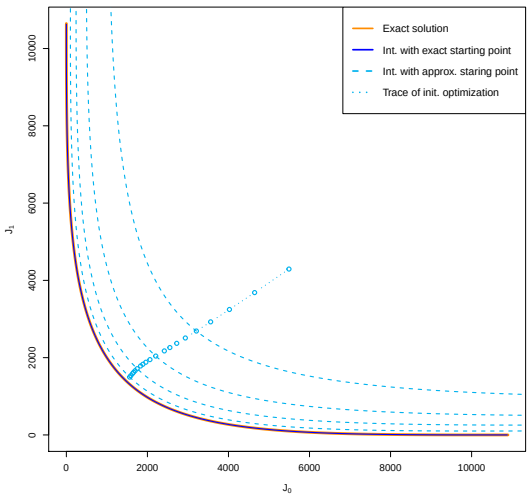
with possible initial values $x_0 = x(\lambda_0) = \chi_1$ (for $\lambda_0 = 1$) or $x_0 = x(\lambda_0) = \chi_0$ (for $\lambda_0 = 0$).

**Remark 6.40.** *The Hessian of* (6.63) *is then given as* $\nabla_x^2 J_\lambda(x) = (1-\lambda)Q_0 + \lambda Q_1$ *and is as such independent of* $x$*. Since it is further positive definite for all* $\lambda \in (0,1)$*, its smallest eigenvalue* $\Lambda(\lambda, x)$ *is bounded from below by some* $\varepsilon > 0$ *on* $(0,1)$*, i.e., for* $I = (0,1)$*, we have* $\Lambda'(I) = \inf_{\lambda \in I} \Lambda(\lambda, x(\lambda)) \geq \varepsilon > 0$*. This implies that uniform constants* $L_\sharp$*,* $\sharp = H, f, \lambda$*, can be chosen in Lemma 6.24 and Proposition 6.31, where one can set* $L_H = 0$*. Furthermore, the analysis above yields that* $f(\lambda, x(\lambda)) \in C^\infty$*, and consequently following Theorem 6.34 and Lemma 6.27 high-order iteration schemes are then applicable for this problem.*

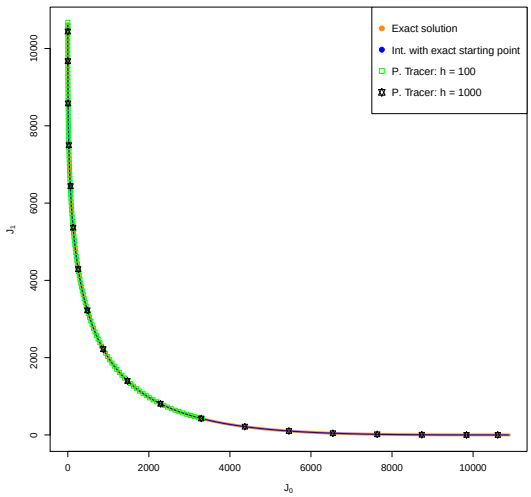
In [149] different classes of biobjective quadratic test functions are proposed. These test functions are of the form

$$J_0(x) = \frac{1}{b}(x - \chi_0)^\top Q_0(x - \chi_0), \quad a, b \in \mathbb{R}, \qquad J_1(x) = \frac{1}{a}(x - \chi_1)^\top Q_1(x - \chi_1)$$

i.e., our test functions are covered by this class of functions, where the difficulty of the biobjective problem can be regulated with the properties of the matrices $Q_0, Q_1$, e.g., orthogonal or permutation matrices, and the vectors $\chi_0, \chi_1$. We do not specify a problem class of varying difficulty as proposed in [149] to investigate, but generate, random matrices $Q_0$ and $Q_1$ by $Q_j = P_j^\top P_j$, where $P_j$ is a sample from a $n \times n$-random matrix with independent standard normal distributed entries, $j = 0, 1$. The vectors $\chi_0$ and $\chi_1$ are also generated in a same way as $n$-dimensional random vectors with independent standard normal entries. Numerical tests are provided for dimension $n = 100$. The 4th-order Runge-Kutta method is used in Parerto tracing algorithm to solve the resulting ODE.
In Figure 6.1a, we compare the analytic solution $x(\lambda)$ (orange, thick solid), i.e., the exact solution obtained from (6.65), with solutions of Pareto tracing by numerical integration




(a) Comparison with the objective values of the numerically integrated solution of the ODE (6.67) started at the exact solution (solid blue, $\lambda_0 = 0.5$) and integrated solutions (dashed light blue, $\lambda_0 = 0.5$) started at the 5th, 10th, 15th, and 20th iteration of a gradient descent algorithm starting at $x_{0,0} = 0$ (dotted light blue). Numerical integration of the ODE uses 4th order Runge-Kutta method with 20 iterations, 10 in each direction (step length $h = 0.05$).




(b) Comparison with the objective values of the numerically integrated solution of the ODE (6.67) and the objective values of solutions computed with Pareto Tracer started in the same exact solution (solid blue) with step lengths $h_{PT} = 100, 1000$ (green and black, respectively).

Figure 6.1: Comparison of the analytic solution (6.65) for the Pareto front (orange, thick solid) with different approximations. The dimension of the problem is $n = 100$. See also [19].

applied on (6.67) with initial value $x_0 = x(0.5)$ (solid blue). One can observe, that the results of the numerical integration are a fairly good approximation of the analytical Pareto front. Furthermore, the Pareto tracing by numerical integration algorithm is applied on approximate starting solutions $x_{0,k}$ for $k \in \{5, 10, 15, 20\}$, which are computed with a gradient based descent algorithm with Armijo step lengths (with parameter $\beta = 0.5$) starting in $x_{0,0} = 0$ to integrate for an approximate Pareto front. One can observe that the solutions $x_5(\lambda), x_{10}(\lambda), x_{15}(\lambda)$ and $x_{20}(\lambda)$, $\lambda \in (0, 1)$, for the 5th, 10th, 15th and 20th iteration of the gradient descent are all $\varepsilon$-Pareto critical approximations of the Pareto front (for different $\varepsilon$ each) and the approximations become more precise, i.e., $\varepsilon$ decreases, for increasing iteration numbers $k$, coinciding with the results of Proposition 6.31 (iii). Moreover, to investigate the robustness of this procedure further 100 randomized biobjective quadratic test functions are investigated analogously. For all instances, the observations from above are reproduced, i.e., good approximations of the Pareto fronts were achieved. Similarly, the results of the integration starting in premature starting solutions $x_{0,k}$ for $k \in \{5, 10, 15, 20\}$ also yield various $\varepsilon$-Pareto critical approximations of the Pareto front for all 100 random instances. Some exemplary solution fronts are illustrated

in Figure 6.2.

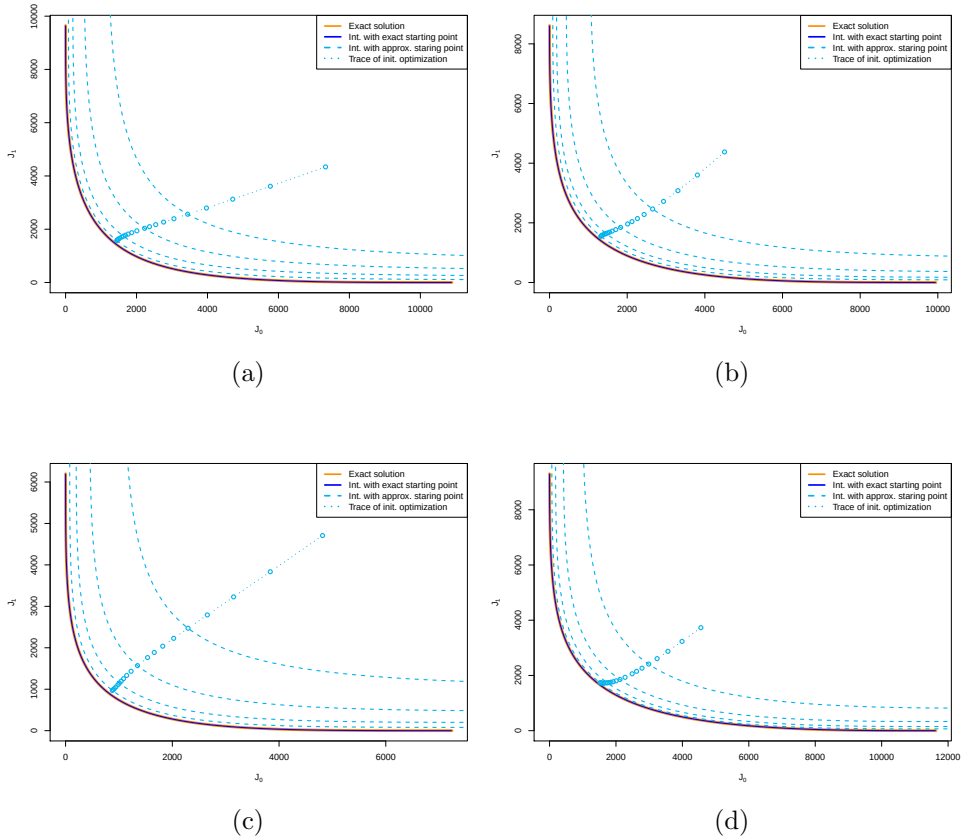



Figure 6.2: Some exemplary fronts of the 100 randomized biobjective quadratic test problems.

In Figure 6.1b we also compare the analytic solution $x(\lambda)$ and the integrated solutions with initial value $x_0$ with solutions obtained by the Pareto Tracer starting in $x_0$ with step lengths $h_{PT} = 100$ and $h_{PT} = 1000$. One can observe that the Pareto Tracer yields a comparable approximation of the Pareto front for both step lengths. For the step length $h_{PT} = 100$, the Pareto Tracer is not able to fully trace the Pareto front, while for $h_{PT} = 1000$, an approximation covering (nearly) the complete range of the front is obtained. Increasing the maximum iteration number for $h_{PT} = 100$ does not improve the range of the approximation, whereas approximations with $h_{PT} = 10$, $h_{PT} = 50$, and $h_{PT} = 150$ covered (slightly) larger ranges than $h_{PT} = 100$. Since the step length of the Pareto Tracer defines the Euclidean distance between two consecutive points on the curve describing the Pareto front, there can be unfavorable choices for step lengths w.r.t. to the local trade-offs. Moreover, a comparison of the needed number of objective function evaluations per solution on the front is given in Table 6.1. One can observe that for the Pareto Tracer the mean number of evaluations increases with bigger step sizes, which is in accordance with the fact that a bigger predictor step may yield a predictor that is farther away from the actual front and hence more corrector steps may be needed. Consequently, a smaller step length reduces the average number of function evaluations,

where the amount of mean Jacobian and Hessian evaluations can become smaller as in the case of Pareto tracing by numerical integration, see $h_{PT} = 10$ in Table 6.1. However, the average number of function evaluations of the approach presented in this paper, Pareto tracing by numerical integration is independent of the step length $h$ and only depends on the order of the chosen Runge-Kutta method. Note that for small steps, the RK 2 method may be sufficient for the integration, further cutting the needed Jacobian and Hessian evaluations in half. Therefore, when aiming at the approximation of small parts of the Pareto front, the Pareto Tracer method may be preferable, while for larger steps on the Pareto front, Pareto tracing by numerical integration is usually a better choice. Note, however, that the error estimates from Theorem 6.37 for RK 4 provide a better asymptotic (quartic) rate as compared to Pareto Tracer since, even when assuming quadratic convergence for the corrector [59], at most a quadratic decrease of the error in the number of evaluations is obtainable.

| | Int. (RK 4) for all $h$ | P. Tracer: $h_{PT} = 10$ | P. Tracer: $h_{PT} = 100$ | P. Tracer: $h_{PT} = 1000$ |
|---|---|---|---|---|
| Mean objective calls | 1 | 23.4300 | 67.2288 | 642 |
| Mean Jacobian calls | 4 | 1.5338 | 8.4576 | 40.9473 |
| Mean Hessian calls | 4 | 1.5338 | 8.4576 | 40.9473 |

Table 6.1: Comparison of the average number of objective function evaluations per computed point for the biobjective convex quadratic problem. See also [19].

### 6.5.2 Pareto Tracing by Numerical Integration for the Biobjective Test Function ZDT3s

The biobjective test function ZDT3, see, for example, [158] is a well-known test function where the Pareto set is not a line segment and the Pareto front consists of various non-contiguous convex parts. For $n \in \mathbb{N}$ and $x \in [0,1]^n$, its objectives are given by

$$J_0(x) := x_1, \tag{6.68}$$

$$J_1(x) := g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\,\pi\,x_1)\right], \tag{6.69}$$

where $g(x) = 1 + \frac{9}{n-1}\left(\sum_{j=2}^{n} x_j\right)$. Hence, the biobjective optimization problem can be stated as

$$\min_{x\in[0,1]^n} J(x) := (J_0(x), J_1(x)), \tag{6.70}$$

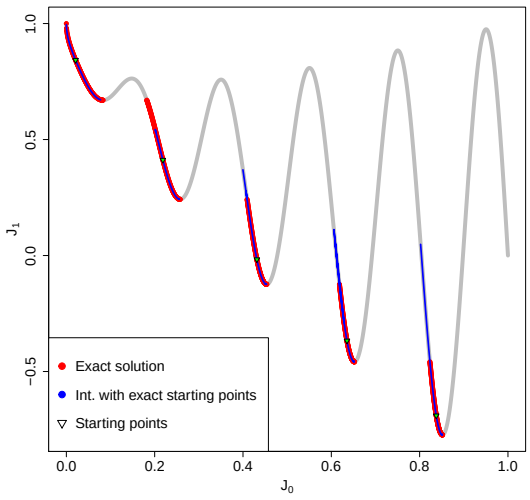
and its Pareto front is formed with $g(x) = 1$, see, e.g., [158]. Since, for these objectives the Hessian of any weighted sum scalarization $\nabla_x^2 J_\lambda(x)$ for $\lambda \in (0,1)$ is singular, we adapt a small modifications to the problem to ensure regular Hessians, i.e., the $x_j$ in the objectives (6.68) and (6.69) are replaced by $x_j^2$, for $j = 1, \ldots, n$. In the following, we refer to this problem as *ZDT3s*. Note that the Pareto front of ZDT3s is equal to the front of ZDT3, while the Pareto set of ZDT3s is given by the component wise square root of the Pareto

set of ZDT3.
For the computation of the right hand side $f(\lambda, x(\lambda))$ of (6.33) for the problem ZDT3s the gradients $\nabla_x J_i(x)$ and Hessians $\nabla_x^2 J_i(x)$, $i = 0, 1$, are approximated by the finite difference method with precisions $\epsilon_G \approx 1.5 \times 10^{-8}$ and $\epsilon_H \approx 6 \times 10^{-6}$, respectively. The gradients and Hessians for Pareto Tracer are computed in the same way. For the numerical integration of the ODE the 4th order Runge-Kutta method with a step length of $h = 0.001$ is used. The dimension of the problem is $n = 100$. In Figure 6.3a, the analytical solution is compared with the results of the numerical integration. Nearly all parts of the Pareto front are approximated reasonably well when starting in five different initial values. The corresponding initial weight $\lambda_0$ for one initial value $x_0$ such that $x_0$ is $J_{\lambda_0}$-critical is computed by solving the equation $\|\nabla J_\lambda(x_0)\| = 0$ for $\lambda_0 \in (0, 1)$. We stop the numerical integration when an inflection point on the local Pareto front is reached, since the Hessians of the weighted sum scalarization become indefinite in these points. Beyond the inflection point on a non convex Pareto front the trade-offs reverse again which would require an ODE solution "backward in time". This is not included in our present algorithm. Also the presented approach does not further guarantee local Pareto optimality from that point on. Integrating past this given barrier may also lead to dominated solutions, which then can be filtered in a post processing step that has to be done anyway since the Pareto front consists of several parts where some local solutions on one part of the curve may dominate some other local solutions on another part of the curve. Further, in Figure 6.3b the analytical solution is compared with the results of Pareto Tracer with a step length of $h_{PT} = 0.02$. In contrast to the numerical integration, Pareto Tracer is not limited by inflection points and can therefore approximate the whole Pareto front. Moreover, in this case as in the other one a post processing filtering step is needed to determine the Pareto front.
Furthermore, in Table 6.2, the mean number of objective evaluations per solution of the numerical integration and Pareto Tracer are compared. Since the parts of the Pareto front are relatively small small step sizes are needed, favoring Pareto Tracer. The mean amount of iterations of Pareto Tracer only exceeds the amount for the numerical integration for step lengths bigger than $\hat{h}_{PT} \approx 0.4$.

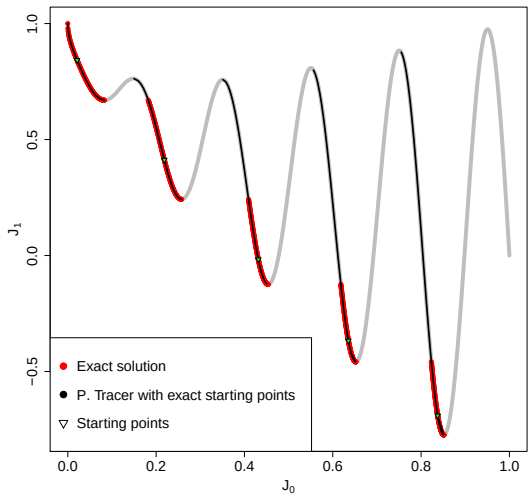
| | Int. (RK 4) for all $h$ | P. Tracer: $h_{PT} = 0.02$ | P. Tracer: $h_{PT} = 0.4$ |
|---|---|---|---|
| Mean objective calls | 20801 | 7085.84 | 25609.75 |

Table 6.2: Comparison of the mean objective function evaluations per solution of the numerical integration and Pareto Tracer for ZDT3s. See also [19].

**Remark 6.41.** *Note that our assumption on the strict positive definiteness of the Hessian of $J_\lambda$ is not satisfied at an inflection point of the Pareto front. Indeed, consider a sequence of weights $\{\lambda_i\}$, $\lambda_i \in (0, 1)$ for all $i$, with $\lim_{i\to\infty} \lambda_i = \bar{\lambda}$, and let $x(\lambda_i)$ be a corresponding sequence of local minima of $J_{\lambda_i}$ with $\lim_{i\to\infty} x(\lambda_i) = \bar{x}$. Now suppose that the outcome vector $J(\bar{x})$ is an inflection point of the Pareto front in the objective space. Then, $\bar{x}$ fulfills $\nabla J_{\bar{\lambda}}(\bar{x}) = 0$ by continuity, but is not a local minimum of $J_{\bar{\lambda}}$ since $J_{\bar{\lambda}}(x(\lambda_i)) < J_{\bar{\lambda}}(\bar{x})$ for $i$ sufficiently large, i.e., for $\lambda_i$ sufficiently close to $\bar{\lambda}$.*




(a) Comparison of the analytic solution of ZDT3s (red) and the numerical integration started in exact solutions (blue). The initial values are marked as triangles.




(b) Comparison of the analytic solution of ZDT3s (red) and Pareto Tracer started in exact solutions (black). The initial values are marked as triangles.

Figure 6.3: Comparison of the analytic solution with solutions of the numerical integration and Pareto Tracer. See also [19].

### 6.5.3 Pareto Tracing by Numerical Integration for Biobjective Shape Optimization

Now we apply the Pareto tracing by numerical integration approach to our biobjective shape optimization problem (3.11), see also [118] for a related work. To this end, recall that we have

$$\begin{aligned} &\min_{\Omega \in \mathcal{O}^{\mathrm{ad}}} \; J(\Omega) := (J_0(\Omega), J_1(\Omega)) \\ &\text{s.t. } u \in H^1(\Omega, \mathbb{R}^2) \text{ solves the state equation,} \end{aligned} \tag{6.71}$$

where $J_1(\Omega)$ is the objective w.r.t. the probability of failure of the shape $\Omega \in \mathcal{O}^{\mathrm{ad}}$ and $J_0(\Omega)$ is its volume. The same discretization, i.e., $J_0(x)$ and $J_1(x)$, that was proposed in Chapter 4 is used. Furthermore, to apply the Pareto tracing by numerical integration approach, the right hand side $f(\lambda, x(\lambda))$ of (6.33) for problem (3.11) has to be computed. Therefore, the $2n_B \times 2n_B$ Hessian matrices $\nabla_x^2 J_i(x^{\mathrm{ml}})$ and $\nabla_x^2 J_i(x^{\mathrm{th}})$, $i = 0, 1$, are approximated with finite differences using a precision of $\varepsilon_H = 10^{-6}$, i.e.

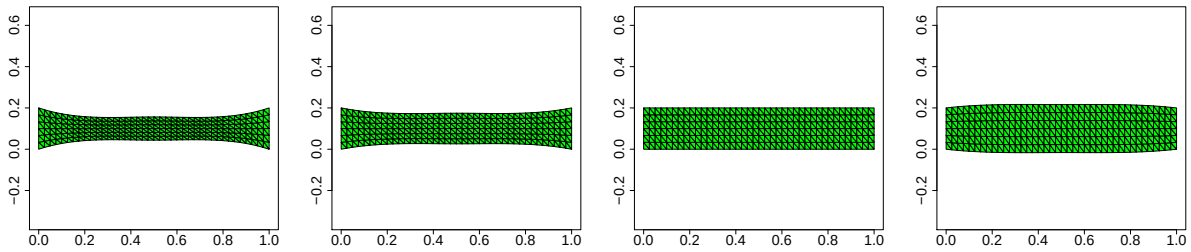
$$\left(\nabla_x^2 J_i(x^\alpha)\right)_{\cdot j} \approx \frac{\nabla_x J_i(x^\alpha + e_j \varepsilon_H) - \nabla_x J_i(x^\alpha)}{\varepsilon_H}, \quad \alpha \in \{\mathrm{ml}, \mathrm{th}\}, \; j = 1, \ldots, n_B \tag{6.72}$$

where $e_j$, $j = 1, \ldots, n_B$, are the standard basis vectors of $\mathbb{R}^{n_B}$. Hence the right hand side is then of the form $f_l(\lambda, x(\lambda))$ for some $l$. This is something that can easily be done in parallel. Toward this end, the R packages "doParallel", version 1.0.15, and "foreach",

version 1.4.7, are used to compute the $n_B+1$ gradient evaluations for the finite differences approximation on $n_B+1$ CPU cores. Further, Proposition 6.31 (iii) establishes the stability of this approach while using an approximate right hand side $f_l(\lambda, x(\lambda))$.
For the numerical solution of the ODE (6.33), we apply an order 2 Runge-Kutta method, thus requiring that the discretized objective functions $J_i$, $i = 0, 1$, are at least 4 times continuously differentiable (c.f. Theorem 6.34). This is clearly satisfied for the discretized volume $J_0(x)$, which is, as a polynomial, infinitely differentiable. For the discretized intensity measure $J_1(x)$, we can build on the analysis for $J_1(Z)$ performed in [79, 21, 80, 73] and Section 4.2. Recall that the discretized state equation (3.2) is of the form $B(Z)\,U(Z) = \hat{F}(Z)$, where $U(Z)$ is the discretized displacement, $B(Z)$ the positive definite stiffness matrix, and $\hat{F}(Z)$ the discretized forces. From the assembly of $B(Z)$ and $\hat{F}(Z)$ in [79, 21, 80, 73] and Section 4.2, it can be seen that $B(Z), \hat{F}(Z) \in C^\infty$. Using the identity $U(Z) = B(Z)^{-1}F(Z)$, where the right hand side is infinitely differentiable, it can be shown iteratively that also $U(Z) \in C^\infty$. Moreover, in [18, Lemma 6.5.5], it is shown that $\zeta(\sigma) = ((n^\top \sigma\, n)^+)^m$ is $m$ times continuously differentiable w.r.t. $\sigma$. We can conclude that this is also the case for $J_1(Z)$. The order 2 Runge-Kutta method is hence applicable for Weibull modules $5 \leq m \leq 30$. We consider the two test cases proposed in Section 4.4, i.e., the straight joint (Subsection 4.4.1) and the s-shaped joint (Subsection 4.4.2), and choose the solutions generated with the weighted sum scalarization in the previous chapter (Chapter 5) as initial solutions for the Pareto tracing by numerical integration algorithm. While this problem may be non convex in general and, as a consequence, the computation of the complete Pareto front cannot be guaranteed a priori, we note that numerical experiments indicate that the Pareto front is at least locally convex and that it can be well approximated by the suggested method.

