RICE UNIVERSITY

High-Dimensional Integration for Optimization Under Uncertainty

by

Timur Takhtaganov

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Arts

Thesis Committee:

Matthias Heinkenschloss, Chairman
Professor of Computational and Applied Mathematics

Danny C. Sorensen
Noah G. Harding Professor of Computational and Applied Mathematics

Yin Zhang
Professor of Computational and Applied Mathematics

Houston, Texas

May, 2015

Abstract

High-Dimensional Integration for Optimization Under Uncertainty

by

Timur Takhtaganov

This thesis focuses on the problem of evaluating high-dimensional integrals arising in optimization under uncertainty. Uncertainties in the input data affect the behavior of the physical system and need to be accounted for at the design stage or in the way the system is controlled. This translates into evaluating integrals of the quantities of interest with respect to the random parameters. This task becomes challenging when the dimension of the random parameters is high. Without guidelines for the choice of favorable integration methods the optimization algorithm might encounter prohibitively high computational cost. This thesis provides a comprehensive overview of methods for high-dimensional integration and exposes their relative strengths and weaknesses. Emphasis is placed on problems with moderately high dimension and with non-smoothness. The performance of integration methods in high dimension is assessed on several simple model problems.

Acknowledgements

This research was supported by AFOSR grant FA9550-12-1-0155 and NSF grant DMS-1115345.

Contents

List of Figures

Introduction

Many engineering applications are affected by uncertainty in the data. This uncertainty might be due to incomplete knowledge (epistemic uncertainty) as is the case, for example, in oil reservoir simulation where the properties of the subsurface media are only known at a few locations and have to be inferred everywhere else. It might also be the case that the physical system has some intrinsic variability and our knowledge cannot be improved by additional measurements (aleatoric uncertainty). An example would be fluctuations of the flow field around the aircraft wing due to the gusts of wind. In either case, in order for our mathematical model of the system to give good predictions we need to account for the uncertainty in the input data and find the ways to "quantify" the uncertainty in the outputs of interest.

This thesis addresses the problem of uncertainty quantification in the context of optimization. The emphasis is placed on the task of efficient evaluation of integrals of the outputs of interest with respect to the random parameters, i.e., expectations, deviations, etc. The challenges addressed are: potentially high dimensionality of the random input parameters, high cost of evaluating the objective function, and non-smoothness in the integrand.

This thesis provides a comprehensive overview of the methods for high-dimensional integration and assesses their applicability for optimization under uncertainty. The first chapter is dedicated to the review of available literature on the topics of numerical solution of PDEs with random inputs, uncertainty quantification via "risk measures", and high-dimensional interpolation and integration methods. The second chapter sets the stage for the discussion of optimization problems with PDE constraints with random inputs. The main goal of this chapter is to provide motivation for the following overview of the integration methods. Chapter 3 provides the detailed description of these methods along with known convergence results. Numerical experiments are presented in chapter 4. Finally, the conclusions drawn from numerical simulations are summarized in the last chapter.

Chapter 1

Literature Review

Optimization problems under uncertainty arise in many application areas. In order to tackle such problems one needs to combine methods from diverse fields, such as optimization theory, probability and statistics, functional analysis and numerical analysis. This thesis presents some of the theoretical background required to understand general approaches to optimization problems under uncertainty. Particular emphasis is placed on one of the main challenges in the numerical solution of such problems - multivariate integration.

In practice many problems are formulated as systems of partial differential equations (PDEs). The uncertainty is incorporated into such systems in the form of random inputs: coefficients, source terms, boundary and initial data, geometry. In general, uncertainty is described as a random field, i.e., a random variable indexed by a spatial or temporal variable. PDEs in which random effects are manifested in parameters are called random PDEs [75, Sec. 4.7]. The optimization problems studied in this thesis are constrained by random PDEs. In contrast, stochastic PDEs (SPDEs) "are forced by 'an irregular process such as a Wiener process or Brownian motion." [75, Sec. 97].

The solutions to random PDEs are also random fields. Representation of random fields is, therefore, one of the first important problems that needs to be considered.

Two main approaches to describing random fields exist and are used in practice: Karhunen-Loève expansion (Karhunen (1947) [39], Loève (1948) [50]) and expansions in terms of global orthogonal polynomials (Wiener (1938) [82], Ghanem and Spanos (1997) [23]). Both types of expansion represent a random field in terms of an infinite number of random variables. In this thesis I will use the Karhunen-Loève (KL) expansion. The KL expansion represents the random field as an infinite series by utilizing the eigenvalues and eigenfunctions of its covariance function. In practice this series is truncated to obtain an approximation of the random field. In particular, for correlated random fields with correlation length comparable to the size of the domain only few terms are needed to describe random field with sufficient accuracy ([18], [51, Ch. 7]). Finite dimensional representation of random fields allows numerical solution of corresponding random PDEs.

Once the inputs of a system, e.g., coefficients and source terms of a random PDE, are described by a finite dimensional random vector, the task of uncertainty quantification becomes that of obtaining statistical information about the outputs of interest that depend on the solution of a random PDE. In many cases in the literature this is limited to obtaining expected value and other statistical moments ([2, 57]). In the context of optimization under uncertainty, for example in optimal control under uncertainty problems, obtaining solution that holds "on average" is usually not sufficient ([66]). For most practical cases the problem needs to be reformulated to account, for example, for rare events or large deviations from the expected value. This results in problems where the goal is to minimize the risk associated with a certain choice of control. The risk in this context is understood as a measure of variation of the objective with respect to the random parameters. Of particular interest is the class of "coherent risk measures" - a term first introduced by Artzner et al. [1] and rigorously studied by many others ([65, 70]). Coherent risk measures are particularly well suited for optimization due to their properties (convexity, monotonicity).

Whether seeking to approximate the expected value of the objective function or estimate some other risk measure associated with it, we are faced with a task of integrating with respect to the random parameters. As the number of random parameters grows, numerical integration becomes challenging. The simple strategy of using tensor products of one-dimensional integration rules fails due to the so-called "curse of dimensionality" - a term coined by Bellman ([4]) that refers to the exponential growth of the number of nodes required to evaluate multivariate integral. For example, evaluating an integral of a function of 10 variables using just 5-point rules requires 5^{10} function evaluations. In the framework considered here, each evaluation of the objective function requires solving an random PDE, hence it becomes crucial to cut the number of integration points as much as possible while preserving the desired accuracy.

In this thesis I consider several methods for multivariate integration starting with the simplest Monte Carlo (MC) approximation which is abundantly covered in literature (e.g., [34, 49]). Although the MC method has some attractive features, such as embarrassingly parallel implementation, in many cases its probabilistic rate of convergence of $\mathcal{O}(1/\sqrt{N})$ is considered too slow. The desire to improve on this rate resulted in the development of quasi-Monte Carlo methods in the 1950s and 1960s. Quasi-Monte Carlo methods are purely deterministic methods that heavily rely on number-theoretic concepts ([54, 11]). Development of QMC methods was further motivated by applications. In particular, option pricing problems have prompted many theoretic developments ([26, 29]). A comprehensive survey of QMC methods is presented in [16].

In the context of solving random PDEs MC methods have been combined with the multilevel methods in order to further reduce computational burden ([25, 14, 46, 47]). The main idea behind these methods is to use the hierarchy of spatial discretizations of the PDE in combination with discretizations of random variables. Although outside the scope of this thesis these methods provide an important direction to consider for the future work.

Another family of methods aimed at circumventing the curse of dimensionality are known under the common name of sparse grid methods. These methods stem from the Smolyak algorithm for approximating the high-dimensional tensor product quadrature ([76]). Smolyak's construction uses the combination of tensor products of one-dimensional quadrature formulas in a way that reduces the dependence of the total number of integration nodes on the dimension up to logarithmic factors (for certain classes of functions). Smolyak algorithm has been applied to the problems of numerical integration ([58, 59, 60, 20]), solution of stochastic PDEs ([8, 56, 57]) and many other problems. In this thesis I review the algorithm and the error estimates that are available in literature (e.g., [3]).

As mentioned above Smolyak's construction is general enough to be applied in different contexts. For the problems of multivariate integration and interpolation, which are the main focus of this thesis, this construction can be applied in two settings. The difference between the two is the type of bases used. The global polynomial bases, e.g., Lagrange polynomials, are good for sufficiently smooth integrands but are ineffective for the integrands that have, for example, sharp gradients or jump discontinuities. By contrast methods based on locally supported piecewise polynomial bases are better suited for non-smooth integrands and allow adaptivity. If the global polynomial approaches achieve greater interpolation accuracy by increasing the polynomial degree, the piecewise polynomial approaches do so by keeping the polynomial degree fixed and refining the grid instead.

The sparse grid approach based on piecewise linear functions was introduced by Zenger in 1991 [83]. It has since been extensively covered in literature (in particular, by Bungartz and Griebel [8]) and generalized to higher order polynomial bases ([5, 6]). The hierarchical construction of such bases proved to be particularly well-suited for implementation of spatial adaptivity ([28, 7]).

Adaptive sparse grids have been a subject of many studies. Gerstner and Griebel in [21] have introduced dimension-adaptive sparse grids that allowed for detection of "important" dimensions, i.e., dimensions corresponding to higher variability of the function of interest. Locally adaptive sparse grids using hierarchical basis functions have been applied in the context of solving stochastic PDEs ([19, 52]), solving regression and classification problems ([63, 64]), and multivariate quadrature ([7]). The attempts to consolidate both dimension and local adaptivity were also reported ([38]).

The implementations of various sparse grid methods that are publicly available include the library `spinterp` by Klimke and Wohlmuth written in Matlab ([40]), the TASMANIAN toolbox developed at the Oak Ridge National Laboratory ([79]), and the SG++ toolbox by a group at the Technische Universität München ([63]).

Chapter 2

Optimization Under Uncertainty

The motivation for this thesis comes from the optimization problems governed by random partial differential equations (PDEs). Random PDEs are PDEs for which the input data (such as advection/diffusion/reaction coefficients, forcing terms,etc.) is modeled as random variables, or more generally, random fields. This thesis does not consider stochastic differential equations which require different techniques for solving and analysis (for explanation of the difference see [75, Sec. 4.7]).

In this chapter I first introduce the general optimization problem with a constraint given by a random PDE. I outline a general theory for solving random PDEs and provide some examples that will be used in numerical simulations. Finally, I describe the notion of the risk measure and motivate the exploration of adaptive integration methods later in the thesis.

2.1 General Problem Setting

As a motivation for the rest of the thesis I first introduce an abstract optimization problem with equality constraints in general Banach spaces.

