Technische Universität Ilmenau
Fakultät für Mathematik und Naturwissenschaften
Arbeitsgruppe Mathematische Methoden des
Operations Research

Efficient numerical solution of chance constrained optimization problems with engineering applications

Dissertation zur Erlangung des akademischen Grades Dr. rer. nat.

Michael Klöppel

betreut von
Prof. Dr. rer. nat. habil. Armin Hoffmann

Dresden, 30.05.2014

urn:nbn:de:gbv:ilm1-2014000087

Zusammenfassung

In der Praxis werden viele Prozesse durch Unsicherheiten beeinflusst. Die Auswirkungen dieser Unsicherheiten können dabei beträchtlich sein. Es ist daher sinnvoll diese Einflüsse bei der Prozessoptimierung zu betrachten. Ein Ansatz dazu ist die Nutzung der wahrscheinlichkeitsrestringierten Optimierung. Diese erfordert die Einhaltung der Nebenbedingungen nur mit einer gewissen Wahrscheinlichkeit und erlaubt damit einen Kompromiss zwischen Profit und Zuverlässigkeit.

In Abhängigkeit des unterliegenden Prozesses sind mehrere Ansätze zur Umwandlung der Wahrscheinlichkeitsrestriktionen in deterministische Restriktionen möglich. Die meisten dieser Ansätze basieren auf der Berechnung hochdimensionaler Integrale. In dieser Arbeit werden entsprechende Methoden zur Berechnung solcher Integrale vorgestellt. Hauptaugenmerk liegt dabei immer auf einer möglichst effizienten numerischen Implementation. Hauptbestandteil der Arbeit ist dabei die Beschreibung von so genannten analytischen Approximationen, welche effizient für eine Vielzahl von Anwendungen eingesetzt werden können. Für diese Verfahren werden Methoden zur Berechnung der Gradienten entwickelt. Eine weitere Verringerung der Rechenzeit wird durch die effiziente Approximierung der unterliegenden Modellgleichungen erreicht.

In Fallstudien aus dem Ingenieurbereich werden die analytischen Approximationen mit anderen Ansätzen verglichen. Dabei stellt sich heraus, dass diese Methoden als genereller Ansatz benutzt werden können, auch wenn andere Methoden zu leicht besseren Ergebnissen führen. Als größere Fallstudie wird eine Problem aus dem Bereich des optimalen Lastflusses gelöst. Hier zeigt sich, dass die vorgeschlagenen Ansätze bessere Ergebnisse liefern als die weithin benutzte Approximation mit normalverteilten Zufallsgrößen. Außerdem kann durch den Einsatz effizienter Methoden selbst dieses größere Beispiel in vernünftiger Rechenzeit gelöst werden.

Abstract

Many practical processes are influenced by uncertainties, which might have a large impact. Therefore, these uncertainties should be considered when optimizing such process. One approach of incorporating uncertain influences is the usage of chance constrained optimization. This approach requires that the constraints are only held with a certain probability level, thereby allowing a compromised decision between reliability and profitability.

Depending on the underlying process, several approaches to transform the chance constraints into deterministic constraints exist. Most of these approaches are based on high-dimensional integrals. In this work, corresponding methods for the evaluation of such integrals will be introduced. In doing so, the focus is always on efficient numerical implementations. An essential part of this thesis is the characterization of so called analytical approximations, which can be efficiently used for a large class of applications. For these approaches, methods to evaluate gradients are described. A further reduction of the computation time can be achieved through an efficient approximation of the underlying model equations.

In the case studies, the analytical approximations are compared with several other approaches. One result is that analytical approximations can act as general purpose approaches, although other methods lead to slightly better optimization results. The largest case study deals with a problem from the area of optimal power flow. Here, it can be shown that the results obtained by the proposed approach is better than the results obtained through the usage of the widely employed Gaussian approximation. Furthermore, by using efficient methods even larger scale case studies can be solved in reasonable computation time.

Nomenclature

$\det(A)$	the determinant of a matrix $A \in \mathbb{R}^{n\times n}$
$\frac{\partial F}{\partial y}$, F_y	partial derivative of F w.r.t. y
$\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}_{++}$	set of real numbers, $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$, $\mathbb{R}_{++} = \{x \in \mathbb{R} \mid x > 0\}$
$\mathbb{N}$	set of natural numbers
$\mathrm{cl}A$	closure of a set $A \subset \mathbb{R}^n$
$C^k(A, \mathbb{R}^m)$	the set of k-times continuously differentiable functions $f : A \to \mathbb{R}^m$, $k \in \mathbb{N} \cup \{\infty\}$, $A \subseteq \mathbb{R}^n$ an open set
$E\left[\cdot\right]$	expectation induced by the corresponding uncertain variables
I_n	$= \mathrm{diag}\left\{1, \ldots, 1\right\} \in \mathbb{R}^{n\times n}$, or I if the dimension is clear from the context
$L^2\left[-1,1\right]^d$	set of all square-integrable functions on the hypercube $\left[-1,1\right]^d$, $d \geq 1$
$Pr\left\{\cdot\right\}$	probability measure induced by the corresponding uncertain variables
$Var\left[\cdot\right]$	variance induced by the corresponding uncertain variables

List of acronyms

AA	Analytical Approximation
AFA	AFine Arithmetic
ANN	Artificial Neural Network
CC	Clenshaw-Curtis
CCOPF	Chance Constrained Optimal Power Flow
CCOPT	Chance Constrained OPTimization
cdf	cumulative distribution function

DS	Distribution System
IA	Interval Arithmetic
MPC	Model Predictive Control
NLP	NonLinear Programming
MDO	Multidisciplinary Design Optimization
OPF	Optimal Power Flow
pdf	probability density function
(Q)MC	(Quasi-)Monte-Carlo
RBDO	Reliability Based Design Optimization
SAA	Sample Average Approximation

Acknowledgments

I would like to thank the Carl-Zeiss-Stiftung for the financial support, my colleagues in the department of Simulation and Optimal Processes as well as in the Institute of Mathematics for their helpful suggestions and fruitful discussions, and, last but not least, my family and friends for their moral support.
The layout of this thesis is based on BergerStyle 2.0 by Thomas Berger.

Contents

1 Introduction

Uncertainties are an inherent property of nearly all practical processes, in engineering as well as in finance. They have a considerable impact on the process in many situations. Therefore, optimization of such processes must be considered under these uncertainties. Several approaches have been proposed to carry out this task. These include (approximated) robust counterparts [10,23], approximate polyhedral dynamic programming [11], nominal solutions [82], measurement-based optimization [81], extended Kalman filter based nonlinear model predictive control approaches [61], worst-case and distributional robustness analysis [62,63], recourse programming [69] and chance constrained optimization [16,17]. In this thesis the focus is on the CCOPT approach.

The main idea of CCOPT is to require the satisfaction of process restrictions with a predefined probability level. An advantage of CCOPT is that a relationship between the optimality, in the sense of achieving an optimal objective function value, and the reliability, in the sense of satisfying process restrictions, can be obtained. Based on this relationship a compromised decision which balances the profitability and reliability can be made.

The last two decades have shown a rising interest in the numerical solution of CCOPT problems. This is partly due to the increased capability of computer hardware as well as software and the availability of efficient computational approaches to large-scale deterministic optimization problems. Recently the demand for high reliability, fault tolerance and risk minimization in structure design [73, 74] and financial risk metrics [70] has led to wider areas of applications of CCOPT.

There exist several available approaches to the solution of CCOPT problems. Until recently, most of the models considered were linear. Analytical solutions can be obtained for linear systems with single chance constraints [78]. A linear CCOPT problem with a joint chance constraint is inherently nonlinear and the probability evaluation can be made by the inclusion-exclusion method proposed by Szantai and Prekopa [69]. Based on this method chance constrained model predictive control was studied [41, 56]. Cannon et al. [14] analyzed a chance constrained control approach for linear time-dependent systems with certain quadratic objective functions.

The solution of nonlinear CCOPT problems is usually carried out in a NonLinear Programming (NLP) framework. To do this, the values and gradients of the objective function and probability constraints need to be computed. For nonlinear systems a direct evaluation of probabilities of holding output restrictions is not promising, since the probability distribution of outputs can hardly be explicitly described due to the propagation through nonlinear model equations. Monte-Carlo sampling was used for computing average sums of function values to approximate chance constraints [13, 40]. Despite the fact that this approach is applicable irrespective of the type of the distribution function of uncertain variables, it requires a very large sample size to yield accurate estimations for the probability values. Importance sampling may provide some improvements over Monte-Carlo sampling. For high dimensional systems, however, it is well-known that sampling methods are not efficient for evaluating gradient values.

Based on a monotonic relation between the constrained output and an uncertain input, a projection approach was proposed by Wendt et al. [89] to evaluate the probability of the output constraint satisfaction in the space of the uncertain inputs. Collocation on finite elements (a so called full-grid method) was used for the numerical multivariate integration. This method was further studied by Arellano-Garcia and Wozny [6] for monotonicity analysis and Flemming et al. [27] for optimization of closed-loop systems under uncertainty as well as Xie et al. [91] for nonlinear model predictive control. The back-mapping approach needs intensive computation due to its demand for repeatedly solving the nonlinear model equations at the grid points. For CCOPT of large-scale systems, these full grid integrations will be prohibitive, in particular when an online implementation is needed.

Recently, Nemirovski and Shapiro [64] and Geletu et al. [34] proposed the usage of so called analytical approximations to extend the solution of CCOPT problems to instances, where a monotonic relation cannot be found or simply does not exist. Whereas the approach of Nemirovski and Shapiro [64] pertains convexity properties of the constraints, the approach of Geletu et al. [34] generally results in much tighter approximations. Similar to the projection approach, analytical approximations require the evaluation of high dimensional integrals, and again the usage of full grid integration will be prohibitive.

As a consequence, one of the main challenges in the efficient solution of large-scale nonlinear CCOPT problems is finding adequate high dimensional integration rules. High dimensional integration approaches can be generally constructed in two different ways.

On the one hand, there are methods based on one-dimensional underlying rules, namely full grid and sparse grid methods. Full grid methods are either a tensor-product of one-dimensional quadrature rules or recursive dimension-wise integration techniques. Such techniques are known to be ineffective for integrals of high dimensions [60]. In contrast, sparse-grid approaches, based on fully-symmetric integration formulas, need very few integration nodes [37]. They are found to provide an efficient evaluation of high dimensional integrations by reducing computation time significantly. Sparse-grids were first proposed by Smolyak in 1963 [80] and have been recently applied to many fields of numerical computation such as stochastic partial differential equations [49, 65], micro-electromechanical systems [3], data mining [30] as well as quantum mechanics [92].

On the other hand, sampling approaches, i.e., (Quasi-)Monte-Carlo ((Q)MC) methods [12, 54], are widely employed for the evaluation of high dimensional integrals. Their main advantage is their wide field of application, since corresponding integration rules can be constructed for a large class of underlying weight functions, and the good convergence in the presence of discontinuities.

From the above it is clear that the solution of CCOPT problems is computationally very demanding, especially in the presence of a higher number of uncertainties. Therefore, the aim of this thesis is to investigate efficient numerical approaches to the solution of such problems. Several methods for treating chance constraints will be introduced and the construction of multivariate integration rules will be discussed. Moreover, the usage of approximation methods can further reduce the computational burden.

The remainder of this thesis is organized as follows. Chapter 2 contains the basic mathematical notions. Chapter 3 introduces the chance constraint optimization problem and gives a basic classification. Chapter 4 presents methods of transforming the probabilistic constraints in deterministic ones. Chapter 5 discusses integration methods in connection with CCOPT. Chapter 6 mainly introduces Newton's method and presents a new idea involving approximation methods. In Chapter 7 several smaller numerical examples and one larger case study from the area of energy network optimization are carried out. A conclusion and an outlook to further work can be found in Chapter 8.

1.1 Related approaches

Here, we will shortly review two additional approaches to optimization under uncertainty, which can be applied in settings slightly different from that of CCOPT.

1.1.1 Robust optimization

Robust optimization [9] deals with the case that no distribution of the uncertainties is known. Instead, uncertainties are contained inside a known compact set. Formally, this leads to the following problem formulation

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & f(u) \\ s.t. \quad & g(u,\xi) \leq 0, \ \forall \xi \in \Omega, \end{aligned}$$

where $f : \mathbb{R}^m \to \mathbb{R}$, $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^l$, $U \subset \mathbb{R}^m$, and $\Omega \subset \mathbb{R}^p$. The difficulty here is that the constraints have to be fulfilled for all realization $\xi \in \Omega$, leading to a semi-infinite optimization problem.

1.1.2 Recourse stochastic programming

Recourse programming [69] can be applied under circumstances similar to that of CCOPT, but requires a certain structure of the optimization problem. Here, we consider only the standard case of two stage optimization. One possible problem formulation is

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & f(u) + E\left[Q(u,\xi)\right] \\ s.t. \quad & g(u) \leq 0 \\ Q(u,\xi) = \min_{t \in \mathcal{T}} \quad & q(t,u,\xi) \\ s.t. \quad & h(t,u,\xi) \leq 0, \end{aligned}$$

where f, ξ, and $\mathcal{U}$ are like above, $g : \mathbb{R}^m \to \mathbb{R}^l$ are the constraints of the first stage, $\mathcal{T} \subset \mathbb{R}^{m_2}$, $Q : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is the optimal solution of the second optimization problem, $q : \mathbb{R}^{m_2} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is the objective function of the second problem and $h : \mathbb{R}^{m_2} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^{l_2}$ are the constraints of the second stage. The operator $E[\cdot]$ is the expectation with regard to the underlying uncertainty ξ. The idea behind the recourse approach is to take a deterministic decision in the first stage and than have a look how the realizations of the uncertainties affect this outcome (second stage). This becomes clear from the well known Newsboy problem [69], p. 252. A newsboy can order an amount $u \in \mathbb{N}$ news papers in the evening, which he is going to sell on the next day. If he orders to much newspapers, he will incur a loss for every newspaper not sold. On the other hand, if he orders to little, he decreases his profit, since more newspapers could have been sold. Given a probability distribution for the amounts of papers sold, recourse programming can now be used to find an optimal solution.

1.2 Contributions of the author

Most of the novel results presented in this thesis were derived by a research group, consisting of Prof. Pu Li, Prof. Armin Hoffmann, Dr. Abebe Geletu, Aouss Gabash, Hui Zhang and the author. The purpose of this section is to clarify the contributions of the author within the research group.

Projection approaches, Chapter 4: The author co-authored a paper [33] on the projection approach. The main contribution was the implementation of the case study.

Analytic approximation, Chapter 4: The author co-authored a paper [34] introducing the analytical approximation approach. The contributions are the evaluation of gradients in the setting of analytical approximations and the implementation of case studies. Moreover, in this thesis the proofs contained in [34] are fleshed out and expanded to higher order derivatives.

Kronrod-Patterson integration, Chapter 5: The experiments on constructing Kronrod-Patterson extensions were carried out solely by the author.

Approximation methods, Chapter 6: The idea of using approximation methods in conjunction with the analytical approximation approach is proposed by the author.

Case studies, Chapter 7: All case studies were conducted by the author. A part of the case studies were previously published in [35, 50].

2 Prerequisites

The main purpose of this chapter is to summarize important mathematical results, which will be required later on, and to act as a reference for the reader. Therefore, it is possible to skip this chapter, especially if one is familiar with the mathematical notions. Since all the result can be found in standard text books no proofs are given here. Instead, references are given.

2.1 Implicit function theorems

In engineering, model equations are either given by or can be discretized to yield a non-linear set of equations in the form $F(x, y) = 0$, where $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$, x are control (input) variables, and y denotes state variables. Implicit function theorems answer the question, if there exists a function $f : \mathbb{R}^m \to \mathbb{R}^n$ mapping the control variables x to the state variables y. More clearly, given a function F and a point $(x_0, y_0) \in \mathbb{R}^m \times \mathbb{R}^n$ with $F(x_0, y_0) = 0$, implicit functions theorems give conditions under which there exists a representation $y = f(x)$ with $F(x, f(x)) = 0$ either locally in a neighborhood of (x_0, y_0) or even globally. Moreover, under additional assumptions, further properties of the function f, e.g., f being bijective, can be assured. The first theorem stated below contains the standard (local) result.

Notation 2.1.1. By $C^k(A, \mathbb{R}^m)$, $0 \leq k \leq \infty$, $A \subset \mathbb{R}^n$ an open set, $n, m \in \mathbb{N}$ we denote the set of all functions $f : A \to \mathbb{R}^m$ which have continuous (partial) derivatives of order up to and including k. By $C_0^k(A, \mathbb{R}^m)$, $0 \leq k \leq \infty$ we denote the set of all functions $f \in C^k(A, \mathbb{R}^m)$ which have a compact support. Whenever A or the dimension m follow from the context, the arguments will be dropped for brevity.

Theorem 2.1.2 (Implicit function theorem, [52] p. 43)**.** *Let $F(x, y)$, $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$ be a mapping of class C^k, $k \geq 1$ defined on an open set $U \subset \mathbb{R}^m \times \mathbb{R}^n$ and taking values in $\mathbb{R}^n$.*

Let (x_0, y_0) be a point of U with $F(x_0, y_0) = 0$. Of course we let (x, y) be any point of $U \times V$. We suppose that

$$\det \left(\frac{\partial F}{\partial y}(x_0, y_0) \right) \neq 0.$$

Then there exists a neighborhood $\tilde{U}$ of (x^0, y^0), and an open set $W \subset \mathbb{R}^m$ containing x_0, and function $f : \mathbb{R}^m \to \mathbb{R}^n$ of class C^k on W such that

$$F(x, f(x)) = 0 \textit{ for every } x \in W.$$

Furthermore f is the unique function satisfying

$$\left\{(x,y) \in \tilde{U} : F(x,y) = 0\right\} = \left\{(x,y) \in \tilde{U} : x \in W, y = f(x), l = 1, \dots, M\right\}.$$

One consequence of the above result is that the implicitly defined function f is at least one time continuously differentiable. The derivate can be obtained by

$$\frac{d}{dx} f(x) = -\left(\frac{\partial F}{\partial y}\right)^{-1} \frac{\partial}{\partial x} F(x,y)$$

for $(x,y) \in U$ with $F(x,y) = 0$. Furthermore, if one assumes that $m \geq n$ and $\frac{\partial F}{\partial x}(x_0, y_0)$ has full rank, then by standard analysis f is injective on an open set $W_0 \subset W$. When restricting f to the open set $f(W_0)$, the function f can be made bijective.

As a next step, one possible formulation of a global implicit function is given.

Theorem 2.1.3 (Global implicit function theorem, [75])**.** *Assume U is an open convex subset of $\mathbb{R}^m$ and V is an open subset of $\mathbb{R}^n$, in which n and m are positive integers such that $m \geq n$. Assume also that $F : U \times V \to \mathbb{R}^n$ is continuously differentiable on $U \times V$ and that the rank of the $n \times m$ matrix $\frac{\partial F}{\partial x}$ is n for $(x,y) \in U \times V$ with $F(x,y) = 0$. Let A be any family of compact subsets of V such that for each compact subset C of V, there is an $S \in A$ such that $C \subset S$, and, similarly, let B denote any collection of compact subsets of U with the property that for any compact subset D in u, there is $T \in B$ such that $D \subset T$.*

Under these conditions there is a unique $f : U \to V$ such that $F(x, f(x)) = 0$ for all $x \in U$, and f is continuously differentiable on V, if and only if

(i) *for some $x_0 \in U$, there is exactly one $y_0 \in V$ such that $F(x_0, y_0) = 0$,*

(ii) $\det \frac{\partial F}{\partial y} \neq 0$ *for $(x,y) \in U \times V$ with $F(x,y) = 0$,*

(iii) *for each $T \in B$, there is an $S \in A$ such that $x \in T$, $y \in V$, and $F(x,y) = 0$ imply that $y \in S$.*

2.2 Partition of the unity

Let $G \subset \mathbb{R}^n$, $n \in \mathbb{N}$ be an open set. Furthermore, assume there exists an open covering $\{G_i \mid i \in I\}$ of the set G, where I is an arbitrary index set. The partition of the unity now allows to partition the unity 1 on the set G using non-negative functions $\alpha_j(x)$, whose support is contained in some G_i. A formal statement of the theorem is given below after some introductory definitions.

Definition 2.2.1 (Topological space, [95] p. 9)**.** A set X is said to be a topological space if a system τ of subsets of X is exhibited (called open sets in X) possessing the following properties:

(i) $\emptyset \in \tau$, $X \in \tau$

(ii) $\tau_i \in \tau$, $i \in I \Rightarrow \bigcup_{i\in I} \tau_i \in \tau$

(iii) $\tau_i \in \tau$, $i \in \{1, \dots, n\} \Rightarrow \bigcap_{i=1}^{n} \tau_i \in \tau$

Definition 2.2.2 (Hausdorff, [95] p. 12)**.** A topological space is Hausdorff if the Hausdorff axiom holds in it: any two distinct points of the space have non-intersecting neighborhoods.

Definition 2.2.3 (σ-compact, [44] p. 339)**.** A topological space X is σ-compact if it is Hausdorff and is a countable union of compact sets.

Definition 2.2.4 (Locally finite cover, [44] p. 340)**.** An open cover $\mathcal{W} := (W_i)_{i\in I}$ of a topological space X is locally finite if every point $x \in X$ has a neighborhood that intersects only finitely many of the W_i.

Definition 2.2.5 (Partition of unity subordinate to an open cover, [32] p. 72)**.** For a topological space X, a set of nonnegative continuous functions $\{\alpha_j\}_{j\in J}$, $\alpha_j : X \to \mathbb{R}$, $j \in J$, is a partition of unity subordinate to an open cover $\{\mathcal{V}_j\}_{j\in J}$ of X if, and only if,

- each α_j is not identically zero,
- $\alpha_j(x) = 0$ if $x \notin \mathcal{V}_j$,
- for each x in X, only finitely many $\alpha_j(x)$ are different from zero, and
- $\sum_{j\in J} \alpha_j(x) \equiv 1$.

Theorem 2.2.6 (Partition of unity, [44] p. 341)**.** *Let X be a second countable finite-dimensional topological space, and let $\mathcal{W}$ be an open cover of X. Then there exists a locally finite refinement $\mathcal{V}$ of $\mathcal{W}$, and a continuous partition of unity subordinate to $\mathcal{V}$. Moreover, if X is C^∞ manifold, the partition of unity can be chosen C^∞.*

Remark 2.2.7. In this work the partition of the unity is applied to subsets $G \subset \mathbb{R}^n$, which are second countable finite-dimensional topological C^∞ spaces.

Given a function $f : G \to \mathbb{R}^m$ of class C^k, $m \in \mathbb{N}$, and using partition of the unity it holds that $f(x) = \sum_{j\in J} \alpha_j(x) f(x)$. Furthermore, the functions $\alpha_j(x)f(x)$ are of class C^k and their support is contained in some G_i.

2.3 Interval and affine arithmetic

Interval Arithmetic (IA) and AFine Arithmetic (AFA) are both generalizations of the standard arithmetic to interval inputs. Both are also methods of self-validated numerical computing, i.e., they can account for round off and truncations errors which occur in numerical calculations when working with floating point numbers. Not accounting for such errors may lead to inaccurate or imprecise results, for an example see [21]. The main application for these methods in this work is to check whether a given interval I does not contain a root of some given function f.

Generally, IA and AFA are approximations, which result in supersets of the actual result, i.e, given a function f it holds for any interval input I that $f(I) \subset f_{IA}(I)$ or $f(I) \subset f_{AFA}(I)$, respectively. Here, f_{IA} and f_{AFA} are the representations of the function f in the respective arithmetic. Due to the nature of the methods, whenever $0 \notin f_{IA/AFA}(I)$ then f contains no root in I. However, $0 \in f_{IA/AFA}(I)$ does not imply that f has a root in I, since both methods overestimate $f(I)$,

Remark 2.3.1. For practical application both methods need a careful choice of rounding mode. Since a discussion of these modes is out of the scope of this work, any influences of rounding are neglected during the description of these methods.

2.3.1 Interval arithmetic

IA was introduced by Moore in 1966 [58]. It is based on the usage of closed intervals $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$, where $a, b \in \mathbb{R} \cup \{-\infty, \infty\}$, $a \leq b$. Scalar values are represented by intervals with the same upper and lower bound, e.g., $3 = [3, 3]$. Let $I_j = [a_j, b_j]$, $a_j, b_j \in \mathbb{R}$, $j = 1, 2$. Then the standard arithmetic operations are defined by

$$\begin{aligned} I_1 + I_2 &= [a_1 + a_2, b_1 + b_2]\,, \\ I_1 - I_2 &= [a_1 - b_2, b_1 - a_2]\,, \\ I_1 \times I_2 &= [\min(a_1a_2, a_1b_2, b_1a_2, b_1b_2), \max(a_1a_2, a_1b_2, b_1a_2, b_1b_2)]\,, \\ I_1/I_2 &= \begin{cases} [a_1/b_2, b_1/a_2] & a_1, a_2 > 0, \\ [a_1/a_2, b_1/b_2] & b_1 < 0,\ a_2 > 0, \\ [a_1/a_2, b_1/a_2] & a_1 < 0,\ b_1 > 0, a_2 > 0, \\ [b_1/b_2, a_1/a_2] & a_1 > 0, b_2 < 0, \\ [b_1/a_2, a_1/b_2] & b_1 < 0,\ b_2 < 0, \\ [b_1/b_2, a_1/b_2] & a_1 < 0,\ b_1 > 0, b_2 < 0, \\ [-\infty, \infty] & \text{otherwise.} \end{cases} \end{aligned}$$

The generalization of elementary functions $f : \mathbb{R} \to \mathbb{R}$ is done depending on whether the given function is monotonic or not. The monotonic case is straightforward, i.e., $f([a, b]) = [f(a), f(b)]$ for f monotonically increasing and $f([a, b] = [f(b), f(a)]$ if f is monotonically decreasing. For non-monotonic f it is necessary to check whether f has a minimum/maximum in the input interval I. The output interval then has to be set accordingly, e.g.,

$$\begin{aligned} f : \mathbb{R} \to \mathbb{R}, x \mapsto 1 - x^2, \\ f([-1, 1]) = [0, 1] = [f(-1), f(0)];\ f \text{ contains a maximum in } [-1, 1], \\ f([1, 2]) = [-3, 0] = [f(2), f(1)];\ f \text{ is monotonic in } [1, 2]. \end{aligned}$$

Please note that in IA $I - I \neq 0$ and $I/I \neq 1$. This is due to the fact that every occurrence of an interval I is treated independently. Such behavior is desired when analyzing for example the influence on round-off errors on calculations (one of the first applications of IA methods), but leads to large overestimation when checking for roots, especially for complicated expressions. To overcome this problem Stolfi and Figueiredo proposed the usage of AFA.

2.3.2 Affine arithmetic

In the AFA approach [21] intervals are expressed using an affine expression of the form

$$\hat{x} = x_0 + x_1\varepsilon_1 + \ldots + x_n\varepsilon_n,$$

where $x_i \in \mathbb{R}$, $i = 0, \ldots, n$. The quantities $\varepsilon_i \in [-1, 1]$ are called noise symbols, the quantity $r_x = \sum_{i=1}^{n} |x_i|$ is called the total deviation, and $\hat{x}$ corresponds to the interval $[x_0 - r_x, x_0 + r_x]$. On the other hand, every interval $[a, b]$ can be expressed as

$$\frac{a+b}{2} + \frac{b-a}{2}\varepsilon_1.$$

Addition and subtraction in AFA are straightforward, since both are affine linear functions, i.e., if $\hat{x} = x_0 + x_1\varepsilon_1 + \ldots + x_n\varepsilon_n$ and $\hat{y} = y_0 + y_1\varepsilon_1 + \ldots + y_n\varepsilon_n$ then

$$\hat{x} \pm \hat{y} = (x_0 \pm y_0) + (x_1 \pm y_1)\varepsilon_1 + \ldots + (x_n \pm y_n)\varepsilon_n.$$

One important property of AFA is that noise symbols can be shared between different expressions (just like $\hat{x}$ and $\hat{y}$ above). This allows for much tighter bounds in contrast to IA. For transformations, which are not affine linear, an affine linear approximation has to be constructed. For standard transformations (e.g., trigonometric functions, multiplication, division) this was already done and approximations are readily available in computer implementations.

2.4 Probability distributions

The main purpose of this section is to give a short introduction on the probability distributions used in this thesis. The idea here is to mention only the most important facts for reference, since more detailed information can be found in various text books, e.g., in [47].

2.4.1 Gaussian distribution

The Gaussian or normal distribution is one of the most commonly found distributions. Its univariate probability density function (pdf) is defined by

$$\varphi(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\xi-\mu)^2}{2\sigma^2}},$$

where μ is the expectation and σ is the standard deviation. There exists no closed form for the cumulative distribution function (cdf). Every univariate Gaussian distributed uncertain variable Y can be transformed to a standard Gaussian distributed uncertain variable by applying the transformation $\frac{Y-\mu}{\sigma}$.

Similar properties hold also true for multivariate Gaussian distributed uncertainties, which pdf is given by

$$\varphi(\xi) = \frac{1}{\sqrt{2\pi}^n \det \Sigma} e^{-\frac{(\xi-\mu)^T \Sigma^{-1} (\xi-\mu)}{2}},$$

where $\xi = (\xi_1, \ldots, \xi_n)$, μ is the vector of expectation and Σ is the covariance matrix. A transformation to the standard case can be achieved by $L(Y - \mu)$, where Y is a vector of jointly Gaussian distributed uncertain variables and L is such that $LL^T = \Sigma$. The matrix L can be obtained from Σ by means of a Cholesky decomposition.

Two additional properties of Gaussian distributed uncertainties are worth mentioning. First, all marginal distributions of a jointly Gaussian distributed uncertain vector are also Gaussian distributed. Second, a linear transformation of a jointly Gaussian distributed uncertain vector always leads to a Gaussian distributed result. This is the reason, why the Gaussian distribution is so widely applied in the context of CCOPT, since this allows to directly compute chance constraints (see Chapter 4).

2.4.2 Beta distribution

The Beta distribution is a large family of rather different probability distributions, as can be seen in Figure 2.1. Depending on the choice of the distribution parameters α and β, the *pdf* may be bounded or unbounded as well as unimodal or bimodal. One special case of the Beta distribution is the uniform distribution, which appears for $\alpha = \beta = 1$. The pdf of a Beta distributed uncertain variable is given by

$$\phi(\xi) = \frac{\xi^{\alpha-1}(1-\xi)^{\beta-1}}{\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx},$$

where $0 \leq \xi \leq 1$, and $\alpha, \beta \in \mathbb{R}_{++}$. If the uncertain variable Y underlies a Beta distribution with parameters α, β, then

$$\mathrm{E}[Y] = \frac{\alpha}{\alpha + \beta}$$

and

$$\mathrm{Var}[Y] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

Higher order moments can also be directly obtained, which is quite useful when constructing Gaussian quadrature rules (see Chapter 5). They are given by

$$\mathrm{E}[Y^k] = \prod_{i=0}^{k-1} \frac{\alpha + i}{\alpha + \beta + i}.$$

The variance of a Beta distributed uncertain variable is always smaller than $\frac{1}{4}$, which is one reason why there exists no standard form of the Beta distribution.

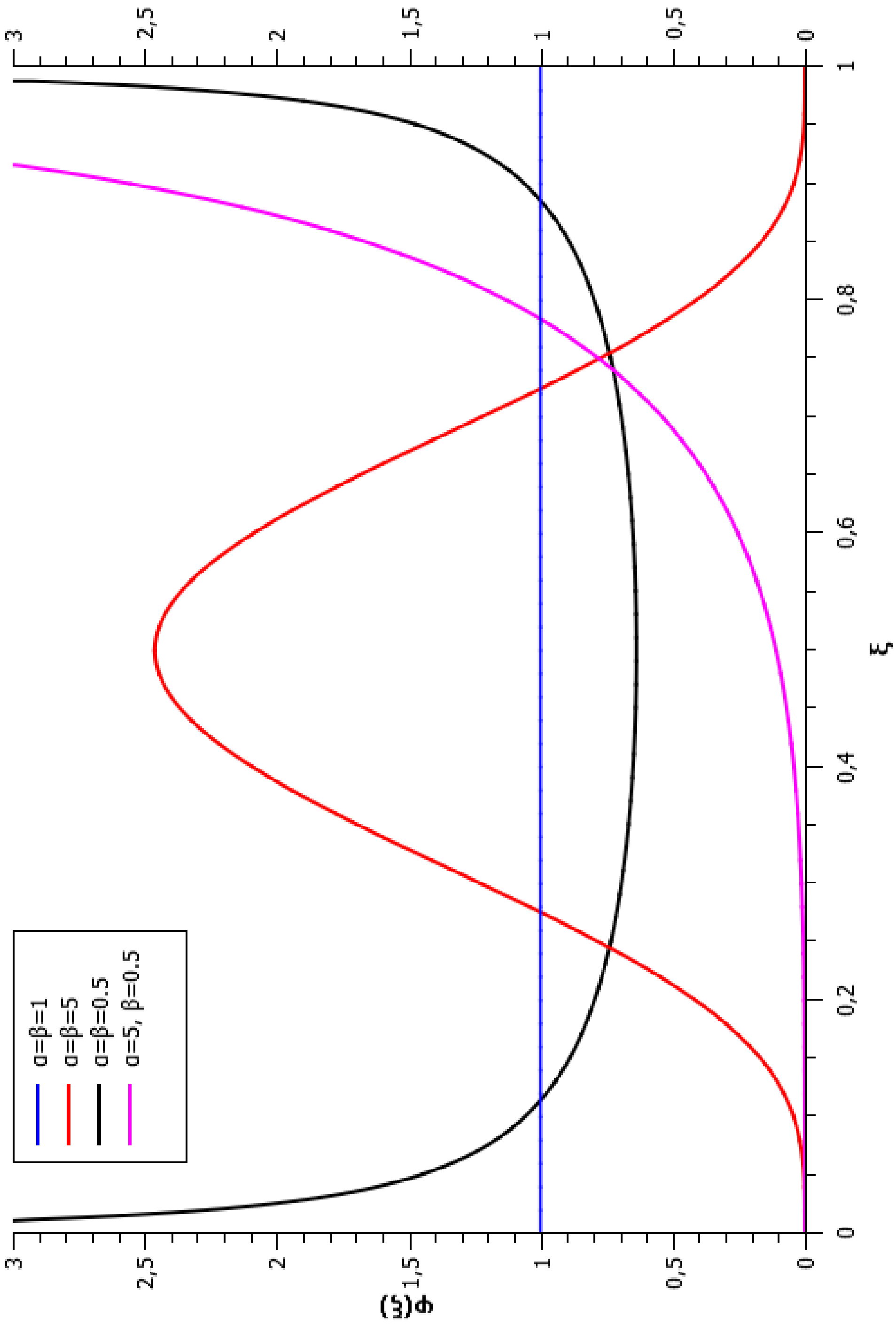

Figure 2.1: *Beta pdf for different distribution parameters* α *and* β

3 Chance Constrained Optimization

In this chapter, the solution process of typical CCOPT problems occurring in process engineering will be discussed. As a first step a classification of CCOPT is given. This is necessary, since different problems may require different solution approaches. Furthermore, some problems can be treated by more than one method. One should keep in mind that the classification given below is not complete, i.e., only problems, which can be solved with the methods presented in this thesis, are listed.

To clarify, which kind of problems can be solved within the scope of this thesis and which cannot, consider a general stochastic optimization problem with single chance constraints. It takes the form

$$\text{(NLP)} \min_{u \in \mathcal{U}} \quad E\left[f(y,u,\xi)\right] \tag{3.0.1}$$

$$s.t. \quad F(y,u,\xi) = 0 \tag{3.0.2}$$

$$Pr\left\{g_i(y,u,\xi) \leq 0\right\} \geq \alpha_i, \quad i = 1,\dots,q, \tag{3.0.3}$$

where $y \in \mathbb{R}^n$ is a vector of output variables, $u \in \mathcal{U} \subset \mathbb{R}^m$ is a vector of control variables, and ξ is a vector of Borel-measurable uncertain input variables from a probability space into $\mathbb{R}^p$. The function $f : \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}^p \to \mathbb{R}$ is the objective function, $F : \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}^p \to \mathbb{R}^n$ describes the model equations, and $g_i : \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}^p \to \mathbb{R}$, $i = 1,\dots,q$ are the constrained quantities. The chance constraints (3.0.3) are to be held with a probability level $\alpha_i \in [0.5, 1]$[1], $i = 1,\dots,q$. Furthermore, the following assumptions are imposed.