### Test Case 1: A Straight Joint

In this test case the left and right boundaries are fixed at the same height and the surface forces $\bar{g}$ act on the right boundary. In the previous chapter, we established that the Pareto critical shapes for this test case are straight rods with varying thickness connecting both boundaries, see Figure 6.4. This motivates the use of a discretized straight rod

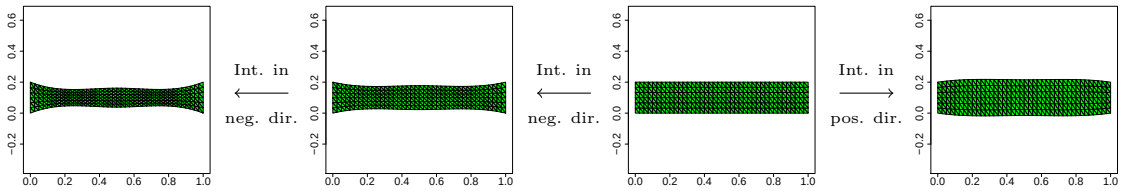



(a) $\lambda = 0.2$ (b) $\lambda = 0.5$ (c) $\lambda_0 = 0.813$; $x_0$ (d) $\lambda = 0.9$

Figure 6.4: Some solutions of the weighted sum method of Chapter 5 and the initial shape $x_0$. See also [19].

with constant thickness of 0.2 m as the initial point $x_0$ for the numerical integration, see

Figure 6.4c, even though this particular shape was not computed with a gradient descent of the weighted sum scalarization. For this shape we do not have an iteration index $k \in \mathbb{N}$, since the initial value $x_0$ is not a approximation but an exact solution. Nevertheless, $x_0$ as a Pareto critical solution is the limit of some unknown sequence $\{x_{0,k}\}$, i.e., $\lim_{k\to\infty} x_{0,k} = x_0$, since the part of the (local) Pareto front that was computed in Chapter 5 seems convex and therefore the weighted sum scalarization should be able to recover the solutions in that part of the front for some weights $\lambda \in (0, 1)$. Further, since the right hand side of the ODE $f_l(\lambda, x(\lambda))$ is an approximation that depends on some parameter $l$ we denote the solution of the ODE as $x_l(\lambda) := x_{\infty;l}(\lambda) = x_{k;l}(\lambda)$. Note that this should not be confused with $x_k(\lambda)$. Next, a corresponding weight $\lambda_0$ such that $x_0$ is $J_{\lambda_0}$-critical is recovered by solving the equation $\|\nabla J_\lambda(x_0)\| = 0$ for $\lambda_0 \in (0, 1)$. From this optimization we obtain $\lambda_0 \approx 0.813$ for the desired weight, for which $\|\nabla J_\lambda(x_0)\| \approx 2.6 \times 10^{-8}$. We therefore have $x_0 \approx x_l(0.813)$ with $\nabla J_0(x_0)$ and $\nabla J_1(x_0)$ pointing in opposite directions ($\cos(\nabla J_0, \nabla J_1) = -0.999998$). We thus consider this solution as Pareto critical up to numerical error. This nicely agrees with the intuition from mechanical engineering that the straight rod should be the optimal form given its volume.
Applying Pareto tracing by numerical integration on the interval $[\lambda_l, \lambda_u] = [\lambda_0 - 0.66, \lambda_0 + 0.1]$ with a step length of $h = 0.01$ then yields shapes of varying thickness that are also straight rods. This is in accordance with the results of Chapter 5, see Figure 6.5 for some exemplary shapes corresponding to $x_l(\lambda)$. The outcome vectors obtained with Pareto




$x_l(0.203)$ $x_l(0.503)$ $x_0$ $x_l(0.903)$

Figure 6.5: Exemplary results of Pareto tracing by numerical integration w.r.t. the ODE (6.33) in negative and positive direction, starting from $x_0$. See also [19].

tracing by numerical integration algorithm and the outcome vectors computed via the two gradient based descent algorithms in Chapter 5 are compared in Figure 6.6. One can observe that not only the weighted sum solutions are covered by the solutions of the numerical integration, but also a larger part of the (local) Pareto front is approximated with this approach.
The statements of Proposition 6.31 are also validated as shown in Figure 6.7, where the first and the second-order optimality conditions were tracked during the numerical integration of Pareto tracing by numerical integration. Indeed, the results nicely display that the solutions of Pareto tracing by numerical integration achieve good results w.r.t. first and second-order optimality tests.
Furthermore, in Figure 6.8 the behaviour of the B-spline coefficients $x$ of the solutions $x_l(\lambda)$, $\lambda \in [\lambda_0 - 0.66, \lambda_0 + 0.1]$ is shown. As all shapes are straight rods, the B-spline coefficients $x^{\text{ml}}$ w.r.t. the meanline values $x_1^{\text{ml}}$ (orange), $x_2^{\text{ml}}$ (red) and $x_3^{\text{ml}}$ (brown) stay unchanged for all solutions. Furthermore, the two thickness coefficients w.r.t. the B-splines

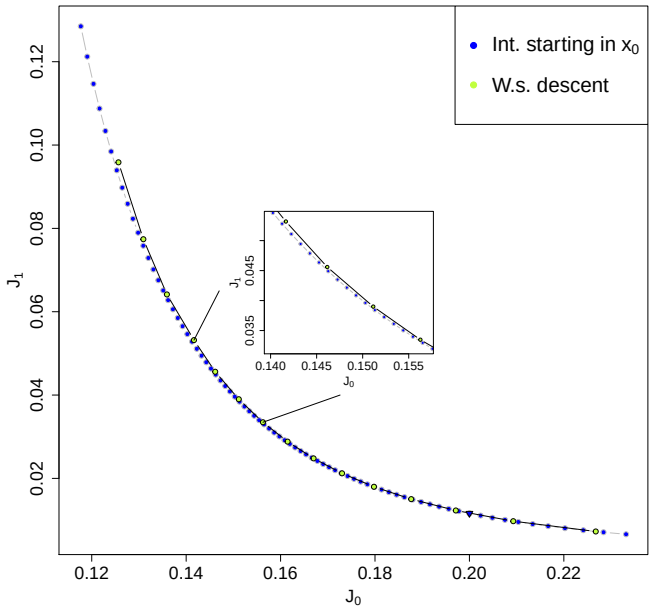
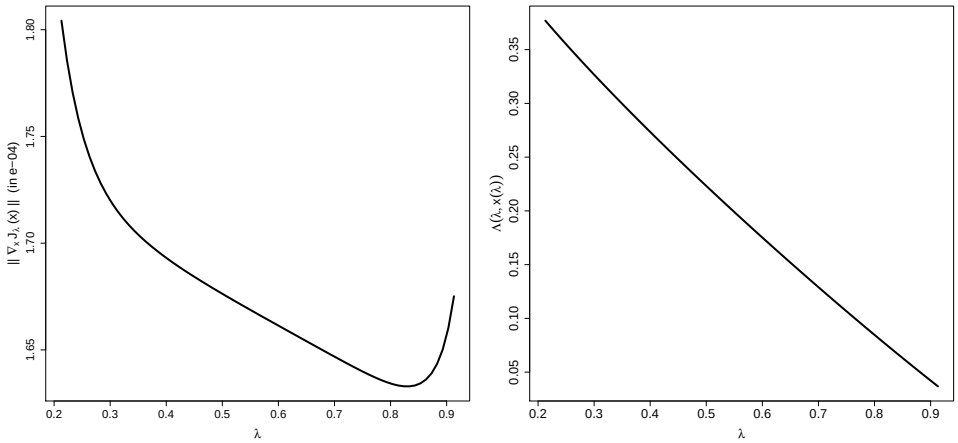



Figure 6.6: Comparison of the outcome vectors of the numerical integration (blue) with initial value $x_0$ (blue triangle) and the outcome vectors obtained in Chapter 5 from the repeated application of gradient descent with the weighted sum scalarization (green). See also [19].




(a) first-order optimality: $\|\nabla_x J_\lambda(x)\|$ (b) second-order optimality: smallest eigenvalue $\Lambda(\lambda, x(\lambda))$ of $\nabla_x^2 J_\lambda(x)$

Figure 6.7: Straight joint: evaluating first and second-order optimality during the numerical integration. See also [19].

on the edges of the shapes $x_1^{\text{th}}$ (green) and $x_3^{\text{th}}$ (purple) behave symmetrically and decrease with smaller $\lambda$, whereas the values of $x_2^{\text{th}}$ (blue) stay nearly unchanged throughout the optimization process, increasing slightly for (relatively) big $\lambda$. Thus, one can assume that

$x_1^{\text{th}}$ and $x_3^{\text{th}}$ mainly control the volume of the shapes via the edges until at some point the middle part of the rod, i.e., $x_2^{\text{th}}$, has to grow, too.

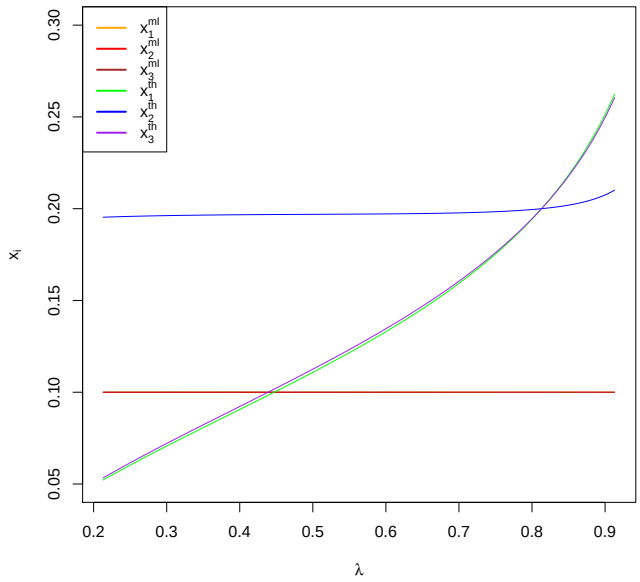
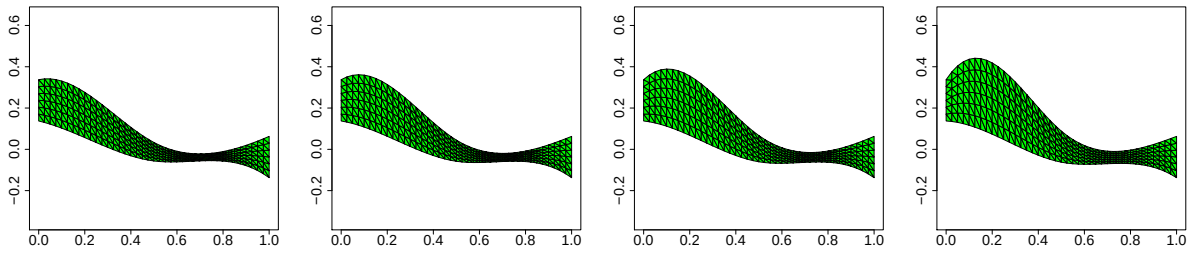



Figure 6.8: Test case 1: behaviour of the B-spline coefficients $x$ of the solution $x_l(\lambda)$ obtained with Pareto tracing by numerical integration for $\lambda \in [\lambda_l, \lambda_u] = [\lambda_0 - 0.66, \lambda_0 + 0.1]$.

### Test Case 2: An S-Shaped Joint

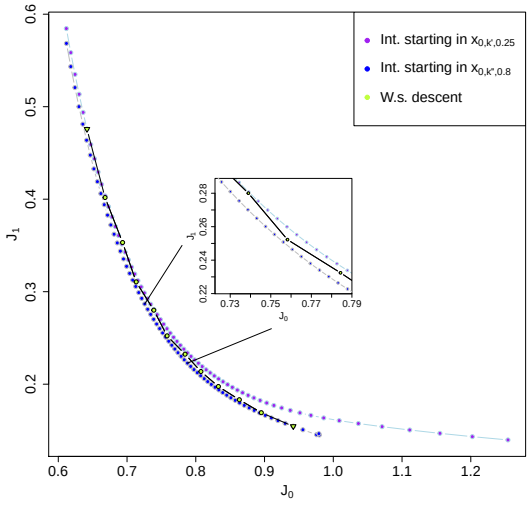
For the test case of the s-shaped joint, the numerical studies of the previous chapter suggest that the (locally) Pareto-optimal shapes resemble the profiles of whales with varying volume. See Figure 6.9 for exemplary solutions of gradient descents with weighted sum scalarizations for weights $\lambda = 0.25, 0.4, 0.6, 0.8$. Note that the gradient descent method did not converge for $\lambda < 0.25$ and $\lambda > 0.8$, hence we omit these solutions in the following comparisons. Since the optimization of this test case was more complex in the previous chapter, we choose two starting solutions for the numerical integration and compare their results. Toward this end, we consider the two solutions of the weighted sum scalarization with the smallest and largest weight $\lambda$ for which the descent algorithm converged in Chapter 5, $x_{0,k';l,0.25} := x_{0,k';l} = x_{k';l}(0.25)$ and $x_{0,k'';l,0.8} := x_{0,k'';l} = x_{k'';l}(0.8)$, as initial values for Pareto tracing by numerical integration. Pareto tracing by numerical integration is then applied on the interval $[\lambda_l, \lambda_u] = [0.20, 0.85]$, starting in $x_{0,k',0.25}$ and $x_{0,k'',0.8}$ and moving in positive (forward) and negative (backward) direction, respectively. In both cases, a step size of $h = 0.01$ is used in the numerical integration.
In Figure 6.10a, the outcome vectors obtained from forward and backward integration and the results of Chapter 5 are compared in the outcome space. Here, the solutions of the gradient descent of weighted sum scalarizations for different weights are illustrated as green points, where the rightmost point on the curve corresponds to $x_{0,k'';l,0.8}$ and where the leftmost point corresponds to $x_{0,k';l,0.25}$, respectively. Most solutions of $x_{k'}(\lambda)$



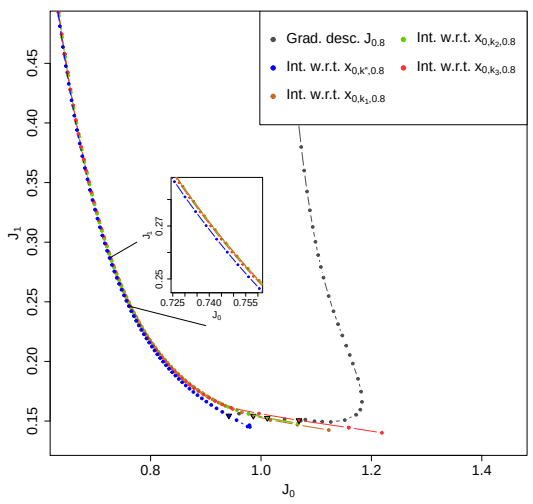
(a) $\lambda = 0.25$; $x_{0,k';l,0.25}$ (b) $\lambda = 0.4$ (c) $\lambda = 0.6$ (d) $\lambda = 0.8$; $x_{0,k'';l,0.8}$

Figure 6.9: Exemplary solutions of the weighted sum method in Chapter 5, including $x_{0,k';l,0.25}$ and $x_{0,k'';l,0.8}$. See also [19].

with initial value $x_{0,k';l,0.25}$ (purple trajectory) are dominated by weighted sum solutions, while the approximative front of weighted sum solutions (obviously excluding $x_{0,k'';l,0.8}$) is dominated by the solutions of $x_{k'';l}(\lambda)$ with initial value $x_{0,k'';l,0.8}$ (blue trajectory). In Figure 6.11, the corresponding shapes to the results of $x_{k'';l}(\lambda)$ are shown. They also resemble the profiles of whales and are therefore consistent with the weighted sum solutions. Furthermore, in Figure 6.12 the behaviour of the B-spline coefficients $x$ of the results of $x_{k'';l}(\lambda)$ for $\lambda \in [0.25, 0.8]$ is shown. One can observe that the meanline coefficients $x_2^{\text{ml}}$ (red), $x_3^{\text{ml}}$ (brown) and the thickness coefficients $x_3^{\text{th}}$ (purple), $x_2^{\text{th}}$ (blue) stay nearly the same during the numerical integration, while $x_2^{\text{th}}$ also slightly increases for bigger $\lambda$. The two coefficients corresponding to the first B-spline on the left boundary, $x_1^{\text{ml}}$ (orange) and $x_1^{\text{th}}$ (green), are the two most sensitive coefficients as they decrease with smaller $\lambda$. Since all shapes resemble whale profiles one can assume that these two variables have the most influence on the volume of the solutions. It is also of interest to investigate the behavior of the solutions when sub optimal initial values obtained from prematurely stopped gradient descents of the weighted sum scalarization in Chapter 5 are used in the Pareto tracing by numerical integration algorithm. The trajectories of three additional numerical integrations starting in suboptimal initial solutions $x_{0,k_1;l,0.8}, x_{0,k_2;l,0.8}$, and $x_{0,k_3;l,0.8}$, with corresponding initial values $\lambda_{0,k_1} \approx 0.808, \lambda_{0,k_2} \approx 0.810$, and $\lambda_{0,k_3} \approx 0.814$, respectively, are shown in Figure 6.10b. To solve the ODEs backward numerical integration with a step size of $h = 0.01$ is applied on $[\lambda_{l,k_i}, \lambda_{u,k_i}] = [\lambda_{0,k_i} - 0.55, \lambda_{0,k_i}]$, $i = 0, 1, 3$, respectively. Here, the gray dots resemble some iterates of the gradient descent method applied to the weighted sum objective $J_{0.8}$. Regardless of the suboptimal choices of the initial values, one can observe that the solutions $x_{0,k_1;l,0.8}(\lambda)$ (brown), $x_{0,k_2;l,0.8}(\lambda)$ (green), and $x_{0,k_3;l,0.8}(\lambda)$ (red) still represent good approximations of the (local) Pareto front, see Figures 6.10b and 6.13. This is not totally surprising, since we observed in Chapter 5 that the gradient descent algorithm applied to the weighted sum objectives $J_\lambda$ first approaches an extension of the Pareto front by improving substantially w.r.t. $J_1$, to then move along almost in parallel to the Pareto front while the trade-off between the potential improvements w.r.t. $J_0$ and $J_1$ changes in favor of $J_0$ during later stages of the optimization. Therefore, the premature solutions $x_{0,k_1;l,0.8}, x_{0,k_2;l,0.8}$, and $x_{0,k_3;l,0.8}$ have better $J_1$-values and thus themselves approximate an extension of the Pareto front, providing excellent starting points for the numerical integration. Note, however, that this




(a) Pareto front tracing by numerical integration started in $x_{0,k',0.25}$ (purple) and $x_{0,k'',0.8}$ (blue), compared to the weighted sum results from [46] (green). The initial solutions are marked as (green) triangles.




(b) Pareto front tracing by numerical integration (backward) started in premature solutions $x_{0,k_1,0.8}$ (brown), $x_{0,k_2,0.8}$ (green) and $x_{0,k_3,0.8}$ (red), compared to the results for $x_{0,k'',0.8}$ (blue; c.f. left figure). The initial solutions are marked as (brown/-green/red/blue) triangles.

Figure 6.10: Comparison of the outcome vectors obtained with Pareto tracing by numerical integration using forward and backward integration (left) and starting from suboptimal initial solutions (right). See also [19].

may not be true in general, since this behaviour largely depends on the initial solution and, even more so, on the relative variability (slopes) of the considered objective functions $J_1$ and $J_0$. It is noticeable that this is a problem-specific observation and should not be expected in general, since, for example, the quadratic case, see Figure 6.1a, behaves differently. Our numerical investigation suggests that the direction of integration matters as we observe drifting away of $(J_0, J_1)$-values from the Pareto front in one $\lambda$-direction and narrowing in the opposite one. This does not come as a surprise, as our algorithm rather provides guarantees for the $\varepsilon$-criticality of $x_k(\lambda)$ but not (directly) for $\|J(x_k(\lambda)) - J(x(\lambda))\|$. In practical applications, it might therefore be interesting to measure the spread of $J$-trajectories, i.e., $\|J(x_k(\lambda)) - J(x_{k'}(\lambda))\|$, close to the initial value $\lambda_0$ in order to identify favorable values for $\lambda_0$ along with a direction of integration that is (initially) contractive in $J$-space.

Moreover, it is apparent that in the above examples Pareto tracing by numerical integration yields a more dense approximation of the (local) Pareto front than the gradient based methods investigated in Chapter 5. This density depends on the choice of the weights for which weighted sum problems are solved in Chapter 5 ([46]), and on the choice of the step length $h$ in the Pareto tracing by numerical integration method, respectively. Note that it is generally difficult to control the distribution of points when solving individual weighted sum problems (see, for example, [38]). The selection of the step length

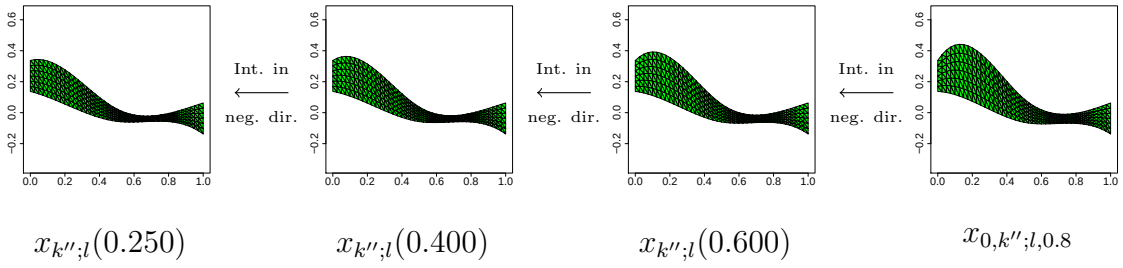



Figure 6.11: Some shapes obtained with backward Pareto tracing by numerical integration in $x_{0,k'';l,0.8}$. See also [19].

$h$ plays a corresponding role for our method. However, the error estimates that we provide give some means of control in the case of the Pareto front tracing by numerical integration method. Moreover, small step lengths induce dense approximations, however, these approximations come at comparably high computational costs. Larger step sizes on the other hand allow to quickly obtain a rough estimate of the Pareto front with rather few and distant solutions. So, it is of particular interest to determine the sensitivity of the Pareto tracing by numerical integration method w.r.t. the step length $h$, and especially for larger values of $h$. Toward this end, we compare the results of three further Pareto tracing by numerical integration solves starting in $x_{0,k'';l,0.8}$ for different step sizes $h = 0.001, 0.04, 0.08$, see Figure 6.14a. Assuming divisibility among the considered step lengths we can observe that the solutions obtained for a larger step size are approximately equal to a subset of the outcome vectors obtained for smaller step sizes. Hence, for this particular problem this enables for a relatively coarse representation of the (local) Pareto front by using a relatively large step length. This was also observed for the simpler Test Case 1. Furthermore, the distance between two consecutive outcome vectors on the approximated (local) Pareto front may differ significantly for a constant step length $h$. This is in accordance with the fact that equidistantly spaced weights $\lambda \in (0, 1)$ do in general not yield equidistantly spaced outcome vectors on the Pareto front, recall Section 3.5.