Let $\mathcal{U}$, $\mathcal{Z}$ be reflexive Banach spaces and $\mathcal{W}$ be a Banach space. Consider optimization problems of the form

$$\min J(u,z) \tag{2.1.1a}$$

subject to

$$e(u,z) = 0, \tag{2.1.1b}$$

$$u \in \mathcal{U}_{ad},\ z \in \mathcal{Z}_{ad}, \tag{2.1.1c}$$

where $\mathcal{U}_{ad}$ and $\mathcal{Z}_{ad}$ are admissible subsets of the state space $\mathcal{U}$ and control space $\mathcal{Z}$, respectively. The objective function $J : \mathcal{U} \times \mathcal{Z} \to \mathbb{R}$ represents the "cost" corresponding to the state u and control z, and the equality constraint $e : \mathcal{U} \times \mathcal{Z} \to \mathcal{W}$ describes the dynamics of the system and represents a PDE or a system of PDEs. Optimization problems in Banach spaces of the type (2.1.1) are discussed, e.g., in the book by Hinze et al. [37].

When the data of the system are not known completely, this uncertainty can be modeled by introducing random variables. Therefore, assume that the state equation $e(u,z)$ depends on some stochastic parameter $\omega \in \Omega$, where Ω is a set of outcomes (future states of knowledge). Assume that the control variable z needs to be chosen before the outcomes of ω become known. Thus z does no depend on ω. The state equation now reads:

$$e(u(\omega), z, \omega) = 0 \;\text{ a.e. in } \Omega \tag{2.1.2}$$

where "a.e." stands for almost everywhere.

Introducing randomness in the constraint equation (e.g., random diffusion coefficient in advection-diffusion equation) forces the state u to be a random variable depending on parameter ω. The solution space for the state equation (2.1.2) is a Bochner space, which I introduce next.

Definition 2.1.1 *Let X be a Banach space and $(\Omega, \mathcal{F}, P)$ be a measure space. The Bochner space $L^p_P(\Omega; X)$ is defined as the space of strongly measurable functions $v : \Omega \to X$ such that*

$$\int_\Omega \|v(\omega)\|^p_X dP(\omega) < \infty \tag{2.1.3}$$

for $p \in [1, \infty)$ *and*

$$\operatorname*{ess\,sup}_{\omega \in \Omega} \|v(\omega)\|_X < \infty \tag{2.1.4}$$

for $p = \infty$.

Let $u \in L^p_P(\Omega; \mathcal{U})$ be a solution to the state equation (2.1.2). For a fixed z the map $\omega \to J(u(\omega), z) : \Omega \to \mathbb{R}$ is now a random variable as well. To handle the uncertainty the problem (2.1.1) is reformulated as follows:

$$\min \sigma(J(u(\cdot), z)) \tag{2.1.5a}$$

subject to

$$e(u(\omega), z, \omega) = 0, \qquad \text{a.e. in } \Omega \tag{2.1.5b}$$

$$u(\omega) \in \mathcal{U}_{ad}, \qquad \text{a.e. in } \Omega \tag{2.1.5c}$$

$$z \in \mathcal{Z}_{ad}, \tag{2.1.5d}$$

where σ is a risk measure ("measure of the risk of loss" in [65]). The precise meaning of this term along with some examples will be provided later in the chapter.

I postpone the discussion of the risk measures until the end of the chapter. First I introduce the theory of random PDEs.

2.2 Random Partial Differential Equations

Let $D \subset \mathbb{R}^d$ with $d = 1, 2,$ or 3. Let $(\Omega, \mathcal{F}, P)$ be a complete probability space, where Ω denotes the set of outcomes, $\mathcal{F}$ is a σ-algebra of events, and $P : \mathcal{F} \to [0, 1]$ is a probability measure. Let $\mathcal{L}$ be a differential operator, linear or nonlinear, that depends on some coefficients $a(\omega, x)$ with $\omega \in \Omega$ and $x \in D$.

Consider the following informal statement of a random boundary value problem: find a function $u : \Omega \times \bar{D} \to \mathbb{R}$, such that P-almost everywhere in ω, or in other words almost surely (a.s.), the following equation holds:

$$\mathcal{L}(a(\omega,\cdot))(u) = f(\omega,\cdot) \qquad \text{in } D, \tag{2.2.1a}$$

$$u = g(\omega,\cdot) \qquad \text{on } \partial D. \tag{2.2.1b}$$

Here the forcing term f and the boundary data g can also be random fields ($f(\omega,x)$, $g(\omega,x)$), or they can be deterministic ($f(x)$, $g(x)$).

The precise statement of (2.2.1) and its well-posedness need to be discussed for the specific examples considered in the following.

Example 2.2.1 Consider the following random linear elliptic partial differential equation:

$$-\nabla\cdot(a(\omega,x)\nabla u(\omega,x)) = f(\omega,x) + z(x) \qquad (\omega,x)\in\Omega\times D, \tag{2.2.2a}$$

$$u(\omega,x) = 0 \qquad (\omega,x)\in\Omega\times\partial D. \tag{2.2.2b}$$

Here $a(\omega,x), f(\omega,x) : \Omega\times D\to\mathbb{R}$ are random fields, i.e., collections of realizations indexed by a random variable. Hence, the solution $u(\omega,x)$ of (2.2.2) is a random field as well.

In the following, I will follow [41] to show that, under suitable conditions, for almost all $\omega\in\Omega$ there exist a weak solution $u(\omega,\cdot)\in H_0^1(D)$ of (2.2.2).

For the purpose of solving (2.2.2) I consider the Bochner space $\mathcal{U} := L_P^2(\Omega; H_0^1(D))$, which is a Hilbert space with the inner product

$$\langle u,v\rangle_{\mathcal{U}} = \int_\omega \langle u(\omega,x), v(\omega,x)\rangle_{H_0^1(D)} dP(\omega) \tag{2.2.3}$$

Multiplying left-hand side of (2.2.2) by a test function $v\in\mathcal{U}$, integrating over $\Omega\times D$ and applying Green's identity we get

$$-\int_\Omega\int_D \nabla\cdot(a(\omega,x)\nabla u(\omega,x))v(\omega,x)dxdP(\omega) =$$
$$\int_\Omega\int_D a(\omega,x)\nabla u(\omega,x)\cdot\nabla v(\omega,x)dxdP(\omega) - \int_\Omega\int_{\partial D}\frac{\partial u}{\partial n}(\omega,x)v(\omega,x)dxdP(\omega),$$

where the second term on the right-hand side is 0 since $v(\omega, x) = 0$ almost surely (a.s.) for $x \in \partial D$.

Thus, the weak formulation of (2.2.2) is: find $u \in \mathcal{U}$ such that

$$\int_\Omega \int_D a(\omega, x)\nabla u(\omega, x) \cdot \nabla v(\omega, x) dx dP(\omega) = \int_\Omega \int_D f(\omega, x)v(\omega, x) dx dP(\omega) + \int_\Omega \int_D z(x)v(\omega, x) dx dP(\omega) \qquad (2.2.4)$$

for all $v \in \mathcal{U}$.

Under the assumption that $a \in L^\infty(\Omega \times D)$ is bounded away from zero almost everywhere on $\Omega \times D$, that is

$$\exists\, a_{min} > 0 \,\text{such that}\; a(\omega, x) \geq a_{min} \text{ a.e. in } \Omega \times D, \qquad (2.2.5)$$

and $f(\omega, x) \in L^2_P(\Omega; (H^{-1}(D)))$, one can apply Lax-Milgram theorem to show the existence of a unique solution $u \in \mathcal{U}$ to the problem (2.2.4) (for details see [41]). In fact, one can show the existence of $c > 0$ such that

$$\|u(\omega, \cdot)\|_{H^1_0(D)} \leq c\|f(\omega, \cdot)\|_{H^{-1}(D)} \quad \text{for almost all } \omega \in \Omega.$$

◇

2.2.1 Finite Noise Assumption

In general, random fields $a(\omega, x)$ and $f(\omega, x)$ (and any other possible sources of uncertainty) are not related to each other, therefore, they are defined on different probability spaces $(\Omega_a, \mathcal{F}_a, P_a)$ and $(\Omega_f, \mathcal{F}_f, P_f)$. The solution u is then defined on the product probability space $(\Omega, \mathcal{F}, P) = (\Omega_a \times \Omega_f, \mathcal{F}_a \times \mathcal{F}_f, P_a \times P_f)$. Thus, a and f are functions of ω_a and ω_f respectively.

For numerical purposes I will need the finite noise assumption, which states that the random fields $a(\omega_a, x)$ and $f(\omega_f, x)$ depend on a finite number of random variables. Formally, I assume that there exists a vector of random variables $Y_a(\omega_a) =$

$[Y_{a,1}(\omega_a), \dots, Y_{a,N_a}(\omega_a)]$ such that $a(\omega_a, x) = a(Y_a(\omega_a), x)$ and a vector of random variables $Y_f(\omega_f) = [Y_{f,1}(\omega_f), \dots, Y_{f,N_f}(\omega_f)]$ such that $f(\omega_f, x) = f(Y_f(\omega_f), x)$.

Further, I can relabel the elements of the random vectors Y_a and Y_f and define a vector of random variables $Y = [Y_1, \dots, Y_N] = (Y_a, Y_f)$ with $N = N_a + N_f$. Denote the image of the random variables Y_i to be $\Gamma_i \subset \mathbb{R}$ for $i = 1, \dots, N$ and set $\Gamma = \Gamma_1 \times \cdots \times \Gamma_N$. Now $Y : \Omega \to \Gamma \subset \mathbb{R}^N$. Assume that there exists a joint probability density function (pdf) for $\{Y_i\}_{i=1}^N$ denoted by $\rho : \Gamma \to \mathbb{R}_+$ with $\rho \in L^\infty(\Gamma)$. In this case the probability measure $dP(\omega)$ can be replaced by the Lebesgue measure $\rho(y)dy$ and one can perform the change of variables in (2.2.4) to obtain the following parametrized deterministic PDE:

$$\begin{aligned}\int_\Gamma \rho(y) \int_D a(y,x)\nabla u(y,x) \cdot \nabla v(y,x)dxdy = &\int_\Gamma \rho(y) \int_D f(y,x)v(y,x)dxdy \\ &+ \int_\Gamma \rho(y) \int_D z(x)v(y,x)dxdy \qquad (2.2.6)\end{aligned}$$

for all $v \in L^2_\rho(\Gamma; H^1_0(D))$. Here the solution space is defined analogously to previously defined Bochner space $L^p_P(\Omega; X)$:

$$L^p_\rho(\Gamma; X) = \{u : \Gamma \to X : u \text{ is strongly measurable}, \int_\Gamma \rho(y)\|u(y)\|^p_X dy < \infty\}. \qquad (2.2.7)$$

Under the finite noise assumption and by the Doob-Dynkin's lemma ([61]) the solution $u(\omega, x)$ can be characterized by the same vector $Y(\omega)$ as are a and f. Therefore, it has a deterministic parametric equivalent $u(y, x)$.