Assumption 3.0.1. The functions $f(\cdot,\cdot,\cdot)$, $F(\cdot,\cdot,\cdot)$, and $g_i(\cdot,\cdot,\cdot)$, $i = 1,\dots,q$ are twice continuously differentiable.

Assumption 3.0.2. The vector of uncertain inputs ξ has a known continuously differentiable pdf with support in a Borel-measurable set $\Omega \subset \mathbb{R}^p$.

Assumption 3.0.3. The derivatives $\nabla_\xi g_i(u,y,\xi)$ are non-zero a.e. for $u \in \mathcal{U}$, $\xi \in \Omega$, $y \in \mathbb{R}^n$, and $i = 1,\dots,q$.

Assumption 3.0.4. For every $u \in \mathcal{U}$ and every $\xi \in \Omega$ there exists a unique $y \in \mathbb{R}^n$ with $F(y,u,\xi) = 0$.

[1] Theoretically, $\alpha_i \in (0,1]$ would be possible, although, for $\alpha_i < 0.5$ the probability of violating the constraint might be actually larger then the probability of holding it. This is not useful from the process engineering point of view.

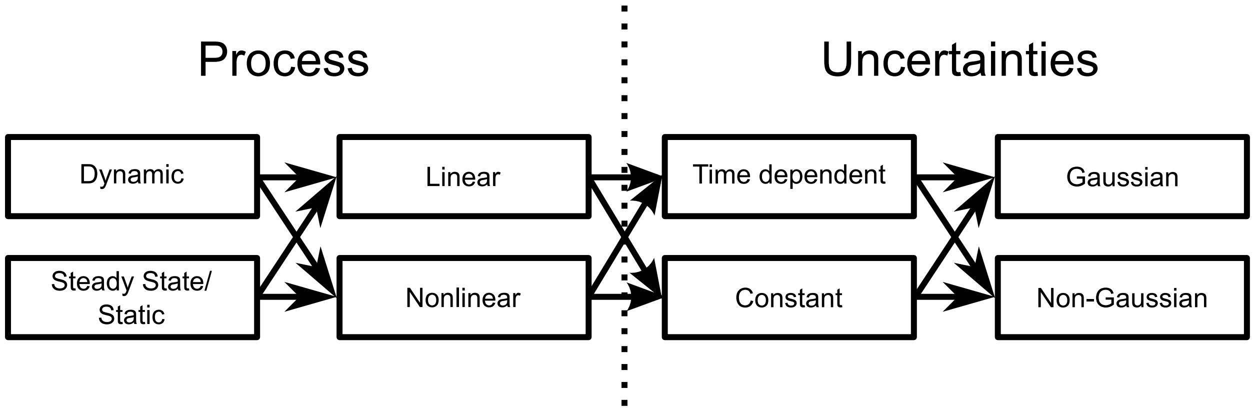

Figure 3.1: *Classification of CCOPT problems. All possible combinations may occur, e.g., static problems with nonlinear process model and time dependent, non-Gaussian distributed uncertainties.*

Notation 3.0.5. A solution y of the model equations $F(u, y, \xi) = 0$ for given $u \in \mathcal{U}$ and $\xi \in \Omega$ is denoted by $y(u, \xi)$.

The first assumption allows to use standard algorithms in the solution process, e.g., SQP- or interior point methods for optimization and Newton's method for solving the model equations. Among others, this excludes problems containing integer variables. The second assumption assures that the integration task necessary to evaluate probabilities, expectations, etc. can be carried out using standard methods, e.g., (Quasi-)-Monte-Carlo integration. Furthermore, uncertainties with discrete distributions and generalizations of CCOPT, where only a family of possible distributions of the uncertain variables is known, are excluded. The third assumption guarantees that the functions $Pr\left\{g_i(\cdot, y, \xi) \leq 0\right\}$ are continuous w.r.t. $u \in \mathcal{U}$. The last assumption guarantees that for every $u \in \mathcal{U}$ and every $\xi \in \Omega$ there exists exactly one possible system state y. This is necessary, since otherwise it is hard to decide how to treat several possible system states, especially if some of these states are inside and others are outside of the desired bounds.

3.1 Classification of CCOPT problems

CCOPT problems can be classified considering several aspects, where one possible scheme for classification can be seen in Figure 3.1 (this figure mimics the presentation in [55]). There, four different aspects are considered, i.e., type of model equations, linearity, temporal behavior of the uncertain variables, and distribution of the uncertain variables. Since all possible combinations of the single components lead to a valid CCOPT problem, overall 16 different types of optimization problems can be found. In the following, the impact of the different classification aspects on the solution of CCOPT problems is discussed.

3.1.1 Type of model equations

The process model can either be static, i.e., they are described by an algebraic equation, or dynamic, i.e., they are described by differential or difference equations. While static problems can be handled directly with the proposed methods, dynamic problems require further treatment. As a first step the following restriction is imposed: For the remainder of this work in connection with dynamic problems only difference equations will be considered, since differential equations can always be discretized to yield difference equations (e.g., using Euler or Runge-Kutta methods). Nonetheless, there are several possible problems, which may occur in connection with difference equations. First, the time horizon may be infinite, resulting in an infinite state space. Second, guaranteeing the feasibility over a long (or even infinite) time horizon might be impossible and even if a feasible solution exists it might be very conservative. To overcome these problems the usage of Model Predictive Control (MPC) methods is proposed. MPC methods require the successive solution of CCOPT problems for a given (changing) time horizon. A schematic of this process can be found in Figure 3.2. Based on measurements from the past, the future behavior of the process is predicted for a certain time horizon (the so called prediction horizon) and the objective function is minimized over the prospective horizon. But instead of applying all obtained optimal controls only a part of the controls (those in the so called control horizon) are applied to the process. This is done to compensate deviations, which might be introduced by the realization of the uncertain variables. The whole process (prediction, optimization, applying part of the optimal controls) is then repeated. It is clear that this approach overcomes the problem of an infinite state space, since the optimization horizon is finite. The problem of guaranteeing the existence of feasible solutions is more involved and topic of ongoing research. For specific problems it can be shown that there exists a certain upper bound for the length of the optimization horizon, where a violation of this bound always leads to an empty feasible set [51]. More general results are not available at this time.

3.1.2 Linearity

The functions $g(u, y, \xi)$ can depend linearly (i.e., $\frac{\partial}{\partial \xi} g(u, y, \xi) \equiv \mathit{const}$) or non-linearly (i.e., $\frac{\partial}{\partial \xi} g(u, y, \xi) \not\equiv \mathit{const}$) on the uncertain input variables ξ. Linear constraints are of special interest in connection with Gaussian distributed uncertainties, since this combination allows to directly determine the distribution of $g(u, y, \xi)$. Furthermore, several approximation schemes exist for linear constraints.

3.1.3 Temporal behavior and distribution of the uncertain variables

The last two categories in the classification determine what kind of integration routines is necessary to evaluate the chance constraints. In a problem with static uncertainties the distribution and possible distribution parameters are known beforehand and a suitable integration routine can be a priori chosen depending on the distribution of the uncertain variables. In the case of dynamic uncertainties either distribution parameters or even the type of distribution may

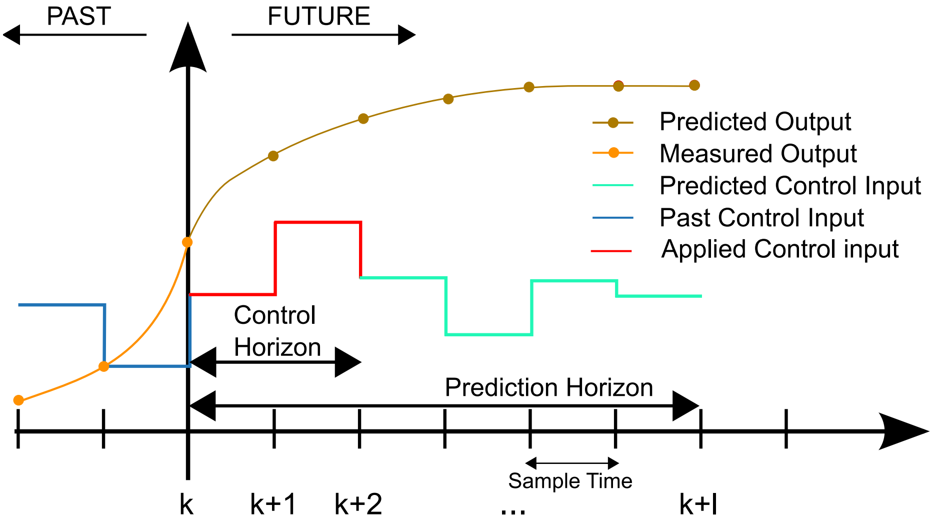

Figure 3.2: *Schematic of the MPC process.*[2]

change over time. In the first case, the type of distribution has a huge influence, i.e., distributions which can be transformed to a standard case (like the Gaussian distribution) do not pose a problem. This is due to the fact that irrespective of the actual distribution parameters it is always possible to transform the problem to the standard case, i.e., only integration routines for the standard case have to be implemented. On the other hand, if no standard case exists, integration rules have to be found online (like in the case of Beta-distributed uncertainties), resulting in a possible higher computational demand. The possibility of a change in the type of the distribution of the uncertainties will not be considered in this work. Nonetheless, an effect similar to the case for distributions without a standardized form is to be expected, i.e., integration routines need to be constructed online.

3.2 Components of solution approaches

Each of the 16 different CCOPT problems defined above has unique properties which allow certain techniques to be used in the solution process. In general, the solution process consists of three parts:

(i) transforming the probabilistic into deterministic constraints,

(ii) solution of the model equations, and

(iii) numerical integration.

The interaction of these three steps is shown in Figure 3.3. In general, a CCOPT solver is based on a standard NLP solver (e.g., interior point or SQP solver) as can be seen in the first

[2] This figure was created based on a work of Martin Behrendt (`http://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg`).

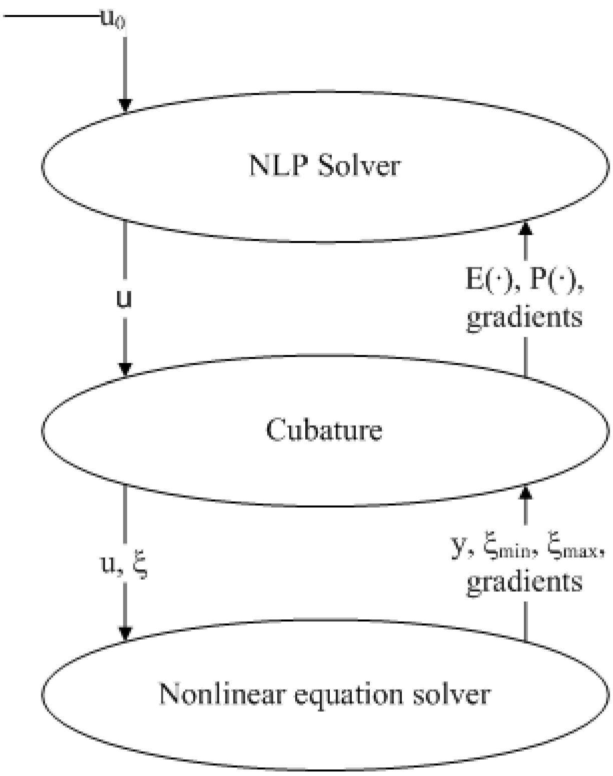

Figure 3.3: *Basic structure of a CCOPT solver.*

layer. In order to be able to employ such solvers, the chance constraints have to be transformed into equivalent deterministic constraints, using for instance one of the methods presented in Chapter 4. These methods typically require the evaluation of multivariate integrals (so called cubature) as shown in the second layer. Several methods for carrying out the integration task are presented in Chapter 5. Finally, in order to evaluate the integrals it is necessary to solve the model equations using for instance one of the methods presented in Chapter 6. Depending on the starting point u_0 of the optimization method the described steps may be carried out repeatedly.

4 Evaluation of Chance Constraints

In this chapter several approaches to the evaluation of chance constraints will be introduced. For brevity, the notation $g(u,\xi) := g(u,y,\xi)$ is used throughout the chapter and the remainder of this thesis. This does not suppress any dependencies, since y also depends on u and ξ through the model equations $F(u,y,\xi) = 0$. Furthermore, any indices on $g(\cdot,\cdot)$ are dropped, since all single chance constraints are treated separately.

4.1 Direct computation

The direct computation approach is the (historically) first one used to solve CCOPT problems, since it was proposed by Charnes et al., who introduced this approach together with CCOPT in [17]. Even today it finds widespread application, due to its ease of use and low demand on computational power. The major disadvantage of this approach is that it only works for (multivariate) Gaussian distributed uncertainties and only for function $g(u,\xi)$ which are linear in ξ. Therefore, the next assumption is necessary for the remainder of this section.

Assumption 4.1.1. The uncertain variables ξ underlie a multivariate normal distribution with expectation μ and covariance matrix Σ. Furthermore, $g(u,\xi)$ can be expressed as $g(u,\xi) = g(u)^T\xi$, where $g : \mathbb{R}^n \to \mathbb{R}^p$.

Notation 4.1.2. The cdf of an univariate standard Gaussian distributed uncertain variable is denoted by $\Phi(\cdot)$, whereas the pdf is denoted by $\varphi(\cdot)$.

Lemma 4.1.3. *Under the previous assumption the chance constraint $Pr\left\{g(u,\xi) \leq 0\right\} \geq \alpha$ is equivalent to $\frac{-g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}} \geq \Phi^{-1}(\alpha)$. Additionally,*

$$\frac{\partial}{\partial u_i} Pr\left\{g(u,\xi) \leq 0\right\} = \\ \varphi\left(-\frac{g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}}\right) \frac{\big(\mu^T g(u)\big) g(u)^T\Sigma \frac{\partial}{\partial u_i} g(u) - \big(\mu^T \frac{\partial}{\partial u_i} g(u)\big) g(u)^T \Sigma g(u)}{\big(g(u)^T\Sigma g(u)\big)^{\frac{3}{2}}}.$$

Proof: For this proof we employ that for a multivariate Gaussian distributed uncertain variable ξ it holds that $g(u)^T\xi$ is also Gaussian distributed with expectation $g(u)^T\mu$ and variance

$g(u)^T \Sigma g(u)$. A short calculation reveals

$$Pr\left\{g(u,\xi) \le 0\right\} = Pr\left\{\frac{g(u,\xi) - g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}} \le -\frac{g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}}\right\} \tag{4.1.1}$$

$$= \Phi\left(-\frac{g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}}\right), \tag{4.1.2}$$

i.e., $Pr\left\{g(u,\xi) \le 0\right\} \ge \alpha$ is equivalent to $\Phi\left(-\frac{g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}}\right) \ge \alpha$. Furthermore, the function $\Phi : \mathbb{R} \to \mathbb{R}$ is bijective and, therefore, $\Phi^{-1}(\cdot)$ exists and the last statement is equivalent to $-\frac{g(u)^T\mu}{\sqrt{g(u)^T\Sigma g(u)}} \ge \Phi^{-1}(\alpha)$, which completes the first part of the proof. For the second part, calculating the derivative of (4.1.2) immediately yields the desired result. □

4.2 Linearization

The idea of linearization methods is to extend the direct computation method to nonlinear problems by using a first order Taylor series expansion. More clearly, for given $u \in \mathcal{U}$ and $\xi_0 \in \Omega$ one uses

$$\begin{aligned} g(u,\xi) &= g(u,\xi_0) + \nabla_\xi g(u,\xi_0)(\xi - \xi_0) + o\left((\xi - \xi_0)^2\right) \\ &\approx g(u,\xi_0) + \nabla_\xi g(u,\xi_0)(\xi - \xi_0). \end{aligned}$$

The next lemma shows that under some assumption on the concavity of $g(u,\xi)$ one can guarantee that a solution of the approximate (linear) problem is feasible to the original problem (NLP).

Lemma 4.2.1. *Let $\mathcal{K}_{lin} = \{u \in \mathcal{U} | Pr\left\{g(u,\xi_0) + \nabla_\xi g(u,\xi_0)(\xi - \xi_0) \le 0\right\} \ge \alpha\}$. If $g(u,\xi)$ is concave w.r.t. ξ for all $u \in \mathcal{U}$ then*

$$\mathcal{K}_{lin} \subset \mathcal{K}.$$

Proof: The case $\mathcal{K}_{lin} = \emptyset$ is trivial. Therefore, assume $\mathcal{K}_{lin} \neq \emptyset$. Then, for $u \in \mathcal{K}_{lin}$ we have $Pr\left\{g(u,\xi_0) + \nabla_\xi g(u,\xi_0)(\xi - \xi_0) \le 0\right\} \ge \alpha$. Furthermore, since $g(u,\xi)$ is concave w.r.t. ξ, we have $g(u,\xi) \le g(u,\xi_0) + \nabla_\xi g(u,\xi_0)(\xi - \xi_0)$ for all $\xi \in \Omega$. As a direct consequence, we get $Pr\left\{g(u,\xi) \le 0\right\} \ge \alpha$. This yields $u \in \mathcal{K}$, which completes the proof. □

Remark 4.2.2. If $g(u,\xi)$ is convex w.r.t. ξ for all $u \in \mathcal{U}$ a similar reasoning is possible. In this case consider the equivalent constraint $Pr\left\{g(u,\xi) > 0\right\} \le 1 - \alpha$.

4.2.1 Implementation

Similar to the direct approach, linearization is very straight-forward to implement. Nonetheless, one should take care when using such techniques in the presence of larger variances of the uncertain variables. As shown in the work of Garnier et al. [31], these might lead to considerable errors in the gradients computation and, hence, to problems in the solution of the optimization problem.

4.3 Projection approaches

Projection methods were first introduced by Wendt et al. [89] and provide an approach for a wide range of nonlinear problems. Additionally, these methods are not restricted to a certain kind of uncertainty. The main idea is to find a monotonic relationship between $g(u,\xi)$ and one of the uncertain input variables ξ. This relation can be used to transform the chance constraint into the domain of the uncertain input variables, where the probabilities can be determined using multivariate integration. A concise description of this process is given in the following.

Notation 4.3.1. The vector ξ without the j-th entry, $j \in \{1,\dots,p\}$ is denoted by ξ_{-j}, i.e.,

$$\xi_{-j} = (\xi_1,\dots,\xi_{j-1},\xi_{j+1},\dots,\xi_p)$$

and $\xi_{-j} \in \mathbb{R}^{p-1}$. The set $\Omega_{-j} \subset \mathbb{R}^{p-1}$ is defined by $\Omega_{-j} := \{\xi_{-j} | \xi \in \Omega\}$. Furthermore, (ξ_{-j},ξ_j) is an additional notation for the vector ξ.

Definition 4.3.2 (Monotonic relation, [33])**.** The function $g(u,\xi)$ is monotonically related with an uncertain input ξ_j, $j \in \{1,\dots,p\}$ on the interval $(-\infty,0]$ if uniformly for arbitrary fixed $u \in \mathcal{U}$ and input ξ_{-j} the following two conditions are satisfied:

(i) For each $y \in (-\infty,0]$ exists an input $\xi_j(y,u,\xi_{-j})$ such that $g\big(u,(\xi_{-j},\xi_j(y,u,\xi_{-j}))\big) = y$ and $(\xi_{-j},\xi_j(y,u,\xi_{-j})) \in \Omega$.

(ii) $\infty < y_1 < y_2 \leq 0$ implies on the whole interval $(-\infty,0]$

a) either $\xi_j(y_1,u,\xi_{-j}) < \xi_j(y_2,u,\xi_{-j})$

b) or $\xi_j(y_1,u,\xi_{-j}) > \xi_j(y_2,u,\xi_{-j})$.

In case (a) the monotonic relation is called positive and is denoted by $g(u,\xi) \uparrow \xi_j$. In case (b) it is called negative and denoted by $g(u,\xi) \downarrow \xi_j$.

If we have a monotonic relation we can project the constraints in the following way. In case (a) we have

$$\begin{aligned} Pr\left\{g(u,\xi) \leq 0\right\} &= Pr\left\{\xi_j \leq \xi_j(0,u,\xi_{-j})\right\} \\ &= \int_{\Omega_{-j}} \int_{-\infty}^{\xi_{-j}(0,u,\xi_{-j})} \phi(\xi) d\xi_j d\xi_{-j}, \end{aligned}$$

whereas in case (b) we get

$$\begin{aligned} Pr\left\{g(u,\xi)\leq 0\right\} &= Pr\left\{\xi_j(0,u,\xi_{-j})\leq \xi_j\right\} \\ &= \int_{\Omega_{-j}}\int_{\xi_{-j}(0,u,\xi_{-j})}^{\infty}\phi(\xi)d\xi_j d\xi_{-j}. \end{aligned}$$

Under the following assumption we can also determine the gradients w.r.t. the controls u.

Assumption 4.3.3. The function $\xi_j(0,u,\xi_{-j})$ is for all $\xi_{-j}\in\Omega_{-j}$ continuously differentiable w.r.t. to $u\in\mathcal{U}$.

Then, using Leibniz's rule, we can find the gradients with respect to u in case (a) as

$$\begin{aligned} \nabla_u Pr\left\{g(u,\xi)\leq 0\right\} &= \nabla_u\int_{\Omega_{-j}}\int_{-\infty}^{\xi_{-j}(0,u,\xi_{-j})}\phi(\xi)d\xi_j d\xi_{-j} \\ &= \int_{\Omega_{-j}}\phi(\xi_{-j},\xi_j)|_{\xi_j=\xi_j(0,\xi_{-j})}\nabla_u\xi_{-j}(0,u,\xi_{-j})d\xi_{-j} \end{aligned}$$

and in case (b) as

$$\begin{aligned} \nabla_u Pr\left\{g(u,\xi)\leq 0\right\} &= \nabla_u\int_{\Omega_{-j}}\int_{\xi_{-j}(0,u,\xi_{-j})}^{\infty}\phi(\xi)d\xi_j d\xi_{-j} \\ &= -\int_{\Omega_{-j}}\phi(\xi_{-j},\xi_j)|_{\xi_j=\xi_j(0,\xi_{-j})}\nabla_u\xi_{-j}(0,u,\xi_{-j})d\xi_{-j}. \end{aligned}$$

Note that in comparison with the computation of the probability values the dimension of integration is reduced by one.

Remark 4.3.4. Royset and Polak [74] proposed a similar approach under the context of Sample Average Approximation (SAA), using Monte-Carlo integration for the evaluation of the involved integrals.

As a next step a method to determine monotonic relationships is described. This method is based on a global implicit function theorem and was proposed by Geletu et al. [33]. It can be used in the special case that $g_i(u,\xi)=y_{j(i)}$, i.e., some or all of the state variables are constrained. The method is based on the global implicit function theorem 2.1.3 applied to the model equations $F(y,u,\xi)=0$. If the three conditions of Theorem 2.1.3 are fulfilled for all $u\in\mathcal{U}$, then there exists a global implicit representation

$$y=\zeta(u,\xi),\ \xi\in\Omega.$$

This equation can be used to determine a monotonic relation between a single coordinate y_i of the state variables y and a single coordinate ξ_j of ξ in the following way. By means of the local implicit function theorem we have

$$F(\zeta(u,\xi),u,\xi)=0,\ u\in\mathcal{U},\ \xi\in\Omega. \tag{4.3.1}$$

Calculating the total partial derivate of (4.3.1) with respect to ξ_j results in

$$\frac{dF(\zeta(u,\xi),u,\xi)}{d\xi_j} = \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial y_i}\frac{\partial y_i}{\partial \xi_j} + \sum_{\substack{k=1\\k\neq i}}^{n} \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial y_k}\frac{\partial y_k}{\partial \xi_j} + \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial \xi_j} = 0, \tag{4.3.2}$$

where $\frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial \xi_j}$ are the partial derivatives with respect to ξ_j. By the global implicit function theorem, the matrix $\frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial y}$ is regular for all $\xi \in \Omega$ and all $u \in \mathcal{U}$ with $F(y,u,\xi) = 0$. Therefore, the linear system of equations

$$\eta^T \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial y_k} = \delta_{ki},\ k = 1,\dots,n$$

has a unique C^1-solution $\eta(u,\xi)$ for $\xi \in \Omega$ and $u \in \mathcal{U}$. Here, $\delta_{ki} = 1$ if $i = k$ and $\delta_{ki} = 0$, otherwise. Multiplying both sides of (4.3.2) with $\eta(u,\xi)$ results in

$$\frac{\partial y_i}{\partial \xi_j} = -\eta(u,\xi)^T \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial \xi_j},$$

which can be used to derive necessary and sufficient conditions for strict monotonicity of y_i to ξ_j. This gives rise to the following theorem.

Theorem 4.3.5 (Geletu et al. 2011 [33]). *Suppose the assumptions for the global implicit function theorem 2.1.3 hold true. Then y_i is globally monotonically to ξ_j if either*

- $\xi_j \uparrow y_i$, *i.e.,*

$$\eta(u,\xi)^T \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial \xi_j} < 0,\ \xi \in \Omega,\ u \in \mathcal{U},$$

or

- $\xi_j \downarrow y_i$, *i.e.,*

$$\eta(u,\xi)^T \frac{\partial F(\zeta(u,\xi),u,\xi)}{\partial \xi_j} > 0,\ \xi \in \Omega,\ u \in \mathcal{U}.$$

As a direct consequence of the theorem, it is only necessary to check the sign of the scalar product

$$\eta^T \left(\frac{\partial F}{\partial \xi_j}\right)$$

to determine monotonicity relations. Nevertheless, for larger scale systems an analysis of all possible combinations y_i and ξ_j is impractical, due to the pure amount of such combinations. One should also keep in mind that monotonicity relations need not necessarily exist.

4.4 Analytical approximations

Analytical Approximation (AA) are one approach for handling CCOPT problems, including for instance highly nonlinear and non-monotonic problems. Their main advantage is the possibility to solve medium and large scale problems without detailed knowledge (e.g., monotonicity relations) of the underlying system.

The section is loosely based on [34], additional material includes a new method to evaluate gradients when using AA.

The main idea of AA is based on the fact that probability functions can be alternatively expressed using expectations, i.e., if $p(u) = Pr\left\{g(u,\xi) \leq 0\right\}$ then

$$1 - p(u) = Pr\left\{g(u,\xi) > 0\right\} = E\left[\mathbb{1}(g(u,\xi))\right], \text{ where } \mathbb{1}(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

Consequently, the constraint

$$Pr\left\{g(u,\xi) \leq 0\right\} \geq \alpha$$

is equivalent to

$$1 - p(u) \leq 1 - \alpha \text{ and finally to } E\left[\mathbb{1}(g(u,\xi))\right] \leq 1 - \alpha.$$

Furthermore, defining $\mathcal{K} = \{u \in \mathcal{U} \; p(u) \geq \alpha\}$ and $\mathcal{M} = \{u \in \mathcal{U} | E\left[\mathbb{1}(g(u,\xi))\right] \leq 1 - \alpha\}$ one has $\mathcal{M} = \mathcal{K}$, i.e., the optimal solution sets of the problems $\min_{u \in \mathcal{M}} E\left[f(u,\xi)\right]$ and $\min_{u \in \mathcal{K}} E\left[f(u,\xi)\right]$ are the same. Please note that the existence of optimal solution sets is guaranteed by the assumptions made on the involved functions (see Chapter 3).

Remark 4.4.1. The more general case of two-sided constraints $Pr\left\{a \leq g(u,\xi) \leq b\right\}$ for $a, b \in \mathbb{R}$, $a < b$ can be handled by AA as well. In this case,

$$\begin{aligned} Pr\left\{a \leq g(u,\xi) \leq b\right\} &= 1 - Pr\left\{g(u,\xi) - b > 0\right\} - Pr\left\{a - g(u,\xi) > 0\right\} \\ &= 1 - E\left[\mathbb{1}(g(u,\xi) - b)\right] - E\left[\mathbb{1}(a - g(u,\xi))\right]. \end{aligned}$$

The approximation methods introduced in this section are then applied to the two subtrahends.

Despite the fact that there are now two equivalent descriptions of the CCOPT problem, this does not directly simplify the process of solution. This is due to the fact that the function $\mathbb{1}(\cdot)$ is discontinuous, which leads to difficulties in the numerical computation. Nonetheless, the function $E\left[\mathbb{1}(g(u,\xi))\right]$ can be used to construct tractable approximations to the CCOPT problem. More clearly, assume there exists a continuous function $\psi : \mathbb{R}_{++} \times \mathbb{R}^m \to \mathbb{R}_+$ and a $\tau_{max} \in \mathbb{R}_{++}$ with the following properties:

- P1: $E\left[\mathbb{1}(g(u,\xi))\right] \leq \psi(\tau, u)$ for all $0 < \tau < \tau_{max}$ and all $u \in \mathcal{U}$,
- P2: $\inf_{\tau > 0} \psi(\tau, u) = E\left[\mathbb{1}(g(u,\xi))\right]$ for all $u \in \mathcal{U}$,
- P3: $\psi(\tau, u)$ is non-decreasing w.r.t. τ, $0 < \tau$.

Here, the property P1 guarantees that a solution $u^*_{NLP_\tau}$ of the approximate problem

$$\begin{aligned}(NLP_\tau)\ \min_{u\in\mathcal{U}}\ & E\left[f(u,\xi)\right]\\ s.t.\ & \psi(\tau,u)\leq 1-\alpha\end{aligned}$$

is also feasible to the original problem, since $E\left[\mathbb{1}(g(u^*_{NLP_\tau},\xi))\right]\leq\psi(\tau,u^*_{NLP_\tau})\leq 1-\alpha$. A function $\psi(\cdot,\cdot)$ having only property P1 would suffice to generate an approximate solution, but regardless of the choice of the parameter τ it is difficult to relate a solution $u^*_{NLP_\tau}$ of the approximate problem to a solution of the original problem. This is evident in an AA approach proposed by Nemirovski and Shapiro [64], where only the property P1 is fulfilled. For general problems, the aforementioned approach grossly underestimates the actual probability of holding the constraints leading to robust but also conservative solutions of the CCOPT problem. To overcome such difficulties, it is desirable to find approximations which are arbitrarily close (in a certain sense) to the original problem. Therefore, properties P2 and P3 are also desired, which guarantee (in connection with the continuity of $\psi(\cdot,\cdot)$) that

$$\inf_{\tau>0}\psi(\tau,u)=\lim_{\tau\to 0^+}\psi(\tau,u)=E\left[\mathbb{1}(g(u,\xi))\right].$$

The advantage of these properties is made clear in the next lemma, which is based on the following definitions.

Definition 4.4.2 (Regularity)**.** Let $\alpha\in\left[\frac{1}{2},1\right]$. A chance constraint $Pr\left\{g(u,\xi)\leq 0\right\}\geq\alpha$ is called regular if for each $u\in\mathcal{U}$ with $Pr\left\{g(u,\xi)\leq 0\right\}=\alpha$ there exists a sequence $(u_k)_{k\in\mathbb{N}}$ such that $\lim_{k\to\infty}u_k=u$ and $Pr\left\{g(u_k,\xi)\leq 0\right\}>\alpha$ for all $k\in\mathbb{N}$.

Definition 4.4.3 (Convergence of set valued maps, [71] p. 152)**.** .For X, Y metric spaces, $F\to\mathcal{P}(Y)$ a set-valued map, and $x_0\in X$

$$\begin{aligned}\limsup_{x\to x_0}F(x)&:=\left\{y\mid\exists x_n\to x_0,\ \exists y_n\to y,\ y_n\in F(x_n)\right\},\\ \liminf_{x\to x_0}F(x)&:=\left\{y\mid\forall x_n\to x_0,\ \exists y_n\to y,\ y_n\in F(x_n)\right\},\end{aligned}$$

and, if $\limsup_{x\to x_0}F(x)=\liminf_{x\to x_0}F(x)=\hat{Y}$, $\hat{Y}\subset Y$ then

$$\lim_{x\to x_0}F(x)=\hat{Y}.$$

Lemma 4.4.4. *Consider a regular chance constraint and suppose $\psi:\mathbb{R}_{++}\times\mathbb{R}^m\to\mathbb{R}_+$ is a continuous function having the properties P1–P3. Let $\mathcal{M}_\tau=\{u\in\mathcal{U}|\psi(u,\tau)\leq 1-\alpha\}$. Then*

$$\lim_{\tau\to 0^+}\mathcal{M}_\tau=\mathcal{M}=\mathcal{K}$$

in the sense of convergence of set-valued maps as in Definition 4.4.3.

Proof: Let us start by observing that $\mathcal{M}$ is compact since $\mathcal{U}$ is compact. Now, for

$$0 < \tau_1 < \tau_2 < \tau_{max}$$

the following inclusion holds $\mathcal{M}_{\tau_2} \subset \mathcal{M}_{\tau_1} \subset \mathcal{M}$. Since $\mathcal{M}$ is compact one automatically gets $\liminf_{\tau \to 0^+} \mathcal{M}_\tau \subset \limsup_{\tau \to 0^+} \mathcal{M}_\tau \subset \mathcal{M}$, i.e., in order to complete the proof it is sufficient to show $\mathcal{M} \subset \liminf_{\tau \to 0^+} \mathcal{M}_\tau$.

Assume $\mathcal{M} \not\subset \liminf_{\tau \to 0^+} \mathcal{M}_\tau$, then there exists $\hat{u} \in \mathcal{M}$ and a sequence $(\tau_k)_{k \in \mathbb{N}}$, $\tau_k \to 0^+$ such that for all sequences $(u_k)_{k \in \mathbb{N}}$ with $u_k \to \hat{u}$ exists a subsequence $(u_{k_l})_{l \in \mathbb{N}}$ with $u_{k_l} \notin \mathcal{M}_{\tau_{k_l}}$, i.e., for all $l \in \mathbb{N}$: $\psi(\tau_{k_l}, u_{k_l}) > 1 - \alpha$. This implies that $Pr\left\{g(\hat{u}, \xi) \le 0\right\} = \alpha$ or equivalently $E\left[\mathbb{1}(g(\hat{u}, \xi))\right] = 1 - \alpha$, since $\psi(\cdot, \hat{u})$ is a non-decreasing and continuous function and $\lim_{\tau \to 0^+} \psi(\tau, \hat{u}) = E\left[\mathbb{1}(g(\hat{u}, \xi))\right]$.

Indeed, if $E\left[\mathbb{1}(g(\hat{u}, \xi))\right] < 1 - \alpha$ then there exists an index $k_0 \in \mathbb{N}$ such that for all $k > k_0$: $\psi(\tau_k, \hat{u}) \le 1 - \alpha$ and, therefore, $\hat{u} \in \mathcal{M}_{\tau_k}$. Taking the constant sequence $u_k = \hat{u}$ results in a direct contradiction of the assumption.

Due to the regularity assumption, we now have a sequence $(\bar{u}_k)_{k \in \mathbb{N}}$ with $\lim_{k \to \infty} \bar{u}_k = \hat{u}$ and $E\left[\mathbb{1}(g(\bar{u}_k, \xi))\right] < 1 - \alpha$. Again, due to the continuity of $\psi(\cdot, \bar{u}_k)$ we have that for sufficiently small τ_k we can find a $\bar{u}_{l(k)} \in (\bar{u}_k)_{k \in \mathbb{N}}$ with $\psi(\tau_k, \bar{u}_{l(k)}) \le 1 - \alpha$, i.e., $u_{\bar{l(k)}} \in \mathcal{M}_{\tau_k}$. Now the sequence $u_k = \bar{u}_{l(k)}$ yields a contradiction to the assumption. Therefore, $\mathcal{M} \subset \liminf_{\tau \to 0^+} \mathcal{M}_\tau$ and finally $\mathcal{M} = \lim_{\tau \to 0^+} \mathcal{M}_\tau$. □

The above result indicates that the feasible sets of the approximate problems (NLP_τ) converge to the feasible set of the original problem as $\tau \to 0^+$. The next lemma shows that this convergence is uniform with respect to the Haussdorf metric.