From a practical point of view, when dealing with computationally expensive problems like the given biobjective shape optimization problem (3.11) rough approximations of the Pareto front are of particular interest. Recall from Chapter 5, that computing one weighted sum solution for a weight $\lambda \in (0, 1)$ starting in the same initial shape came with the cost of $k_W + 1$ gradient computations and $k_W \cdot k_A + 1$ objective function evaluations, where $k_W$ denotes the number of iterations of the gradient descent algorithm and $k_A$ denotes the number of Armijo iterations. One optimization run for Test Case 2 needed on average 106.7 iterations, and per iteration on average 5.3 Armijo iterations to compute a solution for a given weight, i.e., 107.7 gradient computations and 566.5 objective function evaluations in total. In contrast, the Pareto tracing by numerical integration algorithm needs only 14 gradient computations and one objective function evaluation, i.e., 29 PDE evaluations, to compute one further solution, if a sufficiently good initial solution is at hand. This is significantly cheaper than using one common initial shape for all weights $\lambda$ of the weighted sum descent as investigated in Chapter 5 (see also [46]). But one can easily speedup the weighted sum descent by choosing the optimal solution for weight $\lambda_i$ as the initial solution for the next weight $\lambda_{i+1} = \lambda_i \pm h$. The presented approach is also

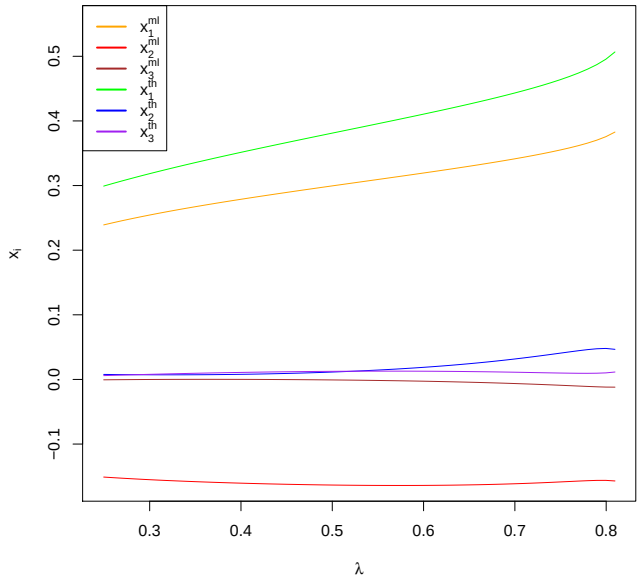
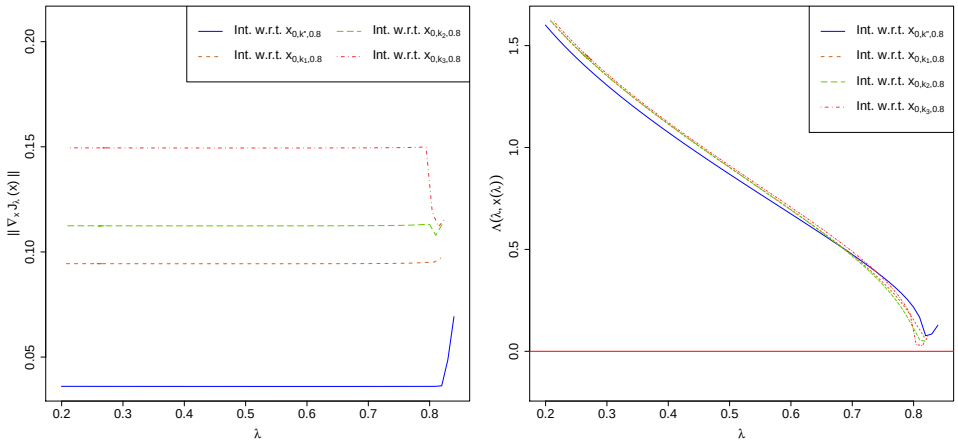



Figure 6.12: Test Case 2: Behavior of the B-spline coefficients $x$ of the results of $x_{k'';l}(\lambda)$ obtained with Pareto tracing by numerical integration for $\lambda \in [\lambda_l, \lambda_u] = [0.25, 0.8]$.

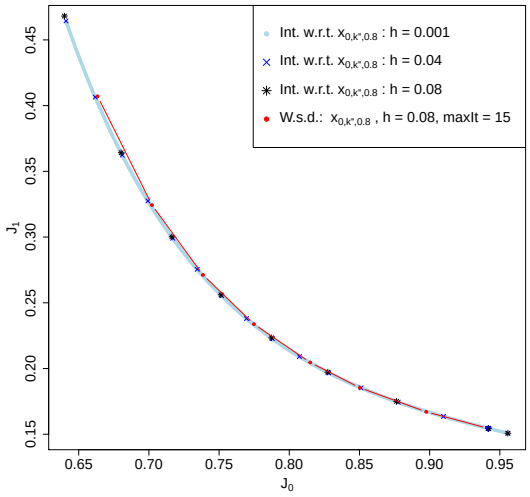
compared with the case where the optimal shape for $\lambda_0 = 0.8$, $x_{0,k'',0.8}$, is used as the initial solution for a weighted sum descent w.r.t. $\lambda_1 = \lambda_0 - h$ and the solution of that descent is then used as an initial value to compute the next solution for $\lambda_2 = \lambda_1 - h$ and so on. This scheme is then repeated with a step length of $h = 0.08$ until a solution for the weight $\lambda = 0.24$ is computed. Further, to better compare this approach with the Pareto front tracing by numerical integration solutions obtained by backward numerical integration starting in $x_{0,k'',0.8}$ utilizing a step size of $h = 0.08$, we set the maximum iteration number of the weighted sum descents as 15 according to the needed 29 PDE evaluations (14 gradient computations and one objective function evaluation) per iteration of Pareto front tracing by numerical integration, i.e., the on average 5.3 Armijo iterations per iteration of the weighted sum descent are not counted as PDE evaluations for the weighted sum descent approach. The red points in Figure 6.14a illustrate the solutions of this scheme. One can observe that the solutions of these consecutive descents with the weighted sum scalarization do not cover the same range of solutions as the Pareto front tracing by numerical integration solutions and lose accuracy during the scheme. Further, setting the step length $h = 0.04$, i.e., choosing a more dense weight distribution, yields a more accurate approximation, but when limited to 15 iterations also fails to cover the same range of solutions as Pareto front tracing by numerical integration with the same step length. Increasing the maximal iteration number to 50 for both step lengths $h = 0.04$ and $h = 0.08$ improves the solutions and their range, respectively (see Figure 6.14b), but none of the descents managed to converge in the given 50 iterations. This nicely shows that for the Pareto front tracing by numerical integration approach, the number of PDE evaluations is independent of the step length $h$, while the error bounds of Theorems 6.34 and 6.37 hold in each step. In contrast, the weighted sum descent is slow to converge



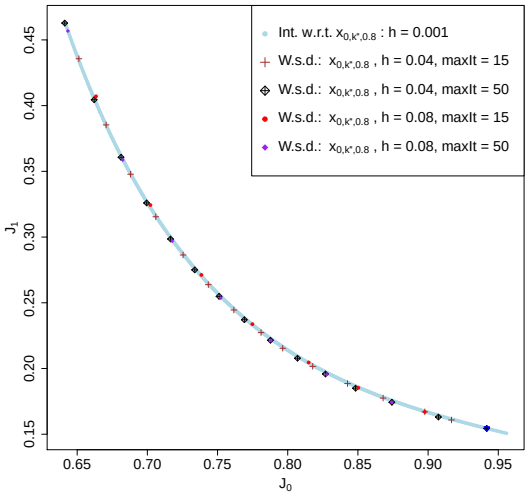

(a) first-order optimality: $\|\nabla_x J_\lambda(x)\|$ (b) second-order optimality: smallest eigenvalue $\Lambda(\lambda, x(\lambda))$ of $\nabla_x^2 J_\lambda(x)$

Figure 6.13: S-Shaped Joint: Tracking first and second-order optimality during Pareto tracing by numerical integration in reliance of the quality of the initial value. See also [19].

and may lose accuracy during the scheme.
All in all, this is a significant speedup that, in combination with the robustness w.r.t. the step length and the guarantee of the error bounds of Theorems 6.34 and 6.37, allows for a reasonably good approximation of a wide range of solutions at reasonable computational cost. Furthermore, in the case that a decision maker is involved, the robustness w.r.t. the step length allows one to first compute a coarse representation of the Pareto front and then in a next step generate a denser representation of the part of the solution front that aligns the most with the preferences of the decision maker.




(a) Comparison of the outcome vectors obtained with Pareto front tracing by numerical integration (backward) started in $x_{0,k'',0.8}$ (blue triangle) with different step lengths $h = 0.001, 0.04, 0.08$ and the outcome vectors of successive weighted sum descents also started in $x_{0,k'',0.8}$ with step length $h = 0.08$.




(b) Comparison of the outcome vectors obtained with successive weighted sum descents started in $x_{0,k'',0.8}$ (blue triangle) with different step lengths $h = 0.04, 0.08$ and maximum iteration numbers 15 and 50. The Pareto front tracing by numerical integration results w.r.t. $h = 0.001$ (light blue, c.f. left figure) are added for reference.

Figure 6.14: Comparison of the outcome vectors obtained with Pareto front tracing by numerical integration and successive weighted sum descents, where both approaches started in the same initial shape $x_{0,k'',0.8}$. See also [19].

# 7 EGO and Gradient Enhanced Kriging

In this chapter, the solutions of the weighted sum gradient descent of Chapter 5 and the solutions of Pareto tracing using numerical integration of Chapter 6 are compared with weighted sum solutions computed via the global surrogate based optimization method *Efficient Global Optimization (EGO)* utilizing *Kriging* and *Gradient Enhanced Kriging (GEK)* as surrogate models, respectively. For expensive to compute objective functions surrogate based optimization methods provide a way to, in comparison, cheaply obtain an optimum for the use case. The central idea of surrogate based optimization is that the optimization takes place on a surrogate model that approximates the expensive objective function by a cheap to evaluate objective function. A thorough introduction to this field and an incorporation into multiobjective optimization can be found in [123]. The history of surrogate based optimization in multiobjective optimization is discussed in [143]. We choose the Efficient Global Optimization algorithm, see, e.g., [92, 90], that utilizes Kriging [98, 105, 130, 34, 35] or GEK [111] as surrogate models as benchmarks, respectively, since they are widely used in applications [157]. The Kriging predictor is an interpolator and a *best linear unbiased predictor*, [130, 92]. ParEGO [96, 36] extends the EGO algorithm to the multiobjective case. In [155], a multiobjective gradient descent is applied on a Kriging model. A common field in which the EGO algorithm is applied, is the gradient-free optimization of aerodynamic design problems [117, 102, 11, 94, 63, 64, 133]. If cheap to compute gradients are available, e.g., from an adjoint approach, they also have been incorporated into surrogate models for aerospace design problems [31, 8, 82, 157, 12]. In [111], the *Direct Gradient Enhanced Kriging* approach was first established, and further developed, see, e.g., [65, 22, 37]. Note that GEK methods are not as commonly used as ordinary Kriging, since cheap gradients are needed and the numerical stability of GEK methods is an issue [157]. The novelty of this work is that now that adjoint gradients for structural mechanic problems of ceramic components are available, see Chapter 3, a GEK approach can be applied on the biobjective shape optimization problem (3.11). Furthermore, since EGO and GEK are widely used in (aerodynamic design) applications they provide proper benchmarks for the numerical results from the Chapters 5 and 6. We use the EGO, Kriging and GEK implementations of the open source optimization and uncertainty quantification software toolbox *Dakota* of Sandia National Laboratories [56, 57, 37]. This chapter is structured as follows. In Section 7.1, a brief overview of random variables, random fields, in particular Gaussian random fields, and their covariance functions is given. In Section 7.2, the continuity and differentiability of random fields is discussed. In Section 7.3, the Kriging and GEK models are introduced and the EGO algorithm is stated. Next, the Dakota toolbox and the coupling is described in Section 7.5. Subsequently, the numerical results are compared in Section 7.6.

## 7.1 Random Variables and Random Fields

From a probabilistic point of view, outcomes of experiments are handled as *random variables* which are uncertain quantities that are observed. A *random field*, as a family of random variables, can be associated with a simulation in which many experiments can be done. Therefore, observing the outcome of the random field is equivalent to observing all experiments [150].
To analyze random fields, tools from elementary probability theory are needed which are presented in this section. A definition of a random variable and afterwards the definition of a random field and some basic insights from *probability theory* are provided. Note that most of this section is based on [95, 150, 1].

**Definition 7.1** (Random Variable)**.** *Let $(\Omega, \mathcal{A}, P)$ be a probability space and $(E, \mathcal{F})$ a measurable space. A measurable function $Z : (\Omega, \mathcal{A}, P) \to (E, \mathcal{F})$ is called a $E$-valued* random variable *and $Z(\omega)$ with $\omega \in \Omega$ is called a* realization *of $Z$.*

Every random variable has a distribution. In the later parts of this section, we fix the distribution as Gaussian.

**Definition 7.2** (Distribution of a Random Variable)**.** *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(E, \mathcal{F})$ a measurable space and $Z : (\Omega, \mathcal{A}, P) \to (E, \mathcal{F})$ a random variable. Then*

$$P_Z(B) := P(Z^{-1}(B)) = P(Z(\omega) \in B) \quad \text{for } B \in \mathcal{F} \tag{7.1}$$

*defines a probability measure $P_Z$ on $(E, \mathcal{F})$ and is called the* distribution of $Z$ under $P$.

For real random variables, i.e., $(E, \mathcal{F}) = (\mathbb{R}^k, \mathcal{B}^k)$, $k \in \mathbb{N}$, the probability of the event $\{Z \leq z\}$ has a special meaning in probability theory. It defines the *(cumulative) distribution function of $Z$*. Recall that for $x, y \in \mathbb{R}^k$ the inequality $x \leq y$ is equivalent to $x_i \leq y_i$ for all $i = 1, \ldots, k$.

**Definition 7.3** (Cumulative Distribution Function (c.d.f))**.** *Let $(\Omega, \mathcal{A}, P)$ be a probability space and $(\mathbb{R}, \mathcal{B})$ and $(\mathbb{R}^k, \mathcal{B}^k)$ measurable spaces.*

1. *Let $Z : (\Omega, \mathcal{A}, P) \to (\mathbb{R}, \mathcal{B})$ be a random variable. The function*

$$\begin{aligned} F \equiv & F_Z : \mathbb{R} \to [0, 1] \\ z \mapsto & F_Z(z) := P(Z \leq z) \end{aligned} \tag{7.2}$$

   *is called the* (cumulative) distribution function of $Z$.

2. *Let $Z = (Z_1, \ldots, Z_k) : (\Omega, \mathcal{A}, P) \to (\mathbb{R}^k, \mathcal{B}^k)$ be a random vector. The function*

$$\begin{aligned} & F \equiv F_Z \equiv F_{Z_1,\ldots,Z_k} : \mathbb{R}^n \to [0, 1] \\ & z = (z_1, \ldots, z_k) \mapsto F_Z(z) := P(Z_1 \leq z_1, \ldots, Z_k \leq z_k) \end{aligned} \tag{7.3}$$

   *is called the* (cumulative) distribution function of the random vector $Z$.

If the cumulative distribution function is differentiable, then the *probability density function* exists:

**Definition 7.4** (Probability Density Function (p.d.f.)). *Let $(\Omega, \mathcal{A}, P)$ be a probability space and let $(\mathbb{R}^k, \mathcal{B}^k)$ be a measurable space.*

1. *Let $Z = (Z_1, \ldots, Z_k) : (\Omega, \mathcal{A}, P) \to (\mathbb{R}^k, \mathcal{B}^k)$ be a random vector. The function $f \equiv f_Z = f_{Z_1,\ldots,Z_k}$ is called the* probability density function of the random vector $Z$ *if*

$$F_Z(z_1, \ldots, z_k) = \int_{-\infty}^{z_1} \ldots \int_{-\infty}^{z_k} f_{Z_1,\ldots,Z_k}(y_1, \ldots, y_k)\ dy_1 \ldots dy_k. \tag{7.4}$$

2. *Let $f \equiv f_Z = f_{Z_1,\ldots,Z_k}$ be a probability density function. The* marginal probability density function of $z_i$ *for $i \in S \subseteq \{1, \ldots, k\}$, is obtained by integrating over all possible values of $z_j$ with $j \notin S$*

$$f(z_i | i \in S) \equiv f_Z(z_i | i \in S) = \int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} f_{Z_1,\ldots,Z_k}(z_1, \ldots, z_k) \prod_{j \notin S} dz_j. \tag{7.5}$$

One can further construct a conditional sample space by imposing conditions on random variables. We state some general definitions and theorems for a random vector $Z$ that is partitioned into two vectors $Z^1$ and $Z^2$. For more detail, we refer to [95]

**Definition 7.5** (Conditional c.d.f. and p.d.f.). *Let $Z = (Z_1, \ldots, Z_k) \in \mathbb{R}^k$ be a random vector that is partitioned into two vectors $Z^1 = (Z_1, \ldots, Z_p) \in \mathbb{R}^p$ and $Z^2 = (Z_{p+1}, \ldots, Z_k) \in \mathbb{R}^q$ and let further be $z^1 \in \mathbb{R}^p$ and $z^2 \in \mathbb{R}^q$ be realizations of the random vectors $Z^1$ and $Z^2$, respectively. Given a condition $\{Z^1 = z^1\}$ with $P(\{Z^1 = z^1\}) > 0$ we then have:*

1. *The* conditional c.d.f. of $Z^2$ given $\{Z^1 = z^1\}$ *is given by*

$$F_{Z^2|Z^1}(z^2|z^1) := P(Z^2 \leq z^2 | Z^1 = z^1). \tag{7.6}$$

2. *The corresponding* conditional p.d.f. $f_{Z^2|Z^1}(z^2|z^1)$ *exists if*

$$\begin{aligned} F_{Z^2|Z^1}(z^2|z^1) &= \int_{-\infty}^{z_{p+1}} \ldots \int_{-\infty}^{z_k} f_Z(z_1, \ldots, z_p, y_{p+1}, \ldots, y_k) \prod_{j=p+1}^{k} dy_j \\ &= \int_{-\infty}^{z_{p+1}} \ldots \int_{-\infty}^{z_k} f_{Z^2|Z^1}(y^2|z^1) \prod_{j=p+1}^{k} dy_j. \end{aligned} \tag{7.7}$$

3. *We have the following relation*

$$f_Z(z^1, z^2) = f_{Z^2|Z^1}(z^2|z^1) f_Z(z^1) = f_{Z^1|Z^2}(z^1|z^2) f_Z(z^2). \tag{7.8}$$

4. *Bayes' theorem for conditional probabilities (see for e.g. [150]) can then be rewritten as*

$$f_{Z^2|Z^1}(z^2|z^1) = \frac{f_Z(z^1, z^2)}{f_Z(z^1)} = \frac{f_{Z^1|Z^2}(z^1|z^2) f_Z(z^2)}{f_Z(z^1)}. \tag{7.9}$$

5. *Two real random vectors $Z^1$ and $Z^2$ are called* independent *if*

$$F_Z(z) = F_{Z^1}(z^1) F_{Z^2}(z^2). \tag{7.10}$$

*Further, if the p.d.f.ś $f_Z, f_{Z^1}, f_{Z^2}$ exist one has equivalently*

$$f_Z(z) = f_{Z^1}(z^1) f_{Z^2}(z^2). \tag{7.11}$$

Note that one can replace the vector $z^2$ with the random vector $Z^2$, transforming the conditional p.d.f. to a *function of the random vector* $Z^2$. A formal definition of a *random field* can be stated as:

**Definition 7.6** (Random Field)**.** *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(E, \mathcal{F})$ a measurable space, and $\mathcal{X}$ a non-empty parameter set. A* random field *is a indexed family $Z := \big(Z(\omega, x)\big)_{x \in \mathcal{X}}$ of $E$-valued random variables, i.e., for every fixed $x \in \mathcal{X}$, $Z(\cdot, x)$ is a measureable function of $\omega \in \Omega$. So, we have*

$$\begin{aligned} Z(\cdot, x) : (\Omega, \mathcal{A}, P) &\to (E, \mathcal{F}) \\ \omega &\mapsto Z(\omega, x). \end{aligned} \tag{7.12}$$

*For notational reasons the dependency on the underlying probability space will be omitted*

$$Z_x := Z(x) := Z(\cdot, x), \ x \in \mathcal{X}.$$

*Furthermore, for $x^i \in \mathcal{X}$, $i \in \mathbb{N}$, we write*

$$Z_i := Z_{x^i}.$$

*For a fixed $\omega \in \Omega$, $Z(\omega, x)$ is a deterministic function of $x$ which is called a* sample path *and is denoted by $z_x$*

$$\begin{aligned} z_\cdot(\omega) : \mathcal{X} &\to (E, \mathcal{F})^{\mathcal{X}} \\ x &\mapsto z_x := z_x(\omega) := Z(\omega, x). \end{aligned} \tag{7.13}$$

*Analogously, for $x^i \in \mathcal{X}$, $i \in \mathbb{N}$, we write*

$$z_i := z_{x^i}.$$

The parameter set $\mathcal{X}$ can in principle be a far more general set than $\mathbb{R}^n$. But in this work we assume that the parameter set is a *linear space* and choose $\mathcal{X} \subseteq \mathbb{R}^n$.

### 7.1.1 Finite-Dimensional Distributions

In this subsection, *finite-dimensional distributions* of random fields are discussed. Now, that we have seen the c.d.f. and p.d.f of random variables and random vectors, we will extend this for random fields. The parameter set $\mathcal{X} \subseteq \mathbb{R}^n$ of a random field in general has an infinite and uncountable amount of points. To describe the distribution of random fields, one uses its *finite-dimensional (cumulative) distributions*.

**Definition 7.7** (Finite-dimensional Distributions)**.** *Let $Z = \big(Z(x)\big)_{x\in\mathcal{X}}$ be a real valued random field with $\mathcal{X} \subseteq \mathbb{R}^n$. Then, for $k \in \mathbb{N}$ its* finite-dimensional (cumulative) distributions *are defined as*

$$F_{x^1,\dots,x^k}(z_1,\dots,z_k) = P(Z_1 \leq z_1,\dots,Z_k \leq z_k), \tag{7.14}$$

*where $\{x^1,\dots,x^k\} \subseteq \mathcal{X}$ and $z_1,\dots,z_k \in \mathbb{R}$.*

Finite-dimensional distributions of random fields have to satisfy two *consistency requirements.*

1. *Symmetry Condition*: For every permutation $\pi$ of the set $\{1,\dots,k\}$

$$F_{x^1,\dots,x^k}(z_1,\dots,z_k) = F_{x_{\pi 1},\dots,t_{\pi k}}(z_{\pi 1},\dots,z_{\pi k}) \tag{7.15}$$

   must hold.

2. *Compatibility condition*:

$$F_{x_1,\dots,x_{k-1}}(z_1,\dots,z_{k-1}) = F_{t_1,\dots,t_{k-1},t_k}(z_1,\dots,z_{k-1},\infty) \tag{7.16}$$

   has to be satisfied.

Conversely, if there exist distribution functions that satisfy (7.15) and (7.16), then there also exists a random field having these distributions. This is the result of the following famous theorem.

**Theorem 7.8** (Kolmogorov's Existence Theorem)**.** *If a system of finite-dimensional distributions, $F_{x^1,\dots,x^k}$, satisfies the consistency requirements* (7.15) *and* (7.16)*, then there exists on some probability space $(\Omega,\mathcal{A},P)$ a random field $\big(Z(x)\big)_{x\in\mathcal{X}}$ having $F_{x^1,\dots,x^k}$ as its finite-dimensional distributions.*

For a proof we refer to [1, 16].

### 7.1.2 Expected Value and Covariance

For this subsection we assume, that we have a random field $Z = \big(Z(x)\big)_{x\in\mathcal{X}}$ and a non-empty parameter set $\mathcal{X} \subseteq \mathbb{R}^n$ at hand. Recall, that for a given $x \in \mathcal{X}$, $Z(x)$ is a real valued random variable. The expected value at a location $x \in \mathcal{X}$ for a random field is defined as a *expectation function* $m(\cdot)$.