I provide two examples of the random field $a(\omega, x)$ that satisfy the finite noise assumption. In the first example the random field is finite-dimensional by definition. In the second, the infinite-dimensional random field is represented by an expansion of which only a few terms are retained.

Example 2.2.2 Piecewise constant random fields

Consider problem (2.2.2) where the spatial domain $D = \cup_{i=1}^N D_i$, $D_i \cap D_j = \emptyset$ for $i \neq j$. Let the diffusion coefficient $a_N(\omega, x)$ be piecewise constant and random on each subdomain D_i, i.e.,

$$a_N(\omega, x) = a_{min} + \sum_{i=1}^{N} \sigma_i Y_i(\omega) \chi_{D_i}(x) \tag{2.2.8}$$

where $\chi_{D_i}(x)$ is a characteristic function of the set D_i, $\sigma_i, a_{min} > 0$ and random variables Y_i are nonnegative with unit variance. ◇

Example 2.2.3 Karhunen-Loève expansion

According to Mercer's theorem (Theorem 11, chapter 30, section 5 of [48]), any second-order correlated random field $a(\omega, x)$ with continuous covariance function $\mathbb{COV}(x_1, x_2)$ can be represented as an infinite sum of random variables. One example of such an expansion is the Karhunen-Loève (KL) expansion ([39, 50, 51]), which is an extension of singular value decomposition. In this case the random field $a(\omega, x)$ can be approximated by a truncated KL expansion:

$$a(\omega, x) \approx a_N(\omega, x) = \mathbb{E}[a](x) + \sum_{n=1}^{N} \sqrt{\lambda_n} b_n(x) Y_n(\omega) \tag{2.2.9}$$

where λ_n and $b_n(x)$ for $n = 1, \ldots, N$ are the dominant eigenvalues and corresponding eigenfunctions of the covariance function and $Y_n(\omega)$ are uncorrelated real-valued random variables.

In order to keep the property that a random input coefficient is bounded away from zero, I will use KL expansion for the logarithm of the random field

$$\log((a_N - a_{min})(\omega, x)) = b_0(x) + \sum_{n=1}^{N} \sqrt{\lambda_n} b_n(x) Y_n(\omega) \tag{2.2.10}$$

where a_{min} is the lower bound on a.

◇

Similar expansions are used for the random field $f(\omega, x)$ when applicable.

2.3 Risk Measures

Consider the general optimization problem under uncertainty (2.1.5). Under the finite noise assumption it becomes:

$$\min \sigma(J(u(\cdot), z)) \tag{2.3.1a}$$

subject to

$$e(u(y), z, y) = 0, \qquad \text{a.e. in } \Gamma \tag{2.3.1b}$$

$$u(y) \in \mathcal{U}_{ad}, \qquad \text{a.e. in } \Gamma \tag{2.3.1c}$$

$$z \in \mathcal{Z}_{ad}, \tag{2.3.1d}$$

Assume that for every $z \in \mathcal{Z}_{ad}$ equation $e(u(y), z, y) = 0$ has a unique solution denoted by $u(y; z)$. Then the mapping $y \mapsto J(u(y; z), z) : \Gamma \to \mathbb{R}$ is a random variable. This randomness in the objective is handled using risk measures. By a risk measure we understand an operator that acts on the space of functions with domain Γ and codomain $\mathbb{R}$ such as $L^q_\rho(\Gamma) \equiv L^q_\rho(\Gamma; \mathbb{R})$ for $q = [1, \infty)$ or $q = \infty$. Denote it by $\sigma : L^q_\rho(\Gamma) \to \mathbb{R}$.

Next I introduce several examples of risk measures σ. In order to simplify notation I will denote the random variable $J(u(y; z), z)$ for a fixed z by $X(y)$. The assumptions on $X(y)$ will be stated separately for each of the risk measures.

2.3.1 Examples

Expected value

The simplest and most commonly used choice is

$$\sigma(X) = \mathbb{E}[X] \tag{2.3.2}$$

This choice requires that $X \in L^1_\rho(\Gamma)$. Although easy to deal with, the expected value is not sensitive to the possibilities of high values of X, and, thus, incorporates no level of risk averseness, i.e., does not take into account possibility of high cost and associated consequences. ([65]).

Worst case

The choice

$$\sigma(X) = \sup_{y \in \Gamma} X(y)\rho(y) \tag{2.3.3}$$

for $X \in L^{\infty}_{\rho}(\Gamma)$ accounts for the "worst case scenario". In most situations it is overly conservative (infinite for normal and exponential distributions), as it ignores all information about the distribution of X except its highest possible realization.

Mean plus deviation

The expected value risk function can be modified by adding a penalty term. To this end define the L^q norm of a random variable $X \in L^q_{\rho}(\Gamma) = L^q_{\rho}(\Gamma; \mathbb{R})$ by

$$\|X\|_q = \begin{cases} \mathbb{E}[|X|] & \text{for } \ q = 1 \\ (\mathbb{E}[|X|^q])^{1/q} & \text{for } \ 1 < q < \infty \\ \sup\limits_{y \in \Gamma} |X| & \text{for } \ q = \infty \end{cases} \tag{2.3.4}$$

and consider

$$\sigma(X) = \mathbb{E}[X] + c\|X - \mathbb{E}[X]\|_q \tag{2.3.5}$$

with $c \geq 0$. For $q = 2$ this corresponds to mean plus c units of standard deviation. This risk measure introduces safety margins for the expected value, thus, assuring that high costs are incurred only for the outcomes corresponding to the upper part of the distribution of X lying more than c deviation units above the mean. However, it does so in a symmetrical manner, thus, penalizing not only outcomes with cost higher than average but also those that are lower.

Mean plus semi-deviation

The mean plus semi-deviation risk measure is defined as $\sigma(X) = \mathbb{E}[X] + c\|[X -$

$\mathbb{E}[X]]_+\|_q$ with $c \geq 0$. Here we denote by $[\cdot]_+$ the function $\max(\cdot, 0)$. Using this risk measure as an objective in the minimization we penalize only costs that are higher than average.

We can rewrite $\sigma(X)$ in terms of auxiliary function $\hat{\sigma}_q : \mathbb{R} \times L^q_\rho(\Gamma) \to \mathbb{R}$ defined by

$$\hat{\sigma}_q(t, X) = t + c\|[X - t]_+\|_q \tag{2.3.6}$$

by simply taking $\sigma(X) = \hat{\sigma}_q(\mathbb{E}[X], X)$.

Conditional-value-at-risk

In order to account for tail probabilities and extreme events one might want to employ the conditional-value-at-risk (CVaR), which is defined as

$$\sigma(X) = \min_{t \in \mathbb{R}} t + (1 - \alpha)^{-1} \, \mathbb{E}[[X - t]_+] \tag{2.3.7}$$

with $\alpha \in (0, 1)$ ([67]).

It too can be written in terms of the auxiliary function (2.3.6) as $\sigma(X) = \min_{t \in \mathbb{R}} \hat{\sigma}_1(t, X)$ with $c = (1 - \alpha)^{-1}$.

The conditional-value-at-risk satisfies the following property ([65]):

$\mathrm{CVaR}_\alpha(X)$ depends continuously on $\alpha \in (0, 1)$ with

$\lim_{\alpha \to 1} \mathrm{CVaR}_\alpha(X) = \sup_{y \in \Gamma} X(y)\rho(y)$ and $\lim_{\alpha \to 0} \mathrm{CVaR}_\alpha(X) = \mathbb{E}[X]$.

2.3.2 Application to optimization problems

Among the introduced risk measures some are more useful in the context of optimization under uncertainty than the others. One desired property is coherency.

Definition 2.3.1 (Coherent Risk Measure). *A functional $\sigma : L^q_\rho(\Gamma) \to \mathbb{R}$ is called a coherent measure of risk in the sense of Artzner, Delbaen, Eber, and Heath [1], if it satisfies the following properties:*

- *convexity: for all* $X_1,\ X_2 \in L^q_\rho(\Gamma)$ *and* $\lambda \in [0,1]$

$$\sigma(\lambda X_1 + (1-\lambda)X_2) \leq \lambda\sigma(X_1) + (1-\lambda)\sigma(X_2)$$

- *monotonicity: for all* $X_1,\ X_2 \in L^q_\rho(\Gamma)$ *such that* $X_1 \leq X_2$ *a.s.*

$$\sigma(X_1) \leq \sigma(X_2)$$

- *translation equivariance: for all* $X \in L^q_\rho(\Gamma)$ *and* $c \in \mathbb{R}$

$$\sigma(X + c) = \sigma(X) + c$$

- *positive homogeneity: for all* $c > 0$ *and* $X \in L^q_\rho(\Gamma)$

$$\sigma(cX) = c\sigma(X)$$

The main consequences of coherency for optimization problem (2.3.1) are summarized in the following theorem from [65]:

Theorem 2.3.2 *Suppose the functional* σ *in* (2.3.1) *is a coherent measure of risk.*

- *Preservation of convexity. If the mapping* $z \mapsto J(u(y;z),z)$ *is convex for each* y*, then* $z \mapsto \sigma(J(u(\cdot,z),z))$ *is convex. Thus, if the problem without uncertainty were a problem of convex programming, this advantage would be inherited by the formulation* (2.3.1)*.*

- *Preservation of certainty. If* $J(u(y;z),z)$ *is a constant random variable for each* z*, i.e.,* $J(u(y;z),z) = J(u(z),z)$ *with no contibution from* y*, then* $\sigma(J(u(\cdot,z),z)) = J(u(z),z)$*.*

- *Insensitivity to scaling. The problem* (2.3.1) *remains the same even if the values of* $J(u(y;z),z)$ *are denominated or rescaled.*

Among the risk measures introduced earlier the expected value, the supremum, the mean plus semi-deviation with $c \in [0, 1]$ and CVaR are coherent measures of risk [65, 70]. The mean plus deviation risk measure for $q > 1$ violates the monotonicity property [71], [74, Sec. 6.3].

Another property of risk measures that is often considered in literature is averseness [68], which is simply understood as

$$\sigma(X) > \mathbb{E}(X) \tag{2.3.8}$$

In this sense the expected value is a risk-neutral risk measure, while mean plus semi-deviation and CVaR are risk-averse.