Definition 4.4.5 (Hausdorff distance, [26] p. 393)**.** Let $A, B \subset \mathbb{R}^n$ bounded sets and $x \in \mathbb{R}^n$. The distance of x to the set A is defined by

$$\mathrm{dist(x, A)} := \inf_{\mathrm{y \in A}} \|\mathrm{x - y}\|_2 .$$

The Hausdorff distance of the sets A, B is given by

$$H(A, B) := \max\left\{\sup_{x \in A} \mathrm{dist(x, B)}, \sup_{\mathrm{x \in B}} \mathrm{dist(x, A)}\right\}.$$

Lemma 4.4.6. *If the chance constraint $Pr\left\{g(u, \xi) \le 0\right\}$ is regular then*

$$\lim_{\tau \to 0^+} H(\mathcal{M}_\tau, \mathcal{M}) = 0.$$

Proof: Since $\mathcal{M}_\tau \subset \mathcal{M}$ for all $\tau > 0$ the expression $H(\mathcal{M}_\tau, \mathcal{M})$ simplifies to $\sup_{u \in \mathcal{M}} \mathrm{dist(u}, \mathcal{M}_\tau)$. Choose an arbitrary $u \in \mathcal{M}$. Then, similar to the proof of Lemma 4.4.4, either $Pr\left\{g(u, \xi) \le 0\right\} > \alpha$ or $Pr\left\{g(u, \xi) \le 0\right\} = \alpha$. In the first case we have $\psi(\hat{\tau}, u) \le 1 - \alpha$ for a sufficiently small $\hat{\tau}$, i.e., $u \in \mathcal{M}_{\hat{\tau}}$ and $\mathrm{dist(u}, \mathcal{M}_\tau) = 0$ for all $0 < \tau \le \hat{\tau}$. In the second case we find again a sequence $(u_k)_{k \in \mathbb{N}}$, $u_k \to u$ and $Pr\left\{g(u_k, \xi) \le 0\right\} > \alpha$ for all $k \in \mathbb{N}$. Due to the continuity of $\psi(\cdot, u_k)$, for any u_k we can find a τ_k such that $\psi\left(\tau, u_k\right) \le 1 - \alpha$ for all $\tau \le \tau_k$, i.e., $u_k \in \mathcal{M}_\tau$ for all $\tau \le \tau_k$. This implies that $\mathrm{dist(u}, \mathcal{M}_\tau) \le \mathrm{dist(u, u_k)}$ for all $\tau \le \tau_k$. For $k \to \infty$ we get $\tau_k \to 0$ and $u_k \to u$, i.e., $\mathrm{dist(u, u_k)} \to 0$, which implies $\lim_{\tau \to 0^+} H(\mathcal{M}_\tau, \mathcal{M}) = 0$. □

Till now, we have shown that the feasible set $\mathcal{M}$ of the original problem can be approximated arbitrarily close by sets $\mathcal{M}_\tau$. Here, the natural question arises, whether such behavior can be also shown for the solutions of the approximate problems, i.e., if one has a sequence of solutions of approximate problems (NLP_{τ_k}) with $\tau_k \to 0$, does this sequence converge to a solution of the original problem and, conversely, if one has a solution u^*_{NLP} of the original problem is there a sequence of solutions $u^*_{NLP_{\tau_k}}$ of the approximate problems (NLP_{τ_k}) with $u^*_{NLP_{\tau_k}} \to u^*_{NLP}$. The following theorem will shed some light on this issue.

Theorem 4.4.7 (Geletu et al. 2013 [34])**.**

(i) *Let $(\tau_k)_{k\in\mathbb{N}}$, $\tau_k \to 0$ for $k \to \infty$ and $(u_k)_{k\in\mathbb{N}}$ be sequences such that u_k is a local optimal solution of (NLP_{τ_k}), $k \in \mathbb{N}$. Then there exists a subsequence $(u_{k_l})_{l\in\mathbb{N}}$ of $(u_k)_{k\in\mathbb{N}}$ such that $u_{k_l} \to u^*$. Let $\bar{B}(u^*)$ be a closed ball around u^*, $u^* \in \mathcal{M} \cap \bar{B}(u^*)$. If for all k_l with $u_{k_l} \in \bar{B}(u^*) \cap \mathcal{M}$ the quantity u_{k_l} is a global minimizer of the problem*

$$\min_{u\in\mathcal{M}_{\tau_{k_l}}\cap\bar{B}(u^*)} E\left[f(u,\xi)\right] \tag{4.4.1}$$

then

$$E\left[f(u^*,\xi)\right] = \min_{u\in\mathcal{M}\cap\bar{B}(u^*)} E\left[f(u,\xi)\right], \tag{4.4.2}$$

i.e., u^* is a local optimal solution of (NLP).

(ii) *Conversely, if u_0 is a strict local minimizer of (NLP), then there is a sequence of local minimizers u_k of (NLP_{τ_k}) which converges to u_0.*

Proof:

(i) Assume that u^* is not a local optimal solution of (NLP), i.e., there exists $\hat{u} \in \bar{B}(u^*) \cap \mathcal{M}$ with $E\left[f(\hat{u},\xi)\right] < E\left[f(u^*,\xi)\right]$. By Lemma 4.4.6 we can find a sequence $(z_k)_{k\in\mathbb{N}}$ with $z_k \in \mathcal{M}_{\tau_k} \cap \bar{B}(u^*)$ and $z_k \to \hat{u}$. For a subsequence $(z_{k_l})_{l\in\mathbb{N}}$ we get

$$E\left[f(z_{k_l},\xi)\right] \geq E\left[f(u_{k_l},\xi)\right],$$

because of the optimality of u_{k_l}. This implies that

$$E\left[f(\hat{u},\xi)\right] = \lim_{k_l\to\infty} E\left[f(z_{k_l},\xi)\right] \geq \lim_{k_l\to\infty} E\left[f(u_{k_l},\xi)\right] = E\left[f(u^*,\xi)\right] > E\left[f(\hat{u},\xi)\right].$$

Hence we get a contradiction.

(ii) Since u_0 is a local optimal solution, we have $E\left[f(u,\xi)\right] \geq E\left[f(u_0,\xi)\right]$ for all $u \in clB(u_0) \cap \mathcal{M}$. Furthermore, for a sequence $(\tau_k)_{k\in\mathbb{N}}$ with $\tau_k \to 0$ it holds that

$$H(\mathcal{M}_{\tau_k} \cap clB(u_0), \mathcal{M} \cap clB(u_0)) \to 0, \tag{4.4.3}$$

especially for k sufficiently large $\mathcal{M}_{\tau_k} \cap clB(u_0)$ is non-empty. By compactness of these sets and the continuity of the objective function, one can find $u_k \in \mathcal{M}_{\tau_k} \cap clB(u_0)$ such that for all $u \in \mathcal{M}_{\tau_k} \cap clB(u_0)$: $E\left[f(u,\xi)\right] \geq E\left[f(u_k,\xi)\right]$. The sequence $(u_k)_{k\in\mathbb{N}} \subset \mathcal{M}$ has a convergent subsequence $(u_{k_l})_{l\in\mathbb{N}}$ with $u_{k_l} \to u^*$ for a $u^* \in \mathcal{M}$. Due to (4.4.3) and the continuity of $E\left[f(\cdot,\xi)\right]$ it follows that $E\left[f(u,\xi)\right] \geq E\left[f(u^*,\xi)\right]$ for all $u \in clB(u_0) \cap \mathcal{M}$. Hence $E\left[f(u^*,\xi)\right] = E\left[f(u_0,\xi)\right]$ and since u_0 is a strict local solution it follows that $u^* = u_0$. Additionally, there exists a k_0 such that for all $k > k_0$: $u_k \in B(u_0)$, i.e., u_k is a local solution of (NLP_{τ_k}).

□

Remark 4.4.8. The above theorem requires that the u_{k_l} are global minimizers of certain optimization problems. This assumption is fulfilled if for instance $E\left[f(\cdot,\xi)\right]$ is convex on the set $\bar{B}(u^*) \cap \mathcal{M}$ and for all k_l with $u_{k_l} \in \bar{B}(u^*) \cap \mathcal{M}$ the sets $\mathcal{M}_{\tau_{k_l}}$ are convex.

The theorem stated above guarantees under some assumptions that for an arbitrary sequence $(\tau_k)_{k\in\mathbb{N}}$ with $\tau_k \to 0$ and corresponding local optimal solutions u_k of the approximate problems (NLP_{τ_k}) we can find a subsequence $(u_{k_l})_{l\in\mathbb{N}}$ with $u_{k_l} \to u^*$ and u^* is a local optimal solution of (NLP). Since the solutions u_k are always feasible to the original problem we now have a practical way of generating solutions for (NLP) by solving a sequence of problems (NLP_{τ_k}). By P2 and P3 we have the point-wise convergence of $\psi(\tau,u)$ towards $E\left[\mathbb{1}(g(u,\xi))\right]$. If we can furthermore guarantee the uniform convergence of the derivatives $\nabla_u\psi(\tau,u)$ for $\tau \to 0$, then we have, by standard analysis, that

$$\lim_{\tau_k\to 0^+} \nabla_u\psi(\tau_k,u) = \nabla_u E\left[\mathbb{1}(g(u,\xi))\right].$$

In the next two parts two specific choices for the function $\psi(\tau,u)$ will be explored. The first was proposed by Nemirovski and Shapiro [64] and only fulfills P1, i.e., most of the results of this section cannot be guaranteed. As a contrast, we will also explore an approach proposed by Geletu et al. [34], which fulfills P1–P3. For this case, a new method for evaluating the gradients numerically will be introduced.

4.4.1 Some details on an analytic approximation approach proposed by Nemirovski and Shapiro

Here, we consider an analytic approximation approach proposed by Nemirovski and Shapiro [64]. They use

$$\psi_{NS}(\tau,u) = E\left[\exp(\tau^{-1}g(u,\xi))\right] \tag{4.4.4}$$

for some $\tau > 0$. The main idea is to approximate a chance constraint $Pr\left\{g(u,\xi) \leq 0\right\} \geq \alpha$ by $\inf_{\tau>0} \tau E\left[\exp(\tau^{-1}g(u,\xi))\right] - \tau(1-\alpha) \leq 0$, which is a convex constraint as long as $g(u,\xi)$ is convex w.r.t. $u \in \mathcal{U}$ and $\xi \in \Omega$. The next lemma summarizes some properties of (4.4.4).

Lemma 4.4.9. *Assuming* $\Pr\{g(u,\xi) > \epsilon\} > \underline{\alpha}_1$ *and* $\Pr\left\{-\frac{1}{2} < g(u,\xi) \leq 0\right\} > \underline{\alpha}_2$ *for an* $\epsilon > 0$, $\underline{\alpha}_1, \underline{\alpha}_2 \in (0,1)$, *and at least one* $u \in \mathcal{U}$, *the approximation* $\psi_{NS}(\tau,u)$ *has property P1, but does fulfill neither P2 nor P3.*

Proof: For all $\tau > 0$ and all $u \in \mathcal{U}$ we have $\exp(\tau^{-1}g(u,\xi)) \geq \mathbb{1}(g(u,\xi)) \geq 0$. This implies $\psi_{NS}(\tau,u) = E\left[\exp(\tau^{-1}g(u,\xi))\right] \geq E\left[\mathbb{1}(g(u,\xi))\right]$ and, hence, property P1. Taking $u \in \mathcal{U}$ with $Pr\{g(u,\xi) > 0\} > 0$ we get

$$\begin{aligned}
& E\left[\exp(\tau^{-1}g(u,\xi))\right] - E\left[\mathbb{1}(g(u,\xi))\right] \\
&= \int_{\Omega} \left(\exp(\tau^{-1}g(u,\xi)) - \mathbb{1}(g(u,\xi))\right)\phi(\xi)d\xi \\
&= \underbrace{\int_{g(u,\xi)\leq 0} \exp(\tau^{-1}g(u,\xi))\phi(\xi)d\xi}_{\geq 0} + \underbrace{\int_{g(u,\xi)>0} \left(\exp(\tau^{-1}g(u,\xi)) - 1\right)\phi(\xi)d\xi}_{>0} \\
&\geq \int_{g(u,\xi)\leq 0} \max\left\{0, 1+\tau^{-1}g(u,\xi)\right\}\phi(\xi)d\xi + \underbrace{\int_{0<g(u,\xi)\leq\epsilon} \tau^{-1}g(u,\xi)\phi(\xi)d\xi}_{\geq 0} + \underbrace{\int_{g(u,\xi)>\epsilon} \tau^{-1}g(u,\xi)\phi(\xi)d\xi}_{\geq\epsilon\underline{\alpha}_1\tau^{-1}} \\
&\geq \int_{-\frac{1}{2}<g(u,\xi)\leq 0} \max\left\{0, 1+\tau^{-1}g(u,\xi)\right\}\phi(\xi)d\xi + \epsilon\underline{\alpha}_1\tau^{-1} \\
&\geq \underline{\alpha}_2 \max\left\{0, 1-\frac{\tau^{-1}}{2}\right\} + \epsilon\underline{\alpha}_1\tau^{-1} \qquad (4.4.5) \\
&\geq \min\left\{\underline{\alpha}_2, 2\epsilon\underline{\alpha}_1\right\},
\end{aligned}$$

where for $\tau \leq \frac{1}{2}$ the first summand of (4.4.5) is zero and $\epsilon\underline{\alpha}_1\tau^{-1} \geq 2\epsilon\underline{\alpha}_1$ and for $\tau > \frac{1}{2}$ the sum (4.4.5) equals $\underline{\alpha_2} + \tau^{-1}\left(\epsilon\underline{\alpha}_1 - \frac{\underline{\alpha}_2}{2}\right)$, which is either greater than $\underline{\alpha}_2$ if the expression in the brackets is positive or greater than $2\epsilon\underline{\alpha}_1$ if the expression is negative. In summary, this yields that $\inf_{\tau>0}\psi_{NS}(\tau,u) > E\left[\mathbb{1}(g(u,\xi))\right]$.

For the third part of the proof take $u \in \mathcal{U}$ such that the assumptions of the lemma hold and $0 < \tau_1 < \tau_2$. Since $\Pr\left\{-\frac{1}{2} < g(u,\xi) \leq 0\right\} > \underline{\alpha}_2$ it holds that $Pr\{g(u,\xi) \leq 0\} = \alpha_3$ for an

$\alpha_3 \geq \underline{\alpha}_2$. Similarly, $\Pr\{g(u,\xi) > \epsilon\} = \alpha_4$ for an $\alpha_4 \geq \underline{\alpha}_1$. Now, it follows that

$$\begin{aligned}
&\psi_{NS}(\tau_2,u) - \psi_{NS}(\tau_1,u)\\
&= \int_\Omega \left(\exp(\tau_2^{-1}g(u,\xi)) - \exp(\tau_1^{-1}g(u,\xi))\right)\phi(\xi)d\xi\\
&= \underbrace{\int_{g(u,\xi)\leq 0} \left(\exp(\tau_2^{-1}g(u,\xi)) - \exp(\tau_1^{-1}g(u,\xi))\right)\phi(\xi)d\xi}_{\geq 0}\\
&\quad + \underbrace{\int_{g(u,\xi)>0} \left(\exp(\tau_2^{-1}g(u,\xi)) - \exp(\tau_1^{-1}g(u,\xi))\right)\phi(\xi)d\xi}_{\leq 0} \qquad (4.4.6)\\
&\leq \alpha_3 - \int_{g(u,\xi)>\epsilon} \left(\exp(\tau_1^{-1}g(u,\xi)) - \exp(\tau_2^{-1}g(u,\xi))\right)\phi(\xi)d\xi \qquad (4.4.7)\\
&\leq \alpha_3 - \alpha_4(\exp(\tau_1^{-1}\epsilon) - \exp(\tau_2^{-1}\epsilon)). \qquad (4.4.8)
\end{aligned}$$

To get from (4.4.6) to (4.4.7) we use that $1 \geq \exp(\tau_2^{-1}g(u,\xi)) - \exp(\tau_1^{-1}g(u,\xi)) \geq 0$ and $\alpha_3 = \int_{g(u,\xi)\leq 0}\phi(\xi)d\xi \geq \int_{g(u,\xi)\leq 0}\left(\exp(\tau_2^{-1}g(u,\xi)) - \exp(\tau_1^{-1}g(u,\xi))\right)\phi(\xi)d\xi$. Somewhat similar is the step from (4.4.7) to (4.4.8) were $\exp(\tau_1^{-1}g(u,\xi)) - \exp(\tau_2^{-1}g(u,\xi)) \geq \exp(\tau_1^{-1}\epsilon) - \exp(\tau_2^{-1}\epsilon)$ for all $g(u,\xi) > \epsilon$ is employed. Now assume τ_2 is fixed. Then we can choose

$$0 < \tau_1 < \min\left\{\frac{\epsilon}{\ln\left(\frac{\alpha_3}{\alpha_4} + \exp(\tau_2^{-1}\epsilon)\right)}, \tau_2\right\}$$

and a simple calculation reveals $\psi_{NS}(\tau_2,u) - \psi_{NS}(\tau_1,u) < 0$, i.e., $\psi_{NS}(\tau,u)$ is not non-decreasing with respect to τ. □

The assumptions in the previous lemma are necessary to avoid pathological cases, e.g., the cases $\Pr\{g(u,\xi) \leq 0\} = 1$ for all $u \in \mathcal{U}$ or $\Pr\{g(u,\xi) > 0\} = 1$ for all $u \in \mathcal{U}$. As such, the requirements are rather weak. As a direct consequence of the lemma we do not have $\inf_{\tau>0}\psi(\tau,u) = \lim_{\tau\to 0}\psi(\tau,u)$.

4.4.2 A novel analytical approximation approach

In this section, an AA based on the parametric function

$$\Theta(\tau,u,s) = \frac{1 + m_1\tau}{1 + m_2\tau\exp(-\frac{s}{\tau})}, \text{ with } \psi_G(\tau,u) := E\left[\Theta(\tau,u,g(u,\xi))\right], \qquad (4.4.9)$$

where $0 < \tau < \tau_{max}$, $m_1, m_2, \tau_{max} \in \mathbb{R}$ are given positive constants, is analyzed. Important properties of the function $\Theta(\tau,u,s)$ are given in the following proposition.

Proposition 4.4.10 (Geletu et. al. 2013 [34]). *Suppose $m_1, m_2, \tau_{max} \in \mathbb{R}$, $0 < \tau \leq \tau_{max}$, $m_2 < \frac{m_1}{1+m_1\tau_{max}}$ and $\Theta(\tau,u,s)$ defined as above. Then the following holds:*

(i) $\Theta(\tau, u, s) > 0$ *for all* $s \in \mathbb{R}$*,*

(ii) $\Theta(\tau, u, s) > 1$ *for* $s \geq 0$*,*

(iii) $\Theta(\tau, u, \cdot) > 0$ *is a strictly increasing function w.r.t.* $s \in \mathbb{R}$*,*

(iv) $\Theta(\cdot, u, s) > 0$ *is a strictly increasing function w.r.t.* τ*,* $0 < \tau < \tau_{max}$*,*

(v) *and*

$$\lim_{\tau \to 0^+} \Theta(\tau, u, s) = \begin{cases} 1, & \textit{if } s \geq 0, \\ 0, & \textit{if } s < 0, \end{cases}$$

uniformly for $u \in \mathcal{U}$ *and for each* $\epsilon > 0$ *uniformly for* $s \in (-\infty, -\epsilon) \cup [0, \infty)$*.*

Proof:

(i) Trivial.

(ii) To verify this, observe that $m_2 < m_1$ as a direct consequence of the assumption $m_2 < \frac{m_1}{1+m_1\tau_{max}}$. Then,

$$m_2\tau \exp(-\frac{s}{\tau}) \leq m_2\tau < m_1\tau,$$

since $\exp(-\frac{s}{\tau}) \leq 1$ for $s \geq 0$. This implies

$$1 + m_2\tau \exp(-\frac{s}{\tau}) < 1 + m_1\tau \Rightarrow \frac{1 + m_1\tau}{1 + m_2\tau \exp(-\frac{s}{\tau})} > 1.$$

(iii) Calculating the derivative

$$\begin{aligned} \frac{\partial}{\partial s}\Theta(\tau, u, s) &= \frac{(1 + m_1\tau)(m_2 exp(-\frac{s}{\tau}))}{(1 + m_2\tau \exp(-\frac{s}{\tau}))^2} \\ &> 0 \end{aligned}$$

directly delivers the desired result.

(iv) The derivative $\frac{\partial}{\partial\tau}\Theta(\tau, u, s)$ is

$$\frac{\partial}{\partial\tau}\Theta(\tau, u, s) = \frac{m_1(1 + m_2\tau \exp(-\frac{s}{\tau})) - (1 + m_1\tau)(m_2 \exp(-\frac{s}{\tau}) + m_2\frac{s}{\tau} \exp(-\frac{s}{\tau}))}{(1 + m_2\tau \exp(-\frac{s}{\tau}))^2}$$

Since the denominator $(1 + m_2\tau \exp(-\frac{s}{\tau}))^2 > 1$, i.e., positive, only the numerator has to be further analyzed. Using the well known inequality $\exp(x) \geq 1 + x$ for all $x \in \mathbb{R}$ in the form $1 \geq (1 + \frac{s}{\tau})\exp(-\frac{s}{\tau})$ the following approximations can be made:

$$\begin{aligned}
&m_1(1 + m_2\tau \exp(-\frac{s}{\tau})) - (1 + m_1\tau)(m_2 \exp(-\frac{s}{\tau}) + m_2\frac{s}{\tau}\exp(-\frac{s}{\tau})) \\
&\quad > m_1 - m_2(1 + m_1\tau)(1 + \frac{s}{\tau})\exp(-\frac{s}{\tau}) && \left(m_2\tau\exp(-\frac{s}{\tau}) > 0\right) \\
&\quad \geq m_1 - m_2(1 + m_1\tau) && \left(1 \geq (1 + \frac{s}{\tau})\exp(-\frac{s}{\tau})\right) \\
&\quad \geq m_1 - m_2(1 + m_1\tau_{max}) && (\tau \leq \tau_{max}) \\
&\quad \geq 0 && \left(m_2 < \frac{m_1}{1 + m_1\tau_{max}}\right)
\end{aligned}$$

In summary, it follows that $\frac{\partial}{\partial\tau}\Theta(\tau, u, s) > 0$.

(v) This proof consists of two parts. First, assume $s \geq 0$. Then $\Theta(\tau, u, s) > 1$ and

$$\begin{aligned}
0 < \Theta(\tau, u, s) - 1 &= \frac{1 + m_1\tau}{1 + m_2\tau\exp(-\frac{s}{\tau})} - 1 \\
&< 1 + m_1\tau - 1 && \left(1 + m_2\tau\exp\left(-\frac{s}{\tau}\right) > 1\right) \\
&< m_1\tau.
\end{aligned}$$

Now assume $s < -\epsilon$ for an arbitrary $\epsilon > 0$. Then,

$$\begin{aligned}
0 < \Theta(\tau, u, s) &= \frac{1 + m_1\tau}{1 + m_2\tau\exp(-\frac{s}{\tau})} \\
&< \frac{1 + m_1\tau}{1 + m_2\tau\exp(\frac{\epsilon}{\tau})}.
\end{aligned}$$

This concludes the proof.

□

Using this proposition, we can now show that $\psi_G(\cdot, \cdot)$ has the desired properties P1–P3.

Corollary 4.4.11. *The analytic approximation* $\psi_G(\tau, u) = E\left[\Theta(\tau, u, g(u, \xi))\right]$ *has properties P1–P3.*

Proof: Take arbitrary $0 < \tau_1 < \tau_2 < \tau_{max}$. Then, by Proposition 4.4.10 1., 2. and 4., we have

$$\begin{aligned}
\psi_G(\tau_2, u) &= E\left[\Theta(\tau_2, u, g(u, \xi)\right] \\
&> E\left[\Theta(\tau_1, u, g(u, \xi)\right] = \psi_G(\tau_1, u) \\
&> E\left[\mathbb{1}(g(u, \xi))\right]
\end{aligned}$$

for all $u \in \mathcal{U}$, and, therefore, properties P1 and P3. To get P2, first observe that the uncertain variables ξ underlie a continuous distribution and that the function $g(\cdot, \cdot)$ is also continuous. Hence, for all $u \in \mathcal{U}$ it holds that $\lim_{\delta \to 0^+} Pr\left\{-\delta < g(u,\xi) < 0\right\} = 0$. Therefore, for every $u \in \mathcal{U}$ and $\epsilon > 0$ we find a $\delta(u) > 0$, such that $Pr\left\{-\delta(u) < g(u,\xi) < 0\right\} < \epsilon$. Observe that

$$\psi_G(\tau,u) - E\left[\mathbb{1}(g(u,\xi))\right] = \int_{g(u,\xi)\leq -\delta(u)} \Theta(\tau,u,g(u,\xi))\phi(\xi)d\xi$$
$$+ \underbrace{\int_{-\delta(u)<g(u,\xi)<0} \Theta(\tau,u,g(u,\xi))\phi(\xi)d\xi}_{\leq \int_{-\delta(u)<g(u,\xi)<0}(1+m_1\tau_{max})\phi(\xi)d\xi \leq (1+m_1\tau_{max})\epsilon} + \int_{g(u,\xi)\geq 0} \left(\Theta(\tau,u,g(u,\xi)) - 1\right)\phi(\xi)d\xi.$$

Due to Proposition 4.4.10 5. we can find a $\hat{\tau} > 0$ such that $\Theta(\hat{\tau},u,g(u,\xi)) - \mathbb{1}(g(u,\xi)) \leq \epsilon$ for all $u \in \mathcal{U}$ and for $g(u,\xi) \leq -\delta(u)$ or $g(u,\xi) \geq 0$. It follows that

$$\psi_G(\tau,u) - E\left[\mathbb{1}(g(u,\xi))\right] \leq (2 + m_1\tau_{max})\epsilon$$

for all $\tau < \hat{\tau}$, i.e., $\lim_{\tau\to 0} \psi_G(\tau,u) - E\left[\mathbb{1}g(u,\xi)\right] = 0$. This implies property P2. □

By Corollary 4.4.11 we have $\lim_{\tau\to 0} \psi_G(\tau,u) = E\left[\mathbb{1}g(u,\xi)\right]$, i.e., the approximation $\psi(\tau,u)$ converges point-wise towards $E\left[\mathbb{1}g(u,\xi)\right]$ as $\tau \to 0$. As a next step, we will show that the gradients $\nabla_u\psi_G(\tau,u)$ converge uniformly for $\tau \to 0$, since this implies that $\nabla_u\psi(\tau,u) \to \nabla_u E\left[\mathbb{1}g(u,\xi)\right]$. In order to do this, observe that $\Theta(\tau,u,g(u,\xi))\phi(\xi)$ is continuous w.r.t. all variables and the partial derivatives

$$\nabla_u\Theta(\tau,u,g(u,\xi))\phi(\xi) = \frac{(1+m_1\tau)m_2\exp(-\tau^{-1}g(u,\xi))}{\left(1+m_2\tau\exp(-\tau^{-1}g(u,\xi))\right)^2}\nabla_u g(u,\xi)\phi(\xi)$$

exist and are also continuous w.r.t. all variables (due to the differentiability of $g(u,\xi)$). Together with the next assumption this allows to exchange differentiation and integration.

Assumption 4.4.12. For the remainder of this section, we assume that the support Ω of the pdf ϕ is a compact subset of $\mathbb{R}^p$.

Using this, we get

$$\nabla_u\psi_G(\tau,u) = \int_\Omega \nabla_u\Theta(\tau,u,g(u,\xi))\phi(\xi)d\xi$$

The next results determine properties of $\nabla_u\psi_G(\tau,u)$.

Lemma 4.4.13. *Let* $B_\epsilon(u) = \{\xi \in \Omega | \; |g(u,\xi)| < \epsilon\}$. *Then,*

$$\lim_{\tau\to 0^+} \sup_{u\in\mathcal{U}} \left| \int_{\Omega\setminus B_\epsilon(u)} \nabla_u\Theta(\tau,u,g(u,\xi))\phi(\xi)d\xi \right| = 0.$$

Proof: Since Ω and $\mathcal{U}$ are compact and g is continuous we have a constant $\gamma > 0$ such that $\sup_{u\in\mathcal{U},\xi\in\Omega} \|\nabla_u g(u,\xi)\| = \gamma < \infty$. Hence, we get

$$\begin{aligned}
&\sup_{u\in\mathcal{U}} \left| \int_{\Omega\setminus B_\epsilon(u)} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi \right| \\
&= \sup_{u\in\mathcal{U}} \left| \int_{\Omega\setminus B_\epsilon(u)} \frac{(1+m_1\tau)m_2\exp(-\tau^{-1}g(u,\xi))}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))^2} \nabla_u g(u,\xi)\phi(\xi)d\xi \right| \\
&\leq \sup_{u\in\mathcal{U}} \int_{\Omega\setminus B_\epsilon(u)} \frac{(1+m_1\tau)m_2\exp(-\tau^{-1}g(u,\xi))}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))^2} \|\nabla_u g(u,\xi)\| \phi(\xi)d\xi \\
&\leq \sup_{u\in\mathcal{U}} \int_{\Omega\setminus B_\epsilon(u)} \frac{\gamma(1+m_1\tau)}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))\left(m_2^{-1}\exp(\tau^{-1}g(u,\xi))+\tau\right)}\phi(\xi)d\xi \\
&= \sup_{u\in\mathcal{U}} \left[\int_{g(u,\xi)<-\epsilon} \frac{\gamma(1+m_1\tau)}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))\left(m_2^{-1}\exp(\tau^{-1}g(u,\xi))+\tau\right)}\phi(\xi)d\xi \right. \\
&\qquad \left. + \int_{g(u,\xi)>\epsilon} \frac{\gamma(1+m_1\tau)}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))\left(m_2^{-1}\exp(\tau^{-1}g(u,\xi))+\tau\right)}\phi(\xi)d\xi \right] \\
&\leq \sup_{u\in\mathcal{U}} \left[\int_{g(u,\xi)<-\epsilon} \frac{\gamma(1+m_1\tau)}{(1+m_2\exp(\tau^{-1}\epsilon))\tau}\phi(\xi)d\xi + \int_{g(u,\xi)>\epsilon} \frac{\gamma(1+m_1\tau)}{m_2^{-1}\exp(\tau^{-1}\epsilon)+\tau}\phi(\xi)d\xi \right] \\
&\leq \sup_{u\in\mathcal{U}} \left[\frac{\gamma(1+m_1\tau)}{(1+m_2\exp(\tau^{-1}\epsilon))\tau} + \frac{\gamma(1+m_1\tau)}{m_2^{-1}\exp(\tau^{-1}\epsilon)+\tau} \right].
\end{aligned}$$

As an immediate consequence, we obtain

$$\lim_{\tau\to 0^+} \sup_{u\in\mathcal{U}} \left| \int_{\Omega\setminus B_\epsilon(u)} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi \right| = 0.$$

□

Assumption 4.4.14. Let $\Omega_g(u) = \{\xi \in \Omega |\ g(u,\xi) = 0\}$. For all $u \in \mathcal{U}$ and all $\xi \in \Omega_g(u)$ it holds that$\nabla_\xi g(u,\xi) \neq 0$.

Lemma 4.4.15. *The limit*

$$\lim_{\tau\to 0^+} \nabla_u \psi_G(\tau, u)$$

exists and the convergence is uniform for $u \in \mathcal{U}$.

Proof: For this proof we consider the transformation $t = g(u,\xi)$ on the set $\mathcal{U}\times\Omega$. Assume that there is a point $(\tilde{u},\tilde{\xi}) \in \mathcal{U}\times\Omega$ such that $g(\tilde{u},\tilde{\xi}) = 0$. Otherwise, we can find an $\epsilon > 0$ with $|g(u,\xi)| > \epsilon$ for all $(u,\xi) \in \mathcal{U}\times\Omega$ and by the previous lemma $\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}} \nabla_u\psi_G(\tau,u) = 0$. Let

$$\Gamma_g := \{(u,\xi,t)|\ t = g(u,\xi) = 0, (u,\xi) \in \mathcal{U}\times\Omega\}.$$

For each $(\tilde{u},\tilde{\xi},0)\in\Gamma_g$, we find an index $\beta\in\{1,\ldots,p\}$ with $\frac{\partial}{\partial\xi_\beta}g(u,\xi)\neq 0$ since $\nabla_\xi g(u,\xi)\neq 0$ by assumption. Furthermore, set $\tilde{\eta}_j=\tilde{\xi}_j$ for $j\in\{1,\ldots,p\}\setminus\{\beta\}$. Due to the implicit function theorem, we find open l^∞-balls $\tilde{U}$, $\tilde{V}$, $(-\hat{t},\hat{t})$, and (a,b) around $\tilde{u}$, $\tilde{\eta}_j$, $\tilde{\xi}_\beta$, and 0 and a unique function $q:\tilde{U}\times\tilde{V}\times(-\hat{t},\hat{t})\to(a,b)$ with $q(\tilde{u},\tilde{\eta},0)=\tilde{\xi}_\beta$.

As a next step, we introduce the coordinate transformation

$$\xi_j=\begin{cases}\eta_j & j\neq\beta\\ q(u,\eta,t) & j=\beta\end{cases}\tag{4.4.10}$$

for $j=1,\ldots,p$. By construction, we have

$$\begin{aligned}
&\frac{\partial\xi_j}{\partial\eta_i}=\delta_{ij},\ i,j\in\{1,\ldots,p\}\setminus\{\beta\}\\
&\frac{\partial\xi_j}{\partial t}=0,\ j\in\{1,\ldots,p\}\setminus\{\beta\}\\
&\frac{\partial\xi_\beta}{\partial\eta_i}=\frac{\partial q(u,\eta)}{\partial\eta_i},\ i\in\{1,\ldots,p\}\setminus\{\beta\}\\
&\frac{\partial\xi_\beta}{\partial\eta_t}=\frac{\partial q(u,\eta)}{\partial t}=\left(\left.\frac{\partial g(u,\xi)}{\partial\xi_\beta}\right|_{\xi_\beta=q(u,\eta,t)}\right)^{-1}\neq 0
\end{aligned}$$

The last non-vanishing property holds for sufficiently small neighborhoods $\tilde{U}\times\tilde{V}\times(-\hat{t},\hat{t})$ due to the continuity of g. Therefore, the functional determinant

$$\Delta(u,\eta,t):=\left|\frac{\partial\xi}{\partial(\eta,t)}\right|=\left|\frac{\partial\xi_\beta}{\partial t}\right|=\left|\left(\frac{\partial g}{\partial\xi_\beta}\right)^{-1}\right|>0.$$

Hence, for each $u\in\tilde{U}$ we have a one-to-one C^1-mapping from $\left\{\xi\in\tilde{V}\times(a,b)\middle|\ -\hat{t}<g(u,\xi)<\hat{t}\right\}$ to the set $\tilde{V}\times(-\hat{t},\hat{t})$, which does not depend on u.