**Definition 7.9** (Expected Value)**.** *Let $Z(x)$ be a real valued random variable with p.d.f. $f_{Z(x)}$. If $\int_{-\infty}^{+\infty} |z|\, f_{Z(x)}(z)\, dz < \infty$ then the* expected value of $Z(x)$ *is defined as*

$$m(x) = \mathbb{E}[Z(x)] = \int_{-\infty}^{+\infty} z\, f_{Z(x)}(z)\, dz. \tag{7.17}$$

*Let $Z = (Z_1, \ldots, Z_k) = (Z(x^1), \ldots, Z(x^k))$ be a real random vector. Then, the* expected value of $Z$ *is given by*

$$\mathbb{E}[Z] = \mathbb{E}[(Z_1, \ldots, Z_k)] = (\mathbb{E}[Z_1], \ldots, \mathbb{E}[Z_k])^\top. \tag{7.18}$$

Further, one has a *covariance function* $C(\cdot, \cdot)$ for the covariance of the values of a random field at two locations.

**Definition 7.10** (Covariance)**.** *Let $Z_1 = Z(x^1)$ and $Z_2 = Z(x^2)$ be two real valued random vectors. Then, the* covariance of $Z_1$ and $Z_2$ *is defined as*

$$C(x^1, x^2) = Cov(Z(x^1), Z(x^2)) = Cov(Z_1, Z_2) = \mathbb{E}[Z(x^1)Z(x^2)] - \mathbb{E}[Z(x^1)]\,\mathbb{E}[Z(x^2)], \tag{7.19}$$

*further, the* variance of $Z_1$ *is defined as*

$$Var(x^1) = Var(Z_1) = Cov(Z_1, Z_1) = \sigma(x^1)^2 = \mathbb{E}[Z(x^1)^2] - \mathbb{E}[Z(x^1)]^2. \tag{7.20}$$

*Let $Z = (Z(x^1), \ldots, Z(x^k)) = (Z_1, \ldots, Z_k)$ be a random vector. Then, the by definition symmetric and positive definite* covariance matrix of $Z$ *is given by*

$$\Sigma_Z = \begin{bmatrix} C(x^1, x^1) & \ldots & C(x^1, x^k) \\ \vdots & \ddots & \vdots \\ C(x^k, x^1) & \ldots & C(x^k, x^k) \end{bmatrix} = \begin{bmatrix} Cov(Z_1, Z_1) & \ldots & Cov(Z_1, Z_k) \\ \vdots & \ddots & \vdots \\ Cov(Z_k, Z_1) & \ldots & Cov(Z_k, Z_k) \end{bmatrix}. \tag{7.21}$$

**Definition 7.11** (Correlation)**.** *Let $Z_1 = Z(x^1)$ and $Z_2 = Z(x^2)$ be two real valued random variables. Then, the* correlation of $Z_1$ and $Z_2$ *is defined as*

$$\rho(Z_1, Z_2) = \frac{C(Z_1, Z_2)}{\sigma(Z_1)\ \sigma(Z_2)} = \frac{\mathbb{E}[Z_1 Z_2] - \mathbb{E}[Z_1]\mathbb{E}[Z_2]}{\sqrt{\mathbb{E}[Z_1^2] - \mathbb{E}[Z_1]^2} \cdot \sqrt{\mathbb{E}[Z_2^2] - \mathbb{E}[Z_2]^2}}. \tag{7.22}$$

The Kriging model makes use of *conditional expectations* for predictions.

**Definition 7.12** (Conditional Expectation [146])**.** *Let $Z^1$ and $Z^2$ be $(E, \mathcal{F}) = (\mathbb{R}^k, \mathcal{B}^k)$ valued random vectors. Furthermore, let $z^1, z^2 \in \mathbb{R}^k$ and let $f_{Z^1|Z^2}(z^1|Z^2 = z^2)$ be the conditional p.d.f. under the condition $\{Z^2 = z^2\}$. The* conditional expectation of $Z^1$ given $Z^2$ *is defined as*

$$\begin{aligned} m_{Z^1|Z^2} = \mathbb{E}[Z^1|Z^2 = z^2] &= \int_{-\infty}^{\infty} z^1\, f_{Z^1|Z^2}(z^1|Z^2 = z^2)\, dz^1 \\ &= \int_{-\infty}^{\infty} z^1\, \frac{f_{Z^1,Z^2}(z^1, z^2)}{f_{Z^2(z^2)}}\, dz^1 \end{aligned} \tag{7.23}$$

Since the conditional expectation depends on $Z^2$, it itself is also a random vector. It becomes a constant if $Z^2$ takes a specific value.

### 7.1.3 Positive Definiteness

For Gaussian random fields the concept of *positive definiteness* is crucial. To establish consistent finite-dimensional distributions, the positive definiteness of the covariance function gives a necessary and sufficient condition. Now, recall the definition of positive definiteness of functions.

**Definition 7.13.** *Let $k \in \mathbb{N}$, and let $x^i \in \mathcal{X}$ and $c_i \in \mathbb{R}$ for $i = 1, \dots, k$. Then, a function $C$ on $\mathcal{X} \otimes \mathcal{X}$ is called* positive definite *on $\mathcal{X}$ if*

$$\sum_{i=1}^{k} \sum_{j=1}^{k} c_i c_j C(x^i, x^j) \geq 0$$

*for any choice of $k$, $\{x^1, \dots, x^k\}$ and $\{c_1, \dots, c_k\}$.*

For a random field $\big(Z(x)\big)_{x \in \mathcal{X}}$ with covariance function $C$ consider $k > 0$ arbitrary random variables $\{Z(x^1), \dots, Z(x^k)\}$. We then have for arbitrary $c_i \in \mathbb{R}$, $i = 1, \dots, k$,

$$Var(c_1 Z(x^1) + \cdots + c_k Z(x^k)) = \sum_{i=1}^{k} \sum_{j=1}^{k} c_i c_j C(x^i, x^j) \geq 0.$$

The following theorem and its corollary characterize the class of covariance and correlation functions, respectively. For proofs we refer to [1].

**Theorem 7.14.** *The class of covariance functions coincide with the class of positive definite functions.*

**Corollary 7.15.** *The class of correlation functions coincide with the class of positive definite functions where $C(x, x) = 1$.*

### 7.1.4 Gaussian Random Fields

In this subsection Gaussian random fields and symmetry properties are introduced. Recall, that we chose $\mathcal{X} \subseteq \mathbb{R}^n$ for the parameter set.

**Definition 7.16** (Multivariate Normal Distribution)**.** *A random vector $Z = (Z_1, \dots, Z_k)$ of $k$ Gaussian random variables is characterized by the multivariate normal p.d.f.*

$$f_Z(z) = (2\pi)^{-k/2} \, |\Sigma_Z|^{-1/2} \exp\Big\{-\frac{1}{2}(z - m_Z)^\top \, {\Sigma_Z}^{-1} \, (z - m_Z)\Big\}, \tag{7.24}$$

*where $m_Z$ is the vector of mean values and $\Sigma_Z$ the $k \times k$ covariance matrix.*

**Theorem 7.17** (Multivariate Gaussian Characteristic Function)**.** *Let $Z$ be a multivariate Gaussian random variable with expectation $m_Z$ and covariance matrix $\Sigma_Z$. Then, the* characteristic function of $Z$ *is given by*

$$\phi_Z(u) = \exp\Big\{iu^\top m_Z - \frac{1}{2} u^\top \Sigma_Z u\Big\}. \tag{7.25}$$

And the *conditional multivariate normal distribution* is given by:

**Definition 7.18** (Conditional Multivariate Normal Distribution)**.** *Let $Z$ be a vector of $k$ Gaussian variables and assume that it is partitioned into two vectors $X$ and $Y$ with dimensions $k_1$ and $k_2$, respectively. Then, we have*

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix}, \quad m_Z = \begin{pmatrix} m_X \\ m_Y \end{pmatrix}, \quad \text{and} \quad \Sigma_Z = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}. \tag{7.26}$$

*The* conditional Gauss p.d.f. of $Y$ given $X$, $(Y|X)$, *is then given by*

$$f(y|X) = (2\pi)^{-k_2/2} \left|\Sigma_{Y|X}\right|^{-1/2} \exp\Big\{-\frac{1}{2}(y - m_{Y|X})^\top \, {\Sigma_{Y|X}}^{-1} \, (y - m_{Y|X})\Big\}, \tag{7.27}$$

*where $m_{Y|X} = \mathbb{E}[Y|X]$ is the conditional expectation of $Y$ given $X$ and $\Sigma_{Y|X}$ the conditional covariance matrix. We have*

$$Y|X \sim \mathcal{N}(m_{Y|X},\ \Sigma_{Y|X}), \tag{7.28}$$

$$m_{Y|X} = m_Y + \Sigma_{YX}\ {\Sigma_X}^{-1}\ (X - m_X), \tag{7.29}$$

$$\Sigma_{Y|X} = \Sigma_Y - \Sigma_{YX}\ {\Sigma_X}^{-1}\ \Sigma_{XY}. \tag{7.30}$$

A complete deduction is provided in [150]

**Definition 7.19** (Gaussian Random Field)**.** *A real valued random field $Z = \big(Z(x)\big)_{x\in\mathcal{X}}$, i.e.*

$$\begin{aligned} Z(\cdot, x) : (\Omega, \mathcal{A}, P) &\to (\mathbb{R}, \mathcal{B}) \\ \omega &\mapsto Z(\omega, x), \end{aligned} \tag{7.31}$$

*is called a* Gaussian random field*, if all of its finite-dimensional distributions $F_{x^1,\dots,x^k}$, $\{x^1, \dots, x^k\} \subseteq \mathcal{X}$, $k > 0$, are multivariate normal distributions.*

Like the multivariate normal distribution a Gaussian random field is fully characterized by its first two moments, i.e., the expectation function $m(x)$ and the covariance function $C(x^1, x^2)$. Therefore, following [1] and Subsection 7.1.3, it is sufficient to restrict the covariance function $C$ to be positive definite to ensure that the consistency requirements (7.15) and (7.16) hold and consequently that the Gaussian random field is well defined. To simplify the verification of positive definiteness, we restrict the class of possible covariance functions.

**Definition 7.20** (Stationarity)**.** *Let $\big(Z(x)\big)_{x\in\mathcal{X}}$ be a real valued random field, where $\mathcal{X}$ is a linear space, e.g., $\mathcal{X} \subseteq \mathbb{R}^n$.*

1. **Stationarity in the strict sense:**
   *If all finite-dimensional distributions of $\big(Z(x)\big)_{x\in\mathcal{X}}$ are invariant under arbitrary translation, i.e.*

$$F_{x^1+s,\dots,x^k+s}(z_1, \dots, z_k) = F_{x^1,\dots,x^k}(z_1, \dots, z_k), \quad \text{for all } s \in \mathcal{X}, \tag{7.32}$$

*then it is called a* stationary random field in the strict sense.

2. **Stationarity to 2nd order:**
*If we have for* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *that*

$$m(x) = m \quad \text{and} \quad C(x,s) = C(\tau) = C(x-s) \tag{7.33}$$

*then it is called a* stationary random field to 2nd order. *The covariance function that only depends on the separation vector* $\tau = x - s$ *is then called* stationary covariance function.

For Gaussian random fields these conditions are equivalent. Note that every stationary covariance function must have constant variance, i.e., $C(x,x) = \sigma^2(x) = \sigma^2$ for all $x \in \mathcal{X}$, so that

$$C(\tau) = \sigma^2 \rho(\tau).$$

This means, that one can investigate the correlation function or the covariance function without loss of generality.

## 7.2 Analytical Properties of Random Fields

In this section, analytical properties, i.e., *continuity* and *differentiability*, of random fields are discussed. These properties can not be described by the finite dimensional distributions, therefore the underlying function space $(E, \mathcal{F})$ has to be taken into account [1]. Most of this section is based on [1, 2, 33].

### 7.2.1 Continuity

Based on the different stochastic convergence types *almost sure convergence* and *mean square convergence* one has different types of continuity for random fields.

**Definition 7.21.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a random field with* $\mathcal{X} \subseteq \mathbb{R}^n$ *and* $Z(x) \in L^2(\Omega, \mathcal{A}, P)$ *for all* $x \in \mathcal{X}$.

1. $\big(Z(x)\big)_{x\in\mathcal{X}}$ *has* continuous sample paths with probability one*, or* sample path continuous*, in* $\mathcal{X}$ *if for every series* $\big(x^k\big)$ *with* $x^k \to x$ *as* $k \to \infty$*, then*

$$P(\{\omega : |Z(\omega, x^k) - Z(\omega, x)| \to 0,\ k \to \infty, \quad \text{for all } x \in \mathcal{X}\}) = 1. \tag{7.34}$$

2. $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is* almost surely continuous *in* $\mathcal{X}$ *if for every series* $\big(x^k\big)$ *with* $x^k \to x$ *as* $k \to \infty$*, then*

$$P(\{\omega : |Z(\omega, x^k) - Z(\omega, x)| \to 0,\ k \to \infty\}) = 1 \quad \text{for all } x \in \mathcal{X}. \tag{7.35}$$

3. $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is* mean square continuous *in* $\mathcal{X}$ *if for every series* $\big(x^k\big)$ *with* $x^k \to x$ *as* $k \to \infty$*, then*

$$\mathbb{E}[|Z(\omega, x^k) - Z(\omega, x)|^2] \to 0,\ k \to \infty \quad \text{for all } x \in \mathcal{X}. \tag{7.36}$$

Note that sample path continuity means that for all $t \in \mathcal{X}$ there are, with probability one, no discontinuities. Whereas for almost sure continuity there exists a measurable set $N$ with $P(N) = 0$, on which discontinuities are allowed. Hence, sample path continuity is a stronger property than almost sure continuity. In general, neither does mean square continuity imply sample path continuity, nor does sample path continuity imply mean square continuity. However, for Gaussian random fields mean square continuity is a necessary and almost sufficient condition for continuous sample paths with probability one [1]. Furthermore, mean square continuity and almost sure continuity imply *continuity in probability*, i.e.

$$\lim_{k\to\infty} P(\{\omega : |Z(\omega, x^k) - Z(\omega, x)| > \delta\} = 0, \quad \text{for all } \delta > 0.$$

**Definition 7.22** (Version of a Random Field)**.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *and* $\big(\bar{Z}(x)\big)_{x\in\mathcal{X}}$ *be two random fields.* $Z$ *and* $\bar{Z}$ *are called* versions *of each other if*

$$P(\{\omega : Z(\omega, x) = \bar{Z}(\omega, x)\}) = 1 \quad \text{for all } x \in \mathcal{X}. \tag{7.37}$$

Note that two versions have the same finite dimensional distributions, but are not necessarily identical.

**Definition 7.23** (Separable Random Fields [3])**.** *A random field* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is called* separable*, if there exists a countable set* $D \subset \mathcal{X}$ *and a fixed event* $N$ *for which* $P(N) = 0$*, such that for any closed set* $A \subset \mathbb{R}^n$ *and open set* $I \subset \mathcal{X}$ *the two sets*

$$\{\omega : Z(\omega, x) \in A, t \in I\} \quad \textit{and} \quad \{\omega : Z(\omega, x) \in A, t \in I \cap D\}$$

*differ by a subset of* $N$*.*

In [1, 47], it is shown that to any given random field $Z$, one can always find a version $\bar{Z}$ which is separable. Therefore, from here on we assume that we have separable random fields.
As already mentioned mean square continuity is implied by the sufficient conditions for sample path continuity. Further, there exists a relation between mean square continuity of a random field and the continuity of its covariance function. Proofs for the following theorems can be found in [1, 2].

**Theorem 7.24.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a random field with continuous expectation* $m(x)$*.* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is mean square continuous at* $x \in \mathbb{R}^n$ *if and only if its covariance function* $C(s, \hat{s})$ *is continuous at* $x = s = \hat{s}$*.* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is everywhere continuous if* $C(s, \hat{s})$ *is continuous at every diagonal point* $(s, \hat{s} = s)$*.*

It directly follows for stationary random fields.

**Corollary 7.25.** *A stationary random field* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is mean square continuous if and only if its covariance function* $C(\tau)$ *is continuous at* $0$*.*

Now some sufficient conditions for sample path continuity are discussed. The following theorem helps to determine if a random field has continuous sample paths.

**Theorem 7.26.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a random field with* $\mathcal{X}\subseteq\mathbb{R}^n$*. If*

$$\mathbb{E}[|Z(x)-Z(s)|^{\alpha}]\leq\frac{c\,\tau^{2d}}{|\log(\|\tau\|)|^{1+\beta}}, \tag{7.38}$$

*where* $\tau=x-s$*,* $c>0$*,* $\alpha>0$*, and* $\beta>\alpha$*, then* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *will have continuous sample paths with probability one.*

For Gaussian random fields we then have the following theorem.

**Theorem 7.27.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a zero-mean, Gaussian random field with continuous covariance function. Then, if for some* $0<c<\infty$ *and some* $\epsilon>0$*,*

$$\mathbb{E}[|Z(x)-Z(s)|^{2}]\leq\frac{c}{|\log(\|\tau\|)|^{1+\epsilon}}, \tag{7.39}$$

*for all* $\tau$ *with* $\|\tau\|<1$*, then* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *has continuous sample paths with probability one.*

For stationary Gaussian random fields this can be simplified.

**Corollary 7.28.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a stationary Gaussian random field with continuous correlation function* $\rho$*. Then, if for some* $0<c<\infty$ *and some* $\epsilon>0$*,*

$$\rho(0)-\rho(\tau)\leq\frac{c}{|\log(\|\tau\|)|^{1+\epsilon}}, \tag{7.40}$$

*for all* $\tau$ *with* $\|\tau\|<1$*, then* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *has continuous sample paths with probability one.*

### 7.2.2 Differentiability

Comparable to the continuity there are also different types of differentiablity corresponding to different forms of convergence. In this work, we focus on *mean square differentiability* corresponding to mean square continuity. Most proofs of the results stated in this subsection can be found in [2, 33].

**Definition 7.29.** *1. A random field* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *with* $\mathcal{X}\subseteq\mathbb{R}^n$ *and* $Z(x)\in L^2(\Omega,\mathcal{A},P)$ *is* differentiable in the mean square sense (m.s.s.) *with respect to the component* $x_i$ *of* $x=(x_1,\ldots,x_i,\ldots,x_n)$*, if there exists a random field* $\dot{Z}_i(x)$*, s.t.*

$$\mathbb{E}\left[\left|\frac{Z(x+h\,e_i)-Z(x)}{h}-\dot{Z}_i(x)\right|^2\right]\to 0\quad\text{as } h\to 0, \tag{7.41}$$

*where* $e_i$ *is the* $i$*-th unit vector.* $\dot{Z}_i(x)$ *is also called the* $i$-th partial derivative *of* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *and is also denoted as*

$$\dot{Z}_i(x)=\frac{\partial Z(x)}{\partial x_i}=\lim_{h\to 0}\frac{Z(x+h\,e_i)-Z(x)}{h}. \tag{7.42}$$

*If* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is differentiable at every location* $x$*, then it is differentiable everywhere.*

2. $\big(Z(x)\big)_{x\in\mathcal{X}}$ *has* differentiable sample paths with probability one *in* $\mathcal{X}$ *if for every series* $\left(x^k\right)$ *with* $x^k \to x$ *as* $k \to \infty$

$$P(\{\omega : |\dot{Z}(\omega, x^k) - \dot{Z}(\omega, x)| \to 0,\ k \to \infty, \quad \text{for all } x \in \mathcal{X}\}) = 1. \tag{7.43}$$

3. $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is* almost surely differentiable *in* $\mathcal{X}$ *if for every series* $\left(x^k\right)$ *with* $x^k \to x$ *as* $k \to \infty$

$$P(\{\omega : |\dot{Z}(\omega, x^k) - \dot{Z}(\omega, x)| \to 0,\ k \to \infty\}) = 1 \quad \text{for all } x \in \mathcal{X}. \tag{7.44}$$

Note that the same implications for continuity following Definiton 7.21 also apply for differentiability. As with mean square continuity, there is a link between the differentiability in the m.s.s. of a random field and the differentiability of its covariance function.

**Theorem 7.30.** *A random field* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *with* $\mathcal{X} \subseteq \mathbb{R}^n$ *and covariance function* $C$ *and differentiable expectation is m.s. differentiable, if the derivative* $\partial^2 C(s,x)/\partial s_i \partial x_i$ *exists and is finite for all* $i = 1, \ldots, n$ *at all diagonal points* $x = s$*. The covariance function of* $\big(\dot{Z}(x)\big)_{x\in\mathcal{X}}$ *is then given by* $\partial^2 C(s,x)/\partial s_i \partial x_i$*.*

A proof can be found in [33, 30]. For stationary random fields we have consequently the following result.

**Corollary 7.31.** *Let* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *be a stationary random field with* $\mathcal{X} \subseteq \mathbb{R}^n$ *and covariance function* $C$*. If the derivative* $\partial^2 C(\tau)/\partial \tau_i^2 = \partial^2 C(x-s)/\partial s_i \partial x_i$ *exists and is finite for all* $i = 1, \ldots, n$ *at* $\tau = 0$*, then* $\big(Z(x)\big)_{x\in\mathcal{X}}$ *is m.s. differentiable. The covariance function of* $\big(\dot{Z}(x)\big)_{x\in\mathcal{X}}$ *is then given by* $-\partial^2 C(\tau)/\partial \tau_i^2$*.*

Note that the negative sign comes from $\partial C(x-s)/\partial s_i = -\partial C(\tau)/\partial \tau_i$. When the partial derivatives of the sample paths are continuous then the sample paths are differentiable. Sufficient conditions for differentiable sample paths are that the partial derivatives of the sample paths are continuous [1]. Thus, one obtains sufficient conditions for continuous sample paths by applying Theorem 7.26 or 7.27 to the covariance or correlation function of the gradient field $\big(\dot{Z}(\omega, t)\big)_{x\in\mathcal{X}}$.

## 7.3 Gradient Enhanced Kriging

In this section, we introduce the method of gradient enhanced Kriging. This method is based on *Design and Analysis of Computer Experiments (DACE)* [130], where an expensive to compute function, for which we also have gradient information, is estimated using Gaussian random fields while incorporating the available gradient information. This estimated surrogate function is then used in the optimization process to determine new points which then are evaluated with the expensive original function. The aim of this approach is to reduce the computational costs of the optimization and to be able to do a more global optimization.
In the following, a brief overview of *Latin hypercube sampling (LHS)* to generate an initial sampling is given. Then, the stochastic model and the Kriging approach are introduced, subsequently it is shown how gradient information can be incorporated in the model.

Most of this section is based on [65, 130, 92, 90, 37]. Further, details about the Dakota Kriging implementation *surfpack* can be found in [57, 71].

### 7.3.1 Latin Hypercube Sampling

For a Kriging surrogate model and our numerical studies with Dakota an initial sampling is needed. We choose Latin hypercube sampling (LHS) as our method to generate an initial sampling, since this is the default option in the Dakota implementation [56]. The LHS technique is a common choice for a variety of computer models since 1979. In that year, [106] introduced Latin hypercube sampling, which was further developed, e.g., in [88, 84]. In [148], the LHS implementation of Dakota is presented. Following [148], we give a brief overview of LHS and refer to [106, 88, 84] for more details.
Let $J$ be a function of $x_1, x_2, \ldots, x_n$, that may be very complicated, e.g., our objective function w.r.t. PoF $J_1$. It is of interest to investigate how $J$ varies when $x_1, x_2, \ldots, x_n$ vary, assuming a joint probability distribution. Applying Monte Carlo sampling yields answers to this question. Given the distribution of $J$ some of its properties, e.g., its mean, can be estimated by repeated random sampling from the assumed joint probability density function of $x_1, x_2, \ldots, x_n$ and by evaluating the $J$ values for each sample. This yields for $k$ Monte Carlo iterations a set of $k$ $n$-dimensional vectors of input variables. These $k$ vectors can then be used as an initial sampling for optimization schemes. For $k$ sufficiently large this method yields reasonable estimates for the distribution of $J$. However, large $k$ come with a high computational cost. Therefore, other sampling methods, like the Latin hypercube sampling, were sought.
Latin hypercube sampling is a constrained Monte Carlo sampling method that was developed in [106]. In LHS, $k$ different values from each of the $n$ variables $x_1, x_2, \ldots, x_n$ are selected as follows. The range of each variable is divided into $k$ disjoint intervals of equal marginal probability $1/k$. From each interval one value is randomly, i.e., w.r.t. the probability density in the interval, selected. In a next step, the $k$ values obtained for $x_1$ are randomly paired with the $k$ values obtained for $x_2$. These $k$ pairs are then combined randomly with the $k$ values obtained for $x_3$, and so on, until $k$ $n$-dimensional vectors are generated. These $k$ $n$-dimensional vectors form an initial sampling, as described above, and are the result of the Latin hypercube sampling. For better understanding one can think of this sample as generating an $(k \times n)$ matrix of input values where the $i$-th row contains specific values for each of the $n$ input variables on the $i$-th run of the computer simulation [148].
Note that there exists ways to restrict the described random pairing of variables. But these are beyond the scope of this work and therefore we omit them and refer to [148] instead for further details.