Some of the risk measures that might be used in the context of optimization under uncertainty introduce non-smoothness in the objective via the $[\cdot]_+$ function. My goal is to adress this issue carefully when considering discretization methods. In particular, I will compare performance of various methods for high-dimensional integration on the functions with non-smoothness introduced by $[\cdot]_+$ function.

Chapter 3

High-dimensional Integration Methods

In the problems of evaluating risk measures of quantities of interest that depend on the solution of a PDE we have to deal with integrating over a high-dimensional domain of random parameters. Moreover, for certain risk measures, such as mean plus semi-deviation and CVaR, the integrand becomes non-smooth ($[\cdot]_+$ function). The convergence of integration methods based on global polynomial interpolation deteriorates for such problems, that is such methods converge slowly or do not converge at all. Therefore, I consider methods based on local approximations.

In the following I give an overview of methods for evaluating multivariate integrals, starting with Monte Carlo and quasi-Monte Carlo methods. Next I describe quadrature methods based on global polynomials. Finally, I focus on methods based on local polynomial approximations and adaptive refinement strategies.

Let $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{y} = (y^{(1)}, \ldots, y^{(d)})$. Consider a problem of integrating a function $f : \mathbb{R}^d \to \mathbb{R}$ from a function class $\mathcal{F}$ to be specified later over $\Gamma \subset \mathbb{R}^d$.

Denote the exact integral of f with respect to the weight function $\rho(\mathbf{y})$ over the domain Γ by

$$I_\rho^d[f] = \int_\Gamma f(\mathbf{y})\rho(\mathbf{y})d\mathbf{y} \tag{3.0.1}$$

3.1 Monte Carlo Method

The classical Monte Carlo (MC) method is an equal-weight cubature rule given by

$$Q_N^{MC}[f] = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{y}_i) \tag{3.1.1}$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_N$ are independent and identically distributed (i.i.d.) random samples drawn from $\rho(\mathbf{y})$.

Theorem 3.1.1 (MC root-mean-square error) *For all square-integrable functions f,*

$$\sqrt{\mathbb{E}[|I_\rho^d[f] - Q_N^{MC}[f]|^2]} = \frac{\sqrt{\mathbb{VAR}[f]}}{\sqrt{N}} \tag{3.1.2}$$

where the expected value is taken with respect to the random samples $\mathbf{y}_1, \ldots, \mathbf{y}_N$, and

$$\mathbb{VAR}[f] = I_\rho^d[f^2] - (I_\rho^d[f])^2 \tag{3.1.3}$$

is the variance of f.

For proof see, e.g., Theorem 2.1 in [16].

It is easy to see that $\mathbb{E}[Q_N^{MC}[f]] = I_\rho^d[f]$, i.e., that MC method is unbiased, and $\mathbb{VAR}[Q_N^{MC}] = \mathbb{E}[|Q_N^{MC}[f] - I_\rho^d[f]|^2] = \frac{\mathbb{VAR}[f]}{N}$. By the Central Limit Theorem, if $0 < \mathbb{VAR}[f] < \infty$, then

$$\lim_{N\to\infty} P\left(|I_\rho^d[f] - Q_N^{MC}[f]| \le C\sqrt{\frac{\mathbb{VAR}[f]}{N}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-C}^{C} \exp(-\frac{x^2}{2})dx. \tag{3.1.4}$$

(See e.g. [11]).

This gives a probabilistic error bound with a convergence rate of $\mathcal{O}(N^{-1/2})$. Note, that the convergence rate is independent of the dimension.

Perhaps one of the main advantages of MC method is its simplicity that allows for embarassingly parallel implementation. However, the convergence rate of $\mathcal{O}(N^{-1/2})$ is in many cases too slow, especially, in the context of solving random PDEs. The efficiency of MC methods can be improved using variance reduction techniques, such as, importance sampling, stratified sampling, correlated sampling (see, for example, [11, Sec. 4]), or multi-level MC method (see [36, 24, 25]). But often MC method is still considered too slow. This is the main motivation for the switch to quasi-Monte Carlo methods.

3.2 Quasi-Monte Carlo Methods

In the following assume that $\Gamma = [0,1]^d$ and $\rho(\mathbf{y}) \equiv 1$. Thus, we are concerned with approximating $I_1^d[f]$.

Quasi-Monte Carlo (QMC) methods are equal-weight cubature rules of the form

$$Q_N^{QMC}[f] = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{y}_i) \tag{3.2.1}$$

just like MC method, however, the points $\mathbf{y}_1, \ldots, \mathbf{y}_N$ are now chosen deterministically.

There are two types of QMC methods:

- the “open” type rules use the first N points of an infinite series, thus, increasing N only requires evaluation of the integrand at the newly added points;
- the “closed” type rules use a finite point set depending on N. Different values of N require evaluating integrand at completely different points.

In order to provide some examples of QMC rules I need the following definition.

Definition 3.2.1 *Let $i \geq 0$ and $b \geq 2$ be integers. The radical inverse function $\phi_b(i)$ is given by*

$$\phi_b(i) = \sum_{a=1}^{\infty} \frac{i_a}{b^a} \tag{3.2.2}$$

where $i = \sum\limits_{a=1}^{\infty} i_a b^{a-1}$ and $i_a \in \{0, 1, \ldots, b-1\}$. In other words, if $i = (\ldots i_2 i_1)_b$ is a representation of i in base b, then $\phi_b(i) = (0.i_1 i_2 \cdots)_b$.

Example 3.2.2 van der Corput sequence

The van der Corput sequence in base b is the one-dimensional sequence

$$\phi_b(0), \phi_b(1), \phi_b(2), \ldots$$

For example, let $b = 2$. Write down the natural numbers $0, 1, 2, \ldots$ in base 2:

$$0, 1_2, 10_2, 11_2, 100_2, 101_2, 110_2, \ldots$$

Applying ϕ_2 to each number we obtain the sequence

$$0, 0.1_2, 0.01_2, 0.11_2, 0.001_2, 0.101_2, 0.011_2, \ldots$$

which in decimal form is a sequence

$$0, 0.5, 0.25, 0.75, 0.125, 0.625, 0.375, \ldots$$

◇

Example 3.2.3 Halton sequence

Let $p_1, p_2, \ldots, p_d$ be the first d prime numbers. The Halton sequence $\mathbf{y}_1, \mathbf{y}_2, \ldots$ in d dimensions is given by

$$\mathbf{y}_{i+1} = (\phi_{p_1}(i), \phi_{p2}(i), \ldots, \phi_{p_d}(i)), \quad i = 0, 1, 2, \ldots \tag{3.2.3}$$

that is, the j-th components of points in the Halton sequence form the van der Corput sequence in base p_j, where p_j is the j-th prime. The Halton sequence is an example of an “open” QMC method. Explicitly,

$$\begin{aligned}
\mathbf{y}_1 &= (0, 0, 0, \ldots, 0), \\
\mathbf{y}_2 &= (0.1_2, 0.1_3, 0.1_5, \ldots, 0.1_{p_d}), \\
\mathbf{y}_3 &= (0.01_2, 0.2_3, 0.2_5, \ldots, 0.2_{p_d}), \\
\mathbf{y}_4 &= (0.11_2, 0.01_3, 0.3_5, \ldots, 0.3_{p_d}), \\
&\vdots
\end{aligned}$$

◇

The Halton sequence satisfies the error bound

$$|I_1^d[f] - Q_N^{QMC}[f]| \leq C_d \frac{(\log N)^d}{N} V[f], \text{ for } N \geq 2$$

as shown in [33] with the constant C_d depending on dimension d, and $V[f]$ being a total variation in the sense of Hardy and Krause (to be defined later).

Example 3.2.4 Hammersley point set

Let $p_1, p_2, \ldots, p_{d-1}$ be the first $d-1$ prime numbers. The Hammerseley point set $\mathbf{y}_1, \ldots, \mathbf{y}_N$ in d dimensions is given by

$$\mathbf{y}_{i+1} = (\frac{i}{N}, \phi_{p_1}(i), \phi_{p_2}(i), \ldots, \phi_{p_{d-1}}(i)), \quad i = 0, 1, \ldots, N-1 \tag{3.2.4}$$

The Hammersley point set leads to a "closed" QMC method. Explicitly,

$$\begin{aligned}
\mathbf{y}_1 &= (0, 0, \ldots, 0), \\
\mathbf{y}_2 &= (\frac{1}{N}, 0.1_2, 0.1_3, \ldots, 0.1_{p_{d-1}}), \\
\mathbf{y}_3 &= (\frac{2}{N}, 0.01_2, 0.2_3, \ldots, 0.2_{p_{d-1}}), \\
&\vdots \\
\mathbf{y}_N &= (\frac{N-1}{N}, \ldots).
\end{aligned}$$

◇

The Hammersley point set satisfies the error bound

$$|I_1^d[f] - Q_N^{QMC}[f]| \leq C_d \frac{(\log N)^{d-1}}{N} V[f], \text{ for } N \geq 2 \tag{3.2.5}$$

as shown in [33] with the constant C_d depending on the dimension d. Note that there is one less power of $\log N$ compared to the error bound for the Halton sequence. In general, the error bounds for QMC methods based on "closed" point sets are better than those based on "open" sequences.

The examples provided so far and the ones that will follow are examples of so-called "low discrepancy sequences". In the next subsection some of the main notions related to the theory of low discrepancy sequences are defined.

3.2.1 Theory of low discrepancy sequences

The infinite sequence of numbers is called a low-discrepancy sequence if for all values of N its subsequence $\mathbf{y}_1, \dots, \mathbf{y}_N$ has a low discrepancy. Let $\mathbf{y}_1, \dots, \mathbf{y}_N$ be N numbers in $\Gamma = [0,1]^d$. If B is a subset of Γ, then

$$A(B;N) = \sum_{n=1}^{N} \chi_B(\mathbf{y}_n) \tag{3.2.6}$$

counts the number of points $\mathbf{y}_n$ that fall into B.

Definition 3.2.5 *The discrepancy D_N of the N numbers $\mathbf{y}_1, \dots, \mathbf{y}_N$ is defined, using Niederreiter's notation ([54]), as*

$$D_N = \sup_{B \in J} \left| \frac{A(B;N)}{N} - |B| \right| \tag{3.2.7}$$

where J is the family of d-dimensional boxes of the form $\prod_{i=1}^{d} [a_i, b_i) = \{\mathbf{y} \in \mathbb{R}^d \ : \ a_i \leq y^{(i)} < b_i\}$ contained in Γ and $|B|$ denotes the d-dimensional Lebesgue measure of B.