The open sets

$$\left\{\tilde{U}\times\left(\tilde{V}\times(a,b)\right)\times(-\hat{t},\hat{t})\right\}_{(\tilde{u},\tilde{\xi},0)\in\Gamma_g}$$

create an infinite open covering of the compact set Γ_g, therefore, we can find a finite open subcovering

$$\{U_i\times(V_i\times(a_i,b_i))\times(-t_i,t_i)\}_{i=1,\ldots,z}$$

Now, choose an $\epsilon>0$ such that $\epsilon<t_i$ for $i=1,\ldots,z$ and each $(u,\xi)\in\mathcal{U}\times\Omega$ with $|g(u,\xi)|<\epsilon$ belongs to the covering. This is possible, since $\|\nabla g(u,\xi)\|\geq\|\nabla_\xi g(u,\xi)\|\geq C>0$ for a constant $C\in\mathbb{R}_{++}$ and for all $(u,\xi,0)\in\Gamma_g$. For

$$N:=\bigcup_{i=1,\ldots,z}U_i\times(V_i\times(a_i,b_i))$$

and its corresponding open covering exists a smooth partition of the unity $\{\mu_i\}_{i=1,\dots,z}$. Now, we can use the partition of the unity and the parameter transformations constructed above to analyze the integral

$$\int_{B_\epsilon(u)} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi = \int_{B_\epsilon(u)} \underbrace{\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}g(u,\xi))}{(1+m_2\tau\exp(-\tau^{-1}g(u,\xi)))^2}\nabla_u g(u,\xi)\phi(\xi)}_{:=Q(g(u,\xi),\tau)} d\xi \tag{4.4.11}$$

$$= \sum_{i=1}^{z} \int_{V_i\times(a_i,b_i),|g(u,\xi)|<\epsilon} Q(g(u,\xi),\tau)\mu_i(u,\xi)d\xi \tag{4.4.12}$$

$$= \sum_{i=1}^{z} \int_{V_i}\int_{-\epsilon}^{\epsilon} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta. \tag{4.4.13}$$

Regardless of the coordinate used for the transformation (which might be different for each summand) we use (η, t) for the new variables and V_i for the set of vectors η. In the same sense we understand the substitution $\xi = (\eta, q(u,\eta,t))$. Continuing, we obtain

$$\begin{aligned}
&\sum_{i=1}^{z} \int_{V_i}\int_{-\epsilon}^{\epsilon} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))dtd\eta \\
&= \sum_{i=1}^{z} \int_{V_i}\int_{-t_i}^{t_i} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta \\
&\quad - \sum_{i=1}^{z} \int_{V_i}\int_{-t_i}^{-\epsilon} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta \\
&\quad - \sum_{i=1}^{z} \int_{V_i}\int_{\epsilon}^{t_i} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta.
\end{aligned}$$

By the previous lemma, the second and third summand tend uniformly towards zero for $\tau \to 0^+$, i.e.,

$$\begin{aligned}
&\lim_{\tau\to 0^+} \sup_{u\in\mathcal{U}} \nabla_u \psi_G(\tau, u) \\
&= \lim_{\tau\to 0^+} \sup_{u\in\mathcal{U}} \sum_{i=1}^{z} \int\limits_{V_i} \int\limits_{-t_i}^{t_i} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta.
\end{aligned}$$

Define $\lambda(u,\eta,t) := \nabla_u g(u,(\eta,q_i(u,\eta,t)))\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)$ and $\hat{\phi}(u,\eta,t) := \phi(\eta,q_i(u,\eta,t))$.

Then

$$
\begin{aligned}
&\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{V_i}\int\limits_{-t_i}^{t_i} Q(t,\tau)\mu_i(u,\eta,q_i(u,\eta,t))\Delta(u,\eta,t)dtd\eta\\
&=\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{V_i}\int\limits_{-t_i}^{t_i}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}\lambda(u,\eta,t)\hat{\phi}(u,\eta,t)dtd\eta\\
&=\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{\tau\log(\tau^2 m_2)}^{\tau\log\left(\frac{m_2}{\tau}\right)}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}\int\limits_{V_i}\lambda(u,\eta,t)\hat{\phi}(u,\eta,t)dtd\eta\\
&+\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{\tau\log\left(\frac{m_2}{\tau}\right)}^{t_i}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}\int\limits_{V_i}\lambda(u,\eta,t)\hat{\phi}(u,\eta,t)dtd\eta\\
&+\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{-t_i}^{\tau\log(\tau^2 m_2)}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}\int\limits_{V_i}\lambda(u,\eta,t)\hat{\phi}(u,\eta,t)dtd\eta
\end{aligned}
$$

Using the mean value theorem for integrals and the assumption that ϕ is continuous for all $\xi\in\Omega$, the above translates to

$$
\begin{aligned}
&\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{\tau\log(\tau^2 m_2)}^{\tau\log\left(\frac{m_2}{\tau}\right)}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}dt\int\limits_{V_i}\lambda(u,\eta,t_1(\tau))\hat{\phi}(u,\eta,t_i^1(\tau))d\eta\\
&+\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{\tau\log\left(\frac{m_2}{\tau}\right)}^{t_i}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}dt\int\limits_{V_i}\lambda(u,\eta,t_i^2(\tau))\hat{\phi}(u,\eta,t_i^2(\tau))d\eta\\
&+\lim_{\tau\to 0^+}\sup_{u\in\mathcal{U}}\sum_{i=1}^{z}\int\limits_{-t_i}^{\tau\log(\tau^2 m_2)}\frac{(1+m_1\tau)m_2\exp(-\tau^{-1}t)}{\left(1+m_2\tau\exp(-\tau^{-1}t)\right)^2}dt\int\limits_{V_i}\lambda(u,\eta,t_i^3(\tau))\hat{\phi}(u,\eta,t_i^3(\tau))d\eta,
\end{aligned}
$$

where $\tau\log(\tau^2 m_2) < t_i^1 < \tau\log\left(\frac{m_2}{\tau}\right)$, $\tau\log\left(\frac{m_2}{\tau}\right) < t_i^2 < t_i$, and $-t_i < t_i^3 < \tau\log(\tau^2 m_2)$ for

$i = 1, \ldots, z$. A short calculation reveals

$$\int\limits_{\tau \log(\tau^2 m_2)}^{\tau \log\left(\frac{m_2}{\tau}\right)} \frac{(1+m_1\tau)m_2 \exp(-\tau^{-1}t)}{(1+m_2\tau\exp(-\tau^{-1}t))^2} dt = \frac{1+m_1\tau}{1+\tau^2} - \frac{\tau(1+m_1\tau)}{1+\tau} \longrightarrow_{\tau\to 0^+} 1$$

$$\int\limits_{\tau \log\left(\frac{m_2}{\tau}\right)}^{t_1} \frac{(1+m_1\tau)m_2 \exp(-\tau^{-1}t)}{(1+m_2\tau\exp(-\tau^{-1}t))^2} dt = \frac{1+m_1\tau}{1+m_2\tau\exp(-\frac{t_1}{\tau})} - \frac{1+m_1\tau}{1+\tau^2} \longrightarrow_{\tau\to 0^+} 0$$

$$\int\limits_{-t_1}^{\tau \log(\tau^2 m_2)} \frac{(1+m_1\tau)m_2 \exp(-\tau^{-1}t)}{(1+m_2\tau\exp(-\tau^{-1}t))^2} dt = \frac{\tau(1+m_1\tau)}{1+\tau} - \frac{1+m_1\tau}{1+m_2\tau\exp(\frac{t_1}{\tau})} \longrightarrow_{\tau\to 0^+} 0.$$

Furthermore, for the second and third summand it holds that the second factor is uniformly bounded for $u \in \mathcal{U}$, together with the calculation above this indicates that the second and third summand go uniformly towards zero as $\tau \to 0^+$. Additionally, the first factors of the first summand goes to one independent of u, whereas the second factors converges uniformly towards some integrals

$$J_i := \int_{V_i} \lambda(u, \eta, 0)\hat{\phi}(u, \eta, 0) d\eta$$

and, finally,

$$\lim_{\tau\to 0^+} \nabla_u \psi_G(\tau, u) = \sum_{i=1}^{z} J_i$$

uniformly for $u \in \mathcal{U}$. □

From the previous lemma, by standard arguments of analysis, immediately follows the following statement.

Theorem 4.4.16. *The function $\psi_G(\tau, u)$ is continuously differentiable and it holds that*

$$\lim_{\tau\to 0^+} \sup_{u\in\mathcal{U}} \|\nabla_u \psi_G(\tau, u) - \nabla_u E\left[\mathbb{1}(g(u, \xi))\right]\| = 0.$$

The method employed to show the uniform convergence of the derivatives can be seen as a generalization of the projection approach presented in Section 4.3. Whereas the original approach relies on the existence of a global monotonic relation between $g(u, \xi)$ and one of the ξ_j, the proof given above requires the existence of such relations only on a finite number of sets. Furthermore, the uncertain variable which induces monotonicity may change between this sets. The transformation (4.4.10) can be seen as an inverse back projection, i.e., a forward projection from the domain Ω into $\tilde{V} \times (-\hat{t}, \hat{t})$. Consequently, like in the case of back-projection methods, the derivatives (in the limit) can be found by integrating over a subset of $\mathbb{R}^{p-1}$.

The proof presented above can be naturally extended to cover also higher order derivatives. Under the assumptions made here and in Chapter 3 the convergence of the Hessian can be shown, giving rise to the following corollary.

Corollary 4.4.17. *The limit*

$$\lim_{\tau \to 0^+} \nabla_u^2 \psi_G(\tau, u)$$

exists and the convergence is uniform for $u \in \mathcal{U}$*.*

Proof: We start with

$$\begin{aligned}
&\nabla_u^2 \psi_G(\tau, u) \\
=& \int_{B_\epsilon(u)} \nabla_u^2 \Theta(\tau, u, g(u,\xi))\phi(\xi) d\xi \\
=& \int_{B_\epsilon(u)} \left(\nabla_s^2 \Theta(\tau, u, s)\big|_{s=g(u,\xi)} \nabla_u g(u,\xi) \nabla_u g(u,\xi)^T + \nabla_s \Theta(\tau, u, s)|_{s=g(u,\xi)} \nabla_u^2 g(u,\xi) \right) \phi(\xi) d\xi,
\end{aligned}$$

which can be directly obtained by calculating the derivative of (4.4.11) with respect to u. Using a partition of the unity and the transformation constructed in the proof of Lemma (4.4.15), we get

$$\begin{aligned}
&\nabla_u^2 \psi_G(\tau, u) \\
=& \sum_{i=1}^z \int_{V_i} \int_{-\epsilon}^{\epsilon} \nabla_t^2 \Theta(\tau, u, t) \nabla_u \hat{g}(u,\eta,t) \nabla_u \hat{g}(u,\eta,t)^T \hat{\phi}(u,\eta,t) \mu_i(u,\eta,q_i(u,\eta,t)) \Delta(u,\eta,t) dt d\eta + \\
&\ldots + \int_{V_i} \int_{-\epsilon}^{\epsilon} \nabla_t \Theta(\tau, u, t) \nabla_u^2 \hat{g}(u,\eta,t) \hat{\phi}(u,\eta,t) \mu_i(u,\eta,q_i(u,\eta,t)) \Delta(u,\eta,t) dt d\eta \\
=& \sum_{i=1}^z \int_{V_i} \int_{-\epsilon}^{\epsilon} \nabla_t \Theta(\tau, u, t) \underbrace{\nabla_t \left(\nabla_u \hat{g}(u,\eta,t) \nabla_u \hat{g}(u,\eta,t)^T \hat{\phi}(u,\eta,t) \mu_i(u,\eta,q_i(u,\eta,t)) \Delta(u,\eta,t) \right)}_{:=Q_1(u,\eta,t)} dt d\eta + \\
&\ldots + \int_{V-i} \underbrace{\left(\nabla_t \Theta(\tau, u, t) \nabla_u \hat{g}(u,\eta,t) \nabla_u \hat{g}(u,\eta,t)^T \hat{\phi}(u,\eta,t) \mu_i(u,\eta,q_i(u,\eta,t)) \Delta(u,\eta,t) \right) \Big|_{t=-\epsilon}^{\epsilon}}_{:=Q_2(u,\eta)} d\eta \\
&\ldots + \int_{V_i} \int_{-\epsilon}^{\epsilon} \nabla_t \Theta(\tau, u, t) \underbrace{\nabla_u^2 \hat{g}(u,\eta,t) \hat{\phi}(u,\eta,t) \mu_i(u,\eta,q_i(u,\eta,t)) \Delta(u,\eta,t)}_{:=Q_3(u,\eta,t)} dt d\eta,
\end{aligned}$$

where $\hat{\phi}(u,\eta,t)$ is defined as in the proof of Lemma 4.4.15 and $\hat{g}(u,\eta,t) := g(u,\eta,q_i(u,\eta,t))$. The last equation holds by integration by parts. Due to the assumptions made, Q_1 and Q_3 are at least continuous and, therefore, bounded for $u \in \mathcal{U}$ and $\xi \in \Omega$. This allows to continue in the same way as in the proof of Lemma 4.4.15. By doing this, we obtain

$$H_i := \int_{V_i} Q_1(u,\eta,0) + Q_2(u,\eta) + Q_3(u,\eta,0) d\eta$$

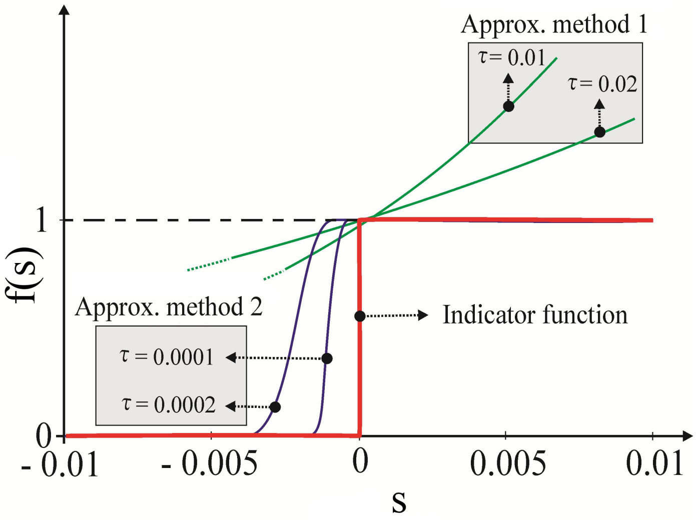

Figure 4.1: *Comparison between the functions* $\exp(\tau^{-1}s)$ *(method 1) employed by Nemirovski and Shapiro and the function* $\Theta(\tau, u, s)$ *(method 2) employed by Geletu et al.*

and, finally,

$$\lim_{\tau \to 0} \nabla_u^2 \psi_G(\tau, u) = \sum_{i=1}^{z} H_i$$

uniformly for $u \in \mathcal{U}$. □

Remark 4.4.18. Under the assumption of sufficient smooth functions g and ϕ, the convergence of even higher derivatives can be shown in the same way, by repeatedly applying integration by parts.

Remark 4.4.19. The proofs above hold only for uncertainties, where the corresponding pdf is continuous (and continuously differentiable) on the compact set Ω. One should be aware that this property is not guaranteed for all common distributions, e.g., for Beta distributed uncertainties with parameters $\alpha, \beta < 1$ the pdf is only continuous in the interior of Ω. In application it is, therefore, necessary to check whether such cases occur.

Remark 4.4.20. It is possible to define a generalized version of $\psi_G(\tau, u)$, where the parameters m_1 and m_2 are functions of the controls u, for details see [34]. Since this yields no immediate gain for practical applications it was not considered in this work.

Figure 4.1 shows a comparison between the approximations methods proposed by Nemirovski and Shapiro as well as Geletu et al. It can be clearly seen that the approximation of the step function $\mathbb{1}(g(u, \xi))$ with $\Theta(\tau, u, s)$ improves as τ decreases for the second method (Geletu et al.), whereas the same is not true for the first method (Nemirovski/Shapiro).

4.4.3 Implementation

Approach of Nemirovski and Shapiro ($\psi_{NS}(\tau, u)$)

All in all, the approach proposed by Nemirovski and Shapiro is rather straight-forward to implement. The approximations $\psi_{NS}(\tau, u)$ can be evaluated using (Q)MC or sparse grid integration methods. The derivatives $\nabla_u \psi_{NS}(\tau, u)$ can be evaluated by

$$\begin{aligned}\nabla_u \psi_{NS}(\tau, u) &= \int_\Omega \nabla_u \exp(\tau^{-1} g(u, \xi))\phi(\xi) d\xi \\ &= \int_\Omega \exp(\tau^{-1} g(u, \xi))\tau^{-1} \nabla_u g(u, \xi)\phi(\xi) d\xi,\end{aligned}$$

again using (Q)MC or sparse grid methods. The choice of the parameter τ poses the biggest difficulty, since for general problems either one chooses a fixed τ, probably leading to a more conservative approximation, or one has to solve a two-stage optimization problem where in the lower stage an optimal value of τ has to be determined.

Approach of Geletu et al. ($\psi_G(\tau, u)$)

In comparison to $\psi_{NS}(\tau, u)$ the implementation of $\psi_G(\tau, u)$ is more involved. This begins to show in the computation of $\psi_G(\tau, u)$, since (Q)MC methods can be successfully employed, but sparse grid rules are no longer an option. The reason for this is that for $\tau \to 0^+$ the integrand $\Theta(\tau, u, s)$ becomes rather steep, which leads to large errors in sparse grid integration rules. Although it is clear from the theory that one should solve a sequence of optimization problems (NLP_{τ_k}), $(\tau_k)_{k\in\mathbb{N}}$ with $\tau_k \to 0^+$ one should keep in mind that the integration rule has to be adapted to τ_k as $k \to \infty$ (i.e., more grid points) in order to avoid that the integration error actually invalidates the approximation property. For practical purposes it might be useful to determine a $\tau_{min} > 0$, subject to the condition that the problem $(NLP_{\tau_{min}})$ has a non-empty feasible set and solve a sequence of problems with $\tau_k \to \tau_{min}$.

Another challenge when implementing $\psi_G(\tau, u)$ is the evaluation of derivatives. Although the proof of Lemma 4.4.15 indicates a method for the evaluation, it is difficult to implement. There are several reasons for this, including the difficulty of determining the open coverings and the partition of the unity. For τ sufficiently large, finite differences can be used to approximate the derivatives, e.g., central differences

$$\frac{\partial}{\partial u_i} \psi_G(\tau, u) \approx \frac{\psi_G(\tau, u + \Delta u) - \psi_G(\tau, u - \Delta u)}{2\Delta u}.$$

In order to minimize the influence of integration errors on the derivatives Δu has to be chosen sufficiently large. A second approach is the direct evaluation of

$$\nabla_u \psi_G(\tau, u) = \int_\Omega \nabla_u \Theta(\tau, u, g(u, \xi))\phi(\xi) d\xi$$

using cubature methods. But due to Lemma 4.4.13 we have

$$\int_{|g(,u\xi)|>\epsilon} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi = 0,$$

i.e., a very large number of grid points would be necessary to sufficiently determine and integrate the set $B_\epsilon(u)$, especially for small values of τ. To overcome this difficulty, here a new branch-and-bound algorithm is proposed. The idea is to determine a finite number N of disjoint subsets Ω_i of Ω such that for given $u \in \mathcal{U}$ and for each $\xi \in \Omega$ with $g(u,\xi) = 0$ there exists an index j, such that $\xi \in \Omega_j$. The derivatives are evaluated by

$$\nabla_u \psi_G(\tau, u) = \sum_{i=1}^{N} \int_{\Omega_i} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi,$$

thereby eliminating subsets of Ω, which do not contribute to the integral. In order to find such sets, we have to be able to efficiently determine, whether $0 \in g(u, \tilde{\Omega})$ holds for a given subset $\tilde{\Omega} \subset \Omega$: . This can be done using interval or affine arithmetic (see Chapter 2). To use these methods, we need an explicit description of $g(u,\xi)$. If such description is not available, we can construct an explicit approximation using for instance (generalized) Fourier series or Artificial Neural Network (ANN). These methods were shown to be uniformly convergent for a wide class of functions (see Chapter 6), i.e., for every $\epsilon > 0$ we can find an approximation $\tilde{g}(u,\xi)$, such that

$$\|\tilde{g}(u,\xi) - g(u,\xi)\| < \epsilon, \text{ for all } \xi \in \Omega.$$

Using this approximation we can now formulate the following algorithm.

Algorithm 4.4.21. Let $\Omega = [\xi_1^{min}, \xi_1^{max}] \times \ldots \times \left[\xi_p^{min}, \xi_p^{max}\right]$ be the Cartesian product of compact intervals $[\xi_i^{min}, \xi_i^{max}]$, $i = 1, \ldots, p$, let $\tilde{g}(u,\xi)$ be an approximation of $g(u,\xi)$ with $\|\tilde{g}(u,\xi) - g(u,\xi)\| < \epsilon$ for a given $\epsilon > 0$, and $\delta > 0$ a parameter. Set $A = \{\Omega\}$.

(i) Choose $\tilde{\Omega} \in A$ with $Pr\left\{\xi \in \tilde{\Omega}\right\} = \max_{\hat{\Omega} \in A} Pr\left\{\xi \in \hat{\Omega}\right\}$ and set $A := A \backslash \left\{\tilde{\Omega}\right\}$.

(ii) Test, if $0 \in \tilde{g}(u, \tilde{\Omega}) + (-\epsilon, \epsilon)$ using interval or affine arithmetic. If this is the case, dissect $\tilde{\Omega}$ into 2^p hypercubes $\tilde{\Omega}_i$, $i = 1, \ldots, 2^p$ with $Pr\left\{\xi \in \tilde{\Omega}_1\right\} = \ldots = Pr\left\{\xi \in \tilde{\Omega}_{2^p}\right\}$ and set $A := A \cup \bigcup_{i=1}^{2^p} \tilde{\Omega}_i$.

(iii) If A is empty return $\nabla_u \psi_G(\tau, u) = 0$. If $Pr\left\{\xi \in \tilde{\Omega}\right\} < \delta$ for all $\tilde{\Omega}$ in A go to (iv), else go to (i).

(iv) Return $\nabla_u \psi_G(\tau, u) = \sum_{\tilde{\Omega} \in A} \int_{\tilde{\Omega}} \nabla_u \Theta(\tau, u, g(u,\xi))\phi(\xi)d\xi$.

Proof: We first consider the termination and then the correctness of the algorithm. If A is empty in step (iii) the algorithms terminates, therefore, assume that A is non-empty in this step. Then, for all $\tilde{\Omega} \in A$ it holds that $Pr\left\{\xi \in \tilde{\Omega}\right\} \leq \frac{1}{2^p}$ as a result of step (ii). After at maximum 2^p iterations of the algorithm we have $Pr\left\{\xi \in \tilde{\Omega}\right\} < \left(\frac{1}{2^p}\right)^2$ for all $\tilde{\Omega} \in A$, since by then every $\tilde{\Omega} \in A$ with $\left(\frac{1}{2^p}\right)^2 < Pr\left\{\xi \in \tilde{\Omega}\right\} \leq \frac{1}{2^p}$ would have been chosen in step (i) and be either discarded or dissected in step (ii). By a similar argument we have $Pr\left\{\xi \in \tilde{\Omega}\right\} < \left(\frac{1}{2^p}\right)^3$ for all $\tilde{\Omega} \in A$ after a maximum of $2^p + (2^p)^2$ iterations. Generalizing this notion we have $Pr\left\{\xi \in \tilde{\Omega}\right\} < \left(\frac{1}{2^p}\right)^k$ for all $\tilde{\Omega} \in A$ after $\sum_{i=1}^{k}(2^p)^i$ iterations, i.e., with $\tilde{k} = \left\lceil -\frac{\log \delta}{p \log 2} \right\rceil$ we get $Pr\left\{\xi \in \tilde{\Omega}\right\} < \delta$ for all $\tilde{\Omega} \in A$ after a maximum of $\sum_{i=1}^{\tilde{k}}(2^p)^i$ iterations. The algorithm then terminates in step (iv).

Now consider the correctness. In step (i), A is always non-empty since either $A = \{\Omega\}$ in the first iteration or the algorithms would have terminated in the third step with an empty set A. Consider $\xi \in \Omega$ with $g(u, \xi) = 0$. By step (ii) we find $0 \in \tilde{g}(u, \tilde{\Omega}) + (-\epsilon, \epsilon)$ and, therefore, $\xi \in \tilde{\Omega}_j$ for some $j \in \{1, \ldots, 2^p\}$. As a consequence we have

$$\{\xi \in \Omega |\ g(u, \xi) = 0\} \subset \bigcup_{\tilde{\Omega} \in A} \tilde{\Omega},$$

which concludes the proof. □

Remark 4.4.22. If the multivariate pdf can be decomposed as in 5.2.2 and cdf and inverse cdf are available (e.g., for standard distributions) then the dissection in step (ii) can be carried in a straight forward fashion.

In the worst case, no set is ever discarded in step (ii). Then, after $\sum_{i=1}^{\tilde{k}}(2^p)^i$ iterations, A contains $(2^p)^{\tilde{k}+1}$ sets. It limits the presented approach to smaller values of the dimension p. In practical applications this usually means that p should not exceed 12–15.

5 Numerical Integration

Purpose of this chapter is to give an overview over integration rules which are suitable for the application in CCOPT. We will also shortly discuss why certain well known rules are not useful and give some ideas for future work. First, one dimensional integration (or quadrature rules) will be treated. Based on these, we will then come to multivariate integration (or cubature) rules. A section on sampling algorithms will conclude this chapter.

5.1 Univariate integration (quadrature)

Quadrature is concerned with the numerical evaluations of integrals of the form

$$\int_a^b f(\xi)w(\xi)d\xi,$$

where $-\infty \leq a < b \leq \infty$, $f : [a,b] \to \mathbb{R}$ is at least piecewise continuous with a finite number of singularities, $w : [a,b] \to \mathbb{R}$ is a piecewise continuous weight function with $w(\xi) > 0$ for $\xi \in (a,b)$. Furthermore, we require that $\int_a^b w(\xi)d\xi < \infty$. This does not lead to restrictions in the course of this work, since pdf will act as weight functions and therefore $\int_a^b w(\xi)d\xi = 1$ if supp $w \subset (a,b)$.

Generally, an quadrature rule $I[\cdot]$ is described by N integration nodes ξ_i, $i = 1,\ldots,N$ and corresponding weights ω_i, which may depend on the weight function $w(\cdot)$ and the integral bounds, but not on the function $f(\cdot)$ to be integrated. Numerically computing the integral consists of evaluating the sum

$$I[f] = \sum_{i=1}^{N} \omega_i f(\xi).$$

A helpful tool in comparing integration routines is the concept of polynomial accuracy:

Definition 5.1.1 (Polynomial accuracy)**.** An integration rule $I[f]$ is said to have a polynomial accuracy of n, if all polynomials up to order n, denoted by Π_n are integrated exactly, i.e.,

$$\int_a^b w(\xi)f(\xi)d\xi - I[f] = 0, \quad f \in \Pi_n.$$

5.1.1 Newton-Cotes quadrature

As a first step, we shortly review the Newton-Cotes rules without going into detail, since these kind of rules are generally not suitable for application in CCOPT. Nonetheless, they are useful for getting to know the basic ideas of numerical integration and are also quite common in numerical libraries. Furthermore, will we use them to examine what are useful/suitable properties of an integration rule in the context of CCOPT. An in depth introduction to these kind of integration rules (upon which the following summary is based) can be found in [83], p. 126 ff.

Newton-Cotes quadrature rules are based on a uniform partition of a bounded interval of integration $[a, b]$ and are commonly used with a constant weight function $w \equiv 1$. As a consequence, in this section only the computation of

$$\int_a^b f(\xi)d\xi \tag{5.1.1}$$

is considered. The results can easily be extended to the "weighted" case by setting

$$\hat{f}(\xi) = w(\xi)f(\xi)$$

and evaluating $\int_a^b \hat{f}(\xi)d\xi$. The single integration nodes (or points) are given by

$$\begin{aligned} h &= \frac{b-a}{N-1} \\ \xi_i &= a + h(i-1), \;\; i = 1, \dots, N, \end{aligned}$$

where $N \geq 2$ is the number of integration nodes. The integral (5.1.1) is then evaluated using

$$NC(N, a, b)\,[f] = h \sum_{i=1}^{N} \omega_i f(\xi_i). \tag{5.1.2}$$

Since $NC(N, a, b)$ depends on the parameter h it is useful to determine how this length influences the integration error. This gives rise to the notion of the *order* of a Newton-Cotes quadrature rule.

Definition 5.1.2 (Order of a Newton-Cotes quadrature rule)**.** A Newton-Cotes quadrature rule $NC(N, a, b)$ is said to of order l, if and only if, there exists a function K depending on a, b, and f but not on h or N, such that for all functions f in an appropriate function space

$$\left| \int_a^b f(\xi)d\xi - NC(N, a, b)[f] \right| \leq K(a, b, f)h^l.$$

There exist several approaches to determine the weights ω_i in (5.1.2) for a given number of grid points N. The main idea in all approaches is to interpolate the integrand $f(\cdot)$ with a polynomial either on the whole interval $[a, b]$ or on subintervals of equal length.

In the first case, the interpolating polynomial is of degree $N-1$ and the N weights ω_i can be chosen, such that all polynomials up to degree $N-1$ are integrated exactly. In general, weights can be generated this way for any number of N, but in practice a larger choice of this value leads to negative weights ω_i, which in turn lead to numerical difficulties due to cancellation. One common example of this class of rules is the well known trapezoidal integration rule

$$NC(2,a,b)[f] = \frac{h}{2}\left(f(a)+f(b)\right),$$

which appears when choosing $N=2$. The integration error in this case is given by

$$\int_a^b f(\xi)d\xi - NC(2,a,b) = \frac{h^3}{12} f^{(2)}(\tilde{\xi}),$$

for functions $f \in C^2[a,b]$, where $\tilde{\xi} \in (a,b)$, but unknown. As a direct consequence, the trapezoidal rule has polynomial accuracy one and is of order three. A list of similar rules and the corresponding integration errors can be found in [83], p. 128. Generally, it is difficult to obtain higher order rules (i.e., rules with an order higher than eight), due to the problem of negative weights occurring for larger numbers of N.

Another approach to the determination of the weights ω_i is to apply the integration rules obtained in the first case to subintervals of $[a,b]$, e.g., given integration nodes $\xi_0,\ldots,\xi_{N-1}$ using the trapezoidal rule on $[\xi_i,\xi_{i+1}]$, $i=0,\ldots,N-2$ and summing up the results to obtain an (composite) integration rule of the form

$$NC_{comp}(N,a,b)[f] = h\left[\frac{f(a)}{2}+\frac{f(b)}{2}+\sum_{i=1}^{N-2} f(\xi_i)\right].$$

The integration error of this rule is bound by

$$\frac{b-a}{12}h^2 f^{(2)}(\xi)$$

for a $\xi \in [a,b]$, i.e., this rule is of order two and has polynomial accuracy one. In contrast to the trapezoidal rule for the whole interval the order is decreased by one. On the other hand it is now possible to increase the number of integration points to improve the result. Using for example twice as many integration nodes decreases the bound on the integration error by a factor of four. Moreover, this approach allows the construction of series of nested integration rules, using N, $2N-1$, $4N-3$, $8N-7$, $\ldots$ integration nodes. The advantage of such construction is that every integration node in a rule with fewer nodes is also existent in rules with a higher node count, i.e., the computed results can be reused. One possible usage of such nested integration rules is to act as termination criterion in an integration routine. If integration results for two consecutive integration rules (e.g., the rules using N and $2N-1$ nodes) differ less than a previously chosen value ϵ than the result of the integration rule with the higher node count is accepted as result. Otherwise, the result of the integration rule with the lower node count is discarded and the whole process is repeated starting with the integration rule with the initially higher node count and its consecutive rule (e.g., taking the rules with $2N-1$ and $4N-3$ integration nodes). Due to the fact that the integration rules derived with this second approach are based on the results of the first approach they suffer from the same fate of being numerically unstable for methods of higher polynomial accuracy.

Summing up, we find that the rules generated by the first approach are more of theoretical nature, since their application is confined to integration on a relatively small interval $[a, b]$ in order to keep the step size h low. Furthermore, choosing more integration nodes with the goal of reducing the step size h may lead to numerically unstable integration rules. In contrast, the second approach allows for an arbitrary step size by choosing a suitable value N for the number of integration nodes. In addition, the construction of series of nested integration rules is possible. What renders these rules unsuitable in the given setting is the fact that increasing N neither increases the polynomial accuracy nor the order of the rule. Moreover, similar to the first approach, integration rules with higher polynomial accuracy can only be generated at the cost of numerical instabilities.

Remark 5.1.3. Generally, Newton-Cotes rules can be designed to include derivative information at the integration nodes. Since in the case of CCOPT the calculation of derivatives typically includes the solution of a possibly medium to large scale linear system in addition to the solution of a nonlinear system such an approach is not useful in the given setting, due to the high computational demand.

5.1.2 Clenshaw-Curtis quadrature

Clenshaw-Curtis (CC) rules are typically generated for integrals of the form

$$\int_{-1}^{1} f(\xi) d\xi,$$

but other bounded intervals of integration are possible (by transforming the integral from $[a, b]$ to $[-1, 1]$ using a linear transformation) as well as extensions for the weighted case. The integration nodes in the standard case are given by

$$\xi_i = \cos\left(\frac{i-1}{N-1}\pi\right), \ i = 1, \dots, N$$

for an $N \geq 2$ and the weights can be obtained by

$$\omega_i = \frac{c_i}{N-1}\left(1 - \sum_{j=1}^{\lceil \frac{N-1}{2} \rceil} \frac{b_j}{4j^2-1} \cos\left(2j\frac{i\pi}{N-1}\right)\right),$$

where

$$b_j = \begin{cases} 1, & j = \frac{N-1}{2} \\ 2, & j < \frac{N-1}{2} \end{cases}, \ c_i = \begin{cases} 1, & i = 0 \mod (N-1) \\ 2, & \text{otherwise} \end{cases}.$$

These weights are obtained by requiring that all polynomials up to degree N-1 are integrated exactly [86]. This is equivalent to the requirement that all monomials are integrated exactly, which leads to linear system of equations of the form

$$\underbrace{\begin{pmatrix} 1 & \dots & 1 \\ \xi_1 & \dots & \xi_N \\ \xi_1^2 & \dots & \xi_N^2 \\ \vdots & & \vdots \\ \xi_1^{N-1} & \dots & \xi_n^{N-1} \end{pmatrix}}_{A} \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_N \end{pmatrix} = \begin{pmatrix} \int_a^b 1 d\xi \\ \int_a^b \xi d\xi \\ \vdots \\ \int_a^b \xi^{N-1} d\xi \end{pmatrix}. \tag{5.1.3}$$

Since the nodes ξ_i, $i = 1, \dots, N$ are all distinct, the $N \times N$-matrix A has full rank and there always exists a unique solution of (5.1.3). As a direct consequence, an N-point CC rule has at least a polynomial accuracy of $N - 1$. An extension of standard CC rules to the weighted case is possible by changing the right-hand side of (5.1.3) to

$$\begin{pmatrix} \int_a^b w(\xi) d\xi \\ \int_a^b \xi w(\xi) d\xi \\ \vdots \\ \int_a^b \xi^{N-1} w(\xi) d\xi \end{pmatrix}.$$

One should be aware that in this generalized approach negative weights can occur (resulting in cancellation and numerical instabilities), whereas the same can not happen in the standard case.

Remark 5.1.4. Although the system (5.1.3) could be used to determine the weights in a CC rule this is not a practical approach, since the matrix A has a bad condition, especially for larger values of N. This leads to a reduced accuracy when trying to solve the system with numerical methods. Practical approaches to the computation of the weights usually involve the usage of a discrete Fourier transform [86].

In summary, N-node CC rules have a polynomial accuracy of $N - 1$, i.e., in contrast to the composite Newton-Cotes rules, increasing the number of node points actually increases the polynomial accuracy. Furthermore, generating CC rules for arbitrary polynomial accuracy does not pose a problem. Similar to the Newton-Cotes formulas the construction of nested sequences of CC rules is possible.

5.1.3 Gauß quadrature

Gauss quadrature offers a natural framework to overcome the limitations of the previously mentioned quadrature rules. For instance, weight functions can be included without the occurrence of negative weights. Moreover, Gauss quadrature achieves the highest polynomial accuracy among all integration rules for a given number of integration nodes N. This is achieved by choosing not only the weights but also the integration nodes in an optimal way.

Gaussian quadrature rules are based on the notion of orthogonal polynomials in the Hilbert space $L^2[a,b]$ of all functions for which

$$\int_a^b (f(\xi))^2 \, w(\xi) d\xi$$

is well defined. The corresponding scalar product is defined by

$$\langle f, g \rangle := \int_a^b f(\xi) g(\xi) w(\xi) d\xi,$$

where $f, g : [a,b] \to \mathbb{R}$ and two polynomials p_1, p_2 are said to be orthogonal if $\langle p_1, p_2 \rangle = 0$. Starting with the polynomial $p_0 \equiv 1$ it is now possible to construct a orthonormal system $p_0.p_1, \ldots$ of polynomials, such that $\langle p_i, p_j \rangle = 0$ whenever $i \neq j$. One way of doing this is to use the following recursive construction (see [83], p. 151)

$$p_{-1} \equiv 0, \tag{5.1.4}$$

$$p_0 \equiv 1, \tag{5.1.5}$$

$$p_{i+1}(\xi) = (\xi - \delta_{i+1}) p_i(\xi) - \gamma_{i+1}^2 p_{i-1} \xi, \; i \geq 0, \tag{5.1.6}$$

$$\text{with } \delta_{i+1} = \frac{\langle \xi p_i, p_i \rangle}{\langle p_i, p_i \rangle} \text{ and } \gamma_{i+1} = \left\{ \begin{array}{cc} 0, & i = 0 \\ \frac{\langle p_i, p_i \rangle}{\langle p_{i-1}, p_{i-1} \rangle}, & i \geq 1 \end{array} \right. . \tag{5.1.7}$$

The roots of these polynomials together with suitable weights can be used as integration rules, which becomes clear from the following theorem.