### 7.3.2 Stochastic Model

The Kriging model is based on a positive definite radial basis function, i.e., a real function which only depends on the distance of the argument to a fixed point, e.g., the origin, of the form

$$\psi(x^i, x^j) = \psi(x^i - x^j) = \exp\Big\{-\sum_{h=1}^{n} \theta_h |x_h^i - x_h^j|^{\varsigma_h}\Big\},\ \theta_h \geq 0,\ \varsigma_h \in [1, 2]. \tag{7.45}$$

Following Theorem 7.14 and Corollary 7.15, this radial basis function is a correlation function [65]. The surrogate model is based on a stationary Gaussian random field $Z(x)$ that estimates the function of interest

$$Z(x) \sim \mathcal{N}(m(x), C(x^i, x^j)), \tag{7.46}$$

where

$$m(x) = \mu \quad \text{and} \quad C(x^i, x^j) = \sigma^2 \exp\Big\{-\sum_{h=1}^{n} \theta_h |x_h^i - x_h^j|^{\varsigma_h}\Big\}. \tag{7.47}$$

Further, it is assumed that the expectation function $m(x)$ is constant for every random variable and the covariance function depends on the Euclidean distance of two points, i.e., the radial basis function $\psi(x^i, x^j)$ is the correlation function $\rho(x^i, x^j)$ of this random field. The model has $2n + 2$ parameters $(\mu, \sigma, \theta_1, \ldots, \theta_n, \varsigma_1, \ldots, \varsigma_n)$, where in general it is assumed that $\theta_h \geq 0$ and $\varsigma_h \in [1, 2]$ for all $h = 1, \ldots, n$. The parameters $\varsigma_h$ determine the smoothness of the covariance function. For $\varsigma_h = 2$ the covariance function is smooth and for smaller $\varsigma^h$ the smoothness decreases, i.e., one assumes less covariance between the points.

### Estimation of the Model Parameters

The maximum-likelihood-method is used to estimate the model parameters. Additionally to the parameters $(\theta_1, \ldots, \theta_n, \varsigma_1, \ldots, \varsigma_n)$ , we have $k$ sample points $\{x^1, \ldots, x^k\} \subseteq \mathbb{R}^n$ and their associated function values $\{z_1, \ldots, z_k\} \subseteq \mathbb{R}^n$ which are a realization of the random field $\underline{Z}(x) := (Z(x^1), \ldots, Z(x^k))$. These random variables are multivariate normal distributed with expectation $\vec{1}\mu$ and covariance matrix $\Sigma_{\underline{Z}(x)}$. Therefore, the likelihood function takes the following form

$$\mathcal{L}(\mu, \sigma, \theta_1, \ldots, \theta_n, \varsigma_1, \ldots, \varsigma_n) = \frac{\exp\Big\{-\frac{(\underline{Z}(x)-\vec{1}\mu)^\top \Sigma_{\underline{Z}(x)}^{-1} (\underline{Z}(x)-\vec{1}\mu)}{2\sigma^2}\Big\}}{(2\pi\sigma^2)^{\frac{k}{2}} |\Sigma_{\underline{Z}(x)}|^{\frac{1}{2}}}. \tag{7.48}$$

The parameters $(\theta_1, \ldots, \theta_n, \varsigma_1, \ldots, \varsigma_n)$ are imbedded in the covariance matrix $\Sigma_{\underline{Z}(x)}$. Further, if the parameters $(\theta_1, \ldots, \theta_n, \varsigma_1, \ldots, \varsigma_n)$ are known, then a maximization of (7.48) yields:

$$\hat{\mu} = \frac{\vec{1}^\top \Sigma_{\underline{Z}(x)}^{-1} \underline{Z}(x)}{\vec{1}^\top \Sigma_{\underline{Z}(x)}^{-1} \vec{1}}, \tag{7.49}$$

$$\hat{\sigma}^2 = \frac{(\underline{Z}(x) - \vec{1}\mu)^\top \Sigma_{\underline{Z}(x)}^{-1} (\underline{Z}(x) - \vec{1}\mu)}{k}. \tag{7.50}$$

Re-substituting (7.49) and (7.50) in (7.48) yields the *concentrated likelihood*:

$$\frac{\exp\Big\{-\frac{k}{2}\Big\}}{(2\pi\sigma^2)^{\frac{k}{2}} |\Sigma_{\underline{Z}(x)}|^{\frac{1}{2}}} \tag{7.51}$$

For the optimization, i.e., estimation of the model parameters, the logarithm of the concentrated likelihood is taken and constants are omitted

$$-\frac{k}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln(|\Sigma_{\underline{Z}(x)}|). \tag{7.52}$$

Since we want to incorporate gradients to our model we set $\varsigma_1 = \ldots = \varsigma_n = 2$ and change $|x_h^i - x_h^j|^2$ in (7.45) to $(x_h^i - x_h^j)^2$ to ensure differentiability of the covariance function. The remaining $d+2$ model parameters are estimated with the maximum-likelihood method. Toward this end, the following algorithm is used.

### 7.3.3 DIvision of RECTangles (DIRECT) Algorithm

The DIRECT algorithm is a widely used heuristic for global optimization problems and was first introduced in [91]. Since the Dakota implementation of DIRECT is based on [67], we will introduce the algorithm from this reference. It consists of two main parts. The first defines how to divide the domain, and the second defines how to decide which hyperrectangles are divided in the next iteration.

#### Dividing the Domain

DIRECT uses division based on $n$-dimensional trisection. In the following, it is shown how this division is done for a hypercube and a hyperrectangle.
Let $c$ be the center point of a hypercube. DIRECT evaluates the function at the locations $c \pm \delta e_i$, where $\delta$ is 1/3 of the side length of the cube and $e_i$ is the $i$-th standard basis vector, $i = 1, \ldots, n$. Then, $w_i$ is defined by

$$w_i = \min\{f(c - \delta e_i), f(c + \delta e_i)\}, \quad i = 1, \ldots, n. \tag{7.53}$$

Then in a next step the hypercube is divided in the order given by the $w_i$, starting with the smallest $w_i$. First the hypercube is divided perpendicularly to the direction with the lowest $w_i$. The remaining volume is then divided perpendicularly to the direction of the second lowest $w_i$ and so on, until the hypercube is divided in all directions. This approach constructs a hypercube with length $\delta$ centered at $c$. Let $b = \text{argmin}_{i=1,\ldots,n}\{f(c-\delta e_i), f(c+\delta e_i)\}$. Then, $b$ is the center of a hyperrectangle with one side of length $\delta$, and the oder $n-1$ sides of length $3\delta$.
Hyperrectangles are only divided along its longest sides. This ensures that the maximal side length of the hyperrectangle decreases.

#### Choosing a Hyperrectangle to divide

The DIRECT algorithm decides which hyperrectangle to divide in the next iteration by using the concept of *potentially optimal hyperrectangles* [67].

**Definition 7.32.** *Let $\varepsilon > 0$ and let $f_{min}$ be the current best function value. A hyperrectangle $j$ is called* potentially optimal hyperrectangle *if there exists some $C > 0$ such*

*that*

$$f(c_j) - Cd_j \leq f(c_i) - Cd_i, \ \forall i$$
$$f(c_j) - Cd_j \leq f_{min} - \varepsilon\,|f_{min}|.$$

*Here $c_j$ is the center of the hyperrectangle $j$ and $d_j$ is a measure for this hyperrectangle.*

In [91] $d_j$ is chosen as the distance from the center $c_j$ of the hyperrectangle $j$ to its vertices.

### The DIRECT Algorithm

Unlike more traditional optimization methods the DIRECT algorithm has no stopping condition. It terminates when the maximum number of iterations $N_{\text{it}}$ or the maximum number of function evaluations $N_{\text{eval}}$ is reached. Therefore, the variables $m$ for the number of function evaluations and $t$ for the number of iterations are needed in the algorithm formulation. The DIRECT algorithm is then of the following form, see also [67]:

**Algorithm 6:** DIRECT

**Data:** A hyperrectangle, choose $f$, $N_{\text{it}}$, $N_{\text{eval}}$, and $\varepsilon > 0$.
**Result:** Approximation of a global solution (on the hyperrectangle).
Normalize the search space to be the unit hypercube with center point $c_1$;
Evaluate $f(c_1),\ f(c_1) = f_{\min},\ t = 0,\ m = 1$;
**while** $t < N_{it}$ *and* $m < N_{eval}$ **do**
  Identify the set $P$ of potentially optimal hyperrectangles;
  **while** $P \neq \emptyset$ **do**
    Take $j \in P$;
    Sample new points $(c_i \pm \delta e_i)$, evaluate $f$ at these points and divide the hyperrectangle as described above;
    Update $f_{\min},\ m = m + \Delta m$;
    Set $P = P \setminus \{j\}$
  **end**
  t=t+1;
**end**

Following lemma from [67] helps to reformulate Definition 7.32 to identify potentially optimal intervals.

**Lemma 7.33.** *Let $\varepsilon > 0$ and let $f_{min}$ be the current best function value. Let $I$ be the set of all indices of all intervals existing. Let $I_1 = \{i \in I : d_i < d_j\}$, $I_2 = \{i \in I : d_i > d_j\}$ and $I_3 = \{i \in I : d_i = d_j\}$. Interval $j \in I$ is potentially optimal if*

$$f(c_j) \leq f(c_i) \quad \forall i \in I_3, \tag{7.54}$$

*there exists a $C > 0$ such that*

$$\max_{i \in I_1} \frac{f(c_j) - f(c_i)}{d_j - d_i} \leq C \leq \min_{i \in I_2} \frac{f(c_i) - f(c_j)}{d_i - d_j}, \tag{7.55}$$

*and*

$$\varepsilon \leq \frac{f_{min} - f(c_j)}{|f_{min}|} + \frac{d_j}{|f_{min}|} \cdot \min_{i \in I_2} \frac{f(c_i) - f(c_j)}{d_i - d_j}, \ f_{min} \neq 0, \tag{7.56}$$

*or*

$$f(c_j) \leq d_j \cdot \min_{i \in I_2} \frac{f(c_i) - f(c_j)}{d_i - d_j}, \ f_{min} = 0. \tag{7.57}$$

One modification of the DIRECT algorithm of [91] in [67] is that another measure $d_j$, i.e., the length of the longest side of the hyperrectangle, is used. This reduces the number of different groups of hyperrectangles and makes the algorithm more biased toward local search [67]. The second modification is, that when there is more than one hyperrectangle with the same measure only one is divided, instead of all. Both modifications can lead to an improvement of performance of the DIRECT algorithm.

### 7.3.4 The Kriging Model

In this section, the Kriging model and Kriging estimator are introduced. Toward this end, let $x^1, \ldots, x^k \in \mathbb{R}^n$ be a set of sample points with responses $z_1, \ldots, z_k$ which are realizations of the random variables $Z(x^1), \ldots, Z(x^k)$. In the Kriging approach one assumes that the response $Z(x^0)$ of an unknown point $x^0$ is linearly dependent on the observed values $Z(x^1), \ldots, Z(x^k)$, i.e.

$$\hat{Z}(x^0) = \sum_{j=1}^{k} \alpha_j Z(x^j). \tag{7.58}$$

One chooses the weights $\alpha_j$ by minimizing the Mean Square Error of (7.58) under an unbiasedness assumption

$$\begin{aligned}
&\mathbb{E}[\hat{Z}(x^0)] = \mathbb{E}\Big[\sum_{j=1}^{k} \alpha_j Z(x^j)\Big] \stackrel{!}{=} \mu \\
\Leftrightarrow &\sum_{j=1}^{k} \alpha_j \mathbb{E}[Z(x^j)] = \sum_{j=1}^{k} \alpha_j \mu = \mu \\
\Leftrightarrow &\mu \sum_{j=1}^{k} \alpha_j = \mu \\
\Leftrightarrow &\sum_{j=1}^{k} \alpha_j = 1 \\
\Leftrightarrow &\alpha^\top \vec{1} - 1 = 0.
\end{aligned} \tag{7.59}$$

We have the following problem

$$\begin{aligned}
\min \quad & \mathbb{E}[(\hat{Z}(x^0) - Z(x^0))^2] \\
\text{s.t.} \quad & \alpha^\top \vec{1} - 1 = 0 \\
& x^0 \in \mathbb{R}^n
\end{aligned} \quad . \tag{7.60}$$

With $\underline{Z}(x) = (Z(x^1), \ldots, Z(x^k))$, the covariance matrix $\Sigma_x := \Sigma_{\underline{Z}(x)} = \big(\mathrm{Cov}(x^i, x^j)\big)_{i,j}$, and the covariance vector w.r.t. $x^0$, $\Sigma_0 := \Sigma_{x^0,x} := \mathrm{Cov}(x^0, x)$, we get

$$\mathbb{E}[(\hat{Z}(x^0) - Z(x^0))^2] = \sigma^2(\alpha^\top \Sigma_x \alpha - 2\alpha\Sigma_0 + 1). \tag{7.61}$$

Therefore, the problem (7.60) is a convex minimization problem with a unique solution, since $\Sigma_x$ is positive definite. This unique solution can be computed using the KKT conditions of (7.60), which are given as follows.

$$\begin{aligned} 2\Sigma_x\alpha - 2\Sigma_0 + u\vec{1} &= 0 \\ \alpha^\top\vec{1} - 1 &= 0, \end{aligned} \tag{7.62}$$

with $u \geq 0$. This can also be expressed as

$$\begin{pmatrix} \Sigma_x & \vec{1} \\ \vec{1}^\top & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \frac{1}{2}u \end{pmatrix} = \begin{pmatrix} \Sigma_0 \\ 1 \end{pmatrix}. \tag{7.63}$$

Furthermore, we have

$$\begin{pmatrix} \alpha \\ \frac{1}{2}u \end{pmatrix} = \begin{pmatrix} \Sigma_x & \vec{1} \\ \vec{1}^\top & 0 \end{pmatrix}^{-1} \begin{pmatrix} \Sigma_0 \\ 1 \end{pmatrix}. \tag{7.64}$$

Solving (7.64), with $A := -\vec{1}^\top\Sigma_x^{-1}\vec{1}$, yields,

$$\alpha = \Sigma_x^{-1}\Sigma_0 + \frac{\Sigma_x^{-1}\vec{1}\,\vec{1}^\top\Sigma_x^{-1}\Sigma_0}{A} - \frac{\Sigma_x^{-1}\vec{1}}{A}. \tag{7.65}$$

Therefore, we have for (7.58)

$$\hat{Z}(x^0) = \Sigma_0^\top\Sigma_x^{-1}\underline{Z}(x) + \frac{\Sigma_0^\top\Sigma_x^{-1}\vec{1}\,\vec{1}^\top\Sigma_x^{-1}\Sigma_0\underline{Z}(x)}{A} - \frac{\Sigma_x^{-1}\vec{1}\underline{Z}(x)}{A}. \tag{7.66}$$

Using (7.49) and (7.50) we get for the Kriging predictor and the mean square error of the prediction

$$\hat{Z}(x^0) = \hat{\mu} + \Sigma_0^\top\Sigma_x^{-1}(\underline{Z}(x) - \vec{1}\hat{\mu}) \tag{7.67}$$

$$\hat{s}_K^2(x^0) = \hat{\sigma}^2\left(1 - \Sigma_0^\top\Sigma_x^{-1}\Sigma_0 + \frac{(1 - \vec{1}^\top\Sigma_x^{-1}\Sigma_0)^2}{\vec{1}^\top\Sigma_x^{-1}\vec{1}}\right). \tag{7.68}$$

For further detail we refer to [65, 130].

### 7.3.5 Bayesian Approach

Alternatively, for Gaussian processes one can also follow the Bayesian approach to model the predictor. In general, both methods lead to different results. Let $x^1, \ldots, x^k \in \mathbb{R}^n$ be a set of sample points with responses $Z(x^1), \ldots, Z(x^k)$, and let $Z(x^0)$ be the response of an unknown point $x^0$. Following the DACE model [130], $(Z(x^1), \ldots, Z(x^k), Z(x^0))$ are

multivariate normal distributed, i.e.

$$m_Z = (\vec{1}^\top \mu, \mu) \quad and \quad \Sigma_Z = \begin{pmatrix} \Sigma_x & \Sigma_0 \\ \Sigma_0^\top & \sigma^2 \end{pmatrix}. \tag{7.69}$$

Now, one can calculate the conditional Gaussian distribution (Definition 7.18) of $Z(x^0)$ given $Z(x^1), \ldots, Z(x^k)$ and predict $Z(x^0)$ by

$$\hat{Z}(x^0) = \mathbb{E}[Z(x^0)|Z(x^1), \ldots, Z(x^k)]. \tag{7.70}$$

One obtains

$$\hat{Z}(x^0) = \mu + \Sigma_0^\top \Sigma_x^{-1}(\underline{Z}(x) - \vec{1}\mu), \tag{7.71}$$

$$\hat{s}_B^2(x^0) = \sigma^2 \left( 1 - \Sigma_0^\top \Sigma_x^{-1} \Sigma_0 \right). \tag{7.72}$$

Since we do not know $\mu$ and $\sigma$, we use the estimates $\hat{\mu}$ and $\hat{\sigma}$ in the equations (7.71) and (7.72). We have now two different measures of uncertainty of the model prediction $\hat{s}_K^2$ and $\hat{s}_B^2$ for which we have

$$\hat{s}_K^2(x^i) = \hat{s}_B^2(x^i) = 0, \quad i = 1, \ldots, k,$$

i.e., there is no uncertainty in the prediction of the sampled points $(x^1, \ldots, x^k)$. The additional term in $\hat{s}_K^2$ can be seen as the uncertainty of the prediction of $\mu$ and is in general omitted, since its influence usually is not of great magnitude. In practice, the Kriging approach also uses $\hat{s}_B^2$ to calculate the model uncertainty. Therefore, we use $\hat{s}^2 := \hat{s}_B^2$ as the uncertainty measure from here on out.

### 7.3.6 Gradient Enhanced Kriging

If we have additional data on the gradients at the sampled points available, we can add this information to our model. Let $\{x^1, \ldots, x^k\} \in \mathcal{X} \subseteq \mathbb{R}^n$, with $x^i = (x_1^i, \ldots, x_n^i)$, be our sampled points and $Z_1, \ldots, Z_k$ the random variables associated with the realizations, then the observed data is a $(n+1)k$ column vector:

$$Z = \left[ Z_1, \ldots, Z_k, \frac{\partial Z_1}{\partial x_1^1}, \ldots, \frac{\partial Z_k}{\partial x_1^k}, \ldots, \frac{\partial Z_1}{\partial x_n^1}, \ldots, \frac{\partial Z_k}{\partial x_n^k} \right]^\top \tag{7.73}$$

A second set of Kriging basis functions centered around the sampled data is incorporated into the model. The corresponding radial basis functions are the derivatives of the first $k$ Gaussian basis functions, where we set $\varsigma_h = 2$ for all $h \in \{1, \ldots, n\}$ to ensure differentiability

$$\frac{\partial \psi(x^i, x^j)}{\partial x_l^i} = \frac{\partial \exp\left\{ -\sum_{h=1}^n \theta_h (x_h^i - x_h^j)^2 \right\}}{\partial x_l^i} = -2\,\theta_l\,(x_l^i - x_l^j)\,\psi(x^i, x^j). \tag{7.74}$$

We then have

$$\begin{aligned}
\frac{\partial\psi(x^i,x^j)}{\partial x^i} &= \left[\frac{\partial\psi(x^i,x^j)}{\partial x^i_l} = -2\,\theta_l\,(x^i_l - x^j_l)\,\psi(x^i,x^j)\right]_{l=1,\ldots,k},\\
\frac{\partial\psi(x^i,x^j)}{\partial x^j} &= \left[\frac{\partial\psi(x^i,x^j)}{\partial x^j_l} = 2\,\theta_l\,(x^i_l - x^j_l)\,\psi(x^i,x^j)\right]_{l=1,\ldots,k},\\
\frac{\partial^2\psi(x^i,x^j)}{\partial x^i\partial x^j} &= \left[\frac{\partial\psi(x^i,x^j)}{\partial x^i_m x^j_l} = -4\,\theta_m\theta_l\,(x^i_m - x^j_m)(x^i_l - x^j_l)\,\psi(x^i,x^j)\right]_{l,m=1,\ldots,k}.
\end{aligned} \tag{7.75}$$

Since the derivatives in (7.75) exist and are finite in all diagonal points, it follows with (7.31) that the gradients in the m.s. sense of the random field exist and that their covariance function is given by $\partial^2 C(x^i,x^j)/\partial x^i\partial x^j$, where $C(x^i,x^j) = \sigma^2\psi(x^i,x^j) = \sigma^2\psi(x^i - x^j)$. To show that they also have differentiable sample paths, we have to check if for a finite $C > 0$, $\epsilon > 0$, and $\|\tau\| < 1$, where $\tau = x^i - x^j$, $\partial^2\psi(x^i,x^j)/\partial x^i\partial x^j$ satisfies (7.28):

$$\begin{aligned}
\frac{\partial^2\psi(x^i,x^i)}{\partial x^i\partial x^j} - \frac{\partial^2\psi(x^i,x^j)}{\partial x^i\partial x^j} &= \frac{\partial^2\psi(0)}{\partial x^i\partial x^j} - \frac{\partial^2\psi(x^i - x^j)}{\partial x^i\partial x^j}\\
&= 0 + 4\theta^2\tau^2\exp\Big\{-\sum_{l=1}^{n}\theta_l(\tau_l)^2\Big\}\\
&\le 4\theta^2\tau^2\exp\Big\{-\hat\theta\sum_{l=1}^{n}(\tau_l)^2\Big\}\\
&\le 4\theta^2\tau^2\exp\Big\{-\hat\theta\sqrt{\sum_{l=1}^{n}(\tau_l)^2}\Big\}\\
&= 4\theta^2\tau^2\exp\Big\{-\hat\theta\|\tau\|\Big\}\\
&\overset{!}{\le} \frac{C}{|\log(\|\tau\|)|^{1+\epsilon}}.
\end{aligned} \tag{7.76}$$

Both sides vanish at $\|\tau\| = 0$, therefore the derivatives have to be inspected. The derivative of the left hand side is finite for any $\|\tau\|$. Consequently, for any $\epsilon$ one can choose a sufficiently large $C$ such that (7.76) holds.
The $(d+1)n \times (d+1)n$ covariance matrix is then of the form

$$\dot\Sigma_Z = \begin{bmatrix}
C(x^1,x^1) & \ldots & C(x^1,x^k) & \frac{\partial C(x^1,x^1)}{\partial x^1} & \ldots & \frac{\partial C(x^1,x^k)}{\partial x^k}\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
C(x^k,x^1) & \ldots & C(x^k,x^k) & \frac{\partial C(x^k,x^1)}{\partial x^1} & \ldots & \frac{\partial C(x^k,x^k)}{\partial x^k}\\
\frac{\partial C(x^1,x^1)}{\partial x^1}^\top & \ldots & \frac{\partial C(x^1,x^k)}{\partial x^1}^\top & \frac{\partial^2 C(x^1,x^1)}{\partial^2 x^1} & \ldots & \frac{\partial^2 C(x^1,x^k)}{\partial x^1\partial x^k}\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
\frac{\partial C(x^k,x^1)}{\partial x^k}^\top & \ldots & \frac{\partial C(x^k,x^k)}{\partial x^k}^\top & \frac{\partial^2 C(x^k,x^1)}{\partial x^k\partial x^1} & \ldots & \frac{\partial^2 C(x^k,x^k)}{\partial^2 x^k}
\end{bmatrix}. \tag{7.77}$$

It contains the covariance between the data with itself, between the gradients and data, between the data and gradients and between the gradients and themselves.
With the covariance vector w.r.t. to an unknown $x^0$

$$\dot{\Sigma}_0 = \left( C(x^0, x^1), \ldots, C(x^0, x^k), \frac{\partial C(x^0, x^1)}{\partial x^1}, \ldots, \frac{\partial C(x^0, x^k)}{\partial x^k} \right)^\top \tag{7.78}$$

one can compute the Kriging predictor for an unknown $x^0$. Let

$$\underline{1} = (\underbrace{1, \ldots, 1}_{k}, \underbrace{0, \ldots, 0}_{kn})^\top \tag{7.79}$$

then the GEK predictor is given by

$$\hat{Z}(x^0) = \hat{\mu} + \dot{\Sigma}_0^\top \dot{\Sigma}_Z^{-1} (\underline{Z}(x) - \underline{1}\hat{\mu}). \tag{7.80}$$

## 7.4 Efficient Global Optimization (EGO)

In this section, *Efficient Global Optimization (EGO)* is introduced. It was first introduced in [92] and is a global optimization method that incorporates a Kriging model as a surrogate model on which the optimization takes place. After acquiring an optimal solution on the surrogate model, the original expensive function value of that solution is evaluated and the stopping condition is checked. If the algorithm does not terminate, then this value and its point are added to the underlying Kriging model, and subsequently the surrogate is updated. Therefore, the acquisition of new points is of interest and is handled in Subsection 7.4.1. The implementation of EGO in Dakota is discussed in Subsection 7.4.2. Most of this section is based on [92, 90, 57, 65].