A useful variant of the above definition is a so-called star-discrepancy.

Definition 3.2.6 *The star-discrepancy D_N^* of the N numbers $\mathbf{y}_1, \dots, \mathbf{y}_N$ in Γ is defined by*

$$D_N^* = \sup_{B \in J^*} \left| \frac{A(B;N)}{N} - |B| \right| \tag{3.2.8}$$

where J^* is the family of boxes of the form $\prod_{i=1}^{d} [0, t_i)$ contained in Γ.

The star-discrepancy is perhaps the more natural one in statistics, since it measures the maximum difference between the empirical cumulative distribution function of the points $\mathbf{y}_1, \dots, \mathbf{y}_N$ and the uniform distribution of measure on the unit cube. If we construct the empirical cumulative distribution function of the points $\mathbf{y}_1, \dots, \mathbf{y}_N$

$$\hat{F}_N(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{\mathbf{y}_n \leq \mathbf{y}\} \tag{3.2.9}$$

where $\mathbb{1}(E)$ is the indicator function of event E, and compare it with the theoretical uniform distribution on Γ given by

$$F(\mathbf{y}) = F(y^{(1)}, \ldots, y^{(d)}) = \min\{1, y^{(1)} \cdot y^{(2)} \cdot \ldots \cdot y^{(d)}\} \text{ if all } y^{(i)} \geq 0, \qquad (3.2.10)$$

then the star-discrepancy D_N^* is the Kolmogorov-Smirnov distance between these two cumulative distribution functions

$$D_N^* = \sup_{\mathbf{y}} |\hat{F}_N(\mathbf{y}) - F(\mathbf{y})|. \qquad (3.2.11)$$

For an infinite sequence of points in Γ the conditions $\lim_{N\to\infty} D_N = 0$ and $\lim_{N\to\infty} D_N^* = 0$ are equivalent to the sequence being uniformly distributed in Γ ([45, Chapter 2, § 1]). Thus, finite sequences of points with small discrepancy provide a good approximation to uniform distribution.

It makes intuitive sense that the points $\mathbf{y}_1, \ldots, \mathbf{y}_N$ should be chosen so that the discrepancy is small for each N. This intuition is supported by theoretical results, at least in the case of smooth integrands with smooth partial derivatives. In order to present these results I need a notion of total variation for a function of several variables.

For a function f on $\Gamma = [0,1]^d$ and a box $B = [y_1^{(1)}, y_2^{(1)}] \times \cdots \times [y_1^{(d)}, y_2^{(d)}] \subseteq \Gamma$ let

$$\Delta(f; B) = \sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} (-1)^{i_1 + \ldots i_d} f(y_{i_1}^{(1)}, \ldots, y_{i_d}^{(d)}).$$

Next define a partition $\mathcal{P}$ of Γ to be a union of boxes of the form $[y_{i_1}^{(1)}, y_{i_1+1}^{(1)}] \times \cdots \times [y_{i_d}^{(d)}, y_{i_d+1}^{(d)}]$ with $0 \leq i_j < m_j$ for $j = 1, 2, \ldots, d$.

Definition 3.2.7 *For a function f on Γ set*

$$V^{(d)}[f] = \sup_{\mathcal{P}} \sum_{B \in \mathcal{P}} |\Delta(f; B)| \qquad (3.2.12)$$

where the supremum is taken over all partitions $\mathcal{P}$ of Γ. If $V^{(d)}[f]$ is finite, then f is said to be of bounded variation on Γ in the sense of Vitali.

For sufficiently regular functions $V^{(d)}[f]$ can be represented by an integral ([54, Sec. 2]):

$$V^{(d)}[f] = \int_0^1 \dots \int_0^1 \left| \frac{\partial^d f}{\partial t_1 \dots \partial t_d} \right| dt_1 \dots d_d \tag{3.2.13}$$

whenever the partial derivative under the integral is continuous.

Note that if function f is constant with respect to some of the variables, then $\Delta(f; B) = 0$ and, hence, $V^{(d)}[f] = 0$. Since such a function might still be irregular, a more suitable notion of variation is required.

The next definition generalizes the notion of total variation to the case when function f does not depend on all d variables.

Definition 3.2.8 *Let f be a function on Γ. For $1 \leq k \leq d$ and $1 \leq i_1 < i_2 < \dots < i_k \leq d$, denote by $V^{(k)}[f; i_1, \dots, i_k]$ the k-dimensional variation in the sense of Vitali of the restriction of f to $\{(t_1, \dots, t_d) \in \Gamma \ : \ t_j = 1 \text{ for } j \neq i_1, \dots, i_k\}$. If all variations $V^{(k)}[f; i_1, \dots, i_k]$ are finite, then f is said to be of bounded variation on Γ in the sense of Hardy and Krause.*

Since $V^{(d)}[f] = V^{(d)}[f; 1, 2, \dots, d]$, a function of bounded variation in the sense of Hardy and Krause is automatically a function of bounded variation in the sense of Vitali. For more information on these types of variation see [45].

Denote by $V[f]$ the total variation of f in the sense of Hardy and Krause defined as

$$V[f] = \sum_{k=1}^{d} \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq d} V^{(k)}[f; i_1, \dots, i_k] \tag{3.2.14}$$

The following inequality provides an estimate of the error in the approximation of the multi-dimensional integral by quasi-Monte Carlo methods.

Theorem 3.2.9 (Koksma-Hlawka inequality) *If f is a function of bounded variation on Γ in the sense of Hardy and Krause and $\mathbf{y}_1, \dots, \mathbf{y}_N$ are points in Γ, then*

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{y}_n) - \int_\Gamma f(\mathbf{y}) d\mathbf{y} \right| \leq V[f] D_N^* \tag{3.2.15}$$

For proof see [45, Chapter 2, § 5].

The bound in the theorem above allows a separation between the regularity properties of the integrand and the degree of uniformity (or level of discrepancy) of the sequence. As stated earlier the sequences used in quasi-Monte Carlo methods aim to achieve low level of discrepancy (on the order of $\frac{(\log N)^d}{N}$).

For the one-dimensional van der Corput sequence introduced earlier, we have that $D_N^* = \mathcal{O}(\frac{\log N}{N})$. The Halton sequence, which is a generalization of van der Corput sequence to multi-dimensional setting satisfies $D_N^* = C_d \frac{(\log N)^d}{N}$ with a constant C_d depending on dimension d. This constant, in fact, is asymptotic to d^d ([55]), thus, the uniformity of the points in the sequence degrades rapidly with increasing dimension. The following examples provide an improvement over Halton sequence for high dimensions.

Example 3.2.10 Faure sequence

Faure sequence is similar to the Halton sequence in that each dimension is a permutation of the van der Corput sequence. However, to construct the Faure sequence the same prime is used as the base b for each component of the vector. This prime is usually chosen to be the smallest prime greater than or equal to the dimension d.

Recall, that in the van der Corput sequence the natural numbers $i = 1, 2, \ldots$ were written in the form $i = \sum\limits_{a=1}^{\infty} i_a b^{a-1}$ and then mapped into the point $\sum\limits_{a=1}^{\infty} i_a b^{-a}$ in the unit interval via the radical inverse function $\phi_b(i)$. For the Faure sequence the same construction is used but with different permutations of the coefficients i_a for each of the coordinates. The first coordinate still repeats the van der Corput sequence. For the j-th coordinate with $j \geq 2$ we generate the point

$$\sum_{a=1}^{\infty} c_a b^{-a} \tag{3.2.16}$$

where

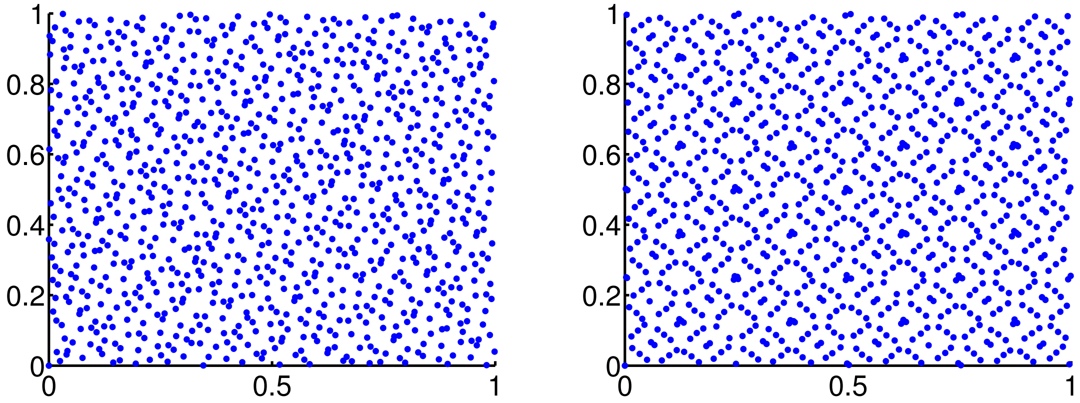
$$c_a = \sum_{m \geq a-1} \binom{m}{a-1} (j-1)^{m-a+1} i_{m+1} \mod b \tag{3.2.17}$$

For example, for the case $b = 2$ the first two coordinates of the first 10 Faure numbers are

$d = 1 : 0$	$1/2$	$1/4$	$3/4$	$1/8$	$5/8$	$3/8$	$7/8$	$1/16$	$9/16$
$d = 2 : 0$	$1/2$	$3/4$	$1/4$	$5/8$	$1/8$	$3/8$	$7/8$	$15/16$	$7/16$

The first two coordinates of the first 1000 points of Faure sequence in $d = 3$ are plotted in Figure 3.1a.

◇



(a) Faure sequence

(b) Sobol' sequence

Figure 3.1: The first two coordinates of the first 1000 QMC points in $d = 3$.

Example 3.2.11 Sobol' sequence

The Sobol' sequence is generated using a set of so-called direction numbers $v_i = \frac{m_i}{2^i}$, $i = 1, 2, \ldots,$ where m_i is an odd positive integer less than 2^i. The values of m_i are chosen to satisfy a recurrence relation using the coefficients of a primitive polynomial in the field $\mathbb{Z}_2$. A primitive polynomial is a polynomial of order p that is irreducible (i.e., cannot be factored into polynomials of smaller degree) and does not divide the polynomial $x^r + 1$ for $r < 2^p - 1$. For example, the polynomial $x^2 + x + 1$ has no non-trivial factors over the field $\mathbb{Z}_2$ and it does divide $x^3 + 1$ but not $x^r + 1$ for $r < 3$.