Theorem 5.1.5 (Gaussian quadrature, [83] p. 153 f.).

(i) *Let $\xi_1, \ldots, \xi_N$ be the real and simple roots of the N-th orthogonal polynomial $p_N(\xi)$, and let $\omega_1, \ldots, \omega_N$ be the solution of the (non-singular) system of equations*

$$\sum_{i=1}^{N} \omega_i p_k(\xi_i) = \left\{ \begin{array}{cc} \langle < p_0, p_0 \rangle >, & k = 0 \\ 0 & k = 1, \ldots, N-1 \end{array} \right. . \tag{5.1.8}$$

Then $\omega_i > 0$ for $i = 1, \ldots; N$ and

$$\int_a^b p(\xi) w(\xi) d\xi = \sum_{i=1}^{n} \omega_i p(\xi_i) \tag{5.1.9}$$

holds for all polynomials $p \in \Pi_{2N-1}$.

(ii) *Conversely, if the numbers ω_i, $i = 1, \ldots, N$ are such that* (5.1.9) *holds for all $p \in \Pi_{2N-1}$, then the ξ_i are the roots of p_N and the weights ω_i satisfy* (5.1.8).

(iii) *It is not possible to find numbers ξ_i, ω_i, $i = 1, \ldots; N$, such that* (5.1.9) *holds for all polynomials $p \in \Pi_{2N}$.*

Remark 5.1.6. Similar to the CC rules it is inappropriate to use the theoretical construction to determine the integration nodes and weights for Gaussian quadrature rules. This is due to the fact that it is numerically difficult to obtain the N roots of the N-th orthogonal polynomial p_N with sufficient accuracy. Additionally, the system (5.1.8) suffers from bad condition for larger numbers of N. For practical purposes one can construct a symmetric matrix from the values δ_i and γ_i used in the recurrence relation. The integration nodes can then be found as eigenvalues of this matrix, whereas the weights can be constructed from the eigenvectors (see [83], p. 156 ff. for a detailed description).

An error bound for N-point Gaussian quadrature of functions $f \in C^{2N}[a, b]$ is given by

$$\frac{f^{(2N)}(\xi)}{2N!} \langle p_0, p_0 \rangle$$

for a $\xi \in (a, b)$.

In summary, Gaussian quadrature rules seem to be the “best” available method for univariate integration, allowing a polynomial accuracy of $2N - 1$ for an N-point rule. Furthermore, the weights are guaranteed to be positive, therefore, no cancellations occur. Nevertheless, the rules under consideration also have a drawback, since they generally cannot be nested like Clenshaw-Curtis or Newton-Cotes rules. Additionally, for non-polynomial functions the performance of Gaussian rules seems to be similar to those of Clenshaw-Curtis rules [84].

5.1.4 Kronrod-Patterson quadrature

Gauss-Kronrod and Kronrod-Patterson integration both try to extend existing Gaussian rules with N nodes by adding $M > N$ further integration nodes. Since the original integration nodes are preserved, this results in a set of nested quadrature rules. This “nested” property comes at the price of a reduced polynomial accuracy, which is usually lower than the polynomial accuracy of $2(N + M) - 1$ achieved by a pure Gaussian rule with the same number of nodes but greater than a polynomial accuracy of $N + M - 1$ achieved by a Clenshaw-Curtis rule with $N + M$ nodes. Kronrod [53] proposed to extend a rule containing N nodes by further $N+1$ nodes, e.g., he extended the seven point Gauss-Legendre rule by additional eight points. Building on the work of Kronrod, Patterson tried to generate extensions to the already extended rules, using a similar approach.

Starting point for the process of Gauss-Kronrod and Kronrod-Patterson quadrature is an arbitrary orthogonal polynomial p_N of a corresponding Gauss quadrature rule. The next step is to construct a polynomial

$$q_M(\xi) = \xi^M + \sum_{i=0}^{M-1} \kappa_i \xi^i, \tag{5.1.10}$$

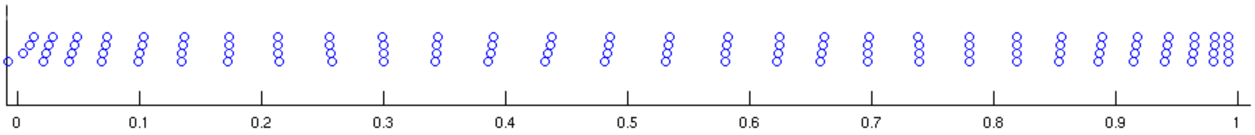

Figure 5.1: *Kronrod-Patterson extensions to four different quadrature rules with Beta weight, constant parameter β and four different choices for the parameter α*

such that

$$\langle q_M p_N, p_i \rangle = 0, \ i = 0, \ldots, N$$

i.e., $q_M p_N$ is orthogonal to all polynomials p_i, $i = 0, \ldots, N$. This is equivalent to the requirement that

$$\langle q_M p_N, \xi^i \rangle = 0, \ i = 0, \ldots, N. \tag{5.1.11}$$

Inserting (5.1.10) into (5.1.11) leads to the linear system

$$\begin{pmatrix} \int_a^b w(\xi)d\xi & \ldots & \int_a^b \xi^{M-1} w(\xi)d\xi \\ \int_a^b \xi w(\xi)d\xi & \ldots & \int_a^b \xi^{M} w(\xi)d\xi \\ \vdots & & \vdots \\ \int_a^b \xi^{N-2} w(\xi)d\xi & \ldots & \int_a^b \xi^{N+M-3} w(\xi)d\xi \\ \int_a^b \xi^{N-1} w(\xi)d\xi & \ldots & \int_a^b \xi^{N+M-2} w(\xi)d\xi \end{pmatrix} \begin{pmatrix} \kappa_0 \\ \kappa_1 \\ \vdots \\ \kappa_{M-2} \\ \kappa_{M-1} \end{pmatrix} = \begin{pmatrix} -\int_a^b \xi^{M} w(\xi)d\xi \\ -\int_a^b \xi^{M+1} w(\xi)d\xi \\ \vdots \\ -\int_a^b \xi^{N+M-1} w(\xi)d\xi \\ -\int_a^b \xi^{N+M} w(\xi)d\xi \end{pmatrix}, \tag{5.1.12}$$

which has a unique solution if $M = N + 1$. The M roots of q_M together with the N roots of p_N form the integration nodes of the Kronrod extensions. In contrast to Gaussian quadrature, the roots of q_M may be neither real nor simple. Furthermore, it is not assured that all roots of q_M are actually inside the region of integration. An example for this behavior is shown in Figure 5.1 , where extensions to several 7-node quadrature rules for Beta weight with the same parameter β but slightly changing parameter α were generated. In the three cases on the top the extensions exist, as all nodes are simple, real and inside the interval of integration. In the last case however, while the nodes are still real and simple, the most left node is outside of the domain of integration. As a consequence, this extension cannot be used for quadrature. More generally, if some of the previously mentioned conditions appear the Kronrod is said to be non-existent. In the case that the extensions exists, it has a polynomial accuracy of $3N + 1$ while having $2N + 1$ integration nodes. This is less than the maximum possible polynomial accuracy of $4N + 1$ achieved by a pure Gaussian rule but more than a polynomial accuracy of $2N$, which could be achieved by a Clenshaw-Curtis rule. Although some measures have been taken to find conditions for the existence of Kronrod extensions, this field of research is wide open. In many instances the only way to find such extensions is by the method of trial-and-error. A second approach is the usage of suboptimal Kronrod extensions [8]. Here the polynomial q_M

is constructed in a certain way to guarantee the existence of the extensions, but this comes at the cost of a reduced polynomial accuracy. A third approach is to choose a value $M > N + 1$, e.g., $M = N + 3$. Since (5.1.12) is now underdetermined, additional conditions of the form $\langle q_M p_N, p_i \rangle = 0$, $i = N + 1, \ldots, M - 1$ can be added. These additional conditions ensure that the constructed rule has the highest possible polynomial accuracy of at least $2M + N - 1$. Fundamentally, this method also relies on trial-and-error.

In the following, yet another approach will be presented. It arose during the work on a practical CCOPT problem, which was solved as part of this thesis (see the Chance Constrained Optimal Power Flow (CCOPF) problem in the case studies). This particular problem includes Beta-distributed uncertainties, whose parameters α, β depend on a forecast. For this reason, the actual values of α and β are not known beforehand. Since the Beta distribution has no standard form, quadrature rules have to be generated for every possible pair of parameters α, β. The proposed approach goes as follows. Starting with a Kronrod extension generated for integrals with Beta weight for some parameters α_0, β_0 by means of another approach, Newton's method can be used to find the nodes of Kronrod rules for weight functions with parameters α, β in some neighborhood of α_0, β_0. Assuming that the already generated extension adds M nodes to an N node Gaussian quadrature rule, the proposed approach consists of two steps:

(i) The N nodes of the Gaussian quadrature rule belonging to the weight function with parameters α and β are sought by means of a standard approach.

(ii) The additional integration nodes are determined by Newton's method. The system to be solved is essentially (5.1.12), but since q_M is now described by its roots instead of its coefficients the system is nonlinear. In the case that $M > N + 1$ the system (5.1.12) has to be padded with further restrictions of the kind $\langle q_M p_N, p_i \rangle = 0$, $i = N + 1, \ldots, M - 1$. As starting point of Newton's method the M roots of the polynomial q_M for weight with parameters α_0, β_0 are used.

Similar to the construction of standard Kronrod extensions, solutions may not exist (i.e., some of the nodes are complex) or some of the nodes may not be contained in the domain of integration. If this happens the extensions by means of Newton's method fails. Otherwise, this approach results in a Kronrod type quadrature rule for weights with parameters α, β. The same approach can also be used to construct Patterson extensions from a given existing sequence of Patterson extensions. In this case, the second step has to be carried out repeatedly. Figure (5.2) shows the approximate area in the domain of the parameters α, β of a Beta-distribution, where a Patterson extension of the type $1 + 2 + 4 + 22$ exists. This figure was obtained by using the aforementioned approach. The type $1 + 2 + 4 + 22$ has to be read as a one node Gaussian rule extend by two nodes to get a Kronrod extensions, further extended by first four and then 22 nodes. As can be seen in Figure (5.2), the domain where this kind of extensions exist is rather irregular. Consequently, the idea to provide a list of precomputed Gauss-Kronrod or Kronrod-Patterson rules from which other rules could then be generated proved impractical, since it is generally unclear for which parameters quadrature rules should be computed beforehand.

Till now we have only discussed the computation of the nodes. The weights ω_i can be found

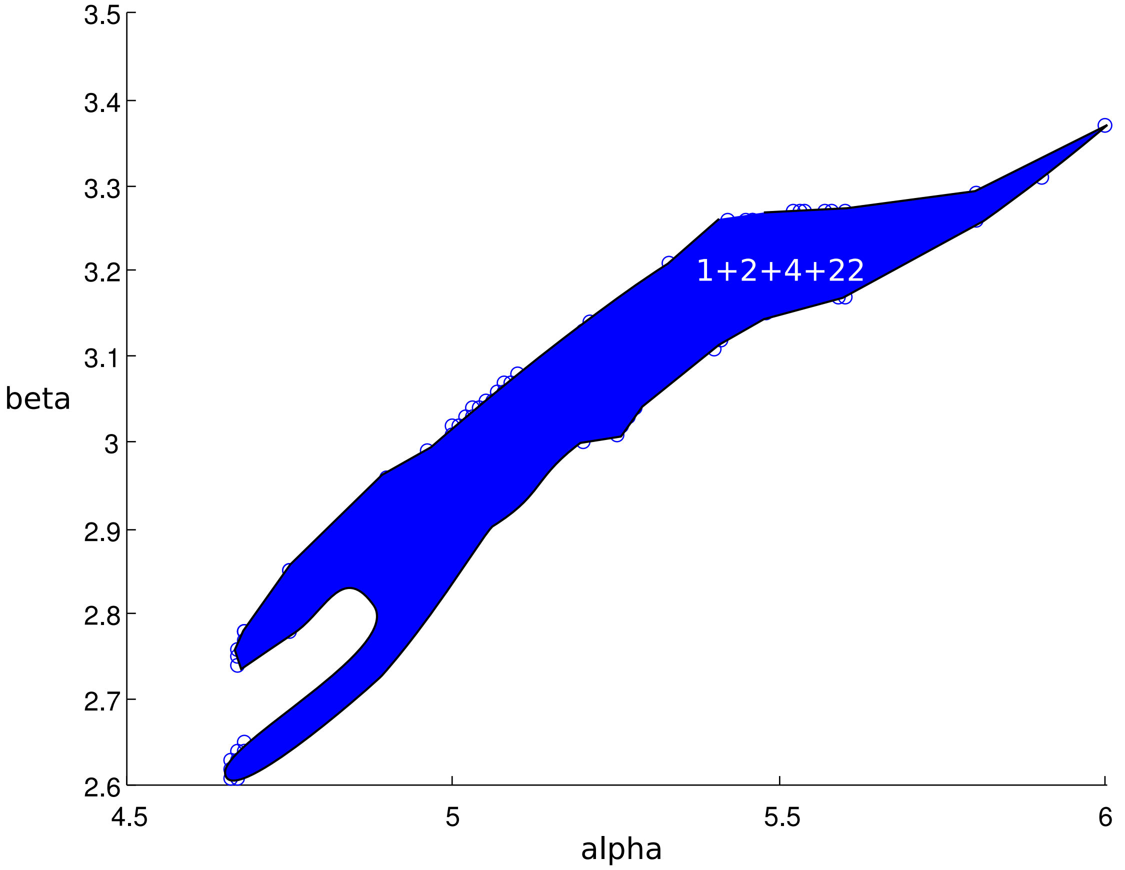

Figure 5.2: *Approximation of the parameters α, β of a Beta-distribution for which a Kronrod-Patterson rule of the type $1+2+4+22$ exists*

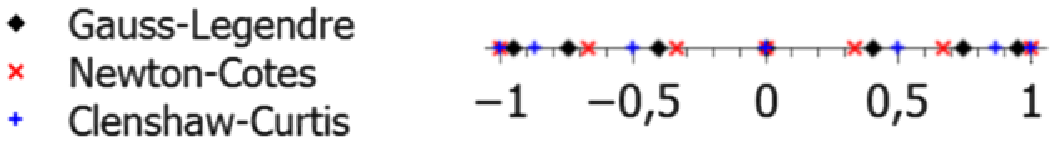

Figure 5.3: *Integration nodes for different quadrature methods on the interval* $[-1,1]$ *with uniform weight*

as the unique solution of (5.1.8), i.e., the same system as in standard Gaussian quadrature. Unlike the standard case, negative weights may appear.

In summary, Kronrod-Patterson quadrature rules can be seen as a kind of compromise between pure Gaussian and Clenshaw-Curtis rules. From the first they inherit the high polynomial accuracy, from the second the property of “nestedness”. While for some weight functions (especially Gaussian and uniform weight) the Kronrod-Patterson rules are readily available, the existence of extensions is not clear for others (e.g., Beta weight). In practice, this limits the usage of such rules mostly to weight functions of the first type.

5.1.5 Some remarks on univariate quadrature in the context of CCOPT

While univariate quadrature is seldom necessary on itself in the context of CCOPT (most problems contain more than one uncertainty), the corresponding rules can be used as building blocks for multivariate quadrature rules as demonstrated in the next section. As a consequence, the methods presented in this section were rated mainly on their suitability for being building blocks. The necessary criteria are

- high polynomial accuracy,
- nestedness (i.e., rules with a higher count of integration nodes should contain the nodes of an integration rule with fewer node count),
- ability to incorporate weight functions,
- positive weights.

As became clear in this section, there is no approach fulfilling all these criteria. While Newton-Cotes rules fulfill only two of the criteria (nestedness and positive weights) and, therefore, are unsuitable for CCOPT, Gauss and Clenshaw-Curtis both fulfill three criteria. These are high polynomial accuracy, ability to incorporate weights functions, and positive weights for Gaussian quadrature and high polynomial accuracy (although to a lesser extent than Gaussian rules), nestedness and positive weights for Clenshaw-Curtis quadrature. For specific weight functions, Clenshaw-Curtis quadrature fulfills all four criteria, which is the same as for Kronrod-Patterson rules. For weight functions were all four criteria are fulfilled by either Kronrod-Patterson or

Clenshaw-Curtis, the respective method is the approach of choice. Otherwise, if the negative weights in the Clenshaw-Curtis rule do not impact the numerical stability of the integration rule (i.e., the negative weights have a small magnitude) then Clenshaw-Curtis is still the method of choice, since the nestedness significantly reduces the number of grid points (multidimensional integration nodes) in an important type of multivariate integration rules. If the numerical stability is impacted in Clenshaw-Curtis rules then Gaussian rules are the method of choice.

We conclude this section with a comparison of seven node Newton-Cotes, Clenshaw-Curtis and Gaussian rules for the interval $[-1, 1]$ with uniform weight. The corresponding nodes are shown in Figure 5.3. While the Newton-Codes nodes are uniformly distributed, the nodes for the other two methods are more dense near the boundary of the domain of integration, which is actually necessary to guarantee a high polynomial accuracy.

5.2 Multivariate integration (cubature)

As discussed above, to compute the probability values and gradients in the chance constraints as well as statistical moments and gradients we have to evaluate multidimensional integrals of the form

$$I = \int\limits_{a_1}^{b_1} \dots \int\limits_{a_n}^{b_n} f(\xi_1, \dots, \xi_n) w(\xi_1, \dots, \xi_n) d\xi_n \dots d\xi_1, \qquad (5.2.1)$$

where $-\infty \leq a_j < b_j \leq \infty$ for $i = 1, \dots, n$, f is at least piecewise continuous, and $w : [a_1, b_1] \times \dots \times [a_b, b_n] \to \mathbb{R}$ is a continuous weight function with $w(\xi_1, \dots, \xi_n) > 0$ for $(\xi_1, \dots, \xi_N) \in (a_1, b_1) \times \dots \times (a_b, b_n)$. Furthermore, we assume that the weight function w can be decomposed in the following way

$$w(\xi_1, \dots, \xi_n) = \prod_{i=1}^{n} w_i(\xi_i), \qquad (5.2.2)$$

where $w_i : [a_i, b_i] \to \mathbb{R}$. For systems with a large number of uncertain variables the computation of such integrals is very time consuming. Similar to the univariate case such integrals can be approximated by cubature formulas

$$\hat{I}[f] = \sum_{i=1}^{N} \omega_i f(\xi_1^i, \dots, \xi_n^i),$$

where N is the number of grid points (multivariate integration nodes), ω_i are weighting factors and $\xi_1^i, \dots, \xi_n^i$ denote grid points $a_j \leq \xi_j^i \leq b_j$ for $j = 1, \dots, n$ and $i = 1, \dots, N$, respectively.

5.2.1 Full grids

Full grids present the trivial approach to multivariate integration. Writing (5.2.1) as

$$\int_{a_1}^{b_1} \left(\int_{a_2}^{b_2} (\ldots)\, w_2(\xi_2) d\xi_2 \right) w_1(\xi_1) d\xi_1$$

allows to use a suitable univariate quadrature rule for the outer integral, then at every integration node the next inner integral is again computed using a univariate rule, and so on. More clearly, let $V_j(N_j) = \sum_{i=1}^{N_j} \omega_j^i f(\xi_j^i)$ denote a one-dimensional integration rule for the integral

$$\int_{a_j}^{b_j} f(\xi_j) w_j(\xi_j) d\xi_j$$

with N_j nodes ξ_j^i and weights ω_j^i. This can be used to construct a rule for the integral (5.2.1) using the tensor product approach. The product (or full-grid) rule [19] over the n-dimensional integral is defined by

$$V_1(N_1) \otimes \ldots \otimes V_n(N_n)[f] = \\ \sum_{i_1=1}^{N_1} \cdots \sum_{i_k=1}^{N_k} \omega_1^{i_1} \ldots \omega_n^{i_n} f(\xi_1^{i_1}, \ldots, \xi_n^{i_n}). \tag{5.2.3}$$

A graphical explanation of the construction of tensor product grids is shown in 5.4. The points on the left-hand-side and below the square represent the one-dimensional integration nodes, while the grid points inside the square are the derived points for a two-dimensional rule.

The number of grid points of the product rule is $N_1 \times \ldots \times N_n$. Assuming that $N_1 = \ldots = N_n = N$ the number of points in the product rule is N^n, i.e., the number of grid points grows exponentially with the dimension of the integral. This behavior is called the "curse of dimension". The polynomial accuracy of a full grid rule depends on the polynomial accuracy of the underlying univariate rule. If these rules have a polynomial accuracy of A_i, $i = 1, \ldots, n$ then all polynomials of the form

$$\prod_{i=1}^{n} \xi_1^{p_1}, \quad 0 \le p_i \le A_i,\ i = 1, \ldots, n$$

are integrated exactly. This leads to cases where a polynomial of degree $\sum_{i=1}^{n} A_i$ is integrated exactly, whereas certain polynomials with lesser degree are not integrated exactly. Nevertheless, the minimum guaranteed polynomial exactness is $\min_{i \in \{1,\ldots,n\}} A_i$.

Remark 5.2.1. A method, comparable to full grid integration, was proposed by Prékopa for Gaussian weights in [69], p. 195. This method is based on the fact that the marginal distributions of a Gaussian distributed uncertain vector also underlie a Gaussian distribution. But as Prékopa himself points out, this method should not be applied due to the high amount of computation required to carry out the necessary transformations.

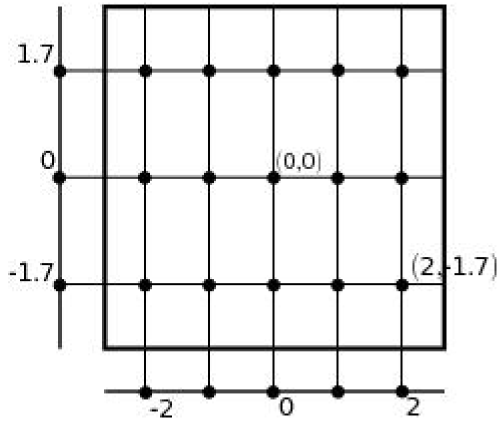

Figure 5.4: *Construction of tensor product grids*

5.2.2 Sparse Grids

In 1963 Smolyak [80] proposed the sparse-grid integration method to overcome the "curse of dimension". The idea is to construct multi-dimensional rules using tensor products of differences of quadrature rules, in contrast to full-grid integration, where only quadrature rules are used. The basis of sparse-grids is a sequence of one-dimensional quadrature rules $V_i(N_i)$ with an increasing polynomial accuracy, $i = 1, \dots, n$, which explains the requirement of a high polynomial accuracy in evaluating the univariate integration rules. Using these integration rules the differences of quadrature rules can be defined by

$$V_0(N_0) = 0, \quad \Delta_0 = 0, \Delta_i = V_i(N_i) - V_{i-1}(N_{i-1}).$$

It is easy to see that $V_n(N_n) = \sum_{i=1}^n \Delta i$ because of the telescopic sum. A second observation is that for continuous integrands and suitable underlying quadrature rules $\lim_{n\to\infty} \sum_{i=1}^n \Delta i = I$, i.e., the series converges toward the exact integral value. This series is used to derive the multi-dimensional cubature formula. As in the full-grid case the series is combined using the tensor-product approach (see (5.2.3)), leading to

$$\sum_{0 \le i_1, \dots, i_n < \infty} \Delta_{i_1} \otimes \dots \otimes \Delta_{i_n}.$$

The series cannot be evaluated practically. Therefore a truncation is used, such that the sparse-grid integration rule has the form

$$A(n,q) = \sum_{0 \le i_1 + \dots + i_n \le q} \Delta_{i_1} \otimes \dots \otimes \Delta_{i_n}. \tag{5.2.4}$$

It is possible to express $A(n,q)$ directly in terms of the underlying quadrature rules as shown by Wasilkowski and Wozniakowski [87].

Sparse-grid rules generated through (5.2.4) generally contain negative weights. Since these negative weights affect the numerical stability of the rules, not all rules generated by (5.2.4) are convenient for integration. In fact, one has to determine grid points and weights where the norm of the negative weights is comparatively small. Such grids have already been evaluated for unweighted integrals over hypercubes and integrals with Gaussian weights. Novak and Ritter [66] have shown that formulas generated with (5.2.4) have a polynomial accuracy of at least $2(q-n)+1$ if integration over a hypercube is considered and the underlying one-dimensional rules are Gaussian quadrature rules. For the same class of integrals it was shown that the number of grid points to obtain the same polynomial accuracy grows only polynomially with the dimension of the integral, in contrast to an exponential growth for the product rule. This is a real advantage, which becomes clear from Figure 5.5, where the numbers of necessary grid points are shown for full and sparse grid rules with the same polynomial accuracy. Whereas the eight-dimensional sparse-grid rule needs only about 1000 grid points, a full grid rule of the same polynomial accuracy needs about 10^8 grid points, which is prohibitive in practical application. The number of grid points in a sparse grid rule can be further reduced by using nested underlying quadrature rules, which explains why nestedness was a desired property in the last section.

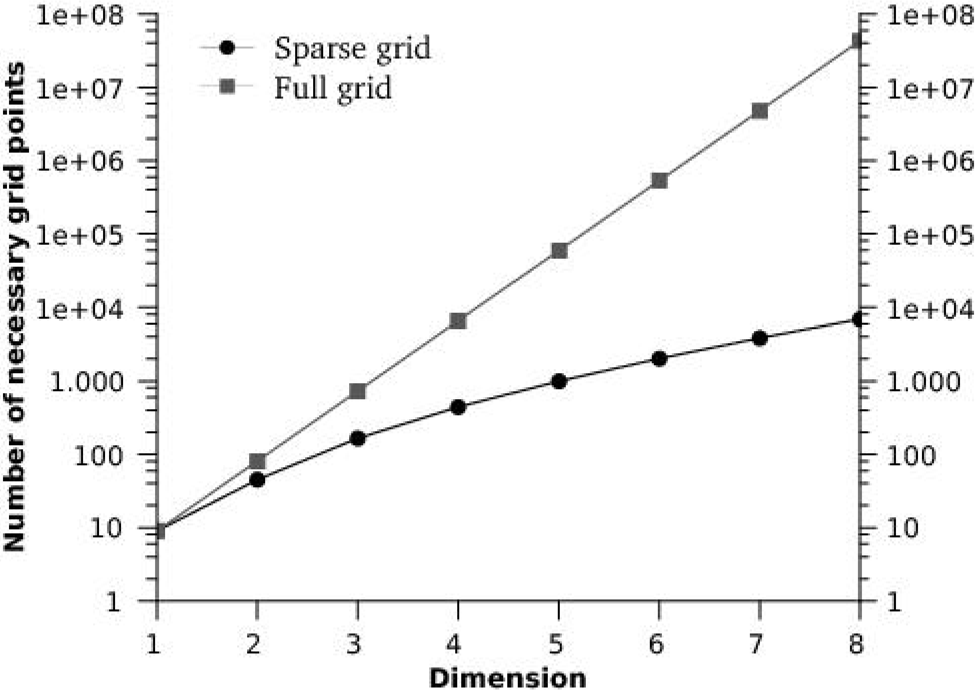

Figure 5.5: *Comparison of the number of required grid points for full and sparse grid integration for different dimensions of integration and the same polynomial accuracy*

Further comparisons between product rules and sparse-grids based on standard Gaussian rules as well as on the extensions proposed by Genz and Keister [36] were given by Heiss and Winschel [39]. More performance results for sparse-grids were obtained by Gerstner and Griebel [37]. The advantages of sparse-grid integration techniques can be summarized as follows:

- Multidimensional integrals can be accurately computed using only a few grid points.
- Integration of polynomials can be evaluated exactly (depending on the order).
- Grid points and weights only depend on the region of integration not on the integrand itself.
- They can be easily implemented when using predefined grid points and weights.
- Existing routines for full-grids can be used without any or with only few modifications.

The disadvantages of sparse-grid rules are basically twofold. First, the existence of negative weights cannot be prevented, therefore, it is necessary to check sparse grid rules for numerical instabilities. Second, the constructed rules are only suitable for very smooth functions, which can be seen by the error estimate

$$\mathcal{O}\left(\hat{N}^{-r/n}(\log \hat{N})^{(n-1)(r/n+1)})\right) \tag{5.2.5}$$

given in [66] for a bounded domain of integration and integrands f with bounded derivatives up to order r, where $\hat{N}$ is the total number of grid points and n is the dimension of the integral.

5.3 (Quasi-)Monte-Carlo cubature

In order to apply Monte-Carlo and Quasi-Monte-Carlo schemes we transform the integrals to the unit cube,

$$\int\limits_{\mathbb{R}^d} f(x)\phi(x)dx = \int\limits_{[0,1]^d} f(\Phi^{-1}(y))dy, \tag{5.3.1}$$

using the inverse cumulative distribution function Φ^{-1}. Then both methods approximate integrals in the following way

$$Q(n,d) = \frac{1}{N}\sum_{i=1}^{N} f\left(\Phi^{-1}(y_i)\right). \tag{5.3.2}$$

Monte-Carlo methods sample the grid points y_i from a uniform distribution. On the other hand, Quasi-Monte-Carlo methods use the concept of low discrepancy for the generation of grid points. Roughly speaking, this means that the empirical distribution induced by the grid points should not deviate too much from the uniform distribution. The main difficulty is to generate such grids in higher dimensions without an exponential growth of the necessary grid points. There exist different strategies to generate such grids. In this work the Sobol sequence was used, because at least for the test functions it yields the best results. A detailed description of Monte-Carlo and Quasi-Monte-Carlo methods can be found in [54].

The integration error of standard Monte-Carlo integration is in the order of

$$\mathcal{O}\left(\frac{1}{\sqrt{\hat{N}}}\right),$$

whereas the integration error of Quasi-Monte-Carlo is in the order of

$$\mathcal{O}\left((\log \hat{N})^n \frac{1}{\hat{N}}\right),$$

where again $\hat{N}$ is the number of grid points and n the dimension of integration [12]. Although the error rate of Quasi-Monte-Carlo integration is better than that of pure Monte-Carlo for moderate dimension n of integration, it should be noted that the Monte-Carlo result could be improved by additional methods like importance sampling [67]. Such methods were not considered in the course of this work, since such methods would need to be carried out during run time of an integration algorithm, thereby further increasing computational demand.

5.4 Comparison of the different methods

5.4.1 Test Results

The goal of these tests is to determine the minimum number of grid points necessary to evaluate the weighted integrals

$$\int\limits_{\mathbb{R}^d} f(\xi)\phi(\xi)d\xi,$$

where $\phi(\xi)$ is the multivariate Gaussian weight, with an absolute numerical error smaller or equal to 10^{-3} for different test functions. Table 5.1 presents the results of the numerical experiments. The table does not list the result of the integration, but instead lists the numerical errors. The first column contains the dimension, whereas the second column contains the description of the routine in the format "method - # of grid points'". The acronym SG stands for sparse grid, MC and QMC for Monte-Carlo and Quasi-Monte-Carlo, respectively. The integration routines (column 2) which are most suitable for application are printed in bold font. The bold face numbers in columns 3–8 indicate violations of the chosen numerical accuracy. As one can see, sparse grids yield the best results (measured in numbers of necessary grid points) except for the step function. Since the step function is discontinuous, such behavior is not surprising in the light of the error estimate 5.2.5. This shows that sparse grid techniques are not suitable if the functions contain discontinuities or are generally not very smooth, which, for example, is the case when using AA methods. Consequently, the integration method of choice in this case are Quasi-Monte-Carlo algorithms. It is also obvious that the Monte-Carlo method requires significantly more grid points than the other integration routines. Therefore, it should not be applied in its pure form.

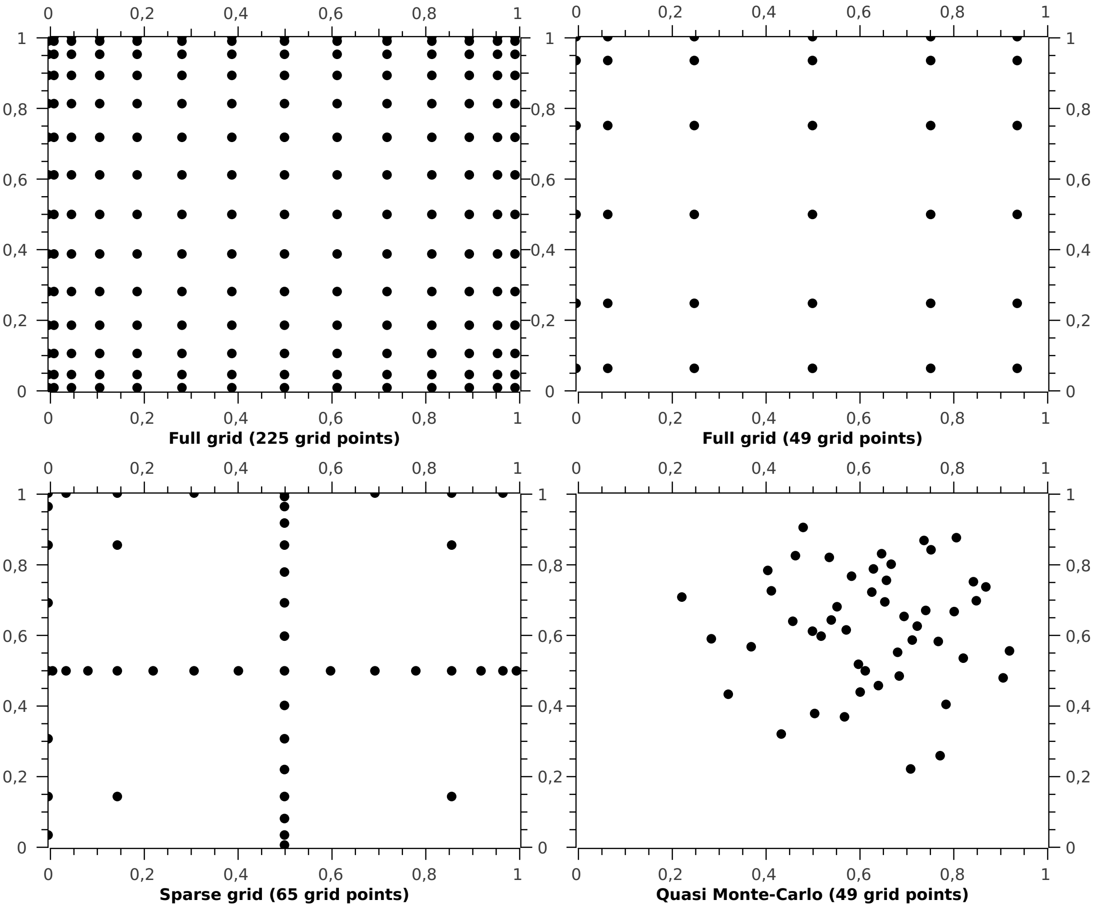

Figure 5.6: *Grid points of different integration methods for a two-dimensional integration with Beta weight.*

We conclude this section with a comparison of the different grid points of two full grids, a sparse grids (all three based on Clenshaw-Curtis rules), and a Quasi-Monte-Carlo grid for two-dimensional integration with Beta weight, which is shown in Figure 5.6. The integration rules connected with the grids shown in the first column have the same polynomial accuracy, but the sparse grid rule in the second row uses only a subset of the grid points of the full grid rule. Furthermore, while the grid points of the (Q)MC rule are uniformly distributed (with respect to the underlying Beta weight), the grid points of the sparse grid rule are concentrated on two axis parallel to the boundary of the domain and on the boundary itself. This explains, why sparse grids rules are only suitable for sufficiently smooth functions, since any discontinuities suitably far away from the grid points cannot be detected.