### 7.4.1 Acquisition Functions

In this subsection, two acquisition functions which determine new points to evaluate are discussed. A minimization process using the Kriging estimator $\hat{Z}(t^0)$ as an objective, can easily lead to a local minimum, especially when the initial sample is not evenly spread. In the parts of the range of the objective function where there is uncertainty in the model, there may be objective values that are smaller than the minimum value of the Kriging estimation. Thus, a minimization process with the predictor as an objective function would only yield an accurate estimate of a local minimum. The reason for this is, that this approach assumes that there is no uncertainty in the model and therefore only *exploits* the predictor, failing to *explore* points where the model is uncertain. So, a proper acquisition function should balance *exploitation*, i.e., optimization of the objective, and *exploration* of uncertain points. This can also be seen as a balance between a local and global search. In the following, the *probability of improvement* and *expected improvement* are introduced.
Let $\{x^1, \ldots, x^k\}$ be a set of sample points with responses $\{z_1, \ldots, z_k\}$ as realizations of the random variables $Z(x^1), \ldots, Z(x^k)$. The response at an unknown $x^0$ is given by a random variable that is normal distributed with mean and standard deviation given by

the Kriging predictor and its standard error, i.e.

$$\tilde{Z}(x^0) := Z(x^0)|Z(x^1), \dots, Z(x^k) \sim \mathcal{N}(\hat{Z}(x^0), \hat{s}^2(x^0)). \tag{7.81}$$

Further, let

$$z_{\min} = \min(z_1, \dots, z_k) \tag{7.82}$$

be the current best objective value.

### Probability of Improvement

The *probability of improvement* assesses for given $\{x^1, \dots, x^k\}$ and responses $\{z_1, \dots, z_k\}$ the probability that the response of an unknown $x^0$ is better than the current best objective value $z_{\min}$. Then, we have for $\hat{s}(x^0) > 0$

$$\begin{aligned} PI(x^0) &:= P(\tilde{Z}(x^0) \leq z_{\min} \,|\, Z(x^1) = z_1, \dots, Z(x^k) = z_k) \\ &= \Phi\left(\frac{z_{\min} - \hat{Z}(x^0)}{\hat{s}(x^0)}\right), \end{aligned} \tag{7.83}$$

where $\Phi(\cdot)$ is the standard normal distribution function, and $PI(x^0) = 0$ in the case $\hat{s}(x^0) = 0$ as this only occurs if and only if $x^0 \in \{x^1, \dots, x^k\}$. Hence, the objective is to maximize the probability of improvement.

### Expected Improvement

Another acquisition function is the *expected improvement* of the current best objective value $z_{\min}$ at an unknown $x^0$. In contrast to the probability of improvement, which assesses the probability of an improvement, the expected improvement assesses the actual expected improvement at a location $x^0$. The *improvement* is formally given as

$$I(x^0) = \max\Big(z_{\min} - \tilde{Z}(x^0), 0\Big), \tag{7.84}$$

and it itself is a random variable since $\tilde{Z}(x^0)$ is a random variable. Hence, the expected improvement is

$$EI(x^0) := \mathbb{E}[I(x^0) \,|\, Z(x^1) = z_1, \dots, Z(x^k) = z_k]. \tag{7.85}$$

Further, one can express the expected improvement after some integration in closed form. First we use a reparameterization of $\tilde{Z}(x^0)$

$$\tilde{Z}(x^0) = \hat{Z}(x^0) + \hat{s}(x^0)\,\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1).$$

Then we have with $K := \frac{z_{\min} - \hat{Z}(x^0)}{\hat{s}(x^0)}$

$$\begin{aligned} EI(x^0) &= \int_{-\infty}^{\infty} I(x)\,\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \int_{-\infty}^{K} \Big(z_{\min} - \hat{Z}(x^0) - \hat{s}(x^0)\,\varepsilon\Big)\,\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \hat{Z}(x^0)\Big)\,\Phi(K) - \hat{s}(x^0) \int_{-\infty}^{K} \varepsilon\,\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \hat{Z}(x^0)\Big)\,\Phi(K) + \frac{\hat{s}(x^0)}{\sqrt{2\pi}} \int_{-\infty}^{K} -\varepsilon\, e^{-\varepsilon^2/2}\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \hat{Z}(x^0)\Big)\,\Phi(K) + \hat{s}(x^0)\,\phi(K). \end{aligned} \tag{7.86}$$

Re-substituting $K$ yields

$$EI(x^0) = \begin{cases} (z_{\min} - \hat{Z}(x^0))\,\Phi\left(\frac{z_{\min} - \hat{Z}(x^0)}{\hat{s}(x^0)}\right) + \hat{s}(x^0)\,\phi\left(\frac{z_{\min} - \hat{Z}(x^0)}{\hat{s}(x^0)}\right) & , \hat{s}(x^0) > 0 \\ 0 & , \hat{s}(x^0) = 0 \end{cases}, \tag{7.87}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal distribution and density function. One can also express the expected improvement for a weighted sum scalarization in closed form. Let $f$ be an additional deterministic objective function and $\lambda \in [0,1]$ an arbitrary weight. The weighted improvement is then of the form

$$I_\lambda(x^0) = \max\Big(z_{\min} - \big(\lambda\,\tilde{Z}(x^0) + (1-\lambda)\,f(x^0)\big), 0\Big), \tag{7.88}$$

and we have for the weighted expected improvement with $K_\lambda := \frac{z_{\min} - \lambda\,\hat{Z}(x^0) - (1-\lambda)\,f(x^0)}{\lambda\,\hat{s}(x^0)}$

$$\begin{aligned} EI_\lambda(x^0) &= \int_{-\infty}^{\infty} I_\lambda(x)\,\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \int_{-\infty}^{K_\lambda} \Big(z_{\min} - \lambda\,\big(\hat{Z}(x^0) - \hat{s}(x^0)\,\varepsilon\big) - (1-\lambda)\,f(x^0)\Big)\,\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \lambda\,\hat{Z}(x^0) - (1-\lambda)\,f(x^0)\Big)\,\Phi(K_\lambda) - \lambda\,\hat{s}(x^0) \int_{-\infty}^{K_\lambda} \varepsilon\phi(\varepsilon)\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \lambda\,\hat{Z}(x^0) - (1-\lambda)\,f(x^0)\Big)\,\Phi(K_\lambda) + \frac{\lambda\,\hat{s}(x^0)}{\sqrt{2\pi}} \int_{-\infty}^{K_\lambda} -\varepsilon e^{-\varepsilon^2/2}\,\mathrm{d}\varepsilon \\ &= \Big(z_{\min} - \lambda\,\hat{Z}(x^0) - (1-\lambda)\,f(x^0)\Big)\,\Phi(K_\lambda) - \lambda\,\hat{s}(x^0)\,\phi(K_\lambda). \end{aligned} \tag{7.89}$$

As with the probability of improvement one want to maximize the expected improvement.

### 7.4.2 The EGO Algorithm

The expected improvement function is highly modular,i.e., and in large parts numerically equal to zero [92]. Therefore, one has to choose a suitable optimization approach to determine the global maximum of $EI$. In our case, we follow the Dakota implementation [57] and use the DIRECT algorithm (Algorithm 6) for the maximization of $EI$. The EGO algorithm as we use it is given as follows.

**Algorithm 7:** EGO

**Data:** $f$ $(,\nabla f)$, $N_{\text{LHS}}$, and $N_{\text{max.it}}$.
**Result:** Approximation of a global solution.
Generate a Latin hypercube sampling of size $N_{\text{LHS}}$ as an inital sampling $X^{(0)}$, see Subsection 7.3.1;
Set $l = 0$;
**while** $l \leq N_{max.it}$ **do**
  Construct a Kriging model from $X^{(l)}$, the MLE of the model parameters is done with the DIRECT algorithm, i.e., Algorithm 6;
  Use the mean $\hat{Z}(t^0)$ and its standard error $\hat{s}^2(t^0)$ of the Kriging model to formulate the expected improvement $EI$ (7.87);
  Maximize $EI$ on the Kriging model by using the DIRECT algorithm and attain the solution $\bar{x}^{(l)}$;
  Evaluate $f$ (and $\nabla f$) at $\bar{x}^{(l)}$;
  Add $\bar{x}^{(l)}$ and $f(\bar{x}^{(l)})$ to $X^{(l)}$, i.e., $X^{(l+1)} = X^{(l)} \cup \{\bar{x}^{(l)}, f(\bar{x}^{(l)})\}$;
  l=l+1;
**end**

Note that the function $f$ is in our case the weighted sum scalarization $J_\lambda$ and the only stopping condition is a maximum number of iterations $N_{\text{max.it}}$. One notable difference between this formulation of the algorithm, that is based on the Dakota implementation, and the first version of [92] is that the DIRECT algorithm is used for the optimization. In [92], this is done by a branch and bound method.

## 7.5 Coupling With Dakota

In this section, we give a brief overview of Dakota and how we coupled Dakota version 6.12 with an R script that computes the objective function values and gradients of $J_0$ and $J_1$ under Ubuntu 16.04.7 LTS. The Dakota software toolbox (open source under GNU LGPL) has many capabilities for optimization and uncertainty quantification (UQ). It includes, see, e.g., [56, 57],

- *optimization* with gradient and nongradient-based methods;
- *uncertainty quantification* with sampling, reliability, stochastic expansion, and epistemic methods;
- *parameter estimation* using nonlinear least squares (deterministic) or Bayesian inference (stochastic); and

- *sensitivity/variance analysis* with design of experiments and parameter study methods.

To access the methods implemented in Dakota one has to specify six blocks in the Dakota input file. These six specification blocks are *variables, interface, responses, model, method,* and *environment*. The relationship of these blocks is described in the user manual of Dakota [56] as follows. In each iteration of its algorithm, a *method* block requests a *variables-to-responses mapping* from its *model*, which the model fulfills through an *interface*. In Figure 7.1, this relationship is visualized. The *params* and *results* blocks are described later on.

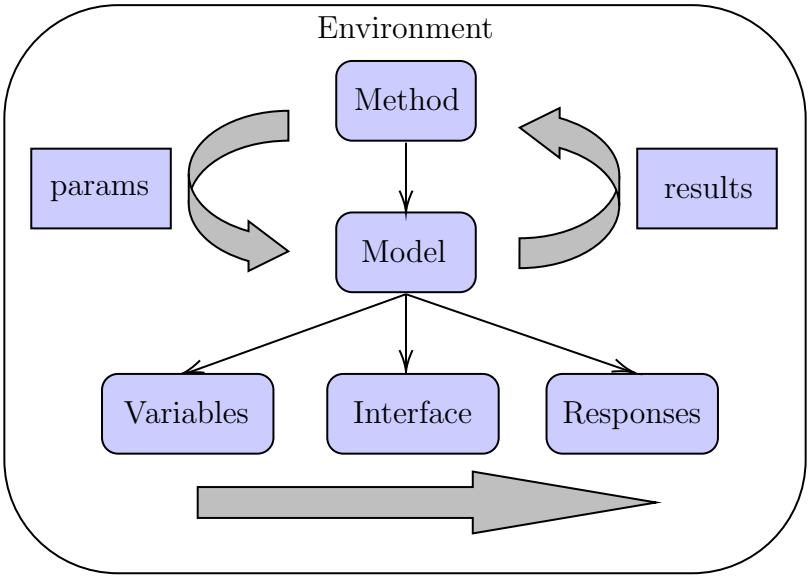



Figure 7.1: Relationship between the six specification blocks of a Dakota input file, see also [56].

These six blocks have to be specified in a *Dakota input file*, see [56]. In the following, we show an input file, Figure 7.2, that we used for the numerical tests and explain based on that file what the specifications of the blocks mean in our case. For a thorough introduction we refer to the Dakota user manual [56].
In Figure 7.2, the *environment* block states that the results should be saved in a tabular data file *'CeramcisRM01.dat'*. We had difficulties with this setting. In our case, this generated empty tabular data files. Therefore, we opt to save the computed values in the coupled R code. In the *method* block, the handle *efficient_global* invokes the implemented EGO Algorithm 7. It includes a LHS as an initial sampling, see [56], for which the handle *initial_samples* corresponds to $N_{LHS}$ in Algorithm 7. The handle *max_iterations* corresponds to $N_{max.it}$. The handle *use_derivatives* incorporates gradient information into the Kriging model, omitting this handle leads to the standard EGO algorithm without GEK. The *variables* block specifies our six optimization variables and their ranges, i.e., the designspace. In the *interface* block, the *fork* handle indicates that an external code is to be used for the evaluation of the objective function and gradients. This external code is accessed via the *analysis_driver* that invokes an Unix-Shell script *'GEKCeramics-DriverRM01.sh'* that starts a R script to compute the requested values. In each iteration $i$ Dakota writes an .txt file *'params.in.i'* with the variable values for the variables 'x1',

```
environment
  tabular_data
    tabular_data_file = 'CeramcisRM01.dat'

method
  efficient_global
  seed = 123456
  use_derivatives
  max_iterations = 211
  initial_samples = 141


variables
  continuous_design = 6
    lower_bounds    -0.05 -0.05 -0.05 0.05 0.05 0.05
    upper_bounds     0.25  0.65 0.25 0.4 0.4 0.4
    descriptors      'x1' 'x2' 'x3' 'x4' 'x5' 'x6'

interface
  analysis_drivers = 'GEKCeramicsDriverRM01.sh'
    fork
     parameters_file = 'params.in'
     results_file    = 'results.out'
     file_tag
     file_save

responses
  objective_functions = 1
  analytic_gradients
  no_hessians
```

Figure 7.2: Dakota input file for EGO with GEK for test case 1.

'x2', 'x3', 'x4', 'x5', 'x6', which correspond to our six optimization variables, and requests corresponding objective value and gradient evaluations from the analysis driver. After computing the values the analysis driver writes them also in a .txt file *'results.out.i'* which then is automatically read by Dakota. The handles *file_tag* and *file_save* are of technical nature and save the .txt files *'params.in.i'* in the workspace such that the R code can access them. Exemplary *'params.in.i'* and *'results.out.i'* files are depicted in Figure 7.3 and Figure 7.4. Finally, the *response* block specifies the requested values. The *objective_function = 1* handle indicates that Dakota expects one objective value for the variables. Note that there can be more than one objective function value requested depending on the optimization method, e.g., multiobjective optimization methods that are implemented in Dakota. The following discussion for the *gradients* handles also applies for the *hessians* handles. There are three options for the *gradients* handle *no_gradients,*

*analytical_gradients* and *numerical_gradients*, where *no_gradients* indicates that no gradient information is expected, i.e., EGO without GEK, *analytical_gradients* implies that the *interface* will return gradient information through the analysis driver, and *numerical_gradients* approximate the gradients with a finite difference method of Dakota.

```
                                6 variables
1.339235474811274e-03 x1
3.021429777979283e-01 x2
6.396233240499141e-02 x3
2.511367462300498e-01 x4
1.953218475344974e-01 x5
2.565019968075593e-01 x6
                                1 functions
                                3 ASV_1:obj_fn
                                6 derivative_variables
                                1 DVV_1:x1
                                2 DVV_2:x2
                                3 DVV_3:x3
                                4 DVV_4:x4
                                5 DVV_5:x5
                                6 DVV_6:x6
                                0 analysis_components
                                1 eval_id
```

Figure 7.3: Exemplary *'params.in.i'* file. The variables are denoted by 'x1', 'x2', 'x3', 'x4', 'x5' and 'x6'. The *functions* handle requests the value of one objective function. The active set value ($ASV$) indicates which values are expected by Dakota. It is ASV $\in \{1, 2, 3, 4, 5, 6, 7\}$ and we have for ASV $= 1$ that the objective function should be computed, for ASV $= 2$ the gradient and for ASV $=$ 4 the hessian. A sum of these values indicate that a combination is requested, e.g., ASV $= 3$ means that the objective value and the gradient are requested. The *derivative_variables* DVV indicate for which variables derivatives should be computed.

```
0.207271
[ 0.037847 0.115888 0.075434 0.183707 0.150483 0.177094 ]
```

Figure 7.4: Exemplary *'results.out.i'* file. The first row consists of the objective value, the second row of the gradient. The gradient is marked by the brackets [...], hessians would be marked by double brackets [[...][...]...[...]].

### 7.5.1 Routine for Consecutive Weighted Sum EGO Runs

In this subsection, we describe our implementation to apply EGO for consecutive weighted sum scalarizations with differing weights. For simplicity we refer to the Dakota input file

as *'Ceramics.in'* and to the R script that is invoked by the Unix-Shell analysis driver and computes the objective values and gradients as *Driver.R*. First, $n$ weights $\lambda$ are chosen for which weighted sum scalarizations should be minimized. These weights are saved in a $n$-dimensional array named *weights*. Then, a for-loop over $k = 1, \ldots, n$ is initiated, in which the $\lambda$ that is at position $k$ in the weights array, i.e., $\lambda = \text{weights}[k]$, is written in a .txt file *'weight.txt'* and saved in the workspace. Then, Dakota is started and the 'Ceramics.in' file is executed, i.e., EGO is started. During the optimization *'params.in.i.txt'* files are written and the analysis driver invokes *Driver.R* which then reads the *'weight.txt'* and *'params.in.i.txt'* files, constructs the weighted sum scalarization w.r.t. $\lambda$ from *'weight.txt'*, computes the objective values and gradients, and subsequently writes *'results.out.i.txt'* files for Dakota. At this point *Driver.R* also saves the computed objective values and gradients in a separate .RData file. This scheme is repeated until $k = n$ is reached. Figure 7.5 illustrates this routine.

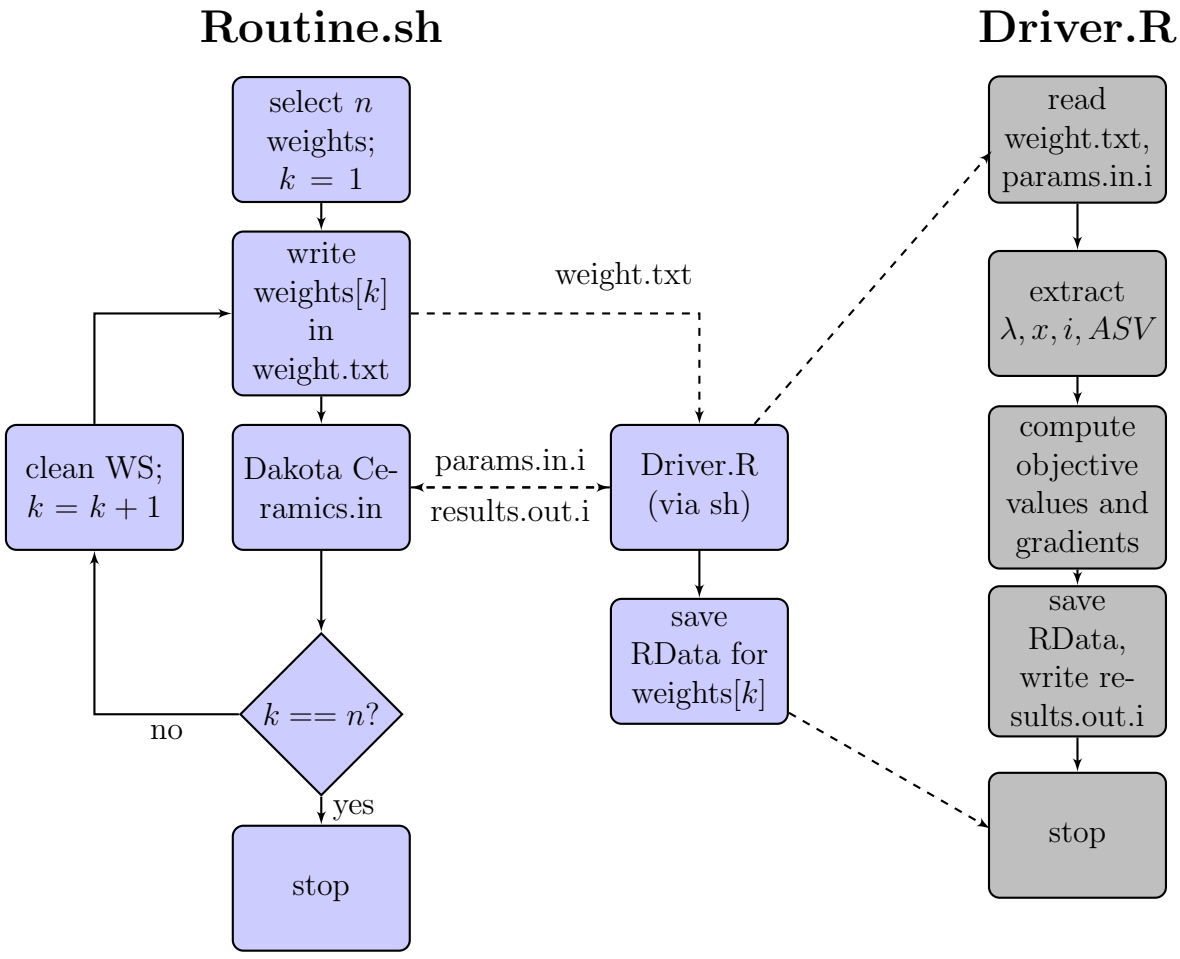



Figure 7.5: Routine to apply EGO to consecutive weighted sum sclarizations.

## 7.6 Numerical Results

In this section, we apply the EGO algorithm with Kriging and GEK as the underlying surrogate model to our biobjective shape optimization problem (3.11), respectively, and

compare the results with the solutions computed in Chapter 3 and Chapter 6. In the following, the results for the test cases introduced in Subsection 4.4.1 and Subsection 4.4.2 are described.