Corresponding to a primitive polynomial

$$z^p + c_1 z^{p-1} + \cdots + c_{p-1} z + c_p$$

is the recursion

$$m_i = 2c_1 m_{i-1} + 2^2 c_2 m_{i-2} + \cdots + 2^p c_p m_{i-p}$$

where the addition is carried out using binary arithmetic. For the Sobol' sequence the binary digit i_a is then replaced by $i_a v_a$.

The first two coordinates of the first 10 Sobol' numbers generated using irreducible polynomials $x + 1$ and $x^3 + x + 1$ are

$d = 1 : 0$	$1/2$	$1/4$	$3/4$	$3/8$	$7/8$	$1/8$	$5/8$	$5/16$	$13/16$
$d = 2 : 0$	$1/2$	$1/4$	$3/4$	$1/8$	$5/8$	$3/8$	$7/8$	$11/16$	$3/16$

The first two coordinates of the first 1000 points of Sobol' sequence in $d = 3$ are plotted in Figure 3.1b.

◇

The Faure and Sobol' sequences are particular cases of (t, s)-nets. In order to define them we need the concept of an elementary interval.

Definition 3.2.12 *An elementary interval in base b is an interval E in $[0, 1)^d$ of the form*

$$E = \prod_{j=1}^{d} \left[\frac{a_j}{b^{d_j}}, \frac{(a_j + 1)}{b^{d_j}} \right), \tag{3.2.18}$$

where $d_j \geq 0$, $0 \leq a_j \leq b^{d_j}$, and a_j, d_j are integers.

Definition 3.2.13 ((t, m, s)-net) *Let $0 \leq t \leq m$ be integers. A (t, m, s)-net in base b is a finite sequence with b^m points from $[0, 1)^d$ such that every elementary interval in base b of volume b^{t-m} (i.e., $d_1 + d_2 + \cdots + d_d = m - t$ in the definition of elementary interval) contains exactly b^t points of the sequence.*

Definition 3.2.14 ((t, s)-sequence) *An infinite set of points $\{\mathbf{y}_i\} \in [0,1)^d$ is a (t,s)-sequence in base b if for all $k \geq 0$ and $m > t$, the finite block $\mathbf{y}_{kb^m}, \ldots, \mathbf{y}_{(k+1)b^m-1}$ of b^m points forms a (t,m,s)-net in base b.*

The van der Corput sequence in base b is a $(0,1)$-sequence in base b; the Sobol' sequence is a (t,s)-sequence in base 2, where t is a non-decreasing function of s; the Faure sequence is a $(0,s)$-sequence ([16]).

For a (t,s)-sequence in base b the low discrepancy is ensured ([55, Theorem 4.17]):

$$D_N^* \leq C\frac{(\log N)^d}{N} + \mathcal{O}\left(\frac{(\log N)^{d-1}}{N}\right). \tag{3.2.19}$$

The theory of low-discrepancy sequences and QMC methods based on them is rich and remains an area of active research. For the purposes of this thesis I only use the sequences introduced as examples in this section. For low to moderate dimension problems there is not much of a difference between Halton, Faure or Sobol' sequences in terms of performance for general problems. Therefore, in my numerical section I will use interchangeably Faure and Sobol' sequences.

3.3 Tensor Product Quadrature and Sparse Grids

This section closely follows [20]. In the following let $\Gamma = \prod_{i=1}^d \Gamma_i = [-1,1]^d$ and $\rho(\mathbf{y}) = \prod_{i=1}^d \rho_i(y^{(i)})$ with $\rho_i(y^{(i)})$ being a one-dimensional weight function of a variable $y^{(i)}$ in Γ_i.

The quadrature rule that integrates f over the domain Γ with respect to the weight function $\rho(\mathbf{y})$ is defined by the nodes $\{\mathbf{y}_i\}_{i=1}^N \subset \Gamma$ and weights $\{w_i\}_{i=1}^N \subset \mathbb{R}$:

$$I_\rho^d[f] \approx \sum_{i=1}^N w_i f(\mathbf{y}_i). \tag{3.3.1}$$

3.3.1 One-dimensional Quadrature Formulas

I provide a short review of univariate quadrature formulas for functions $f \in \mathcal{C}_r$ where

$$\mathcal{C}_r = \{f : \Gamma_i \to \mathbb{R} \ \mid \ \left\| \frac{\partial^s f}{\partial y^s} \right\|_\infty < \infty, \ s \leq r\}. \tag{3.3.2}$$

I will focus on formulas that are based on nested sets of points meaning that the grid corresponding to level l rule is a subset of a more refined, level $l+1$, grid. The reason for this will become clear in the section on sparse grids. Moreover, the number of points at level 1 is always set to 1 and $Q_1[f] = 2 \cdot f(0)$.

Example 3.3.1 Trapezoidal Rule

The Newton-Cotes formulas use equidistant nodes with weights obtained from integration of Lagrange polynomials constructed using these nodes. They are known to become numerically unstable for large number of points, i.e., some of the weights become negative, therefore, in practice iterated versions of low degree formulas are most commonly used. An example is iterated trapezoidal rule: the number of points at level l is defined

$$n_l = 2^{l-1} + 1, \quad l \geq 2$$

and the rule is defined as

$$Q_l[f] = 2^{2-l} \left(\frac{1}{2} f(-1) + \sum_{i=2}^{n_l - 1} \cdot f((i-1) \cdot 2^{2-l} - 1) + \frac{1}{2} f(1) \right).$$

The error bound is well known to be $\mathcal{O}(2^{-2l})$ for the functions $f \in \mathcal{C}_2$ [15].

◇

Example 3.3.2 Clenshaw-Curtis Formulas

The Clenshaw-Curtis formulas [13] use nodes given by the extreme points of the Chebyshev polynomials:

$$y_{l,i} = -\cos \frac{\pi \cdot (i-1)}{n_l - 1}$$

with $n_l = 2^{l-1} + 1$ for $l \geq 2$ and weights given by

$$w_{l,1} = w_{l,n_l} = \frac{1}{n_l(n_l - 2)}$$

$$w_{l,i} = \frac{2}{n_l - 1} \left(1 + 2 \cdot \sum_{j=1}^{(n_l-3)/2} \frac{1}{1 - 4j^2} \cdot \cos\left(\frac{2\pi(i-1)j}{n_l - 1}\right) + \frac{\cos(\pi(j-1))}{n_l(n_l - 2)} \right)$$

for $2 \leq i \leq n_l - 1$.

The polynomial degree of exactness is $n_l - 1$ and the error bound for $f \in \mathcal{C}_r$ is $\mathcal{O}(2^{-lr})$ [15].

◇

Example 3.3.3 Gauss and Gauss-Patterson Formulas

Gauss formulas have the maximum possible polynomial degree of exactness of $2n - 1$ for n points ([77, pp. 172-175]). For the case of the unit weight function the nodes are the zeroes of the Legendre polynomials and the weights are computed by integrating the associated Lagrange polynomials.

In general Gauss-Legendre formulas are not nested. Kronrod [44] extended an n-point Gauss quadrature formula by $n + 1$ points in a way to achieve maximal polynomial degree of exactness of teh resulting $(2n+1)$-point formula. Patterson [62] iterated Kronrod's scheme recursively and obtained a sequence of nested quadrature formulas with maximal degree of exactness. However, Patterson extensions do not exist for all Gauss-Legendre formulas. Gauss-Patterson rules are only available for orders of 1, 3, 7, 15, 31, 63, 127, 255 or 511.

If Q_l for $l \geq 3$ is the $(l-2)$-nd Patterson extension, then the error is $\mathcal{O}(2^{-lr})$ [20].

◇

3.3.2 Tensor Product Quadrature

For $i = 1, \ldots, d$ consider a sequence of one-dimensional quadrature formulas $\{Q_l^{(i)}\}_{l=1}^{\infty}$ defined on $\Gamma_i = [-1, 1]$, where subscript l denotes the level of the rule. When applied to a univariate function f the rule $Q_l^{(i)}$ has a form:

$$Q_l^{(i)}[f] = \sum_{j=1}^{n_l} w_{l,j} f(y_{l,j}^{(i)}). \tag{3.3.3}$$

Here n_l denote the number of nodes of the rule $Q_l^{(i)}$ so that $n_l < n_{l+1}$. Suppose that the rules are chosen so that $Q_l^{(i)}[P] = I_{\rho_i}^1[P]$ holds exactly for all polynomials P of degree at most m_l on Γ_i. We denote the nodes and the weights of the rule $Q_l^{(i)}$ by $\{y_{l,j}^{(i)}\}_{j=1}^{n_l}$ and $\{w_{l,j}\}_{j=1}^{n_l}$ respectively.

Applying the quadrature rules $Q_l^{(i)}$ to evaluate the integral $I_\rho^d[f]$ coordinate-wise we obtain the following approximation

$$I_\rho^d[f] \approx \sum_{j_1=1}^{n_l} \cdots \sum_{j_d=1}^{n_l} w_{l,j_1} \cdot \ldots \cdot w_{l,j_d} \cdot f(y_{l,j_1}^{(1)}, \ldots, y_{l,j_d}^{(d)}) = \bigotimes_{i=1}^{d} Q_l^{(i)}[f]. \tag{3.3.4}$$

This quadrature is typically referred to as a *tensor product* quadrature. Note that the total number of nodes here, and therefore the number of evaluations of f, grows exponentially with d. This phenomenon is referred to as a *curse of dimensionality.*

An improvement over the tensor product rule comes in the form of *Smolyak quadrature.*

3.3.3 Smolyak quadrature

For the construction of Smolyak quadrature [76] consider $f \in \mathcal{C}_r^d$ where

$$\mathcal{C}_r^d = \{f : \Gamma \to \mathbb{R} \ \mid \ \left\| \frac{\partial^{|\boldsymbol{\alpha}|_1} f}{\partial y_1^{\alpha_1} \ldots \partial y_d^{\alpha_d}} \right\|_\infty < \infty, \ \alpha_i \leq r\} \tag{3.3.5}$$

with $|\boldsymbol{\alpha}|_1 = \alpha_1 + \cdots + \alpha_d$, that is f has bounded mixed derivatives of order r. The integer r is called the *regularity* of the function $f \in \mathcal{C}_r^d$.