Table 5.1: *Test functions and error results*

	$f(x) =$	1	x	x^2	x^3	$\sin(x)$	$\frac{x}{x^2+1}$	$\mathbb{1}_{x<0}$
1-dim.	**SG - 9**	0	1.1e-17	4.4e-16	3.9e-17	1.7e-17	1.1e-18	1.3e-1
	QMC - 18,000	8e-7	2.6e-4	1e-3	**2.3e-3**	4.8e-5	4.4e-5	4e-7
	QMC - 20,000	1e-13	1.4e-4	7.6e-4	6.1e-4	6.3e-5	2.9e-5	4.7e-14
	MC - 800,000	1.7e-11	3.8e-4	**1.2e-3**	5.9e-4	5.0e-4	3.7.3e-4	6.5e-5
	MC - 900,000	1.0e-6	1.0e-4	6.9e-4	4.2e-4	1.6e-5	4.3e-4	3.7e-4
	$f(x) =$	1	$x_1 + x_2$	$x_1 * x_2$	$(x_1 - x_2)^2$	$\sin(x_1)\cos(x_2)$	$\cos(x_1)\sin(x_2)$	$\mathbf{1}_{x<0}$
2-dim.	**SG - 45**	0	5.1e-19	6.4e-18	0	1.5e-17	9.6e-18	1.5e-1
	QMC - 30,000	1.0e-6	1.4e-4	9.4e-6	**1.3e-3**	6.4e-5	2.9e-5	2.5e-7
	QMC - 35,000	1.0e-6	2.5e-4	3.5e-5	9.8e-4	3.6e-5	2.3e-5	2.5e-7
	MC - 1,500,000	5.0e-7	4.0e-4	6.7e-5	**2.8e-3**	1.3e-4	5.4e-4	1.8e-4
	MC - 2,500,000	2.5e-11	4.0e-4	5.9e-4	4.4e-4	3.3e-4	6.7e-4	9.0e-5
	$f(x) =$	1	$\sum x_i$	$\prod x_i$	$(x_1 - \sum_2^d x_i)^2$			$\mathbf{1}_{x<0}$
3-dim.	**SG - 165**	6.7e-16	1.7e-17	6.3e-19	3.6e-15			1.1e-1
	QMC - 45,000	1e-6	8.8e-5	6.9e-5	**1.01e-3**			2.2e-5
	QMC - 50,000	7.1e-13	1.3e-4	2.2e-4	9.5e-4			2e-5
4-dim.	**SG - 441**	2.2e-16	7.5e-19	1.5e-19	3.3e-14			6.1e-2
	QMC - 70,000	1e-6	6.4e-5	8.6e-5	**1.2e-3**			1.4e-5
	QMC - 72,000	8.0e-6	2.8e-5	2.6e-4	6.6e-4			1.4e-5
5-dim.	**SG - 993**	1.1e-15	3.4e-16	1.4e-19	6.3e-14			3.1e-2
	QMC - 160,000	1.5e-12	7.5e-5	2.2e-5	**1.02e-3**			1.3e-5
	QMC - 165,000	6.5e-7	3.4e-16	1.4e-19	9.8e-4			7.6e-6
6-dim.	**SG - 2021**	1.9e-14	1.2e-16	0	4.0e-13			1.6e-2
	QMC - 520,000	1.6e-6	4.1e-5	**1.1e-3**	4.6e-4			1.9e-6
	QMC - 550,000	1.0e-6	1.5e-5	4.9e-4	5.4e-4			1.4e-6

6 Solution of Model Equations

This chapter describes the way the model equations $F(y, u, \xi) = 0$ can be solved for y given controls $u \in U$ and uncertain inputs $\xi \in \Omega$. The case that $F(y, u, \xi) = 0$ can be explicitly solved for the state variables y is trivial. Therefore, the following discussion is limited to problems where this is not the case. The usual approach to such problems is the usage of (generalized) Newton methods, which are described below.

6.1 Newton's method

The main idea of the Newton (also Newton-Raphson) method is to use a first order Tailor approximation of the function $F(y, u, \xi)$ to determine a root of the function. More clearly, given a starting point $y_0 \in \mathbb{R}^n$ and using the first order approximation

$$\hat{F}(y, u, \xi) = F(y_0, u, \xi) + F_y(y_0, u, \xi)(y - y_0), \tag{6.1.1}$$

where F_y is a short-hand notation for $\frac{\partial F}{\partial y}$, a new approximation of the root can be obtained by calculating a root of (6.1.1), which is given by

$$y_1 = y_0 - F_y(y_0, u, \xi)^{-1} F(y_0, u, \xi).$$

Repeatedly applying the above approximation leads to

$$y_{k+1} = y_k - F_y(y_k, u, \xi)^{-1} F(y_k, u, \xi), \ k \in \mathbb{N}, \tag{6.1.2}$$

which results in a sequence $(y_k)_{k \in \mathbb{N}}$ being generated, assuming that $F_y(y_k, u, \xi)$ is invertible for $k \in \mathbb{N}$. The next theorem gives assumptions under which the sequence converges towards a root of $F(y, u, \xi)$. Here, $\|\cdot\|$ denotes either the operator norm (for matrix arguments) or the Euclidean norm (for vector arguments).

Notation 6.1.1. Let $y_0 \in \mathbb{R}^n$ and $r > 0$. By $B_r(y_0)$ we denote an open ball of radius r around y_0 in the Euclidean norm, i.e.,

$$B_r(y_0) = \left\{ y \in \mathbb{R}^n \mid \|y - y_0\|_2 < r \right\}.$$

Theorem 6.1.2 (Convergence of Newton's method, [83] p. 270)**.** *Let $C \subset \mathbb{R}^n$ be a given open set. Further, let C_0 be a convex set with* cl $C_0 \subset C$*, and let $F : C \times \mathcal{U} \times \Omega \to \mathbb{R}^n$, $(y, u, \xi) \mapsto F(y, u, \xi)$ be a given function, which is differentiable with respect to y on C_0 for all $u \in \mathcal{U}$ and $\xi \in \Omega$, and continuous on $C \times \mathcal{U} \times \Omega$.*

For $y_0 \in C_0$ let positive constants r, α, β, γ, h be given with following properties:

$$B_r(y_0) \subset C_0$$
$$h := \frac{\alpha\beta\gamma}{2} < 1$$
$$r := \alpha/(1-h)$$

and for arbitrary but fixed $u \in \mathcal{U}$ and $\xi \in \Omega$ let $F(y, u, \xi)$ have the properties

(i) $\|F_y(\hat{y}, u, \xi) - F_y(\tilde{y}, u, \xi)\| \leq \gamma\|\hat{y} - \tilde{y}\|$ *for all $\hat{y}, \tilde{y} \in C_0$;*

(ii) $F_y(y, u, \xi)^{-1}$ *exists and satisfies* $\|F_y(y, u, \xi)^{-1}\| \leq \beta$ *for all $y \in C_0$;*

(iii) $\|F_y(y_0, u, \xi)^{-1}F(y_0, u, \xi)\| \leq \alpha$.

Then

(i) *beginning at y_0, each point*

$$y_{k+1} = y_k - F_y(y_k, u, \xi)^{-1}F(y_k, u, \xi),\ \ k \in \mathbb{N},$$

is well defined and satisfies $y_k \in B_r(y_0)$ for all $k \geq 0$,

(ii) $\lim_{k\to\infty} y_k = \bar{y}$ *exists and satisfies $\bar{y} \in$ cl $B_r(y_0)$ and $F(\bar{y}, u, \xi) = 0$,*

(iii) *for all $k \geq 1$*

$$\|y_k - \bar{y}\| \leq \alpha\frac{h^{2k-1}}{1-h^{2k}}.$$

Since $0 < h < 1$, Newton's method is at least quadratically convergent.

It is clear from the previous theorem that convergence of Newton's method can only be expected when starting in a sufficiently small neighborhood of the root. Several methods to overcome this problem have been proposed. One possible approach is the usage of a line search method, i.e., instead of (6.1.2) the iteration

$$y_{k+1} = y_k - \mu_k F_y(y_k, u, \xi)^{-1}F(y_k, u, \xi),\ \ k \in \mathbb{N}, 0 < \mu_k < 1,$$

is used for suitable choices of the parameter μ_k.

6.2 Approximation approaches

Approximation methods are actually not a method to find an exact solution to a set of equations, but rather a means to find a suitable approximation of such solution. As a consequence, they are not a replacement of Newton's method. Nonetheless, under certain circumstances they can significantly reduce the computational burden in the solution of CCOPT problems. The methods described below are best employed in conjunction with the AA approach proposed by Geletu et al. [34] (see also Chapter 4). As guaranteed by Proposition 4.4.10 (v)

$$\lim_{\tau \to 0^+} \Theta(\tau, u, s) = \begin{cases} 1, & \text{if } s \geq 0, \\ 0, & \text{if } s < 0, \end{cases}$$

uniformly for $u \in \mathcal{U}$ and uniformly for $s \in (-\infty, -\epsilon) \cup [0, \infty)$ and for given $\epsilon > 0$. This means that whenever $g(u, \xi) \geq 0$ or $g(u, \xi) < -\epsilon$ the function $\Theta(\tau, u, g(u, \xi))$, used in the approximation, tends to 1 or 0, respectively. Using an suitably small value of τ , a sufficiently large value of ϵ, and an approximation $\hat{g}(u, \xi)$ of $g(u, \xi)$ with a maximum guaranteed approximation error of ϵ_{approx} leads to the following algorithm:

Algorithm 6.2.1. In the evaluation of $\psi_G(\tau, u)$ use

$$\hat{\Theta}(\tau, u, g(u, \xi)) = \begin{cases} 0, & \hat{g}(u, \xi) < -\epsilon - \epsilon_{approx} \\ \Theta(\tau, u, g(u, \xi)), & -\epsilon - \epsilon_{approx} \leq \hat{g}(u, \xi) \leq \epsilon_{approx} \\ 1, & \hat{g}(u, \xi) > \epsilon_{approx} \end{cases} .$$

Proof: Since the approximation $\hat{g}(u, \xi)$ has a maximum error of ϵ_{approx}, $\hat{g}(u, \xi) < -\epsilon - \epsilon_{approx}$ guarantees that $g(u, \xi) < -\epsilon$ and, therefore, $\lim_{\tau \to 0^+} \Theta(\tau, u, g(u, \xi)) = 0$. The same reasoning can be applied in the case that $\hat{g}(u, \xi) > \epsilon_{approx}$, which yields the desired result. □

Assuming that the approximation algorithm is faster than the solution of the model equations by Newton's method the following approach can be used. Whenever the probability of holding the constraints is computed by means of a numerical integration of

$$\int_{\Omega} \Theta(\tau, u, g(u, \xi)) d\xi$$

the first step is to determine $\hat{g}(u, \xi_k)$ for all grid points ξ_k, $k = 1, \ldots, N$ in a suitable integration rule $I[\cdot]$. The grid points ξ_k can be grouped into three sets:

(i) $S_{<-\epsilon-\epsilon_{approx}} = \{k \mid g(u, \xi_k) < -\epsilon - \epsilon_{approx}\}$,

(ii) $S_{>\epsilon_{approx}} = \{k \mid g(u, \xi_k) > \epsilon_{approx}\}$,

(iii) $S_{other} = \{k \mid -\epsilon - \epsilon_{approx} \leq g(u, \xi_k) \leq \epsilon_{approx}\}$.

The corresponding integration routine then becomes

$$I[\hat{\Theta}(\tau, u, g(u, \xi)] = \sum_{i \in S_{>\epsilon_{approx}}} \omega_i + \sum_{i \in S_{other}} \omega_i \Theta(\tau, u, g(u, xi)),$$

which can possibly be evaluated much faster than the original integral $I[\Theta(\tau, u, g(u, \xi))]$, depending of course on the actual sizes of the sets $S_{<-\epsilon-\epsilon_{approx}}$ and $S_{>\epsilon_{approx}}$. In the following we will shortly examine two methods of approximation.

6.2.1 Artificial Neural Networks

Artificial Neural Network (ANN) are a concept of machine learning and the main idea is to emulate the working of a human brain by using a somewhat similar structure. Here, we will only consider so called feed-forward networks, since a general description of ANN is out of the scope of this work. A feed-forward network generally consists of three type of neurons (input, hidden, and output) and connections between these neurons. A schematic of such network is shown in Figure 6.1. From the mathematical point of view, a neural network is a mapping $F_{ANN} : \mathbb{R}^n \to \mathbb{R}^m$, $m, n \in \mathbb{N}$, mapping the values of the input neurons onto the output neurons. This is done in several steps, depending on the amount of layers in the network. Let i=$0, \dots, M$ number the single layers, where layer 0 contains the input neurons and layer M contains the output neurons. Let further N_i be the number of neurons in layer i, $i = 0, \dots, M$. All layers, except layer 0, describe a mapping $F_i : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$, where $F_i(x) = (F_{i,1}(x), \dots F_{i,N_i}(x))^T$. The j-th neuron in layer i then calculates the entry $F_{i,j}(x^i)$. The single functions $F_{i,j}(x^i)$ can be further decomposed into an activation function $A_{i,j} : \mathbb{R}^{N_{i-1}} \to \mathbb{R}$, and a transfer function $T_{i,j} : \mathbb{R} \to \mathbb{R}$ with $F_{i,j}(x^i) = T_{i,j}(A_{i,j}(x^i))$. The whole network output is then described by

$$F_{ANN}(x) = F_M\left(F_{M-1}\left(F_{M-2}\left(\dots F_1(x)\dots\right)\right)\right).$$

The activation functions $A_{i,j}$ are typically weighted summation, i.e.,

$$A_{i,j}(x^i) = \sum_{k=1}^{N_{i-1}} w_{i,j,k} F_{i-1,k}(x^{i-1}),$$

where $w_{i,j,k} \in \mathbb{R}$ for $1 \leq i \leq M$, $1 \leq j \leq N_i$ and $1 \leq k \leq N_{i-1}$. Transfer functions can be any at least smooth functions, e.g., sigmoidal functions.

Before an ANN can be used for the approximation of a function $F : \mathbb{R}^n \to \mathbb{R}^m$ it has to be "trained". For this step precomputed pairs (x_i, y_i) with $y_i = F(x_i)$, $i = 1, \dots, L$ are required. The training now consists of finding optimal values for the weights $w_{i,j,k}$. This can be done in several ways. One way consists of solving a least square problem of the following form

$$\min_{w_{i,j,k}} \sum_{k=1}^{L} |F(x_i) - F_{ANN}(x_i)|^2,$$

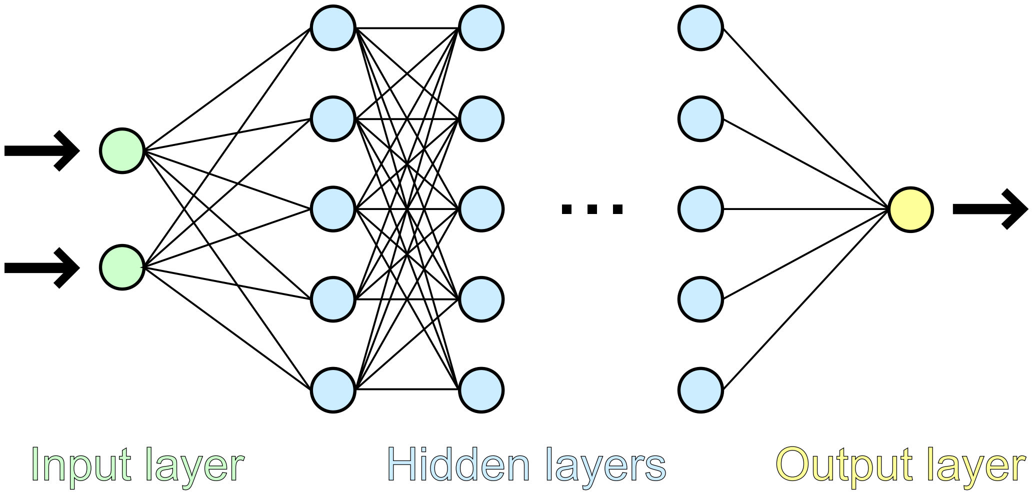

Figure 6.1: *Schematic of a feed-forward network*[1]

another approach is the so called back propagation [72], Chapter 7. Assuming that the number of layers is low enough and all involved functions are given in a closed form, the output of an ANN can be obtained reasonably fast, since only arithmetic operations have to be carried out. The next theorem, proposed by Cybenko [20], guarantees that for continuous functions F on the unit hypercube we can always find a feed-forward network with only one hidden layer and an approximation error less than ϵ_{approx}.

Definition 6.2.2 (Discriminatory function, [20])**.** We say that $\sigma : \mathbb{R} \to \mathbb{R}$ is discriminatory if for a measure μ

$$\int_{[0,1]^n} \sigma(w^T x + \theta) d\mu(x) = 0$$

for all $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ implies that $\mu \equiv 0$.

Remark 6.2.3. One common discriminatory function used in the context of ANN is

$$\sigma(x) = \frac{1}{1 + e^{-\beta x}}$$

for some $\beta \in \mathbb{R}_{++}$.

Theorem 6.2.4 (Cybenko, [20])**.** *Let σ be any continuous discriminatory functions. Then finite sums of the form*

$$F_{ANN}(x) = \sum_{j=1}^{N} \alpha_j \sigma(w_j^T x + \theta_j),$$

[1] This figure was created based on a work of Mysid Dake (`http://commons.wikimedia.org/wiki/File:Neural_network.svg`).

where $\alpha_j, \theta_j \in \mathbb{R}$ for $j = 1, \ldots, n$ and $w_j \in \mathbb{R}^n$ for $j = 1, \ldots, n$, are dense in $C\left([0,1]^m\right)$. In other words, given any $F \in C\left([0,1]^m\right)$ and $\epsilon_{approx} > 0$, there is a sum, $G(x)$, of the above form, for which

$$|F(x) - F_{ANN}(x)| \leq \epsilon_{approx}$$

for all $x \in [0,1]^n$.

6.2.2 (Generalized) Fourier series

The second approach of approximation consists of using a series of orthogonal functions (like the polynomials constructed in Gaussian quadrature), which form a basis of a suitable function space. One commonly known method of this type is the Fourier series expansion [95], Chapter 18, which is defined in the space $L_2([-\pi, \pi])$ of all square-integrable functions on the interval $[-\pi, \pi]$. The corresponding basis functions are $\psi_0 \equiv 1$, $\psi_n^S(x) = \sin(nx)$, and $\psi_n^C(x) = \cos(nx)$, $n \in \mathbb{N}$. Any function $f \in L_2\left[-\pi, \pi\right]$ can now be approximated using

$$s_N(x) = \frac{\langle f, \psi_0 \rangle}{\langle \psi_0, \psi_0 \rangle}\psi_0(x) + \sum_{i=1}^{N} \left(\frac{\langle f, \psi_i^S \rangle}{\langle \psi_i^S, \psi_i^S \rangle}\psi_i^S(x) + \frac{\langle f, \psi_i^C \rangle}{\langle \psi_i^C, \psi_i^C \rangle}\psi_i^C(x) \right)$$

for an $N \in \mathbb{N}$ and

$$\langle f, g \rangle = \int_{-\pi}^{\pi} f(x)g(x)dx,$$

for $f, g \in L_2\left[-\pi, \pi\right]$. The next theorem gives conditions for the uniform convergence of this approximation.

Theorem 6.2.5 (Zorich [95], p. 542). *If the function $F : [-\pi, \pi] \to \mathbb{R}$ is such that*

(i) *$f \in C^{(m-1)}\left[-\pi, \pi\right]$, $m \in \mathbb{N}$,*

(ii) *$f^{(j)}(-\pi) = f^{(j)}(\pi)$, $j = 1, \ldots, m-1$,*

(iii) *f has a piecewise continuous m-th derivative $f^{(m)}$ on $[-\pi, \pi]$, $m \geq 1$,*

then the Fourier series of f converges absolutely and uniformly on $[-\pi, \pi]$ to f, and the deviation of the n-th partial sum $s_N(x)$ of the Fourier series from $f(x)$ has the following estimate on the entire interval:

$$|f(x) - s_N(x)| \leq \frac{\epsilon_N}{N^{m-\frac{1}{2}}},$$

where $\{\epsilon_N\}$ is a sequence of positive numbers tending to zero.

The main problem with this kind of approximation is the fact that uniform convergence is only guaranteed for 2π-periodic functions.

Remark 6.2.6. Approximations can be also defined on a basis of orthogonal polynomials on suitable intervals $[a, b]$. The main concern with these approximations is that uniform convergence of the approximation towards the original function can generally only be shown for compact intervals inside (a, b), see for example [46, 77].

One way to overcome this problem was proposed by Adcock [1]. He used a modified Fourier series expansion for approximation on the interval $[-1, 1]$ with $\bar{\psi}_0^{[0]} \equiv \frac{1}{\sqrt{2}}$, $\bar{\psi}_n^{[0]}(x) = \cos(n\pi x)$, and $\bar{\psi}_n^{[1]}(x) = \sin\left(\left(n - \frac{1}{2}\right)\pi x\right)$ and

$$\bar{s}_N(x) = s_N(x) = \frac{\langle f, \bar{\psi}_0^{[0]}\rangle}{\langle \bar{\psi}_0^{[0]}, \bar{\psi}_0^{[0]}\rangle}\bar{\psi}_0^{[0]}(x) + \sum_{i=1}^{N}\left(\frac{\langle f, \bar{\psi}_i^{[0]}\rangle}{\langle \bar{\psi}_i^{[0]}, \bar{\psi}_i^{[0]}\rangle}\bar{\psi}_i^{[0]}(x) + \frac{\langle f, \bar{\psi}_i^{[1]}\rangle}{\langle \bar{\psi}_i^{[1]}, \bar{\psi}_i^{[1]}\rangle}\bar{\psi}_i^{[1]}(x)\right).$$

In the following theorem we will see that this kind of approximation converges uniformly on $[-1, 1]$ regardless of periodicity.

Definition 6.2.7 (Weak derivative, [48] p. 266)**.** Let $\Omega \subset \mathbb{R}^d$ be open, $f \in L^1_{loc}(\Omega) := \{f \in L^1(\Omega') \mid \Omega' \subset \Omega \text{ compact}\}$. A function v is called the weak (partial) derivative of f in the direction ξ_j if

$$\int_\Omega v(\xi)u(\xi)d\xi = -\int_\Omega f(\xi)\frac{\partial}{\partial \xi_j}u(\xi)d\xi$$

holds for all $u \in C_0^1(\Omega) = \{f \in C^1(\Omega) \mid \text{supp} f \subset \Omega\}$.

Definition 6.2.8. The space $H^1[-1, 1]$ is defined by

$$H^1[-1,1] = \left\{ f \in L^2[-1,1] \mid \int_{-1}^{1} f^2(x) + \left(\frac{\partial}{\partial x}f(x)\right)^2 dx < \infty \right\},$$

where $\frac{\partial}{\partial x}f$ is the weak derivative of f.

Theorem 6.2.9 (Adcock [1])**.** *Suppose that $f \in H^1[-1, 1]$. Then $\|f - \bar{s}_N\|_\infty \to 0$ as $N \to \infty$.*

Another advantage of this approach is that it can easily be extended to the multivariate case of approximating functions over the hypercube $[-1, 1]^d$. Using multi-indices $n = (n_1, \ldots, n_d) \in \mathbb{N}^d$ and $i = (i_1, \ldots, i_d) \subset \{0, 1\}^d$ the d-dimensional basis functions are given by

$$\psi_n[i] = \prod_{j=1}^{d} \psi_{n_j}^{[i_j]}(x_j),$$

where $x = (x_1, \ldots, x_d) \in [-1, 1]^d$. Given a finite index set $I_N \subset \mathbb{N}^d$ an approximation to $f : [-1, 1]^d \to \mathbb{R}$ can be obtained by

$$\hat{s}_N(x) = \sum_{i \in \{0,1\}^d} \sum_{n \in I_N} \langle f, \psi_n^{[i]}\rangle \psi_n^{[i]}(x).$$

Similar to the univariate case the uniform convergence of this approximation can be shown.

Definition 6.2.10.

$$H^1_{mix}[-1,1]^d = \left\{ f \in L^2[-1,1]^d \mid \frac{\partial}{\partial x_j} f \in L^2[-1,1]^d,\ \forall j = 1, \ldots, d \right\},$$

where $\frac{\partial}{\partial x_j} f$, $j = 1, \ldots, d$, are the weak partial derivatives of f.

Theorem 6.2.11 (Adcock [2])**.** *Suppose that $f \in H^1_{mix}[-1,1]^d$ and $I_N \subset \mathbb{N}^d$, $N \in \mathbb{N}$ satisfy $\bigcup_{N \geq 1} I_N = \mathbb{N}^d$ as well as $I_1 \subset I_2 \subset \ldots$. Then, $\hat{s}_N(x)$ converges point-wise to f for all $x \in [-1,1]^d$ as $N \to \infty$. Moreover, the convergence is uniform.*

Whereas in the neural network approach a high count of pairs (x_i, y_i) with $y_i = f(x_I)$ is necessary to train the network, Fourier approaches require a similar amount of function evaluations to determine the coefficients $\langle f, \psi \rangle$, which can be found by numerical integration, least square approaches or Fast Fourier Transform (FFT)[2]. As a consequence, approximation approaches are only suitable if the function f needs to be approximated sufficiently often or under time constraints in an online application.

[2] The authors of [45] argue that numerical integration is more suited for the task than FFT methods.

7 Applications and Numerical Experiments

7.1 Software and implementation

Purpose of this section is to describe, which software and what hardware was used to obtain the results described below. All problems were solved using a personal computer equipped with a hexacore I7-980X processor, 6 GiBi RAM, an NVIDIA GTX 470 consumer graphic card, and running Ubuntu Linux. The problems were implemented in C++ using IpOPT as optimizer. Routines from the GNU Scientific Library were used to generate (Q)MC grid points as well as to solve linear and nonlinear systems of equations. To speed up the computation, parallelization, based on OpenMP, was used. Additionally, GPU computing, based on CUDA, was employed for certain problems. Besides the usage of compiler flags (as "-O3" in the gcc compiler), parallelization, and the choice of suitable data structures no further attempts on optimizing the implementation were made.

7.1.1 Parallelization

Modern processors with more than one core (i.e., so called multicore processors) are MIMD (multiple instructions, multiple data) devices. As such, every core can carry out different computation tasks using totally different data sets as input. The limiting factors are the computation speed of the single cores (e.g., clock speed) and the time it takes to transport data from the main memory to the processor cores. Multicore processor can be used to speed up the solution of CCOPT problems. The approach, taken in this thesis, is to parallelize the evaluation of integrals using OpenMP[1]. As the evaluation on a single grid point does not depend on information from any other grid point this avoids the appearance of so called racing conditions, which appear if one thread in a parallel execution depends on data processed by another thread. One problem in connection with parallel execution is the inherent non-determinism. It occurs since threads are distributed differently over the distinct processor cores every time a parallel part of a program is executed. Since the order of execution is no longer deterministic, problems may occur for example when summing up results of the single grid points using a reduction (one method of summing up the results of different iterations of a loop in OpenMP). Due

[1] openmp.org - visited 31.07.2013

to the changing order in the summation of the results, different rounding occurs, leading to non-deterministic results when the same program is executed repeatedly.

In contrast to processors, graphic cards are so called SIMD (single instruction, multiple data) devices. These cards usually contain a larger number of processor cores (typically 128–2048). The main characteristic is that all processors have to carry out the same operation (or do nothing), leading to a more restricted area of application. In this work, parallelization on an NVDIA graphics card using CUDA [76] was employed.

7.1.2 Implementation

The main difficulty when implementing the AA approach is the choice of a suitable parameter τ. One has to keep in mind that smaller values of τ typically require a more refined integration method, i.e., a method using more grid points, thereby increasing the computation time. In the case studies, the following approach to the choice of τ was used. In a first step an integration routine was generated. Then, by experiment, the smallest value of τ for which the generated integration rule gave reliable results was determined. If the resulting value of τ was deemed too large (e.g., $\tau > 0.001$), the integration method was refined and the whole process of determining the minimal parameter τ was repeated. The whole process was repeatedly carried out until a suitable integration method, allowing a sufficiently small value of τ, was found. This value of τ was the used one throughout the optimization.

7.2 Numerical experiments

The main contribution of this work is the introduction of a novel Analytical Approximation approach (see Chapter 4) and suitable methods for the computation of the corresponding gradients. Furthermore, an approach to decrease the computation time was proposed. In the following, a comparison of the proposed approach with other methods for solving CCOPT problems is conducted. The test problems include applications in finance, chemical process engineering and Reliability Based Design Optimization (RBDO). The first four experiments were not previously published, whereas the last case study was partly published in [35].

We begin with four academic standard examples. As integration methods, suitable (Q)MC rules based on the Sobol sequence are used. The parameters for the AA approach are $\tau_{max} = 0.0005$, $\tau = 0.0004$, $m_1 = 1.0005$, $m_2 = 1$.

7.2.1 Cattle feed problem

The first test problem is a cattle feed problem proposed and solved in [22]. It consists of finding a cost optimal mix of ingredients (barley, oats, sesame flake, and groundnut meal) which satisfies certain constraints on the nutritional content (in this case protein and fat) of the mix. In addition, the content of protein in the single ingredients was assumed to be normally distributed and independent of the protein content in the other ingredients (see Table 7.1 for

Table 7.1: *Data for the cattle feed problem*

Variable	Ingredient	Expected protein content	Variance in protein content	Fat content	Price per ton
X_1	Barley	12.0	0.2809	2.3	24.55
X_2	Oats	11.9	0.1936	5.6	26.75
X_3	Sesame flakes	41.8	20.2500	11.1	39.00
X_4	Groundnut meal	52.1	0.6241	1.3	40.50

Table 7.2: *Comparison of optimization results for the cattle feed problem, plain text numbers represent the results found in [68], results in italic numbers were obtained using the AA approach*

α	X_1	X_2	X_3	X_4	Cost	True probability
0.9	0.6269	0.0100	0.3089	0.0515	29.86	0.944
	0.6408	*0.0100*	*0.3078*	*0.0414*	*29.68*	*0.9004*
0.95	0.6127	0.0100	0.3106	0.0666	30.12	0.979
	0.6273	*0.0100*	*0.3092*	*0.0536*	*29.89*	*0.9503*
0.98	0.5935	0.0100	0.3126	0.0839	30.43	0.995
	0.2933	*0.3939*	*0.1748*	*0.1380*	*30.14*	*0.9802*
0.99	0.0100	0.6995	0.0696	0.2209	30.62	0.999
	0.2101	*0.4874*	*0.1423*	*0.1602*	*30.23*	*0.9901*

the details). The corresponding optimization problem is given as

$$\begin{aligned}
\min\quad & 24.55X_1 + 26.75X_2 + 39.00X_3 + 40.50X_4 && (7.2.1)\\
s.t.\quad & X_1 + X_2 + X_3 + X_4 = 1 && (7.2.2)\\
& 2.3X_1 + 5.6X_2 + 11.1X_3 + 52.1X_4 \geq 5 && (7.2.3)\\
& Pr\left\{21 - \xi_1 X_1 + \xi_2 X_2 + \xi_3 X_3 + \xi_4 X_4 \leq 0\right\} \geq \alpha && (7.2.4)\\
& X_1, X_2, X_3, X_4 \geq 0.01. && (7.2.5)
\end{aligned}$$

Due to the Gaussian distribution of the uncertainties and the linearity of the problem, the chance constraint could be transformed into an equivalent *nonlinear* deterministic constraint. In order to obtain a *linear* deterministic problem, the authors of [68] proposed a linear approximation of the chance constraint. A comparison of the results obtained by the linear approximation and the analytic approximation is given in Table 7.2. It can be seen that both approaches underestimate the true probability value. Nonetheless, the analytic approximation approach results in a tighter bound of the probability values and, consequently, better solutions of the cattle feed problem. The solution of this problem needs five seconds when using parallel computation on the CPU. A comparison with the approach in [68] is not possible, since no performance data was presented there.

7.2.2 Portfolio optimization problem

This problem was considered in [5] and is concerned with the optimal investment of borrowed capital. At the beginning of a time period a capital is borrowed which has to be payed off at the end of this period with an interest rate l. The borrowed capital can be invested at a fixed rate b or at a uncertain rate ξ ($E[\xi] > b$). Furthermore, a part of the capital can be consumed resulting in a satisfaction measured by a concave non-decreasing function f. Goal of the optimization is to maximize the sum of the satisfaction and the expected capital return, subject to the constraint that the borrowed capital and the interest rate can be paid off at the end of the period, at least with a probability α. The optimization problem is stated as follows

$$\begin{aligned} \min \quad & -s(1-u-v)-(1+b)u-(1+E[\xi])v && (7.2.6)\\ s.t. \quad & u+v \leq 1 && (7.2.7)\\ & Pr\{1+l-(1+b)u-(1+\xi)v \leq 0\} \geq \alpha && (7.2.8)\\ & u,v \geq 0, && (7.2.9) \end{aligned}$$

where u describes the amount of capital invested at the fixed rate b, v describes the amount of capital invested at the uncertain rate ξ and the function $s : \mathbb{R} \to \mathbb{R}$ given by

$$s(x) = -\frac{x^2}{2} + 2x$$

describes the satisfaction brought by spending the amount x. The parameters are given as $b = 0.2$, $l = 0.15$ and the cdf of ξ is

$$\Phi(\xi) = \begin{cases} 0 & \xi < -2.6, \\ \frac{1}{16}\left(3\left(\frac{\xi-0.4}{3}\right)^5 - 10\left(\frac{\xi-0.4}{3}\right)^3 + 15\left(\frac{\xi-0.4}{3}\right) + 8\right) & -2.6 \leq \xi \leq 3.4, \\ 1 & \xi > 3.4. \end{cases}$$

One special property of this optimization problem is that for $\alpha \geq 0.7$ an investment at the uncertain rate ξ becomes infeasible, i.e., in this case the chance constraint is reduced to a deterministic constraint, which is satisfied as long as $u \geq 0.95833$. The proposed approach is able to deal with this pathology, i.e., for $\alpha = 0.7$ the optimal values of the decision variables are $u^* = 0.961$ and $v^* = 0$. For $\alpha = 0.24$ the authors of [5] report the optimal values $u^* = 0$ and $v^* = 0.504$ with objective function value -1.5746, whereas the proposed approach leads to optimal values $u^* = 0$ and $v^* = 0.506$ with objective function value -1.5744. Although, the result reported in [5] is better by a small margin, it should be noted that the proposed AA approach does not require any special adaption to neither the pathologies of the optimization problem nor the non-standard distribution of the uncertain variable. The computations require less than one second (using parallelization on the CPU). A comparison with the approach in [5] is not possible, since no performance data was given there.

7.2.3 Multidisciplinary design optimization: Maximum distance problem

Multidisciplinary Design Optimization (MDO) is concerned with the solution of design optimization problems in the case that more than one (engineering) discipline is required to describe the underlying system. Commonly, models from two or more disciplines are coupled to describe the system behavior, leading to possible computational problems when trying to solve the complete system at once. Here, a maximum distance problem as presented in [18] is considered. The problem is defined as

$$\begin{aligned} \min \quad & -x_2 && (7.2.10)\\ s.t. \quad & h_1(x,\xi,u,v) = \xi_2 x_1 + 2x_2 - u + v = 0 && (7.2.11)\\ & h_2(x,\xi,u,v) = 3x_1 - u - v = 0 && (7.2.12)\\ & Pr\left\{-\xi_1 + u(x,\xi) - \frac{1}{2}(\xi_2+1)x_1 \le 0\right\} \ge \alpha && (7.2.13)\\ & Pr\left\{-v(x,\xi) \le 0\right\} \ge \alpha && (7.2.14)\\ & x_1, x_2 \ge 0, && (7.2.15) \end{aligned}$$

where $h_1(\cdot,\cdot,\cdot,\cdot)$ and $h_2(\cdot,\cdot,\cdot,\cdot)$ are the model equations associated with different disciplines and $\xi = (\xi_1, \xi_2)$ are multivariate Gaussian distributed uncertainties with $E[\xi] = (1,1)$ and $\Sigma = I_2$, where I_n is the $n \times n$ identity matrix. The quantities $u(x,\xi)$ and $v(x,\xi)$ are so called intermediate or state variables. The probability level is set to $\alpha = 0.9987$. The optimal solution as derived by Chiralaksanakul and Mahadevan [18] with several different methods is $x^* = (0.378, 0.322)$. The AA approach results in the optimal values $x^* = (0.378, 0.318)$. Again, the result obtained through analytical approximation differs by a small margin from the optimal value. The solution of the problem is computed within one second using parallelization on the CPU. Like above, a comparison of the performance is not possible.

In the context of MDO the proposed AA approach is a so called all-at-once approach, i.e., all model equations are solved simultaneously. This may lead to increased computational costs in comparison to specialized approaches. Nonetheless, the proposed approach is a good choice for a general purpose solver, since it is easy to implement and no in-depth knowledge of the involved disciplines is necessary.