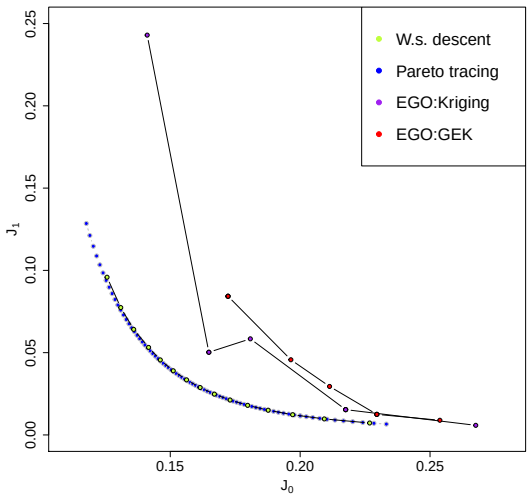
### 7.6.1 Test Case 1: A Straight Joint

We take the same specifications as in Subsection 4.4.1 and utilize the weighted sum scalarization $J_\lambda = (1-\lambda)J_0 + \lambda J_1$ as the objective function that is estimated by the Kriging predictor (7.67). Following Algorithm 7, we have to choose values for $N_{\text{LHS}}$ and $N_{\text{max.it}}$. Toward this end, we take the mean numbers of the objective function evaluations and gradient computations for the gradient descents with the weighted sum scalarization of Chapter 3. Recall that on average one gradient descent with the weighted sum scalarization, i.e., for one weight $\lambda$, needs 93.21 iterations and in each iteration 3.77 Armijo iterations to compute a solution. This equals on average $93.21 \times 3.77 = 351.51$ objective function evaluations, 93.21 gradient computations, and $93.21 + 351.51 \approx 445$ total evaluations. To be able to compare the solutions of EGO with the previously computed solutions w.r.t. the function evaluations, we set $N_{\text{LHS}} + N_{\text{max.it}} = 445$ when utilizing a Kriging model and $N_{\text{LHS}} + N_{\text{max.it}} = 352$ for the GEK approach. The total budget is then divided into $N_{\text{LHS}}$ and $N_{\text{max.it}}$, where $N_{\text{LHS}}$ attributes to 40% and $N_{\text{max.it}}$ to 60% of the budget. Hence, for the Kriging model we set $N_{\text{LHS}} = 178$ and $N_{\text{max.it}} = 267$ and for the GEK model $N_{\text{LHS}} = 141$ and $N_{\text{max.it}} = 211$, respectively. Furthermore, the design space is chosen in a way that it contains the starting shape, see Figure 5.1c, and the solutions of Chapter 3 and Chapter 6. The lower and upper bounds for the six optimization variables is given in Table 7.1.

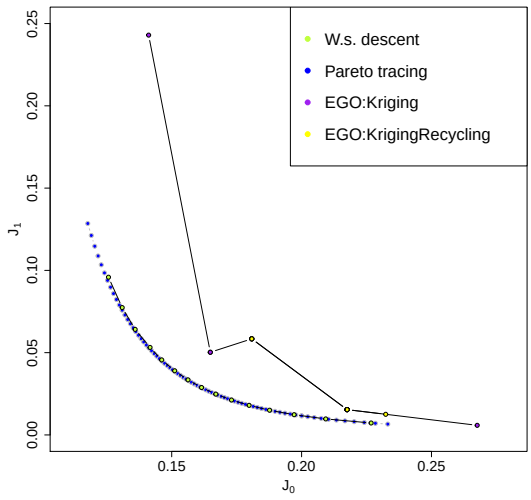
| Variables | $x_1 = \gamma_2^{\text{ml}}$ | $x_2 = \gamma_3^{\text{ml}}$ | $x_3 = \gamma_4^{\text{ml}}$ | $x_4 = \gamma_2^{\text{th}}$ | $x_5 = \gamma_3^{\text{th}}$ | $x_6 = \gamma_4^{\text{th}}$ |
|---|---|---|---|---|---|---|
| Lower bounds | -0.05 | -0.05 | -0.05 | 0.05 | 0.05 | 0.05 |
| Upper bounds | 0.25 | 0.65 | 0.25 | 0.4 | 0.4 | 0.4 |

Table 7.1: Lower and upper bounds for the optimization variables for the first test case, see Subsection 4.4.1.

The EGO algorithm is then applied for the weights $\lambda \in \{0.2, 0.3, \ldots, 0.9\}$. In Figure 7.6a, the results of these optimization runs with a Kriging model (purple) and a GEK model (red) are compared with the weighted sum gradient descent solutions of Chapter 3 (green) and the Pareto tracing by numerical integration solutions (blue).
Further, in Table 7.2 the $J_\lambda$ objective values, for $\lambda \in \{0.2, 0.3, \ldots, 0.9\}$, of the optimal solutions computed by the four methods depicted in Figure 7.6a are compared. In Figure 7.7, some exemplary shapes for these four methods are shown. Solutions of both EGO methods are not quite straight rods. The Kriging model outperformed the GEK model for every $\lambda \in \{0.2, 0.3, \ldots, 0.9\}$. A reason for this is that EGO with the GEK model had a smaller budget of iterations than with the Kriging model. Apparently, the additional gradient information could not compensate for this deficit. All of this, combined with Table 7.2 and Figure 7.6a clearly shows, that the gradient descent of Chapter 3 and Pareto tracing by numerical integration of Chapter 6 outperform the EGO solutions, whether gradient information is incorporated or not. Another observation for both EGO methods is that for some distinct weights the same local Pareto optimal solution is computed, i.e.,




(a) Comparison of the objective values of $J_0$ and $J_1$ of the weighted sum gradient descent solutions of Chapter 3 (green), the Pareto tracing by numerical integration solutions (blue), EGO with a Kriging model (purple) and a GEK model (red).




(b) Comparison of the objective values of $J_0$ and $J_1$ of EGO with a Kriging model without initial sample recycling (purple) and with recycling (yellow).

Figure 7.6: Comparison of the numerical results for test case 1.

there are less distinct solutions than distinct weights. Furthermore, EGO with a Kriging or GEK model computed only five distinct solutions for eight different weights, respectively. Dakota also allows to integrate sampling data into the Kriging model, for GEK this option is unfortunately not available. Doing this comes in our case with an additional cost of 7 function evaluations that Dakota needs to check the incorporated points. We used this capability of Dakota to recycle the Latin hypercube sampling computed for the optimization w.r.t. the first weight $\lambda = 0.2$, i.e., we incorporated the $N_{LHS} = 267$ sample points of this LHS into the models for the weights $\lambda \in \{0.3, \ldots, 0.9\}$ and thereby saved the computational cost of $N_{LHS} - 7 = 260$ evaluations. Note that, since we save the computed weighted sum and both objective function values in an separate R file, we can adjust the weighted sum objective values of this recycled LHS for weight $\lambda = 0.2$ to the other weights $\lambda \in \{0.3, \ldots, 0.9\}$. In Figure 7.6b, the results of this approach (yellow) are shown. The computed solutions coincide with the already computed EGO with Kriging solutions. As with the non recycled case, the method gets stuck in local minima, i.e., eight distinct weights produce only three distinct solutions. This nicely shows that if gradient information is available a gradient descent with a weighted sum scalarization is preferable to a global approach like EGO. For both cases a continuation of a solution with Pareto tracing by numerical integration is in general possible, but since the gradient descent clearly outperforms EGO, it should be used to generate an initial value for Pareto tracing by numerical integration method.

| $\lambda$ | EGO:Kriging | EGO:GEK | Grad. Descent | Pareto Tracing |
|---|---|---|---|---|
| 0.2 | 0.14158868 | 0.15437675 | 0.11969789 | 0.11890384 |
| 0.3 | 0.13012732 | 0.14558050 | 0.11438980 | 0.11392020 |
| 0.4 | 0.13148747 | 0.13567980 | 0.10592921 | 0.10539918 |
| 0.5 | 0.11583851 | 0.11985043 | 0.09486769 | 0.09435064 |
| 0.6 | 0.09562510 | 0.09865657 | 0.08167443 | 0.08118969 |
| 0.7 | 0.07541169 | 0.07695466 | 0.06652542 | 0.06602785 |
| 0.8 | 0.05519827 | 0.05525275 | 0.04925905 | 0.04870324 |
| 0.9 | 0.03122986 | 0.03266942 | 0.02920745 | 0.02853962 |

Table 7.2: Comparison of the objective values of $J_\lambda$, for $\lambda \in \{0.2, 0.3, \ldots, 0.9\}$, of the solutions computed with EGO with Kriging, EGO with GEK, gradient descent (Chapter 3) and Pareto tracing by numerical integration (Chapter 6) for test case 1.

### 7.6.2 Test Case 2: An S-Shaped Joint

We take the same specifications as in Subsection 4.4.2 and also estimate $J_\lambda$ with the surrogate models. For this test case the gradient descents with weighted sum scalarizations of Chapter 3 needs 106.64 iterations and in each iteration 5.25 Armijo iterations to compute a solution. This equals on average $106.64 \times 5.25 = 560.11$ objective function evaluations and $106.64 + 560.11 \approx 667$ total evaluations, i.e., $N_{\text{LHS}} + N_{\text{max.it}} = 667$ for the Kriging model and $N_{\text{LHS}} + N_{\text{max.it}} = 561$ when incorporating gradient information. We therefore set $N_{\text{LHS}} = 267$ and $N_{\text{max.it}} = 400$ for the Kriging model and set $N_{\text{LHS}} = 225$ and $N_{\text{max.it}} = 336$ for GEK, respectively. Here, we also ensure that the starting shape , see Figure 4.5c, and the solutions for this test case of Chapter 3 and Chapter 6 are included in the design space, see Table 7.3.

| Variables | $x_1 = \gamma_2^{\text{ml}}$ | $x_2 = \gamma_3^{\text{ml}}$ | $x_3 = \gamma_4^{\text{ml}}$ | $x_4 = \gamma_2^{\text{th}}$ | $x_5 = \gamma_3^{\text{th}}$ | $x_6 = \gamma_4^{\text{th}}$ |
|---|---|---|---|---|---|---|
| Lower bounds | 0.1 | -0.2 | -0.2 | 0.05 | 0.05 | 0.05 |
| Upper bounds | 0.7 | 0.4 | 0.4 | 0.7 | 0.7 | 0.7 |

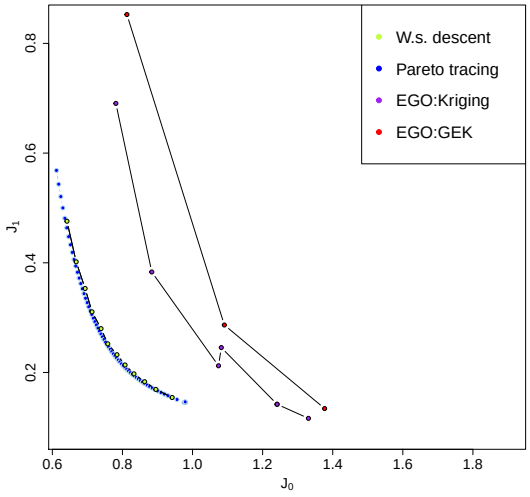
Table 7.3: Lower and upper bounds for the optimization variables for the second test case, see Subsection 4.4.2.

For this test case the EGO algorithm with a Kriging model is applied for the weights $\lambda \in \{0.2, 0.3, \ldots, 0.8\}$. For the GEK runs only the weights $\lambda \in \{0.2, 0.5, 0.8\}$ are used, since each computation took over eight days to complete and we see no added insight from further (time expensive) points. A comparison of the results of EGO computed with a Kriging model (purple) and a GEK model (red), respectively, weighted sum gradient descent solutions of Chapter 3 (green) and the Pareto tracing by numerical integration solutions (blue) is shown in Figure 7.8a. In Table 7.4, the weighted sum objective values are compared. Exemplary shapes are illustrated in Figure 7.9. As with test case 1 one can observe that the gradient descent and Pareto tracing by numerical integration method outperform both EGO methods. The shapes computed by the EGO methods do not really resemble the shapes computed by the other methods. Here, EGO with a Kriging model also outperformed EGO with a GEK model.

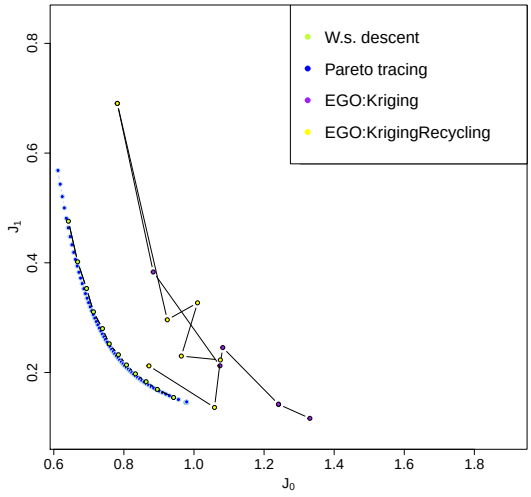
| $\lambda$ | EGO:Kriging | EGO:GEK | Grad. Descent | Pareto Tracing |
|---|---|---|---|---|
| 0.2 | 0.7631164 | 0.8207741 | - | 0.6031819 |
| 0.3 | 0.7336212 | - | 0.5881540 | 0.5849745 |
| 0.4 | 0.7293749 | - | 0.5521766 | 0.5498210 |
| 0.5 | 0.6637988 | 0.6887095 | 0.5052462 | 0.5034985 |
| 0.6 | 0.5817372 | - | 0.4512161 | 0.4483271 |
| 0.7 | 0.4718165 | - | 0.3872412 | 0.3849630 |
| 0.8 | 0.3592713 | 0.3828001 | 0.3119188 | 0.3119188 |

Table 7.4: Comparison of the objective values of $J_\lambda$, for $\lambda \in \{0.2, 0.3, \ldots, 0.8\}$, of the solutions computed with EGO with Kriging, EGO with GEK, a gradient descent (Chapter 3) and Pareto tracing by numerical integration (Chapter 6) for test case 2. Note that the weighted sum gradient descent did not converge for $\lambda = 0.2$ and is therfore omitted from the comparison.

As with the first test case, we applied the recycling of the LHS approach. The results (yellow) are illustrated in Figure 7.8b. The recycling yields better solutions for some weights for in total less evaluations, but still does not reach the quality of solutions of the gradient descent with a weighted sum scalarization or of Pareto tracing by numerical integration.




(a) Comparison of the objective values of $J_0$ and $J_1$ of the weighted sum gradient descent solutions of Chapter 3 (green), the Pareto tracing by numerical integration solutions (blue), EGO with a Kriging model (purple) and a GEK model (red).

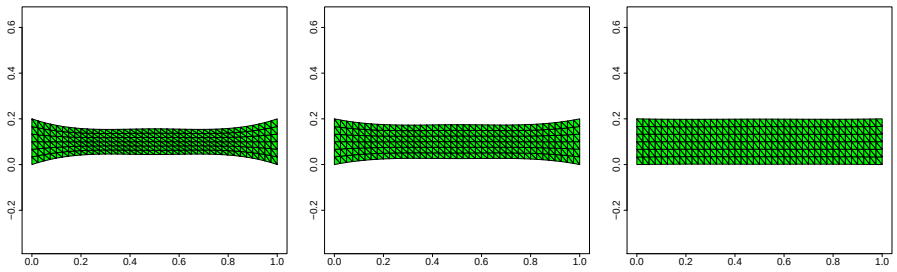



(b) Comparison of the objective values of $J_0$ and $J_1$ of EGO with a Kriging model without initial sample recycling (purple) and with recycling (yellow).

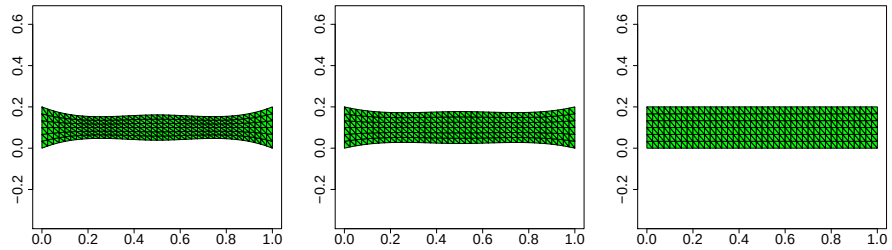
Figure 7.8: Comparison of the numerical results for test case 2.

Wrapping up, one can conclude that for our biobjective shape optimization problem 3.11, where gradient information is available, a gradient descent with a weighted sum

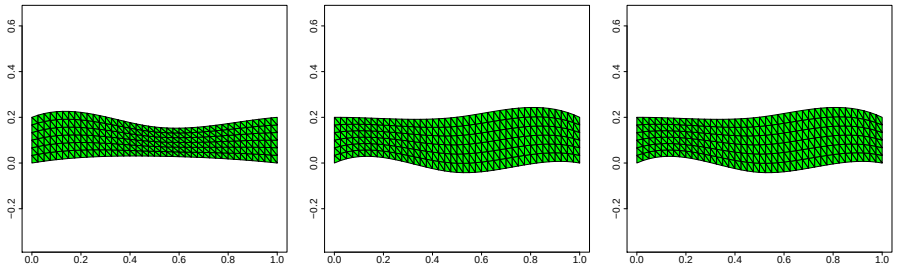
scalarization (Chapter 3) and afterwards applying Pareto tracing by numerical integration (Chapter 6) to the solution is preferable to a global approach like EGO. In aerodynamic design problems similar observations were made. In [153], a gradient descent method also outperformed an evolutionary algorithm, strengthening our argument for the inclusion of gradient information when available.



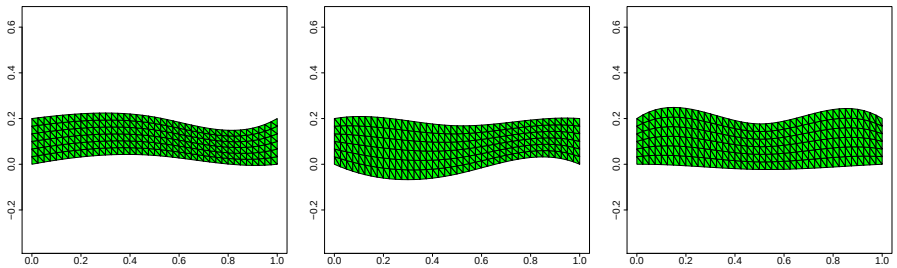
(a) WS Descent: $\lambda = 0.2$ (b) WS Descent: $\lambda = 0.5$ (c) WS Descent: $\lambda = 0.8$



(d) PT: $\lambda = 0.2$ (e) PT: $\lambda = 0.5$ (f) PT: $\lambda = 0.8$

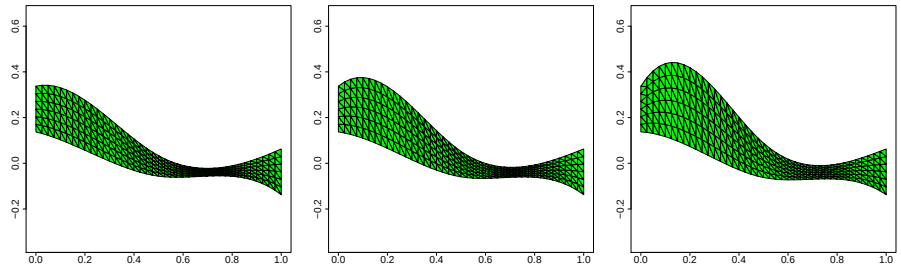


(g) Kriging: $\lambda = 0.2$ (h) Kriging: $\lambda = 0.5$ (i) Kriging: $\lambda = 0.8$

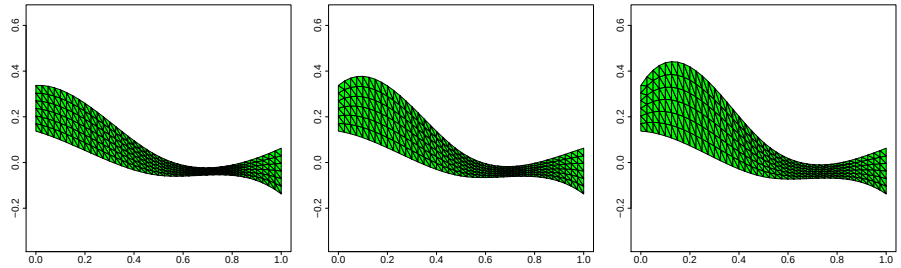


(j) GEK: $\lambda = 0.2$ (k) GEK: $\lambda = 0.5$ (l) GEK: $\lambda = 0.8$

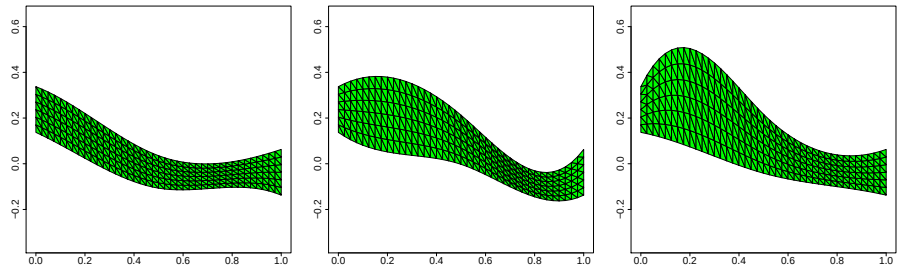
Figure 7.7: Exemplary solutions for test case 1 of the weighted sum method (row 1), Pareto tracing (row 2), EGO with Kriging (row 3) and GEK (row 4) for the weights $\lambda = 0.2, 0.5, 0.8$.



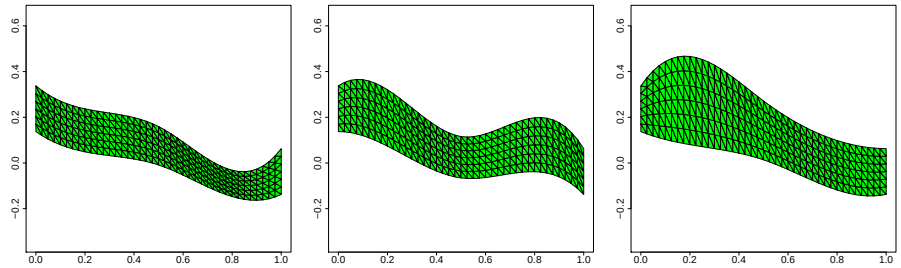
(a) WS Descent: $\lambda = 0.25$ (b) WS Descent: $\lambda = 0.5$ (c) WS Descent: $\lambda = 0.8$



(d) PT: $\lambda = 0.2$ (e) PT: $\lambda = 0.5$ (f) PT: $\lambda = 0.8$



(g) Kriging: $\lambda = 0.2$ (h) Kriging: $\lambda = 0.5$ (i) Kriging: $\lambda = 0.8$



(j) GEK: $\lambda = 0.2$ (k) GEK: $\lambda = 0.5$ (l) GEK: $\lambda = 0.8$

Figure 7.9: Exemplary solutions for test case 2 of the weighted sum method (row 1), Pareto tracing (row 2), EGO with Kriging (row 3) and GEK (row 4) for the weights $\lambda = 0.2, 0.5, 0.8$. Note that the weighted sum descent did not converge for $\lambda = 0.2$, therefore we included the solution for $\lambda = 0.25$.

# 8 Conclusion and Outlook

In this chapter, we give our conclusions and an outlook for each of the gradient-based optimization methods considered in Chapters 5, 6, and 7 separately. Note that some parts were already published in [46, 19].

### Gradient Based Biobjective Shape Optimization to Improve Reliability and Cost of Ceramic Components

We have developed a modelling and solution approach for biobjective PDE constrained shape optimization of ceramic components. The mechanical integrity of the component on one hand, and the cost of the component on the other hand, were considered as two pivotal optimization criteria. A probabilistic approach was used to assess the mechanical integrity (i.e., the reliability) of the component, which allows, in combination with a finite element discretization and an adjoint approach for gradient computations, the efficient calculation of derivative information. Approximations of the Pareto front were computed using two different approaches: (1) parametric weighted sum scalarizations in combination with a single objective gradient descent method, and (2) a biobjective descent algorithm with parametric scalings of the objective functions. Numerical results for 2D test cases visualize the trade-off between the reliability and the cost, and hence pave the way for an informed selection of a most preferred design. A generalization to 3D shapes seems possible and is the next natural step. Moreover, further optimization criteria like, for example, reliability w.r.t. other loading scenarios, minimal natural frequencies, and/or efficiency criteria, can be included into a general multiobjective shape optimization problem.