Let $\{Q_l^{(i)}\}_{l=1}^\infty$ be a sequence of univariate quadrature defined on the intervals Γ_i, $i = 1, \ldots, d$. For each dimension i define the *difference quadrature rule* in Γ_i by

$$\Delta_1^{(i)} = Q_1^{(i)}, \ \ \Delta_{l+1}^{(i)} = Q_{l+1}^{(i)} - Q_l^{(i)} \ \text{ for } \ l = 1, 2, \ldots. \tag{3.3.6}$$

Definition 3.3.4 *The d-dimensional Smolyak quadrature rule of level l in Γ is defined by*

$$Q^{SM}_{l,d}[f] := \sum_{|\boldsymbol{\alpha}| \leq l+d-1} \bigotimes_{i=1}^{d} \Delta^{(i)}_{\alpha_i}[f], \tag{3.3.7}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ is a multi-index denoting the level of difference quadrature rule in each dimension and $|\boldsymbol{\alpha}| = \sum_{i=1}^{d} \alpha_i$.

Using the difference quadrature rules the full tensor product rule can be rewritten as

$$\bigotimes_{i=1}^{d} Q^{(i)}_{l}[f] = \sum_{\max \alpha_i \leq l} \bigotimes_{i=1}^{d} \Delta^{(i)}_{\alpha_i}[f]. \tag{3.3.8}$$

Smolyak's formula can be written in terms of $Q^{(i)}_{\alpha_i}$ instead of $\Delta^{(i)}_{\alpha_i}$ using a so-called *combination technique* [20]:

$$Q^{SM}_{l,d}[f] = \sum_{l \leq |\boldsymbol{\alpha}| \leq l+d-1} (-1)^{l+d-1-|\boldsymbol{\alpha}|} \binom{d-1}{|\boldsymbol{\alpha}| - l} \bigotimes_{i=1}^{d} Q^{(i)}_{\alpha_i}[f]. \tag{3.3.9}$$

It is also important to notice that the Smolyak rule is dimension recursive [81]:

$$Q^{SM}_{l,d}[f] = \sum_{k=1}^{l-1} (Q^{SM}_{l-k,d-1} \bigotimes \Delta^{(d)}_{k})[f]. \tag{3.3.10}$$

3.3.4 Sparse grids

The points of a multivariate Smolyak formula form a so-called *sparse grid* [83].

Denote the set of nodes required to evaluate $Q^{(i)}_{l}$ by $\mathcal{N}^{(i)}_{l}$, that is

$$\mathcal{N}^{(i)}_{l} = \{y^{(i))}_{l,j}\}_{j=1}^{n_l} \subset \Gamma_i. \tag{3.3.11}$$

If the one-dimensional quadrature formulas $Q^{(i)}_{l}$ are nested, i.e., $\mathcal{N}^{(i)}_{l} \subset \mathcal{N}^{(i)}_{l+1}$, then define the one-dimensional difference grid by

$$\mathcal{D}^{(i)}_{l} = \mathcal{N}^{(i)}_{l} \backslash \mathcal{N}^{(i)}_{l-1} \text{ for } l = 1, 2, \ldots \tag{3.3.12}$$

with $\mathcal{N}_0^{(i)} = \emptyset$. The number of elements in $\mathcal{D}_l^{(i)}$ is $p_l := n_l - n_{l-1}$ with $n_0 = 0$. In the non-nested case, set

$$\mathcal{D}_l^{(i)} = \mathcal{N}_l^{(i)} \text{ for } l = 1, 2, \dots \tag{3.3.13}$$

and $p_l := n_l$.

Now, the points of the multivariate Smolyak quadrature are given by the union over pairwise disjoint grids $\mathcal{D}_{\alpha_1}^{(1)} \times \cdots \times \mathcal{D}_{\alpha_d}^{(d)}$:

$$\mathcal{N}_l^{SM} = \bigcup_{|\boldsymbol{\alpha}| \le l+d-1} \bigotimes_{i=1}^{d} \mathcal{D}_{\alpha_i}^{(i)}. \tag{3.3.14}$$

The number of points in a sparse grid can be determined from

$$n_l^{SM} = \sum_{|\boldsymbol{\alpha}| \le l+d-1} p_{\alpha_1} \cdot \ldots \cdot p_{\alpha_d}. \tag{3.3.15}$$

If the number of points in the one-dimensional rule is $\mathcal{O}(2^l)$, then the sparse grid constructed using this rule has $\mathcal{O}(2^l \cdot l^{d-1})$ points. This is in contrast to the full tensor product rule that involves $\mathcal{O}(2^{ld})$ points. The order of n_l^{SM} is the same in both the nested and the non-nested case, however, the constants in the non-nested case are considerably larger, thus, in general the nested sequences are preferrable.

3.3.5 Weights

The Smolyak quadrature formula (3.3.7) can be written as [20]

$$Q_{l,d}^{SM}[f] = \sum_{|\boldsymbol{\alpha}| \le l+d-1} \sum_{j_1=1}^{p_{\alpha_1}} \cdots \sum_{j_d=1}^{p_{\alpha_d}} w_{\boldsymbol{\alpha},\mathbf{j}} f(\mathbf{y}_{\boldsymbol{\alpha},\mathbf{j}}) \tag{3.3.16}$$

where $\mathbf{y}_{\boldsymbol{\alpha},\mathbf{j}} = (y_{\alpha_1,j_1}^{(i)}, \dots, y_{\alpha_d,j_d}^{(i)})$. In the nested case the weights are given by

$$w_{\boldsymbol{\alpha},\mathbf{j}} = \sum_{|\boldsymbol{\alpha}+\boldsymbol{\beta}| \le l+2d-1} v_{(\alpha_1+\beta_1),j_1} \cdot \cdots \cdot v_{(\alpha_d+\beta_d),j_d} \tag{3.3.17}$$

where $\boldsymbol{\beta} \in \mathbb{N}^d$ and

$$v_{(\alpha+\beta),j} = \begin{cases} w_{\alpha,j}, & \text{if } \beta = 1, \\ w_{(\alpha+\beta-1),r} - w_{(\alpha+\beta-2),s}, & \text{if } \beta > 1,\ y_{\alpha,j} = y_{(\alpha+\beta-1),r} = y_{(\alpha+\beta-2),s} \end{cases}$$

In the non-nested case $w_{\boldsymbol{\alpha},\mathbf{j}}$ can be found using

$$w_{\boldsymbol{\alpha},\mathbf{j}} = w_{\alpha_1,j_1} \cdot \ldots \cdot w_{\alpha_d,j_d}. \tag{3.3.18}$$

In both cases weights can be precomputed.

Smolyak quadrature formula can contain negative weights even if the underlying one-dimensional quadrature formulas are positive. The absolute values of the weights, however, remain relatively small. It can be shown [59] that

$$\sum_{|\boldsymbol{\alpha}|\leq l+d-1} \sum_{j_1=1}^{p_{\alpha_1}} \cdots \sum_{j_d=1}^{p_{\alpha_d}} |w_{\boldsymbol{\alpha},\mathbf{j}}| = \mathcal{O}((\log(n_l^{SM}))^{d-1}) \tag{3.3.19}$$

In order to avoid numerical cancellation due to the existence of negative weights it is preferrable to perform summation coordinate-wise [20]

$$\mathcal{Q}_l^d[f] = \sum_{\alpha_1=1}^{l} \sum_{\alpha_2=1}^{l-\alpha_1} \cdots \sum_{\alpha_d=1}^{l-\alpha_1-\ldots-\alpha_d} \sum_{j_1=1}^{p_{\alpha_1}} \cdots \sum_{j_d=1}^{p_{\alpha_d}} w_{\boldsymbol{\alpha},\mathbf{j}} f(\mathbf{y}_{\boldsymbol{\alpha},\mathbf{j}}) \tag{3.3.20}$$

as opposed to summing-up with increasing l:

$$\mathcal{Q}_l^d[f] = \sum_{m=1}^{l} \sum_{|\boldsymbol{\alpha}|=m+d-1} \sum_{j_1=1}^{p_{\alpha_1}} \cdots \sum_{j_d=1}^{p_{\alpha_d}} w_{\boldsymbol{\alpha},\mathbf{j}} f(\mathbf{y}_{\boldsymbol{\alpha},\mathbf{j}}). \tag{3.3.21}$$

3.3.6 Generalized sparse grids

The Smolyak quadrature rule can be generalized. Instead of using the set of multi-indices $\boldsymbol{\alpha}$ such that $|\boldsymbol{\alpha}| \leq l + d - 1$ as in the Smolyak rule we could construct a different set of multi-indices $\mathcal{M}$ and define a generalized sparse-grid quadrature rule

$$Q_{\mathcal{M},d}^{GS}[f] = \sum_{\boldsymbol{\alpha}\in\mathcal{M}} \bigotimes_{i=1}^{d} \Delta_{\alpha_i}^{(i)}[f]. \tag{3.3.22}$$

In order to be able to rewrite (3.3.22) in terms of the original 1D quadrature operators as we did in (3.3.9), we need to restrict possible sets $\mathcal{M}$ to *admissible* ones.

Definition 3.3.5 *The multi-index set $\mathcal{M} \in \mathbb{N}^d$ is admissible if for all $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d) \in \mathcal{M}$ the following holds:*

$$\boldsymbol{\alpha} - \mathbf{e}_j \in \mathcal{M} \;\; \textit{for} \;\; 1 \leq j \leq d, \; \alpha_j > 1 \tag{3.3.23}$$

where $\mathbf{e}_j$ denotes j-th unit vector.

To apply the combination method to the generalized sparse grid quadrature rule (3.3.22) we introduce a characteristic function $\chi_{\mathcal{M}}$ of $\mathcal{M}$:

$$\chi_{\mathcal{M}}(\boldsymbol{\alpha}) = \begin{cases} 1, & \text{if } \boldsymbol{\alpha} \in \mathcal{M} \\ 0, & \text{else.} \end{cases} \tag{3.3.24}$$

Then the combination method leads to

$$Q^{GS}_{\mathcal{M},d}[f] = \sum_{\boldsymbol{\alpha} \in \mathcal{M}} \left(\sum_{\boldsymbol{\beta} \in \{0,1\}^d} (-1)^{|\boldsymbol{\beta}|} \chi_{\mathcal{M}}(\boldsymbol{\alpha} + \boldsymbol{\beta}) \right) \bigotimes_{i=1}^{d} Q^{(i)}_{\alpha_i}[f]. \tag{3.3.25}$$

By defining

$$\vartheta(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\beta} \in \{0,1\}^d} (-1)^{|\boldsymbol{\beta}|} \chi_{\mathcal{M}}(\boldsymbol{\alpha} + \boldsymbol{\beta}) \tag{3.3.26}$$

we can determine the set of points required to evaluate $Q^{GS}_{\mathcal{M},d}$ (i.e., the sparse grid associated with $\mathcal{M}$) from

$$\mathcal{N}^{GS}_{\mathcal{M}} = \bigcup_{\{\boldsymbol{\alpha} \in \mathcal{M} \,:\, \vartheta(\boldsymbol{\alpha}) \neq 0\}} \bigoplus_{i=1}^{d} \mathcal{N}^{(i)}_{\alpha_i}. \tag{3.3.27}$$

The choice of admissible set $\mathcal{M}$ affects the computing cost of the integral. In general, the number of points in $\mathcal{N}^{GS}_{\mathcal{M}}$ is considerably smaller than in the full tensor product grid, as we have already seen in the case of classical Smolyak grid ($\mathcal{M} = \{\boldsymbol{\alpha} \;\mid\; |\boldsymbol{\alpha}| \leq l + d - 1\}$). The cost can be reduced further if the underlying 1D quadrature rules are nested.