7.2.4 Multidisciplinary design optimization: Design of a short column

This problem is concerned with the optimal design of a short structural column and is widely used as a benchmark for numerical methods in RBDO. The objective is to design a structural column with minimal cross section $b \times h$, which is able to withstand certain stresses (so called oblique bending). The problem formulation as presented in [74] is

$$\begin{aligned} \min \quad & bh && (7.2.16)\\ s.t. \quad & \frac{1}{2} \le \frac{b}{h} \le 2 && (7.2.17)\\ & G(b,h,\xi) = 1 - \frac{4\xi_1}{bh^2\xi_4} - \frac{4\xi_2}{b^2hy} - \left(\frac{\xi_3}{bh\xi_4}\right)^2 && (7.2.18)\\ & Pr\left\{-G(b,h,\xi) \le 0\right\} \ge \alpha && (7.2.19)\\ & b, h \ge 0, && (7.2.20) \end{aligned}$$

where b and h are breadth and height of the column, respectively, and $\xi = (\xi_1, \ldots, \xi_4)$ are statistically independent uncertain variables. Details on these variables can be found in Table 7.3. Before presenting the results a similar problem formulation, considered in [24], is introduced. In addition to minimizing the cross sectional area a penalty term for violating the constraint is added. Furthermore, breadth and width are also considered as independently normally distributed random variables ξ_5 and ξ_6 with expectation b and h, respectively, and coefficient of variation 0.01. The corresponding optimization problem is

$$\min \quad bh(1 + 100Pr\{G(\xi) \leq 0\}) \tag{7.2.21}$$

$$s.t. \quad \frac{1}{2} \leq \frac{b}{h} \leq 2 \tag{7.2.22}$$

$$G(\xi) = 1 - \frac{4\xi_1}{\xi_5\xi_6^2\xi_4} - \frac{4\xi_2}{\xi_5^2\xi_6} - \left(\frac{\xi_3}{\xi_5\xi_6\xi_4}\right)^2 \tag{7.2.23}$$

$$Pr\{-G(\xi) \leq 0\} \geq \alpha \tag{7.2.24}$$

$$b, h \geq 0. \tag{7.2.25}$$

The optimization results for both problems can be found in 7.4. For the first problem one can find the usual result, i.e., the direct solution of the problem leads to a better objective function value, but again only by a small margin. Considering the second problem, the proposed AA algorithm actually results in a lower objective function value in comparison to the kriging method presented in [24]. One cause for this might be that the kriging method uses only a small number of model function evaluations and relies on interpolation, introducing conservative approximations of the chance constraints. The computation times are 22 seconds for the first problem and 146 seconds for the second problem. In both cases parallelization on the CPU was employed. Although, no information on the computation time was presented, [24] gives the number of function calls made by different approaches for the second problem. The kriging method requires only 140 model function evaluations, whereas the employed AA approach requires 5.16×10^8 evaluations. To put this number into a context, it should be mentioned that the approach of Dubourg et al. also needs 1.9×10^7 evaluations when used without kriging. This gives rise to the question whether kriging could be used together with the AA approach. Similar to the approximation approaches in Chapter 6, kriging replaces the model equations by a so-called emulator, which is much easier to evaluate. Unlike the other approaches, kriging assumes that the solution of the model equations can be described by a Gaussian process, an assumption, which does not hold in general CCOPT problems. Therefore, kriging can generally not be used in conjunction with the AA approach.

7.2.5 Chemical Process Engineering under Uncertainty

In the last case study a batch reactor with a series reaction of the form

$$A \rightarrow B \rightarrow C$$

is considered, where species B is the desired product, but the reaction of B into C cannot be totally avoided. A similar process was considered as a case-study by [79]. There, only

Table 7.3: *Data for the short column problem*

Variable	Meaning	Distribution	Expectation	Coefficient of variation
ξ_1	biaxial bending moment	log-normal	250 kNm	0.3
ξ_2	biaxial bending moment	log-normal	125 kNm	0.3
ξ_3	axial force	log-normal	2500 kN	0.2
ξ_4	material yield strength	log-normal	40 MPa	0.1

Table 7.4: *Comparison of the solutions for the short column problem*

Objective	b^*	h^*	Cost	Source
min bh	0.313	0.624	0.1953	[74]
min bh	0.313	0.626	0.1957	proposed app.
min $bh(1+100Pr\{G(\xi)\leq 0\})$	0.379	0.547	0.2166	[24]
min $bh(1+100Pr\{G(\xi)\leq 0\})$	0.326	0.630	0.2145	proposed app.

uncertainties in one of the pre-exponential Arrhenius factors were considered. In contrast, we consider uncertainties in both pre-exponential Arrhenius factors as well as in both activation energies. The principal configuration of a batch reactor can be seen in Figure 7.1.

The process in the reactor is described by a system of nonlinear differential equations

$$\dot{x_1} = -k_1 x_1^2 \tag{7.2.26}$$

$$\dot{x_2} = k_1 x_1^2 - k_2 x_2 \tag{7.2.27}$$

$$\dot{T} = \frac{(\Delta Hr_1 k_1 x_1^2 + \Delta Hr_2 k_2 x_2)V_p - hA_T(T-T_M)}{Vp\rho_p c_p} \tag{7.2.28}$$

$$\dot{T}_M = \frac{hA_T(T-2T_M+T_J)}{V_M\rho_M c_M} \tag{7.2.29}$$

$$\dot{T}_J = \frac{F_J\rho_J c_J(T_{J0}-T_J)+hA_T(T_M-T_J)}{V_J\rho_J c_J} \tag{7.2.30}$$

$$k_1 = k_{10}\exp(-\frac{E_1}{RT}) \tag{7.2.31}$$

$$k_2 = k_{20}\exp(-\frac{E_2}{RT}) \tag{7.2.32}$$

with initial conditions

$$\begin{aligned} x_1(0) &= 1, \\ x_2(0) &= 0, \\ T(0) &= T_M(0) = T_J(0) = 320K, \end{aligned}$$

where x_1 and x_2 are the concentration of species A and B, T, T_M and T_J are the temperatures of the reaction mass, wall, jacket, respectively. The k_1, k_2 describe the reaction rates, E_1, E_2 the

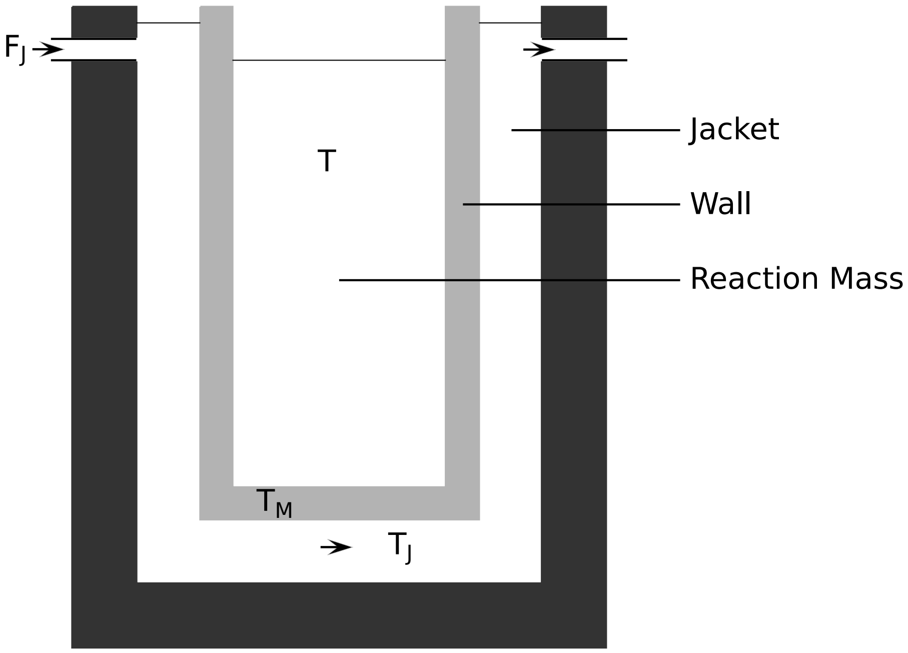

Figure 7.1: *Principal configuration of a batch reactor*

activation energy, k_{10}, k_{20} the pre-exponential Arrhenius factors and F_J is the amount of cooling water per hour. The physical parameters of the model are described in Table 7.5. The behavior of the temperature T for the deterministic system for different constant levels of F_J is shown in Figure 7.2. We assume that the variables E_1, E_2, k_{10} and k_{20} are time-independent and underlie a joint normal distribution with the parameters given in Table 7.6. This is motivated by the fact, that these four quantities are determined experimentally and therefore underlie measurement errors. Using an Euler discretization of the dynamic system (7.2.26)–(7.2.32), a chance constrained dynamic optimization problem can be formulated as

$$
\begin{aligned}
\min \quad & f(F_J, x_1, x_2, T, T_M, T_J, \hat{t}, h) && (7.2.33)\\
s.t. \quad & k_1(t) = k_{10} \exp\left(-\frac{E_1}{RT(t-1)}\right) && (7.2.34)\\
& k_2(t) = k_{20} \exp\left(-\frac{E_2}{RT(t-1)}\right) && (7.2.35)\\
& x_1(t) = x_1(t-1) - \Delta t(k_1(t)x_1^2(t-1)) && (7.2.36)\\
& x_2(t) = x_2(t-1) + \Delta t(k_1(t)x_1(t-1) - k_2(t)x_2(t-1)) && (7.2.37)\\
& T(t) = T(t-1)+ \\
& \quad \Delta t\left(\frac{(\Delta Hr_1 k_1(t)x_1^2(t-1) + \Delta Hr_2 k_2(t)x_2(t-1))V_p - hA_T(T(t-1) - T_M(t-1))}{V_p\rho_p c_p}\right) \\
& && (7.2.38)\\
& T_M(t) = T_M(t-1) + \Delta t\left(\frac{hA_T\left(T(t-1) - 2T_M(t-1) - Z_J(t-1)\right)}{V_M\rho_M c_M}\right) && (7.2.39)\\
& T_J(t) = T_J(t-1) + \Delta t\left(\frac{F_J(t)\rho_J c_J\left(T_{J0} - T_J(t-1)\right) + hA_T\left(T_M(t-1) - T_J(t-1)\right)}{V_J\rho_J c_J}\right) \\
& && (7.2.40)\\
& Pr\left\{T(t) \leq 328K\right\} \geq 0.8 && (7.2.41)\\
& t \in \left\{\hat{t}, \ldots, \hat{t}+h-1\right\} && (7.2.42)\\
& x_1(0) = 1, \quad x_2(0) = 0, \quad T(0) = T_M(0) = T_J(0) = 320K, && (7.2.43)
\end{aligned}
$$

where h is the length of the prediction horizon, $\hat{t}$ is the actual time interval, $\Delta t = 2min$ is the time interval used for the discretization and the other quantities are defined as above. Firstly, a deterministic objective function in the form

$$
f(F_J, x_1, x_2, T, T_M, T_J, \hat{t}, h) = \Delta F_J^T \Delta F_J + \sum_{t=\hat{t}}^{\hat{t}+h-1} F_J^2(t) \qquad (7.2.44)
$$

is considered, where the term $\Delta F_J = \left(F_J(\hat{t}) - F_J(\hat{t}-1), \ldots, F_J(\hat{t}+h-1) - F_J(\hat{t}+h-2)\right)^T$ describes the fluctuation in the cooling water inflow. We use a model predictive control scheme for the optimal control of the process. The objective of the optimization is to reduce the fluctuations in the cooling water stream (described by the first term in the objective function) and at the same time to minimize the amount of cooling water used (described by second term), whereas the temperature of the reaction mass should not exceed 328 K. It is visible from Figure 7.2 that reaction mass temperatures above 328K typically only occur within the first 60 minutes of the process. Therefore, this time period was chosen for optimization together with a prediction horizon $h = 8$. The results for one specific realization of the uncertain variables

Table 7.5: *Physical parameters of the batch reactor*

Parameter	Description	Value
ΔHr_1	heat of reaction $A \rightarrow B$	10000 cal/mol
ΔHr_2	heat of reaction $B \rightarrow C$	50000 cal/mol
V_p	volume of reaction mass	1.5 m^3
V_M	volume of wall	0.25 m^3
V_J	volume of jacket	1.7 m^3
ρ_p	density of reaction mass	700 kg/m^3
ρ_M	density of wall	8220 kg/m^3
c_J	specific heat of jacket	1 $kcal/kgK$
c_p	specific heat of reaction mass	0.5 $kcal/kgK$
c_M	specific heat of wall	0.12 $kcal/kgK$
ρ_J	density of jacket	1000 kg/m^3
h	heat transfer coefficient	100 $kcal/hK$
A_T	heat transfer area	10 m^2
T_{J0}	temperature of cooling water	297 K
R	gas constant	1.985 $cal/molK$

and using different approaches to the calculation of the probabilities (analytical approximation approach proposed by Nemirovski/Shapiro (called convex approximation for the remainder of the section) with sparse grid integration, analytical approximation proposed by Geletu et al. with (Q)MC integration, back-mapping with sparse grid integration) are shown in Figure 7.3. For the back-mapping approach the positive monotonic relationship $k_{20} \uparrow T$ was used. This relation is clear from the equations (7.2.37) and (7.2.38): A higher value of k_{20} leads to an increased reaction $B \rightarrow C$, which produces reaction heat and therefore increases reaction mass temperature as described in (7.2.38). As shown in Figure 7.3, the convex approximation approach uses a higher amount of cooling water, due to the underestimation of the probabilities of holding the constraints, leading to increased costs. Furthermore it should be noted, that no feasible solution could be found in the first five time horizons when using this approach. On the other hand, the cooling water stream profiles for the analytic approximation and back-mapping approach are similar and lead to similar costs (amounting to 5181.2 for the AA proposed by Geletu et al., 5551.1 for the back-mapping, and 15345.3 for the convex approximation, respectively.) The differences in the profiles can be mainly attributed to the behavior of the NLP solver used (fmincon from Matlab$^{\text{TM}}$).

The special structure of the given problem, i.e., state variables can be obtained explicitly, allows the usage of various parallel programming methods in the solution. Table 7.7 shows the computation time using a single core, multiple cores (based on OpenMP), and GPU computing (based on CUDA). Using the six processor cores, the computation was sped up by a factor of 4.5. The theoretic maximum speed up of 6 cannot be reached due to communication overhead. In comparison, CUDA allowed a speed up of a factor 47.6 in comparison with the single-threaded application and a factor 10.6 in comparison with the multi-threaded CPU implementation. It

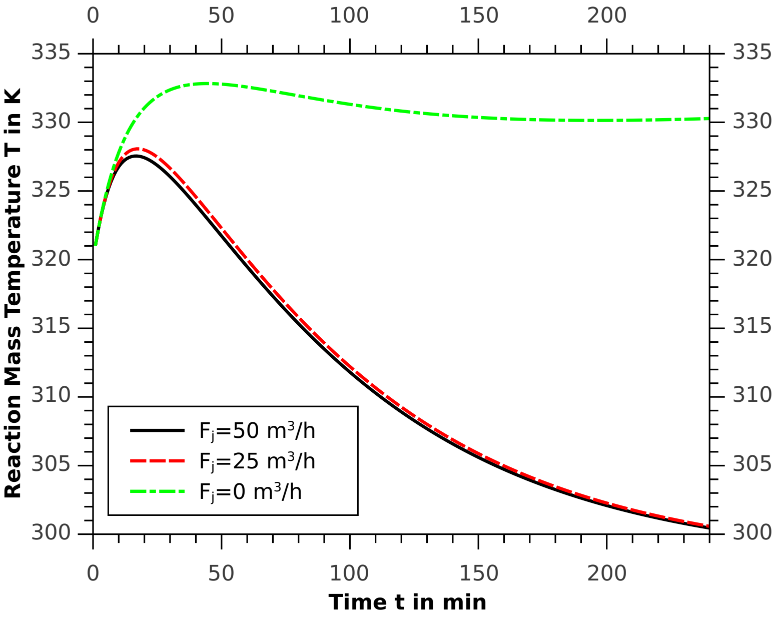

Figure 7.2: *Temperature T of the reaction mass for different levels of the feed F_J*

Variable (unit)	Expectation	Standard deviation	Covariance matrix			
E_1 (cal/mol)	4680	10.26	1	0.4	-0.6	0.1
E_2 (cal/mol)	11000	25.78	0.4	1	0.1	0.31
k_{10} $(l/molh)$	3730	25.67	-0.6	0.1	1	0.5
k_{20} $(1/h)$	480125	1000	0.1	0.31	0.5	1

Table 7.6: *Parameters for the uncertain inputs*

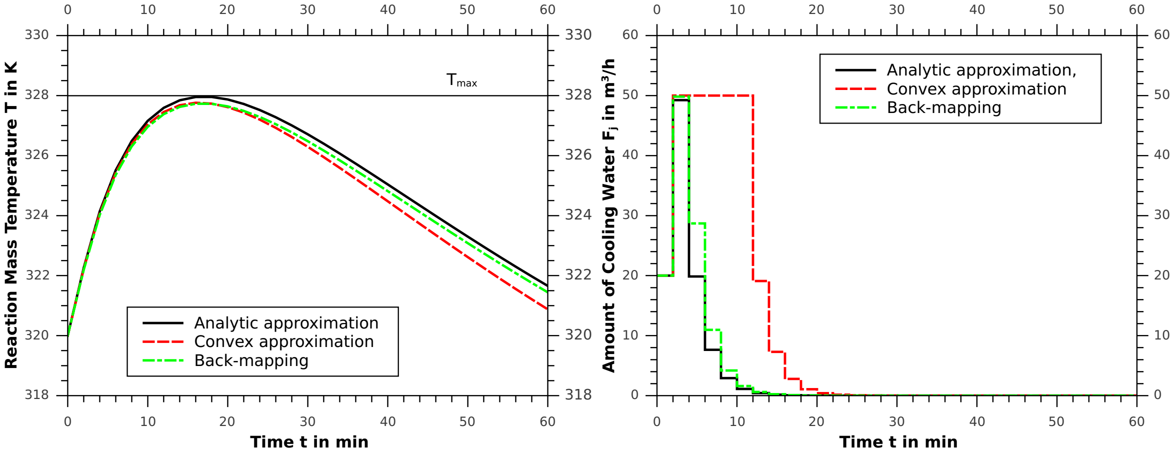

Figure 7.3: *Optimization results for the reactor problem and different approaches to the calculation of the chance constraints*

Method	Computation Time
Single-threaded CPU	238 s
Parallel CPU (OpenMP)	53 s
Parallel GPU (CUDA)	5 s

Table 7.7: *Computation time for 120 time steps and a prediction horizon of 15 time steps for several parallel programming methods using the AA approach of Geletu et al. and (Q)MC integration.*

	GTX 470	GTX Titan	C2050	I7-980X
Cores	448	2688	448	6 + Hyperthreading
Clock (MHz)	1215	837	1150	3300
Memory (MiBi)	1280	6144	3072	6144
Bandwidth (GB/s)	133.9	288.4	144	
Price (Euro)	190.00	880.00	1950.00	890.00

Table 7.8: *Comparison of several graphic cards and the processor employed.*

is interesting to note that the processor employed is still the second fastest (as of August 2013) available in the end user segment, whereas the graphics card is slow in comparison to today's models.

Table 7.8 shows a comparison between several graphics cards and the employed processor, which allows to make some predictions on the possible speed ups obtainable with current cards. Graphic cards data was obtained at *nvidia.com*, whereas data on the processor was obtained at *intel.com*. The C2050 is a card designated for the usage with CUDA. Prices were obtained using *geizhals.eu* on August the 10th, 2013.

Current consumer graphics cards (especially the 600 and 700 series) from NVIDIA (with exception of the Titan models) are not suitable for improving the computation speed, since the double precision computation capability of these cards is artificially restricted in the driver. Nevertheless, taking into account the information in Table 7.8, one could expect a speed up of about 500× in comparison to the single-threaded application when using a Titan model. Interestingly, such card costs even a little less than the CPU employed.

In order to show the viability of the AA approach of Geletu et al., an optimization problem with the stochastic objective function

$$\begin{aligned} f(F_J, x_1, x_2, T, T_M, T_J, \hat{t}, h) =& \omega_1 \left[\sum_{t=\hat{t}}^{\hat{t}+h-1} \big(-E\left[x_2(t)\right] + Var\left[x_2(t)\right] \big) \right] + \ldots \\ & \ldots + \omega_2 \left[F_J^T \Delta F_J + \sum_{t=\hat{t}}^{\hat{t}+h-1} F_J^2(t) \right], \end{aligned} \tag{7.2.45}$$

is solved under the same set of constraints (7.2.34)-(7.2.43). Here, ω_1 and ω_2 are weighting

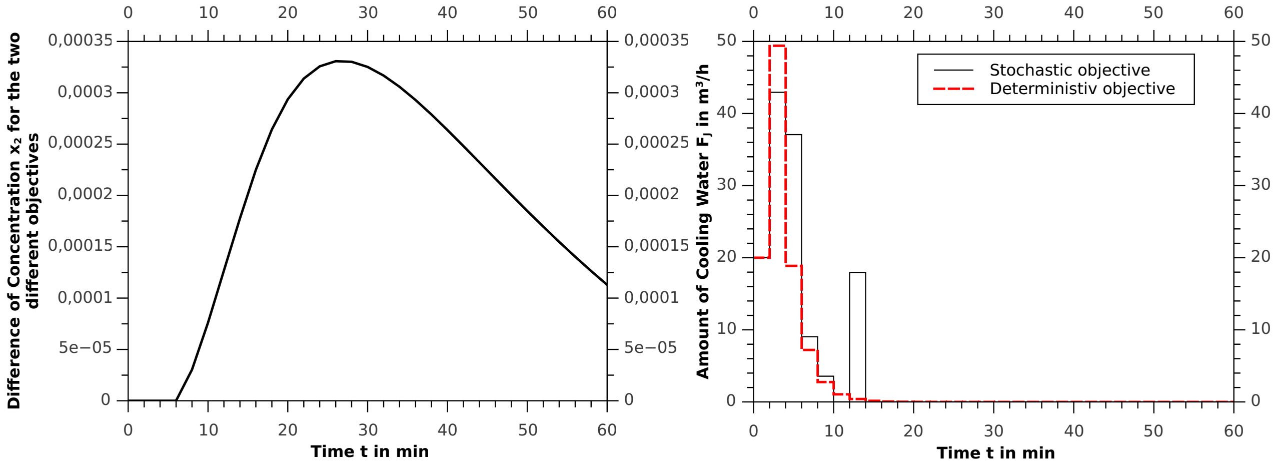

Figure 7.4: *Comparison of the results for optimization with deterministic and stochastic objective function using the AA approach.*

factors. This objective function is a weighted sum: between the function in (7.2.44) and the expected concentration of product B (i.e., $E[x_2(t)]$) while decreasing its variance ($Var[x_2(t)]$). For this problem only the AA approach was considered. The results of the optimization can be found in Figure 7.4. The left hand side presents the difference between the concentration of x_2 for the stochastic objective (7.2.45) and the deterministic objective (7.2.44). It can be seen, that the difference is always non-negative, showing a success in the optimization. The fact that the margin of difference is not very large can be attributed to the model used. The right hand side figure shows that there exist qualitative as well as quantitative differences in the cooling water stream profiles.

Since the presented problems have only four uncertain input variables (the corresponding sparse grid cubature rule for back-mapping uses only 441 grid points and the (Q)MC rule in the AA approach 500 grid points), the computation of probabilities, expectations and variances can be carried out without significant computational load. The typical computation time for the solution of both problems for an optimization horizon of 30 is well below one minute using the Matlab$^{\text{TM}}$optimization toolbox.

To compare the AA approach proposed by Geletu et al. with previously employed methods, the above problem with the objective function (7.2.45) is solved for an optimization horizon of 30 using AA and back-projection under Matlab$^{\text{TM}}$. The corresponding computation time, number of total iterations, and computation time per iteration can be found in Table 7.9. It is clear that the AA approach needs significantly less overall computation time, but since the problem is dynamic this might be misleading. Due to the different optimization results, one method may encounter a situation were a lot of iterations are required. That this is not the case here can be seen from the number of iterations, where the back-projection approach requires fewer iterations in comparison to the AA approach. When comparing the computation time, AA is much faster (about 38×). This can be attributed to the fact that AA is able to evaluate all 15 constraints at once. Furthermore, due to the structure of the problem the system states can

Method	Computation time	Total iterations	Time per Iteration
AA	67.2 s	930	0.07 s
Back-projection	1534.6 s	554	2.77 s

Table 7.9: *Performance results for the AA approach of Geletu et al. and back-projection for the problem with the stochastic objective function* (7.2.45)*, an optimization horizon of 30, and a prediction horizon of 15 time steps.*

be obtained explicitly, thereby further decreasing computation time. In contrast and although the system states can be obtained explicitly, the back-projection approach has to evaluate all 15 constraints separately and the corresponding model equations to carry out the projection are implicit, requiring additional effort to solve the corresponding equations.

To further study the influence of the two approaches on the computation time, the problem was solved for different values of the prediction horizon h. It is clear from the problem formulation that the number of chance constraints equals the length of the prediction horizon. In order to minimize the effect of the dynamical problem on the computation time, the average computation time per iteration was measured. The results of the experiment can be found in Figure 7.5. It is apparent that the back-projection approach needs significantly more computation time per iteration. Moreover, the computation time per iteration in the analytical approximation approach appears to depend only linearly on the length of the prediction horizon, whereas with the back-projection approach it appears to depend quadratically on the parameter. This can be explained by the fact that the analytical approximation approach requires the solution of the model equations only once at every grid point of the underlying integration routine to compute all the chance constraints. Since the computation time to evaluate the model equations also increases linearly with the length of the prediction horizon, overall a linear dependence $\mathcal{O}(h)$ can be expected. In contrast, the back-projection approach requires the solution of h different modified versions of the original model equations at every grid point, resulting in a quadratic dependence of $\mathcal{O}(h^2)$.

More generally, for an arbitrary problem, let n_c be the number of chance constraints, T_{sol} the computation time necessary to solve the model equations, and n_i the number of grid points in the corresponding integration routine. Then, one would expect computation times per iteration of $\mathcal{O}(n_i T_{sol})$ for the AA approach and $\mathcal{O}(n_c n_i T_{sol})$ for the back-projection approach. One should keep in mind that n_i can generally be chosen smaller when using back-projection, since this method allows the usage of sparse grid integration (see Chapter 5). Nevertheless, in the presence of a sufficiently high number of chance constraints, the AA outperforms the previously employed back-projection approach.

7.2.6 Summary

Considering the numerical studies presented above the following conclusion about the AA approach can be found. The advantages are as follows.

- Arbitrary uncertainties can be handled.

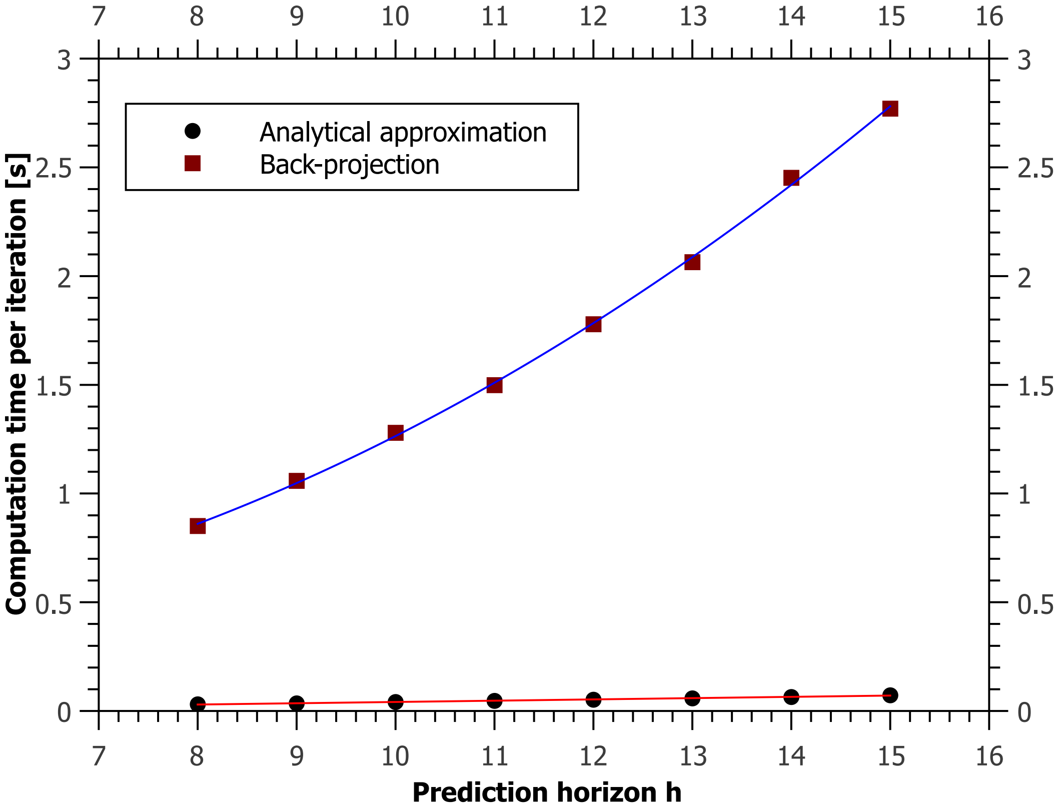

Figure 7.5: *Average computation time per iteration for the chemical reactor example when using analytical approximation and back-projection.*

- No in-depth knowledge about the involved model equations is required.
- Pathological cases can be handled.
- In comparison with certain other approximation methods tighter bounds are generated, resulting in a better approximation of the optimal solution.

Naturally, each approach also contains some disadvantages. For the AA approach, these can be found below.

- Due to the fact that AA only approximates the problem, solutions generated with AA are usually worse than solutions generated by a direct approach. Nonetheless, the experiments have shown that the results of direct and AA approaches differ only by a small margin.
- The proposed AA may require more function evaluations than a more specialized approach, leading to possibly higher computation time.

7.3 Chance constrained optimal power flow

The solution of CCOPT problems in the field of energy networks was one of the main motivations for this thesis. In contrast to the numerical experiments presented above the solution of CCOPT problems in such systems is more involved due to the dimension of the problem as well as the unavailability of monotonicity relations in the power flow equations, which present the main part of the model equations when dealing with energy networks. The results below were previously published in [50].

It is well recognized that many uncertainties have to be considered in the operations planning for power transmission and distribution systems [88]. Well-known uncertainties include power loads due to forecast inaccuracy, system parameters due to variations of the atmospheric temperature, renewable penetrations due to random availability of renewable energies as well as electricity prices due to varying market conditions. Under these uncertainties an optimal decision needs to be determined, which should lead to a both operatively reliable and economically beneficial operation.

In the recent years many studies have been made to investigate probabilistic methods for Optimal Power Flow (OPF) with uncertainty. Monte-Carlo simulation was commonly used to analyze the impacts of uncertain input parameters on the operation of power systems [4]. More recently, the method of CCOPT has been applied to optimization of electrical systems under uncertainty. The unique feature of this method is that the solution can achieve a balanced decision between profitability and reliability [55]. CCOPT was used in generation expansion [57], filter planning [15], wind farm planning [38], stochastic optimal reactive power dispatch [43], transmission network planning [93], optimal scheduling [90], and OPF for transmission networks under demand uncertainty [94].

In almost all of these previous studies the Gaussian (normal) distribution was assumed to describe the probability distribution of uncertain input parameters. However, many studies show that the stochastic distributions of renewable energy penetrations deviate considerably from the Gaussian distribution. In [59] the variation in wind speed was described by the Weibull probability distribution, while other studies indicated that wind power prediction errors can be well represented with Beta or Gamma distribution [25, 85].

This section presents an extension and continuation of the work presented in [94]. Here, OPF under non-Gaussian uncertainties is formulated as a nonlinear CCOPT problem. To solve this problem, the AA approach is used. This approach overcomes the two shortcomings in [94], i.e., a monotonic relation is not required and non-Gaussian distributed uncertain parameters can be treated. The effectiveness of the proposed approach is demonstrate on a real distribution system with a wind power penetration.

7.3.1 OPF under non-Gaussian Uncertainties

We start by relating the general CCOPT problem (3.0.1)–(3.0.3) to an OPF problem in Distribution System (DS). A DS consists of buses, which are connected by feeders. Every bus in an electrical network can be specified by four quantities: active and reactive power, respectively, voltage magnitude and phase angle. Two of these values are always fixed, the other two are state variables. For the slack bus, which acts as reference, voltage magnitude and phase angle are given, active and reactive power are state variables. For PQ buses the contrary is true, active and reactive power are given, voltage magnitude and phase angles are state variables. The aforementioned quantities are computed using active and reactive power flow equations

$$P_i - V_i \sum_{\substack{j=1 \\ j\in i}}^{N} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) = 0, \quad i \in N \tag{7.3.1}$$

$$Q_i - V_i \sum_{\substack{j=1 \\ j\in i}}^{N} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) = 0, \quad i \in N, \tag{7.3.2}$$

where P_i and Q_i are the active and reactive power injection at bus i, respectively. The matrices G and B are the bus conductance and susceptance matrix of the system, V_i is the voltage magnitude at bus i, θ_{ij} is the difference of the phase angles between buses i and j, and N is the number of buses. The active and reactive power injections are given by

$$P_i = P_S + \beta_{curt,i} P_{r,i}(\xi) - P_{d,i}, \quad i \in N \tag{7.3.3}$$

$$Q_i = Q_S - Q_{d,i}, \quad i \in N \tag{7.3.4}$$

where P_S and Q_S denote active and reactive power injected at the slack bus, respectively, $\beta_{curt,i}$ is a curtailment factor for the active power generation $P_{r,i}(\xi)$ of a renewable energy source, $P_{d,i}$ and $Q_{d,i}$ are the active and reactive power demand, respectively. In the case that no curtailment occurs $\beta_{curt,i} = 1$, otherwise $\beta_{curt,i} < 1$. This approach to OPF was proposed in [28].

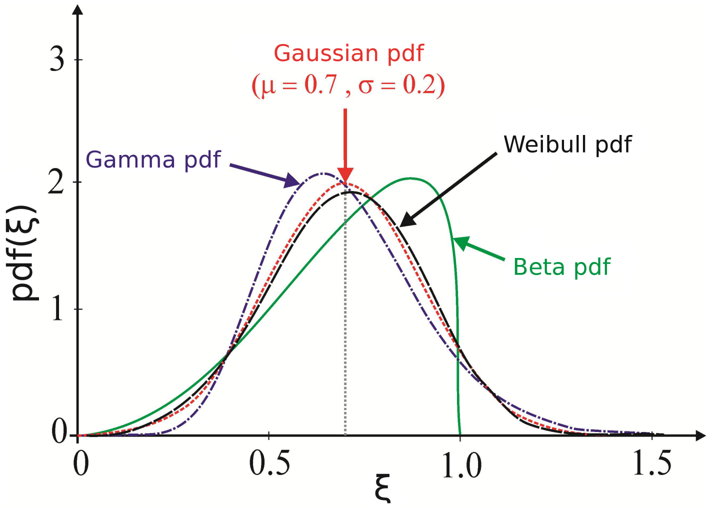

Figure 7.6: *Comparison of Beta, Gamma, Gaussian, and Weibull pdfs with the same mean and variance*

Although other formulations are possible, the minimization of the amount of renewable energy curtailment is considered as objective function (3.0.1) in this study. Chance constraints of the type (3.0.3) are included to prevent inadmissible states by constraining the voltage magnitude of PQ buses, active as well as reactive power at slack bus, and the capacity of the feeders. Since the curtailment factors are considered as control variables, the control constraints can be written as $\beta_{curt,i} \in [0, 1]$.

In this study, only uncertainties from renewable wind energy penetration are considered. Furthermore, we restrict our discussion to Beta-distributed uncertainties, which were proposed for the distribution of wind speed in [25], although additional load uncertainties as well as Gamma- and Weibull-distributed uncertainties could be treated with the presented approach. To illustrate the differences between the different distributions, a Beta pdf, a Gamma pdf, and a Weibull pdf (all with $\mu = 0.7$, $\sigma = 0.2$) and their approximation with a Gaussian pdf are shown in Fig. 7.6. It can be clearly seen that a Gaussian approximation of the Beta-distributed uncertainties is unfeasible and will possibly lead to large errors in the probability computation.