### Pareto Tracing by Numerical Integration

We have presented a novel approach for approximating the Pareto front by tracing it using numerical time integration. The optimality conditions of a scalarization $J_\lambda$ were differentiated w.r.t. the scalarization parameter $\lambda$ to obtain an implicit ODE describing the front. If second order optimality conditions are fulfilled, an explicit ODE is obtained with a Lipschitz right hand side and the existence and uniqueness of the solution that is a representation of the Pareto front was shown. The smoothness of the Pareto front depends on the smoothness of the objective function. Further, we have shown how this extends to $\epsilon$-critical starting points. The use of standard explicit Runge-Kutta methods was established and the well-known convergence estimates can be applied. The technique was demonstrated for a simple biobjective convex quadratic optimization problem, as well as for problems originating from shape optimization.
We have not yet covered the effects of using adapted and/or adaptive step sizes in $\lambda$, e.g., in order to obtain equispaced points on the Pareto front. Different approaches are possible in this respect, see, for example, [53, 132]. Further, we will extend the approach to constrained problems via KKT conditions, and also consider other scalarizations. While

we have only considered the biobjective case here, the approach can also be used to handle more than two criteria. In the case of $q+1$ criteria, the front can be described by a $q$-dimensional functional (using again, e.g., weighted sum scalarizations with $q$ independent scalarization parameters) that can be obtained numerically using a $q$-dimensional mesh and numerical integration starting from some mesh point. This will also be considered in the future.

### EGO and Gradient Enhanced Kriging

We have applied state of the art methods, i.e. the surrogate model based global optimization methods EGO and EGO with GEK, to benchmark the results obtained with the weighted sum method of Chapter 5 and Pareto tracing by numerical integration of Chapter 6. The expected improvement was maximized on the Kriging surrogate model which was fitted with (EGO with GEK) and without (EGO) gradient information, respectively. The DIRECT algorithm was used for the maximization of the expected improvement and the maximum likelihood estimation which is needed for the construction of a Kriging surrogate model. The achieved results with these benchmark methods were all dominated by the results obtained in Chapters 5 and 6. Thus, we conclude that if for a problem gradient information is available it may be favorable to distribute the computational budget more toward gradient descent and gradient-based continuation methods than commonly used surrogate model based methods.
We have not yet considered other state of the art methods like, e.g., biobjective evolutionary algorithms, as additional benchmark methods. In future work, it would be of interest to compare the results of Chapters 5 and 6 with more benchmark methods to have a clearer insight on the performance of gradient-based biobjective optimization methods.

# Bibliography


[1] P. Abrahamsen. Gaussian random fields and correlation functions. Technical report, Norwegian Computing Center, Oslo, Norway, April 1997.

[2] R.J. Adler. *The Geometry of Random Fields (Probability & Mathematical Statistics S.)*. John Wiley & Sons Inc, June 1981.

[3] R.J. Adler. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*. Ims Lecture Series. Institute of Mathematical Statistics, 1990.

[4] R.P. Agarwal and D. O'Regan. *An Introduction to Ordinary Differential Equations*. Universitext. Springer, New York, 2008.

[5] G. Allaire. *Shape Opimization by the Homogenisation Method*. Springer-Verlag, Berlin-Heidelberg-New York, 2001.

[6] G. Allaire and F. Jouve. Minimum stress optimal design with the level set method. *Eng. Anal. Bound. Elem.*, 32:909–218, 2008.

[7] E.L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[8] J. Alonso, H.S. Chung, and T. Juan. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. *40th AIAA aerospace sciences meeting and exhibit*, 03 2002.

[9] W. Arendt and K. Urban. *Partielle Differenzialgleichungen. Eine Einführung in analytische und numerische Methoden*. Spektrum Akademischer Verlag, Heidelberg, 2010.

[10] U.M. Ascher and L.R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[11] M. Aulich and U. Siller. High-dimensional constrained multiobjective optimization of a fan stage. volume Volume 7: Turbomachinery, Parts A, B, and C of *Turbo Expo: Power for Land, Sea, and Air*, pages 1185–1196, 06 2011.

[12] J. Backhaus. *Adjungierte Strömungssimulation und gradientenbasierte Ersatzmodelle in der Turbomaschinenauslegung*. PhD thesis, Ruhr-Universität Bochum, Deutschland, 2020.

[13] M. Bäker, H. Harders, and J. Rösler. *Mechanisches Verhalten der Werkstoffe*. Vieweg+Teubner, 3rd edition, 2008.

[14] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming – Theory and Algorithms*. Wiley, 3rd edition, 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, dec 1971.

[16] P. Billingsley. *Probability and Measure*. John Wiley and Sons, second edition, 1986.

[17] G.R. Bitran. Theory and algorithms for linear multiple objective programs with zero–one variables. *Mathematical Programming*, 17(1):362–390, dec 1979.

[18] L. Bittner. *On shape calculus with elliptic PDE constraints in classical function spaces*. PhD thesis, University of Wuppertal, 2018.

[19] M. Bolten, O.T. Doganay, H. Gottschalk, and K. Klamroth. Tracing locally Pareto-optimal points by numerical integration. *SIAM Journal on Control and Optimization*, 59(5):3302–3328, jan 2021.

[20] M. Bolten, H. Gottschalk, and S. Schmitz. Minimal failure probability for ceramic design via shape control. *J. Optim. Theory Appl. 166*, pages 983–1001, 2015.

[21] Matthias Bolten, Hanno Gottschalk, Camilla Hahn, and Mohamed Saadi. Numerical shape optimization to decrease failure probability of ceramic structures. *Computing and Visualization in Science*, Jul 2019.

[22] M.A. Bouhlel and J.R.R.A. Martins. Gradient-enhanced kriging for high-dimensional problems. *Engineering with Computers*, 35(1):157–173, feb 2018.

[23] D. Braess. *Finite Elements. Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, Cambridge, 1997.

[24] A. Brückner-Foit, T. Fett, D. Munz, and K. Schirmer. Discrimination of multiaxiality criteria with the brasilian disk test. *J. Eur. Ceram. Soc.*, 17:689–696, 1997.

[25] D. Bucur and G. Buttazzo. *Variational Methods in Shape Optimization Problems*. Birkhäuser, 2005.

[26] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd., New York, third edition, 2016. With a foreword by J. M. Sanz-Serna.

[27] A. Charnes and W.W. Cooper. *Management Models and the Industrial Applications of Linear Programming*. John Wiley, New York, 1961.

[28] D. Chenais. On the existence of a solution in a domain identification problem. *Journal of Mathematical Analysis and Applications*, 52:189–289, 1975.

[29] D.V. Chirkov, A.S. Ankudinova, A.E. Kryukov, S.G. Cherny, and V.A. Skorospelov. Multi-objective shape optimization of a hydraulic turbine runner using efficiency, strength and weight criteria. *Structural and Multidisciplinary Optimization*, 58:627–640, 2018.

[30] G. Christakos. *Random Field Models in Earth Sciences*. Elsevier, 1992.

[31] H.S. Chung and J. Alonso. Using gradients to construct response surface models for high-dimensional design optimization problems. *39th AIAA aerospace sciences meeting and exhibit*, 03 2002.

[32] S. Conti, H. Held, M. Pach, M. Rumpf, and R. Schultz. Shape optimization under uncertainty - a stochastic programming perspective. *SIAM J Optim.*, 19 (4):1610–1632, 2008.

[33] H. Cramér and M.R. Leadbetter. *Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications*. Wiley series in probability and mathematical statistics. Tracts on probability and statistics. Wiley, 1967.

[34] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, apr 1990.

[35] N. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, Inc., sep 1993.

[36] C. Cristescu and J. Knowles. Surrogate-based multiobjective optimization : ParEGO update and test. In *Workshop on Computational Intelligence*, 2015.

[37] K. Dalbey. Efficient and robust gradient enhanced Kriging emulators. aug 2013.

[38] 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. *Struct. Optim.*, 14:63–69, 1997.

[39] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[40] K. Deb and T. Goel. Multi-objective evolutionary algorithms for engineering shape design. In *Evolutionary Optimization*, volume 48 of *International Series in Operations Research & Management Science*, pages 147–175. Springer, Boston, MA, 2002.

[41] M.C. Delfour and J.-P. Zolésio. *Shape and Geometries: Analysis, Differential Calculus, and Optimization*. SIAM, 2nd edition, 2011.

[42] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–136, 2005.

[43] J.A. Désidéri. Multiple-gradient descent algorithm (MGDA). Research Report 00389811, INRIA, 2009.

[44] J.A. Désidéri. MGDA II: A direct method for calculating a descent direction common to several criteria. Research Report RR-7922, INRIA, April 2012.

[45] O. T. Doganay. Multicriteria optimization with shape gradients. Master's thesis, University of Wuppertal, 2017.

[46] O. T. Doganay, H. Gottschalk, C. Hahn, K. Klamroth, J. Schultes, and M. Stiglmayr. Gradient based biobjective shape optimization to improve reliability and cost of ceramic components. *Optimization and Engineering*, 2020.

[47] J.L. Doob. *Stochastic Processes*. Probability and Statistics Series. Wiley, 1953.

[48] L.M.G. Drummond and B.F. Svaiter. A steepest descent method for vector optimization. *Journal of Computational and Applied Mathematics*, 175(2):395–414, 2005.

[49] R.G. Duran and M.A. Muschietti. The Korn inequality for Jones domains. *Electronic J. Diff. Equations*, 127:1–10, 2004.

[50] P. Duysinx and M.P. Bendsœ. Topology optimization of continuum structures with local stress constraints. *Internat. J. Numer. Methods Engrg.*, 43 (8):1453–1478, 1998.

[51] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius, and S. Zionts. Multiple criteria decision making, multiattribute utility theory: the next ten years. *Management science*, 38(5):645–654, 1992.

[52] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, 2005.

[53] G. Eichfelder. An adaptive scalarization method in multi-objective optimization. *SIAM J. Optim.*, 19:1694–1718, 2009.

[54] K. Eppler. On Hadamard shape gradient representations in linear elasticity. Unpublished manuscript, 2017.

[55] K. Eppler, H. Harbrecht, and R. Schneider. On convergence in elliptic shape optimization. *SIAM J. Control Optim.*, 45:61–83, 2007.

[56] B.M. Adams et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 6.14 user's manual. Technical report, Sandia National Laboratories, 2021.

[57] K.R. Dalbey et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 6.14 theory manual. Technical report, Sandia National Laboratories, 2021.

[58] J.P. Evans and R.E. Steuer. A revised simplex method for linear multiple objective programs. *Mathematical Programming*, 5(1):54–72, dec 1973.

[59] J. Fliege, L.G. Drummond, and B.F. Svaiter. Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.

[60] J. Fliege and B.F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[61] J. Fliege, A.I.F. Vaz, and L.N. Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 2018. to appear.

[62] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *ICGA*, 1993.

[63] A.I.J Forrester, N.W. Bressloff, and A.J. Keane. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 462(2071):2177–2204, mar 2006.

[64] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, jan 2009.

[65] A.I.J. Forrester, A. Sobester, and A.J. Keane. *Engineering Design via Surrogate Modelling - A Practical Guide.* Wiley, 2008.

[66] N. Fujii. Lower semicontinuity in domain optimization problems. *J. Optim. Theory Appl.*, 59:407–422, 1988.

[67] J. Gablonsky. Direct version 2.0 userguide. Technical report, North Carolina State University, Center for Research in Scientific Computation, Raleigh, NC, 2001.

[68] A.M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3):618–630, 1968.

[69] A.M. Geoffrion, J. Dyer, and A. Feinberg. An interactive approach for multicriterion optimization with an application to the operation of an academic department. *Management Science*, 19:357–368, 12 1972.

[70] M. Giacomini, J.A. Désidéri, and R. Duvigneau. Comparison of multiobjective gradient-based methods for structural shape optimization. Technical Report RR-8511, INRIA, 2014.

[71] A. Giunta, L. Swiler, S. Brown, M. Eldred, M. Richards, and E. Cyr. The surfpack software library for surrogate modeling of sparse irregularly spaced multidimensional data. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, sep 2006.

[72] H. Gottschalk and M. Reese. An analytical study in multi-physics and multi-criteria shape optimization. *J. Optim. Theory Appl. 189*, pages 486–512, 2021.

[73] H. Gottschalk and M. Saadi. Shape gradients for the failure probability of a mechanic component under cyclic loading: a discrete adjoint approach. *Computational Mechanics*, 64(4):895–915, 2019.

[74] H. Gottschalk, M. Saadi, O. Doganay, K. Klamroth, and S. Schmitz. Adjoint method to calculate the shape gradients of failure probabilities for turbomachinery components. *ASME TURBO-Expo*, GT2018-75759, 2018.

[75] H. Gottschalk and S. Schmitz. Optimal reliability in design for fatigue life. *SIAM Journal of Control and Optimization*, 52 (5):2727–2752, 2015.

[76] D. Gross and T. Seelig. *Fracture Mechanics. With an Introduction to Micromechanics.* Springer, 2006.

[77] J. Guddat. Parametric optimization: pivoting and predictor-corrector continuation, a sur- vey. In J. Guddat, H.Th. Jongen, B. Kummer, and F. Nožička, editors, *Parametric Optimization and Related Topics*, pages 125–162, Berlin, 1987. Akademie-Verlag.

[78] J. Hadamard. Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées. *Imprimerie nationale*, 1907.

[79] C. Hahn. Optimal reliability for ceramic structures. Master's thesis, Bergische Universität Wuppertal, 2016.

[80] C. Hahn. *Auto-generated structured meshes for evolving domains.* PhD thesis, University of Wuppertal, 2021.

[81] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I*, volume 8 of *Springer Series in Computational Mathematics.* Springer-Verlag, New York, second edition, 1993. Nonstiff problems.

[82] Z.H. Han. Improving adjoint-based aerodynamic optimization via gradient-enhanced kriging. 01 2012.

[83] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization.* SIAM, 2003.

[84] J.C. Helton and F.J. Davis. Illustration of sampling-based methods for uncertainty and sensitivity analysis. *Risk Analysis*, 22(3):591–622, jun 2002.

[85] C. Hillermeier. *Nonlinear Multiobjective Optimization.* Birkhäuser, Basel, 2001.

[86] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched Pareto genetic algorithm for multi-objective optimization. *Proceedings of the 1st IEEE Conference on Computation Evolutionary*, 1:82 – 87 vol.1, 07 1994.

[87] J.P. Ignizio. *Goal Programming and Extensions.* Lexington Books. Lexington Books, 1976.

[88] R. Iman and M.J. Shortencarier. Fortran 77 program and user's guide for the generation of latin hypercube and random samples for use with computer models. 03 1984.

[89] J. Jahn. Multiobjective search algorithm with subdivision technique. *Comput. Optim. Appl.*, 35:161–175, 2006.

[90] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[91] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, oct 1993.

[92] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[93] O. Kallenberg. *Random Measures*. Akademie-Verlag, Berlin, 1983.

[94] A.J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal*, 44(4):879–891, apr 2006.

[95] A. Klenke. *Probability Theory: A Comprehensive Course*. Springer, 2008.

[96] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. Technical Report TR-COMPSYSBIO-2004-01, University of Manchester, September 2004.

[97] P.J. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24(2):277–287, February 1986.

[98] D. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand, by d.g. krige, published in the journal, december 1951 : introduction by the author. *Journal of The South African Institute of Mining and Metallurgy*, 52:201–203, 1951.

[99] H.W. Kuhn and A.W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.

[100] A. Laurain and K. Sturm. Distributed shape derivative via averaged adjoint method and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50 (4):1241–1267, 2016.

[101] S.M. Lee. *Goal Programming for Decision Analysis*. Management and communications series. Auerbach Publishers, 1972.

[102] I. Lepot, M. Leborgne, R. Schnell, J. Yin, G. Delattre, F. Falissard, and J. Talbotec. Aero-mechanical optimization of a contra-rotating open rotor and assessment of its aerodynamic and acoustic characteristics. volume 225, 03 2011.

[103] A. Martín and O. Schütze. Pareto Tracer: a predictor-corrector method for multi-objective optimization problems. *Eng. Optim.*, 50:516–536, 2018.

[104] B. Martin, A. Goldsztejn, L. Granvilliers, and C. Jermann. On continuation methods for non-linear bi-objective optimization: towards a certified interval-based approach. *Journal of Global Optimization*, 64:3–16, 2016.

[105] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, dec 1963.

[106] M. Mckay, R. Beckman, and W. Conover. A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 05 1979.

[107] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25:86–98, 2003.

[108] P. W. Michor and D. Mumford. Riemannian geometries on spaces of planar curves. *J. Europ. Math. Soc.*, 8:1–48, 2006.

[109] K. Miettinen. *Nonlinear Multiobjective Optimization*. Springer, New York, 1998.

[110] R. Morell. Brevier technical ceramics. Technical report, Verband der Keramischen Industrie e.V, Information Center Technical Ceramics, 2004.

[111] M. Morris, T. Mitchell, and D. Ylvisaker. Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35, 03 1993.

[112] D. Munz and T. Fett. *Ceramics - Mechanical Properties, Failure Behaviour, Materials Selection*. Springer, N.Y., Berlin, Heidelberg, 2001.

[113] K. Nikodem and Z. Páles. Characterizations of inner product spaces by strongly convex functions. *Banach Journal of Mathematical Analysis 5(1):83–87 · January 2011*, 2011.

[114] V. Pareto. *Manual d' éeconomie politique*. F. Rouge, Lausanne, 1896.

[115] S. Peitz. *Exploiting structure in multiobjctive optimization and optimal control*. PhD thesis, Paderborn University, 2017.

[116] S. Peitz and M. Dellnitz. A survey of recent trends in multiobjective optimal control – surrogate models, feedback control and objective reduction. *Math. Comput. Appl.*, 23, 2018.

[117] J. Peter and M. Marcelet. Comparison of surrogate models for turbomachinery design. *WSEAS Transactions on Fluid Mechanics*, 3, 01 2008.

[118] P.Gangl, Köthe S, C. Mellak, A. Cesarano, and A. Mütze. Multi-objective free-form shape optimization of a synchronous reluctance machine. *arXiv, preprint, arXiv:2010.10117*, 2020.

[119] R. Picelli, S. Townsend, C. Brampton, J. Noratoc, and H.A. Kimad. Stress-based shape and topology optimization with the level set method. *Computer Methods in Applied Mechanics and Engineering*, 329:1–23, 2018.

[120] L. Piegl and W. Tiller. *The NURBS Book. Monographs in Visual Communication*. Springer, New York, 2000.

[121] A. Potschka, F. Logist, J.F. Van Impe, and H.G. Bock. Tracing the Pareto frontier in bi-objective optimization problems by ODE techniques. *Numerical Algorithms*, 57:217–233, 2011.

[122] T.H. Pulliam, M. Nemec, T. Holst, and D.W. Zingg. Comparison of evolutionary (genetic) algorithm and adjoint methods for multi-objective viscous airfoil optimization. In *41st Aerospace Science Meeting and Exhibit, 6-9 January 2003, Reno, Nevada*, number 2003-0298 in AIAA Paper, 2003.

[123] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R.V., and P.K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, jan 2005.

[124] J.O. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis with R and MATLAB*. Springer, 2009.

[125] M. Ringkamp, S. Ober-Blöbaum, M. Dellnitz, and O. Schütze. Handling high-dimensional problems with multi-objective continuation methods via successive approximation of the tangent space. *Eng. Optim.*, 44(9):1117–1146, 2012.

[126] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[127] S. Roudi, H. Riesch-Oppermann, and O. Kraft. Advanced probabilistic tools for the uncertainty assessment in failure and lifetime prediction of ceramic components. *Materialwissenschaften u. Werkstofftechnik*, 36:171–176, 2005.

[128] C. Runge. Ueber die numerische Auflösung von Differentialgleichungen. *Math. Ann.*, 46(2):167–178, 1895.

[129] M. Saadi. *Shape sensitivities for the failure probability of mechanical components*. PhD thesis, University of Wuppertal, 2019.

[130] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, nov 1989.

[131] B. Schandl, K. Klamroth, and M.M. Wiecek. Norm-based approximation in multicriteria programming. *Computers and Mathematics with Applications*, 44:925–942, 2002.

[132] S. Schmidt and V. Schulz. Pareto-curve continuation in multi-objective optimization. *Pacific Journal of Optimization*, 4(2):243–257, 2008.

[133] A. Schmitz. *Multifidelity-Optimierungsverfahren für Turbomaschinen*. PhD thesis, Ruhr-Universität Bochum, Deutschland, 2020.

[134] S. Schmitz. *A Local and Probabilistic Model for Low-Cycle Fatigue.: New Aspects of Structural Analysis*. Hartung-Gorre, 2014.

[135] S. Schmitz, T. Beck, R. Krause, G. Rollmann, T. Seibel, and Hanno Gottschalk. A probabilistic model for LCF. *Computational Materials Science*, 79:584–590, 2013.

[136] S. Schmitz, T. Seibel, H. Gottschalk, T. Beck, G. Rollmann, and Rolf Krause. Probabilistic analysis of the LCF crack initiation life for a turbine blade under thermo-mechanical loading. *Proc. Int. Conf LCF 7*, 2013.

[137] J. Schultes. *Multiobjective optimization of shapes using scalarization techniques*. PhD thesis, University of Wuppertal, 2022.

[138] V. Schulz. A Riemannian view on shape optimization. *Foundations of Computational Mathematics*, 14 (3):483–501, 2014.

[139] V. Schulz. Efficient PDE constrained shape optimization based on Steklov–Poincaré-type metrics. *SIAM Journal on Optimization*, 26 (4):2800–2819, 2016.

[140] O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In J. Branke, K. Deb, K. Miettinen, and R. E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[141] O. Schütze, K. Witting, S. Ober-Blöbaum, and M. Dellnitz. Set oriented methods for the numerical treatment of multiobjective optimization problems. In E. Tantar, A.-A. Tantar, P. Bouvry, P. Del Moral, P. Legrand, C.A. Coello Coello, and O. Schütze, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, number 447 in Studies in Computational Intelligence, Berlin, Heidelberg, 2013. Springer.

[142] J.F. Shackelford and W. Alexander, editors. *CRC Materials Science and Engineering Handbook*. CRC Press, Boca Raton, 4th edition, 2015.

[143] T. Simpson, V. Toropov, V. Balabanov, and F. Viana. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come - or not. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, jun 2008.

[144] K. Soetaert, T. Petzoldt, and R. W. Setzer. Solving differential equations in R: package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

[145] J. Sokolovski and J.-P. Zolesio. *Introduction to Shape Optimization - Shape Sensitivity Analysis*. Springer, Berlin Heidelberg, 1992.

[146] E. Spodarev, editor. *Stochastic Geometry, Spatial Statistics and Random Fields*. Springer Berlin Heidelberg, 2013.

[147] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[148] L.P. Swiler and G.D. Wyss. A user's guide to sandia's latin hypercube sampling software : Lhs unix library/standalone version. 7 2004.

[149] C. Toure, A. Auger, D. Brockhoff, and N. Hansen. On bi-objective convex-quadratic problems. In K. Deb, E. Goodman, C.A. Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, editors, *Evolutionary Multi-Criterion Optimization*, pages 3–14. Springer, New York, 2019.

[150] E. Vanmarcke. *Random Fields*. World Scientific, jul 2010.

[151] S. Watanabe. On discontinuous additive functionals and Lévy measures of a Markov process. *Japan J. Math.*, 34, 1964.

[152] E. Weibull. A statistical theory of the strength of materials. *Ingeniörsvetenskapsakedemiens Handlingar*, 151:1–45, 1939.

[153] S. Willeke and T. Verstraete. Adjoint optimization of an internal cooling channel u-bend. 2015.

[154] G.R. Zavala, A.J. Nebro, F. Luna, and C.A. Coello. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, 49:537–558, 2014.

[155] A. Zerbinati, J.A. Desideri, and R. Duvigneau. Comparison between MGDA and PAES for multi-objective optimization. Research Report RR-7667, INRIA, June 2011.

[156] A. Zerbinati, A. Minelli, I. Ghalane, and J.A. Désidéri. Meta-model-assisted MGDA for multi-objective functional optimization. *Computers & Fluids*, 102:116–130, 2014.

[157] R. Zimmermann. On the maximum likelihood training of gradient-enhanced spatial gaussian processes. *SIAM Journal on Scientific Computing*, 35, 01 2013.

[158] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.*, 8(2):173–195, June 2000.

[159] J.-P. Zolesio. *Identification de domaines par déformations*. PhD thesis, Université de Nice-Sophia Antipolis, 1979.