7.3.2 A Case study

In this subsection the proposed approach is applied to a DS as shown in Fig. 7.7. It is a 27.6 kV, 41 bus real radial DS which was studied in [7, 28]. Values in per unit system are given on 10-MVA base, otherwise specified. The DS has three embedded wind parks, located at buses 19, 28 and 40, with rated powers 0.8, 0.4 and 1, respectively. Due to the small local spread of the system, the expected wind speed and its variance are assumed to be the same at all wind parks. Bus 1 is considered to be the slack bus, whereas other buses are considered to be PQ buses.

The operation of this system is considered in the situation that the total rated power of the installed wind parks exceeds the total active power demand. It is assumed that reverse active power flow through the slack bus is not allowed. In this case, curtailment is required to ensure a safe operation of the DS. The goal of the optimization is to minimize the necessary curtailments, in order to minimize spilled wind energy. The CCOPF problem is formulated as follows

$$
\begin{aligned}
\min_{u} \quad & E\left[\sum_{i\in\{19,28,40\}} (1-\beta_{curt,i})P_{W,i}(v)\right] && (7.3.5)\\
\text{s.t.} \quad & Pr\left\{V^{min} \leq V_i \leq V^{max}\right\} \geq \alpha_V, \\
& \qquad i \in N, i \neq S && (7.3.6)\\
& Pr\left\{P^{min} \leq P_S \leq P^{max}\right\} \geq \alpha_{P_S} && (7.3.7)\\
& Pr\left\{Q^{min} \leq Q_S \leq Q^{max}\right\} \geq \alpha_{Q_S} && (7.3.8)\\
& Pr\left\{S_{i,j} \leq S_{max}\right\} \geq \alpha_S, \\
& \qquad i,j \in N,\ i \neq j && (7.3.9)\\
& 0 \leq \beta_{curt,i} \leq 1, \quad i \in \{19,28,40\} && (7.3.10)
\end{aligned}
$$

where $\left(P_{W,19}(v), P_{W,28}(v), P_{W,40}(v)\right)$ are the uncertain variables , V^{min} and V^{max} are the lower and upper bounds of the voltage magnitude at PQ buses, respectively. Similarly, P^{min}, P^{max}, Q^{min}, and Q^{max} are the lower and upper bounds of the active and reactive power at slack bus, respectively, and S^{max} is the upper bound of the apparent power of the main feeder in the system. The apparent power flow between bus i and j is denoted by $S_{i,j}$. Overall, 82 different chance constraints (40 constraints on voltage magnitudes, one constraint on P_S and Q_S, respectively, and 40 constraints on the feeders) are considered in this case study. The objective function describes the total curtailed wind power for the three wind parks and $P_{W,i}(v)$ is the available power from the wind park at bus i for a given wind speed v. According to [42], this quantity can be obtained by

$$
P_{W,i}(v) = \begin{cases} 0, & 0 \leq v \leq v_{cin} \\ P_{Wrated,i}\left(\frac{v-v_{cin}}{v_r-v_{cin}}\right), & v_{cin} \leq v \leq v_r \\ P_{Wrated,i}, & v_r \leq v \leq v_{co} \\ 0, & v_{co} < v, \end{cases} \tag{7.3.11}
$$

where $P_{Wrated,i}$ is the rated power of the wind park at bus i, and the parameters are given as $v_{cin} = 4\ \frac{m}{s}$, $v_r = 14\ \frac{m}{s}$, and $v_{co} = 24\ \frac{m}{s}$. If one considers a variance $Var[v]$ in the wind speed prediction and takes the predicted value

$$
\mu(v) = \begin{cases} 0, & 0 \leq v \leq v_{cin} \\ E\left[\frac{v-v_{cin}}{v_r-v_{cin}}\right], & v_{cin} \leq v \leq v_r \\ 1, & v_r \leq v \leq v_{co} \\ 0, & v_{co} < v, \end{cases}
$$

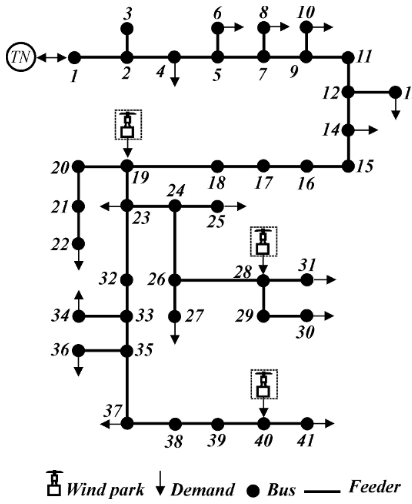

Figure 7.7: *Distribution system for the case study.*

Table 7.10: *Parameters used in the case study*

V^{min}	V^{max}	P^{min}	P^{max}	Q^{min}	Q^{max}	S^{max}
0.95	1.05	0	2	-2	2	1.43
α_V	α_{P_S}	α_{Q_S}	α_S	τ	m_1	m_2
95%	95%	95%	95%	0.001	1	1

as expectation, then $P_{W,i}(v)$ can be considered as an uncertain variable. Following [25], this uncertain variable can be described by the product $\xi P_{Wrated,i}$ in the case that $v_{cin} \leq v \leq v_r$, where ξ is a Beta-distributed uncertain variable. Note that in the other cases either full or no wind power injection is expected. The parameters α and β of the underlying Beta distribution can be computed using (7.3.11), i.e.,

$$\alpha = \mu(v)\left(\mu(v)\frac{(v_r - v_{cin})^2(1-\mu(v))}{Var\,[v]} - 1\right)$$
$$\beta = (1-\mu(v))\left(\mu(v)\frac{(v_r - v_{cin})^2(1-\mu(v))}{Var\,[v]} - 1\right).$$

Computation tests for solving the CCOPT problem are conducted for a scenario with low demand (i.e., a total demand of active power of 0.66) and expected wind speeds between v_{cin} and v_r. Different test scenarios with an increasing expected wind power penetration are shown in Fig. 7.8. The parameters used in the computation tests are listed in Table 7.10 and the results from the different tests are shown in Fig. 7.8. As can be seen from Fig. 7.8 (a), an increasing expected wind speed leads to a higher available wind power. Nonetheless, with an increasing available wind power the necessary curtailment increases. Fig. 7.8 (b) shows the minimum value of the chance constraints at the optimum obtained through a posteriori Monte-Carlo simulation. The simulation results indicate that the chance constraints are held with a probability higher than the desired probability level of 95%, i.e., the solution generated by the proposed approach is feasible.

To show the viability of the AA approach, the same optimization problem is solved using a Gaussian approximation and Quasi-Monte-Carlo integration (using the Sobol sequence), the results can be found in Fig. 7.8. It can be seen from 7.8 (a) that when employing Gaussian approximation, generally, less curtailment occurs in comparison to the proposed approach. While this might indicate that this approximation is more suitable for the task, the contrary is true. When simulating the model it becomes apparent that the chance constraints are violated by a large extent (in the worst case only a satisfaction in 17% is guaranteed in contrast to the desired value of 95%). This indicates that the Gaussian approximation cannot capture the essential details of the Beta-distribution.

In Fig. 7.8 (c) and (d) the expected values of P_s and the total active power losses are shown. It can be seen that the proposed approach imports more active power in comparison to that from the Gaussian approximation (i.e., less available wind power is utilized) in seven of nine scenarios, while the overall active power losses are lower for the proposed method in most scenarios. These results are strongly related to the formulation of the objective function, i.e., in order to accommodate a large amount of wind energy, active power losses are increased [29], e.g., in scenario six of the Gaussian approximation and scenario nine of the proposed approach.

Capability of dealing with high dimensional uncertainties

To test the ability of the proposed approach for handling high dimensional uncertain variables, numerical experiments are carried out with 10, 15 and 20 embedded wind parks at different locations in the DS shown in Figure 7.7. The location and sizes of the wind parks are given in Table 7.11. A medium wind speed scenario ($v = 9\ \frac{m}{s}$) is considered, since it is shown to be the most time consuming case in solving the CCOPT problem with three wind parks. In effect, the limiting factor of the computation expense is the number of grid points, since the power flow equations need to be solved at each grid point to evaluate the integrals for probability as well as gradient values. The obtained results and the corresponding computation time are listed in Table 7.11. It can be seen that, using the proposed approach, a chance constrained OPF with up to 20 wind parks (and, correspondingly, 20 uncertain variables) can be solved in reasonable CPU time on a desktop PC.

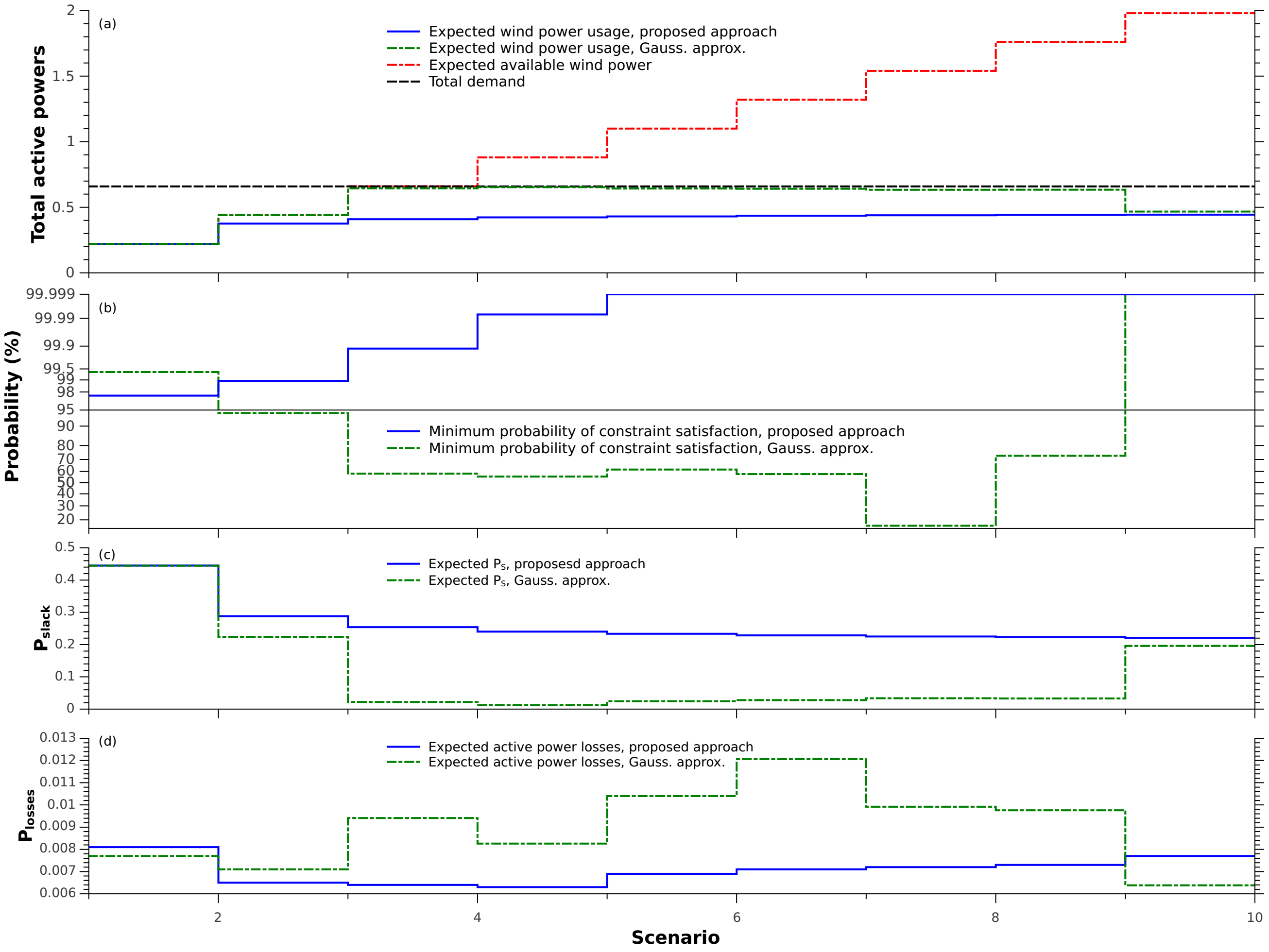

Figure 7.8: *Results for the case study. (a) Expected available wind power generation, active demand and expected wind power generation for the proposed approach and the Gaussian approximation. (b) Minimum chance constraint values obtained from simulation. (c) Expected P_S. (d) Expected active power losses.*

Table 7.11: *Computation time and results for different numbers of uncertain variables. Total expected available wind power is* 1.1*.*

Number of wind parks	Wind parks at buses	Rated power	Curtailed wind power	Computation time
3	19, 28, 40	0.8, 0.4, 1	0.670	4 min
10	17, 18, 19, 22, 26, 28, 29, 36, 40, 41	0.22 each	0.663	3 min
15	Same as for 10, 27, 33, 34, 37, 39	0.146 each	0.662	4 min
20	Same as for 15, 16, 23, 25, 34, 35	0.11 each	0.662	5 min

8 Conclusions and Future Work

The thesis at hand consists of two major parts. In the first part, the notion of a Chance Constrained OPTimization (CCOPT) problem is introduced, followed by a description of a general solution framework. Depending on the actual problem several approaches may be available. This thesis introduces the most important methods for transforming the probabilistic constraints into deterministic ones. The main contribution here consists of a novel Analytical Approximation (AA) approach. Furthermore, the necessary multivariate integration routines are covered.

The second part consists of numerical experiments showing the viability of the AA approach. These experiments can be divided into smaller experiments with less then ten uncertain variables and only few constraints and a larger application in the field of energy network with up to twenty uncertain variables and about 100 constraints. The experiments show that AA can be seen as a general purpose approach, which does not require in-depth knowledge about the system involved. Although the results are generally worse than results obtained by a direct approach, both results usually differ only by a small margin. Additionally, the results can be further improved by decreasing the value of the parameter τ in the Analytical Approximation approach. Even though, this may require a more accurate integration method. In addition, the Chance Constrained Optimal Power Flow problems shows that the commonly used Gaussian approximation of uncertainties is not a valid approach.

8.1 Future work

8.1.1 Theory

Concerning the theoretical background of CCOPT further work is required on several fronts.

First, the proposed approaches are mainly suitable for handling single chance constraints

$$Pr\left\{g_i(u,y,\xi) \le 0\right\} \ge \alpha_i, \quad i = 1, \dots, q.$$

In the future, also the more natural joint chance constraint formulation

$$Pr\left\{g_i(u,y,\xi) \le 0,\ i = 1, \dots, q\right\} \ge \alpha$$

should be treated. This is currently possible with the back-projection approaches, but only for certain systems.

Second, the AA approach proposed by Geletu et al. [34] currently uses a sigmoidal function to approximate a step function. While this functions exhibits good properties, another one, e.g., constructed from composite polynomials, could be even more suitable. Extrapolation methods could be used to obtain the value of the chance constraints as $\tau \to 0$.

Third, the treatment of dynamic problems needs further study. The methods employed today cannot guarantee that in the next time step there exists a feasible solution of the corresponding problem. This could be treated by a hybrid robust optimization/CCOPT approach. The requirement in the robust part consists of guaranteeing a feasible solution in the next time step regardless of the realization of the uncertain variables.

8.1.2 Applications

Prospective work for the practical applications is mainly considered with the CCOPF example. Additional versions could include demand uncertainties, increasing the number of uncertainties in the system. Furthermore, one could examine the influence of batteries in the network, converting the problem into a dynamic one. This does not pose a large difficulty, since the charge/discharge of batteries are included as control variables. Moreover, the power flow problems in the different time horizons are very much decoupled (i.e., the only coupling is the charge of the batteries) allowing to treat every time step separately. Thereby, computation time grows only linearly with the length of the prediction horizon. If the problem is dynamic one could also consider different uncertainty distributions at different time steps, e.g., households have a low demand in the night, but a high demand in the morning and the evening. All these extensions can generally be treated within the existing framework, so the main concern is the implementation and guaranteeing a reasonable computation time.

References

[1] Adcock, B. Univariate modified fourier methods for second order boundary value problems. *BIT Numerical Mathematics 49*, 2 (2009), 249–280. Cited on page 71.

[2] Adcock, B. Multivariate modified fourier series and application to boundary value problems. *Numerische Mathematik 115*, 4 (2010), 511–552. Cited on page 72.

[3] Agarwal, N., and Aluru, N. R. Stochastic analysis of electrostatic mems subjected to parameter variations. *JMEMS 18*, 6 (2009), 1454–1468. Cited on page 2.

[4] Anders, G. J. *Probability concepts in electric power systems.* John Wiley & Sons, 2005. Cited on page 88.

[5] Andrieu, L., Cohen, G., and V'azquez-Abad, F. J. Stochastic programming with probability constraints. In *10th International Conference on Stochastic Programming* (Tucson, USA, October 2004). Cited on page 76.

[6] Arellano-Garcia, H., and Wozny, G. Chance constrained optimization of process systems under uncertainty: I. strict monotonicity. *Comput. Chem. Eng. 33* (2009), 1568–1583. Cited on page 2.

[7] Atwa, Y., and El-Saadany, E. F. Probabilistic approach for optimal allocation of wind-based distributed generation in distribution systems. *IET Renewable Power Generation 5*, 1 (2010), 79–88. Cited on page 90.

[8] Begumisa, A., and Robinson, I. Suboptimal kronrod extension formulae for numerical quadrature. *Numerische Mathematik 58*, 1 (1990), 807–818. Cited on page 52.

[9] Ben-Tal, A., and Nemirovski, A. Robust solutions of uncertain linear programs. *Operations research letters 25*, 1 (1999), 1–13. Cited on page 3.

[10] Ben-Tal, A., and Nemirovski, A. *Lecture Notes on Modern Convex Optimization.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2001. Cited on page 1.

[11] Björnberg, J., and Diehl, M. Approximate robust dynamic programming and robustly stable mpc. *Automatica 42* (2006), 777–782. Cited on page 1.

[12] Caflisch, R. E. Monte carlo and quasi-monte carlo methods. *Acta numerica 1998* (1998), 1–49. Cited on pages 2 and 61.

[13] Calafiore, G., and Campi, M. C. Uncertain convex programs: randomized solutions and confidence levels. *Math. Program., Ser. A 102* (2005), 23–46. Cited on page 2.

[14] Cannon, M., Kouvaritakis, B., and Wu, X. Model predictive control for systems with stochastic multiplicative uncertainty and probabilistic constraints. *Automatica 45*, 1 (2009), 167–172. Cited on page 1.

[15] Chang, G. W., Wang, H. L., and Chu, S. Y. A probabilistic approach for optimal passive harmonic filter planning. *IEEE Trans. Power Syst. 22*, 3 (2007), 1790–1798. Cited on page 88.

[16] Charnes, A., and Cooper, W. Chance-constrained programming. *Management Science 6* (1959), 73–79. Cited on page 1.

[17] Charnes, A., Cooper, W., and Symmonds, G. H. Cost horizons and certainity equivalents: an approach to stochastic programming of heating oil. *Management Science 4* (1958), 235–263. Cited on pages 1 and 19.

[18] Chiralaksanakul, A., and Mahadevan, S. Decoupled approach to multidisciplinary design optimization under uncertainty. *Optim Eng 8* (2005), 261–267. Cited on page 77.

[19] Cools, R. Advances in multidimensional integration. *Journal of computational and applied mathematics 149*, 1 (2002), 1–12. Cited on page 57.

[20] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems 2* (1989), 303–314. Cited on page 69.

[21] de Figueiredo, L. H., and Stolfi, J. *Self-Validated Numerical Methods and Applications.* Brazilian Mathematics Colloquium monographs. IMPA/CNPq, Rio de Janeiro, Brazil, 1997. Cited on pages 7 and 9.

[22] de Panne, C. V., and Popp, W. Minimum cost cattle feed under probabilistic protein constraints. *Mgmt Sci. 9* (1963), 405–430. Cited on page 74.

[23] Diehl, M., Bock, H. G., and Kostina, E. An approximation technique for robust nonlinear optimization. *Math. Program., Ser. B 107* (2005), 213–230. Cited on page 1.

[24] Dubourg, V., Sudret, B., and Bourinet, J.-M. Reliability-based design optimization using kriging surrogates and subset simulation. *Struct. Multidisc. Optim. 44*, 4 (2011), 673–690. Cited on pages 78 and 79.

[25] Fabbri, A., Román, T. G. S., Abbad, J. R., and Quezada, V. H. M. Assessment of the cost associated with wind generation prediction errors in a liberalized electricity market. *IEEE Trans. Power Syst. 20*, 3 (2005), 1440–1446. Cited on pages 89, 90, and 92.

[26] Feigner, U., Herrlich, H., Husek, M., Kanovei, V., Koepke, P., Preuss, G., Purkert, W., and Scholz, E., Eds. *Felix Hausdorff, Gesammelte Werke, Band II.* Springer, Berlin, 2008. Cited on page 26.

[27] Flemming, T., Bartl, M., and Li, P. Set-point optimization for closed-loop control systems under uncertainty. *Ind. Eng. Chem. Res. 46* (2007), 4930–4942. Cited on page 2.

[28] Gabash, A., and Li, P. Active-reactive optimal power flow in distribution networks with embedded generation and battery storage. *IEEE Trans. Power Syst. 27*, 4 (2012), 2026–2035. Cited on pages 89 and 90.

[29] Gabash, A., and Li, P. Active-reactive power flow for low-voltage networks with photovoltaic distributed generation. In *2nd IEEE International Energy Conference and Exhibition (EnergyCon2012)/ Future Energy Grids and Systems(FEGS)* (Florence, Italy, September 2012). Cited on page 93.

[30] Garcke, J., Griebel, M., and Thess, M. Data mining with sparse grids. *Computing 67* (2001), 225–253. Cited on page 2.

[31] Garnier, J., Omrane, A., and Rouchdy, Y. Asymptotic formulas for the derivatives of probability functions and their monte carlo estimations. *Eur. J. Oper. Res. 198*, 3 (2008), 848–858. Cited on page 21.

[32] Gelbaum, B. *Problems in Real and Complex Analysis.* Problem books in mathematics. Springer, New York, 1992. Cited on page 7.

[33] Geletu, A., Hoffmann, A., Klöppel, M., and Li, P. Monotony analysis and sparse-grid integration for nonlinear chance constrained process optimization. *Eng. Optim. 43*, 11 (2011), 1019–1041. Cited on pages 4, 21, 22, and 23.

[34] Geletu, A., Klöppel, M., Hoffmann, A., and Li, P. A tractable approximation of non-convex chance constrained optimization with non-gaussian uncertainties. *To appear in Eng. Optim.* (2014). Cited on pages 2, 4, 24, 27, 28, 30, 40, 67, and 96.

[35] Geletu, A., Klöppel, M., Zhang, H., and Li, P. Advances and applications of chance-constrained approaches to systems optimisation under uncertainty. *Int. J. System Sci.* (2012). Cited on pages 4 and 74.

[36] Genz, A., and Keister, B. D. Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. *J. Comput. Appl. Math. 71*, 2 (1996), 299–309. Cited on page 59.

[37] Gerstner, T., and Griebel, M. Numerical integration using sparse grids. *Numer. Algorithms 18* (1998), 209–232. Cited on pages 2 and 59.

[38] Gu, W., Wang, R., Sun, R., and Li, Q. Wind power penetration limit calculation based on stochastic optimal power flow. *International Review of Electrical Engineering 6*, 4 (2011), 1939–1945. Cited on page 88.

[39] Heiss, F., and Winschel, V. Estimation with numerical integration on sparse grids. Tech. rep., Münchener Wirtschaftswissenschaftliche Beiträge (VWL) 2006-15, 2006. Cited on page 59.

[40] Henrion, R., Li, P., Möller, A., Steinbach, M. C., Wendt, M., and Wozny, G. *Online Optimization of Large Scale Systems.* Springer, Berlin, 2001, ch. Stochastic optimization for operating chemical processes under uncertainty, pp. 455–476. Cited on page 2.

[41] Henrion, R., and Möller, A. Optimization of a continuous distillation process under random inflow rate. *Comput. Math. Appl. 45* (2003), 247–262. Cited on page 1.

[42] Hetzer, J., Yu, D. C., and Bhattarei, K. An economic dispatch model incorporating wind power. *IEEE Trans. Energy Convers. 23*, 2 (2008), 603–611. Cited on page 91.

[43] Hu, Z. C., Wang, X. F., and Taylor, G. Stochastic optimal reactive power dispatch: formulation and solution method. *International Journal of Electrical power & Energy Systems 32*, 6 (2010), 615–621. Cited on page 88.

[44] Hubbard, J. *Teichmüller theory and applications to geometry, topology, and dynamics.* No. Bd. 1 in Teichmüller Theory and Applications to Geometry, Topology, and Dynamics. Matrix Editions, Ithaca, NY, 2006. Cited on page 7.

[45] Iserles, A., and Nørsett, S. P. From high oscillation to rapid approximation i: Modified fourier expansions. *IMA journal of numerical analysis 28*, 4 (2008), 862–887. Cited on page 72.

[46] Jackson, D. *Fourier Series and Orthogonal Polynomials.* Dover Books on Mathematics Series. Dover Publications, Mineola, 2004. Cited on page 71.

[47] Johnson, N. L., Kotz, S., and Balakrishnan, N. *Continuous Univariate Distributions, Volume 1.* John Wiley & Sons, New York, 1994. Cited on page 9.

[48] Jost, J. *Postmodern Analysis.* Universitext (1979). Springer, Berlin, 2005. Cited on page 71.

[49] Keese, A., and Matthies, H. G. Numerical methods and smolyak quadrature for nonlinear stochastic partial differential equations. Tech. rep., Technische Universität Braunschweig, 2003. Cited on page 2.

[50] Klöppel, M., Gabash, A., Geletu, A., and Li, P. Chance constrained optimal power flow with non-gaussian distributed uncertain wind power generation. In *Environment and Electrical Engineering (EEEIC), 2013 12th International Conference on* (2013), pp. 265–270. Cited on pages 4 and 88.

[51] Klöppel, M., Geletu, A., Hoffmann, A., and Li, P. Using sparse-grid methods to improve computation efficiency in solving dynamic nonlinear chance-constrained optimization problems. *Ind. Eng. Chem. Res. 50*, 9 (2011), 5693–5704. Cited on page 15.

[52] Krantz, S. G., and Parks, H. R. *The Implicit Function Theorem.* Birkhäuser, 2002. Cited on page 5.

[53] Kronrod, A. S. *Nodes and Weights of Quadrature Formulas: Sixteen-Place Tables.* Consultants Bureau, New York, 1965. Cited on page 51.

[54] Lemieux, C. *Monte Carlo and Quasi-Monte-Carlo Sampling.* Springer, New York, 2009. Cited on pages 2 and 60.

[55] Li, P., Arellano-Garcia, H., and Wozny, G. Chance constrained programming approach to process optimization under uncertainity. *Comput. Chem. Eng. 32* (2008), 24–45. Cited on pages 14 and 88.

[56] Li, P., Wendt, M., and Wozny, G. Robust model predictive control under chance constraints. *Comput. Chem. Eng. 24* (2000), 829–834. Cited on page 1.

[57] Mazadi, M., Rosenhart, W. D., Malik, O. P., and Aguado, J. Modified chance constrained optimization applied to the generation expansion problem. *IEEE Trans. Power Syst. 24*, 3 (2009), 1635–1636. Cited on page 88.

[58] Moore, R. E. *Interval analysis*, vol. 2. Prentice-Hall Englewood Cliffs, 1966. Cited on page 8.

[59] Musgrove, P. *Wind power.* Cambridge Univ. Press, New York, USA, 2010. Cited on page 89.

[60] Mysovkikh, I. P. On the construction of cubature formulas with the smallest number of nodes. *Soviet Math. Dokl. 9* (1968), 277–280. Cited on page 2.

[61] Nagy, Z. K., and Braatz, R. D. Robust nonlinear model predictive control of batch processes. *AIChE J. 49*, 7 (2003), 1776–1786. Cited on page 1.

[62] Nagy, Z. K., and Braatz, R. D. Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. *IEEE Trans. Control Syst. Technol. 11*, 5 (2003), 694–704. Cited on page 1.

[63] Nagy, Z. K., and Braatz, R. D. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *J. Process Control 14* (2004), 411–422. Cited on page 1.

[64] Nemirovski, A., and Shapiro, A. Convex approximations of chance constrained programs. *SIAM J. Optim. 17*, 4 (2006), 969–996. Cited on pages 2, 25, and 28.

[65] Nobile, F., Tempone, R., and Webster, C. G. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal. 46*, 5 (2008), 2309–2345. Cited on page 2.

[66] Novak, E., and Ritter, K. Simple cubature formulas with high polynomial exactness. *Constr. Approx. 15* (1999), 499–522. Cited on pages 59 and 60.

[67] Oh, M.-S., and Berger, J. O. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation 41*, 3-4 (1992), 143–168. Cited on page 61.

[68] Olson, D. L., and Swenseth, S. R. A linear approximation for chance-constrained programming. *J. Opl Res. Soc. 38*, 3 (1987), 261–267. Cited on page 75.

[69] Prekopa, A. *Stochastic Programming.* Springer, Netherlands, 1995. Cited on pages 1, 3, 4, and 57.

[70] Rockafellar, R. T., and Uryasev, S. Optimization of conditional value-at-risk. *Journal of Risk 2* (2000), 21–41. Cited on page 1.

[71] Rockafellar, R. T., and Wets, R. J.-B. *Variational Analysis*, vol. 317 of *Grundlehren der mathematischen Wissenschaften.* Springer, Dordrecht, 2009. Cited on page 25.

[72] Rojas, R. *Neural Networks - A Systematic Introduction.* Springer, Berlin, 1996. Cited on page 69.

[73] Royset, J. O., Kiureghian, A. D., and Polak, E. Optimal design with probabilistic objective and constraints. *J. of Eng. Mech. (ASCE) 32* (2006), 107–118. Cited on page 1.

[74] Royset, J. O., and Polak, E. Implementable algorithm for stochastic optimization using sample average approximation. *J. Optim. Theory Appl. 122*, 1 (2004), 157–184. Cited on pages 1, 22, 77, and 79.

[75] Sandberg, I. W. Global implicit function theorems. *IEEE Trans. Circuits Syst. 28* (1981), 145–149. Cited on page 6.

[76] Sanders, J., and Kandrot, E. *Cuda by Example: An Introduction to General-purpose GPU Programming.* Addison Wesley Professional, 2011. Cited on page 74.

[77] Sansone, G. *Orthogonal Functions.* Dover Publications, New York, 1991. Cited on page 71.

[78] Schwarm, A. T., and Nikolaou, M. Chance constrained model predictive control. *AIChE J. 45* (1999), 1743–1752. Cited on page 1.

[79] Shah, S. S., and Madhavan, K. P. Design of controllable batch processes in the presence of uncertainity. *Ann. Oper. Res. 132* (2004), 223–241. Cited on page 78.

[80] Smolyak, S. A. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl. 4* (1963), 240–243. Cited on pages 2 and 58.

[81] Srinivasan, B., Bonvin, D., Visser, E., and Palanki, S. Dynamic optimization of batch processes ii. role of measurements in handling uncertainty. *Comput. Chem. Eng. 27* (2002), 27–44. Cited on page 1.

[82] Srinivasan, B., Palanki, S., and Bonvin, D. Dynamic optimization of batch processes i. characterization of the nominal solution. *Comput. Chem. Eng. 27* (2003), 1–26. Cited on page 1.

[83] Stoer, J., and Bulirsch, R. *Introduction to Numerical Analysis.* Springer, New York, 1992. Cited on pages 46, 47, 50, 51, and 66.

[84] Trefethen, L. Is gauss quadrature better than clenshaw-curtis? *SIAM Review 50*, 1 (2008), 67–87. Cited on page 51.

[85] Usaola, J. Probabilistic load flow in systems with wind generation. *IET Generation, Transmission & Distribution 3*, 12 (2009), 1031–1041. Cited on page 89.

[86] Waldvogel, J. Fast construction of the fejér and clenshaw–curtis quadrature rules. *BIT Numerical Mathematics 46*, 1 (2006), 195–202. Cited on page 49.

[87] Wasilkowski, G. W., and Wozniakowski, H. Explicit cost bounds of algorithms for multivariate tensor product problems. *J. Complexity 11* (1994), 1–56. Cited on page 58.

[88] Weber, C. *Uncertainty in the electric industry, methods and model for decision support.* Springer, New York, USA, 2005. Cited on page 88.

[89] Wendt, M., Li, P., and Wozny, G. Nonlinear chance-constrained process optimization under uncertainity. *Ind. Eng. Chem. Res. 41* (2002), 3621–3629. Cited on pages 2 and 21.

[90] Wu, J., Zhu, J., Chen, G., and Zhang, H. A hybrid method for optimal scheduling of short-term electric power generation of cascaded hydroelectric plants based on particle swarm optimization and chance-constrained programming. *IEEE Trans. Power Syst. 23*, 4 (2008), 1570–1579. Cited on page 88.

[91] Xie, L., Li, P., and Wozny, G. Chance constrained nonlinear model predictive control. *Lecture Notes in Control and Inform. Sci. 358* (2007), 295–304. Cited on page 2.

[92] Yserentant, H. Sparse grid spaces for the numerical solution of the electronic schrödinger equation. *Numer. Math. 101* (2005), 381–389. Cited on page 2.

[93] Yu, H., Chung, C. Y., Wong, K. P., and Zhang, J. H. A chance constrained transmission network expansion planning method with consideration of load and wind farm uncertainties. *IEEE Trans. Power Syst. 24*, 3 (2009), 1568–1576. Cited on page 88.

[94] Zhang, H., and Li, P. Chance constrained programming for optimal power flow under uncertainty. *IEEE Trans. Power Syst. 26*, 4 (2008), 2417–2424. Cited on pages 88 and 89.

[95] Zorich, V. *Mathematical Analysis II.* Springer, Berlin, 2004. Cited on pages 6, 7, and 70.

Appendix

Description of the Implementations

Implementations of the problems presented in the case studies and numerical experiments can be obtained from the author[1].

All the implementations are based on IpOPT and share a similar layout. Typical components are

⟨name⟩_main.cpp, which contains the initializations of the optimization (e.g., choice of termination criteria, output options, etc.) and also loads or generates grid points and weights of the integration routine,

⟨name⟩_nlp.cpp, where the actual optimization problem is defined, and

stochopt_global.hpp, which defines problem specific global variables (e.g., for the integration routine and other necessary parameters).

Generally, all IpOPT options are available from the documentation (`http://www.coin-or.org/Ipopt/documentation/`). Nonetheless, some options are especially important when using the software for the solution of CCOPT problems. These are as follows.

Termination tolerances: Due to the usage of integration routines, the objective function and the constraints cannot be calculated exactly. With the exception of some special cases, there always remains a small error, which makes it necessary to increase termination tolerances, since otherwise IpOPT would not converge.

Approximation of the Hessian: While it is mathematically possible to obtain the Hessian, it is very time consuming in a numerical implementation. It is, therefore, advisable to use the limited memory Hessian approximation provided by IpOPT.

Choice of algorithm for the selection of the barrier parameter: For specific problems it may be advisable to use a different algorithm for the computation of the barrier parameter (e.g., the option “adaptive”). This can decrease the number of necessary iterations.

[1] Email: michael.kloeppel@gmx.net

The ⟨name⟩_nlp.cpp files are used to describe the NLP problems, which is done by implementing the methods of the IpOPT NLP-class. The purpose of the single functions is clear from their names. In the given programs, one will usually find additional routines, which evaluate the constraints and possibly also the objective function. This is done by looping over all grid points in the integration routine and solving the model equations at each point. This part of the programs is carried out as parallel computation (recognizable from the "#pragma omp parallel for" directive). If a Newton method is used, one will find additional functions to evaluate the model equations and the partial Jacobian.

There are two additional methods, which might be useful.

finalize_solution Allows to safe the optimal solution to other variables, since these are no longer available after the execution of IpOPT. Also allows to access additional information, e.g. multipliers, diagnostic information, etc.

intermediate_callback Allows to access all available information on the problem between two iterations. Can be either used to terminate the problem with a user-specified termination criterion or to adapt some problem parameters to properties of the optimization routine (e.g., match the parameter τ in the AA approach to the barrier parameter).

With the exception of the OPF example, the problems are usually not scalable. Increasing the number of constraints or uncertainties requires significant changes to the existing code.