The index set $\mathcal{M}$ can be obtained a-priori - if the behavior of the integrand is known - or by adaptive procedure. An algorithm that builds up an index set $\mathcal{M}$ by adaptively selecting most important dimensions has been developed by Gerstner and Griebel in [21].

3.3.7 Error bounds

First, consider polynomial degree of exactness of Smolyak quadrature rule. Let $\mathcal{P}_l^{(i)}$ be the space of one-dimensional polynomials of degree at most l corresponding to dimension i. Define the space of multivariate polynomials as

$$\mathcal{P}_l^d := \left\{ \bigoplus_{i=1}^{d} \mathcal{P}_{\alpha_i}^{(i)}, \;\; |\boldsymbol{\alpha}| = l + d - 1 \right\}. \tag{3.3.28}$$

It can be shown that if $Q_l^{(i)}$ is exact for $\mathcal{P}_l^{(i)}$, then $Q_{l,d}^{SM}$ is exact for $\mathcal{P}_l^d$ (see [58]).

To formulate an error bound for Smolyak's formula first consider error bounds for one-dimensional quadrature rules. Suppose that for a univariate function f with bounded derivatives up to the order r the quadrature error $E_l^1[f] := |I_{\rho_i}^1[f] - Q_l^{(i)}[f]|$ satisfies

$$E_l^1[f] = \mathcal{O}((n_l)^{-r}). \tag{3.3.29}$$

This bound holds for example for interpolatory quadrature formulas with positive weights, such as Clenshaw-Curtis, Gauss-Patterson and Gauss-Legendre formulas [20]. Using one of such formulas as a basis for the Smolyak construction one can show that for $f \in \mathcal{C}_r^d$ and $n_l = \mathcal{O}(2^l)$ the quadrature error is on the order of (see [81])

$$E_l^d[f] = \mathcal{O}(2^{-lr} \cdot l^{(d-1)(r+1)}). \tag{3.3.30}$$

3.4 Hierarchical Interpolation and Locally Adaptive Sparse Grids

Quadrature methods described earlier are based on global Lagrange polynomials and require high regularity of the function of interest f in order to achieve high accuracy. For efficient integration of functions with locally sharp behavior (e.g. steep gradients) I consider methods using hierarchical bases. I start with the piecewise linear interpolation of the function of interest in hierarchical setting. Given the interpolation formula for the function of interest the quadrature formula can be obtained simply by integrating the basis functions. Therefore, in the following I focus on interpolation problem keeping in mind its relation to quadrature.

3.4.1 One-dimensional hierarchical interpolation

Consider an interpolation problem for a function $f : [-1, 1] \to \mathbb{R}$: Given a set of points $(y_{k,j}, f(y_{k,j}))$, where the points $\mathcal{Y}_k = \{y_{k,j}\}_{j=0}^{n_k-1}$ partition the interval $[-1, 1]$, i.e., $y_{0,0} = 0$ for $k = 0$ and

$$-1 = y_{k,0} < y_{k,1} < \cdots < y_{k,n_k-1} = 1, \quad \text{for} \quad k \geq 1 \tag{3.4.1}$$

we seek to construct a linear interpolant $\mathcal{I}_k[f]$ that satisfies $\mathcal{I}_k[f](y_{k,j}) = f(y_{k,j})$ for $j = 1, \ldots, n_k - 1$. The index k stands for the level of refinement of the interval $[-1, 1]$, $k \geq 0$, and n_k is a number of points at level k. Assume that $\{n_k\}_{k=0}^{\infty}$ is an increasing sequence.

I will consider the grid with equidistant nodes defined as follows

$$n_k = \begin{cases} 1, & k = 0 \\ 2^k + 1, & k \geq 1 \end{cases} \tag{3.4.2}$$

$$y_{k,j} = \begin{cases} 0, & j = 0, k = 0 \\ jh_k - 1, & j = 0, \ldots, n_k - 1, k \geq 1 \end{cases} \tag{3.4.3}$$

with $h_k = 1/2^{k-1}$ for $k \geq 1$ denoting the mesh size.

Next I define the usual nodal basis functions $\phi_{k,j}(y)$. They are derived from the “mother” hat function

$$\phi(y) = \max\{0, 1 - |y|\} \tag{3.4.4}$$

using dilation and translation:

$$\phi_{k,j}(y) = \phi\left(\frac{y - y_{k,j}}{h_k}\right) \quad \text{for} \quad k \geq 1. \tag{3.4.5}$$

The resulting basis functions $\phi_{k,j}(y)$ are centered at $y_{k,j}$ and have local support $[y_{k,j} - h_k, y_{k,j} + h_k]$. Set $\phi_{0,0}(y) \equiv 1$.

Using functions $\phi_{k,j}(y)$ the linear interpolant $\mathcal{I}_k[f]$ can be constructed as follows

$$\mathcal{I}_k[f](y) = \sum_{j=0}^{n_k-1} f(y_{k,j})\phi_{k,j}(y). \tag{3.4.6}$$

Now consider the sequence of finite-dimensional subspaces $\{V_k\}_{k=0}^{\infty}$ spanned by the nodal basis functions $\{\phi_{k,j}(y)\}_{j=0}^{2^k}$ corresponding to level k:

$$V_k \coloneqq \text{span}\{\phi_{k,j} \mid j = 0, \ldots, 2^k\} \text{ for } k = 0, 1, \ldots. \tag{3.4.7}$$

Each of the subspaces V_k is a standard finite-element subspace of continuous linear functions on $[-1, 1]$, and $\{\phi_{k,j}(y)\}_{j=0}^{2^k}$ is a standard nodal basis for V_k.

The sequence of subspaces $\{V_k\}$ is nested, that is $V_0 \subset V_1 \subset \cdots \subset V_k \subset V_{k+1}$, and is dense in $L^2([-1, 1])$, i.e., $\cup_{k=0}^{\infty} V_k = L^2([-1, 1])$.

Alternative to the nodal basis for V_k is the *hierarchical* basis, which is constructed in the following manner.

First, define the *interpolation difference operator* $\Delta_k[f]$ as

$$\Delta_k[f] = \mathcal{I}_k[f] - \mathcal{I}_{k-1}[f] \text{ for } k = 0, 1, \ldots \tag{3.4.8}$$

with $\mathcal{I}_{-1}[f] \equiv 0$ and $\mathcal{I}_k[f] \coloneqq \mathcal{I}_k[f](y)$. Recall that $\mathcal{I}_k[f] = \sum_{y_{k,j} \in \mathcal{Y}_k} f(y_{k,j})\phi_{k,j}(y)$. Using the fact that $V_{k-1} \subset V_k$ we have that

$$\mathcal{I}_{k-1}[f] = \mathcal{I}_k[\mathcal{I}_{k-1}[f]]. \tag{3.4.9}$$

Therefore,

$$\Delta_k[f] = \sum_{y_{k,j} \in Y_k} (f(y_{k,j}) - \mathcal{I}_{k-1}[f](y_{k,j}))\phi_{k,j}(y). \tag{3.4.10}$$

Observe also that $f(y_{k-1,j}) - \mathcal{I}_{k-1}[f](y_{k-1,j}) = 0$ for all $y_{k-1,j} \in \mathcal{Y}_{k-1}$. Since $\mathcal{Y}_{k-1} \subset \mathcal{Y}_k$ we can write

$$\Delta_k[f] = \sum_{y_{k,j} \in \Delta\mathcal{Y}_k} (f(y_{k,j}) - \mathcal{I}_{k-1}[f](y_{k,j}))\phi_{k,j}(y), \tag{3.4.11}$$

where $\Delta\mathcal{Y}_k = \mathcal{Y}_k \backslash \mathcal{Y}_{k-1}$. Here $\Delta\mathcal{Y}_k$ consists of the points $y_{k,j}$ with indices j from the set

$$B_k = \begin{cases} \{0\} & \text{for } k = 0 \\ \{0, 2\} & \text{for } k = 1 \\ \{j \in \mathbb{N} \mid j = 1, 3, 5, \ldots, 2^k - 1\} & \text{for } k = 2, 3, \ldots \end{cases} \tag{3.4.12}$$

Thus, $\Delta\mathcal{Y}_0 = \mathcal{Y}_0 = \{0\}$, $\Delta\mathcal{Y}_1 = \{-1, 1\}$, $\Delta\mathcal{Y}_2 = \{-0.5, 0.5\}$ and so on.

Coefficients of the basis functions in (3.4.11) are usually referred to as *hierarchical surpluses*. I denote them by $s_{k,j}$:

$$s_{k,j} \coloneqq f(y_{k,j}) - \mathcal{I}_{k-1}[f](y_{k,j}) \ \text{ for } \ j \in B_k. \tag{3.4.13}$$

Thus,

$$\Delta_k[f] = \sum_{j \in B_k} s_{k,j} \phi_{k,j}(y) \ \text{ for } \ k = 0, 1, \ldots. \tag{3.4.14}$$

With the incremental interpolation operators $\Delta_k[f]$ we can decompose interpolant $\mathcal{I}_k[f]$ at any level k in the form

$$\mathcal{I}_k[f] = \mathcal{I}_{k-1}[f] + \Delta_k[f] = \cdots = \sum_{l=0}^{k} \Delta_l[f]. \tag{3.4.15}$$

For each $\Delta_k[f]$ the set of functions $\{\phi_{k,j}\}_{j \in B_k}$ generate a so-called *hierarchical subspace*:

$$W_k \coloneqq \operatorname{span}\{\phi_{k,j}(y) \mid j \in B_k\} \ \text{ for } \ k = 0, 1, \ldots. \tag{3.4.16}$$

Due to the nestedness of subspaces V_k we have that $V_k = V_{k-1} \oplus W_k$ and $W_k = V_k \backslash \oplus_{l=0}^{k-1} V_l$ for $k = 1, 2, \ldots$. We also have a hierarchical subspace splitting of V_k given by

$$V_k = W_0 \oplus W_1 \oplus \cdots \oplus W_k \ \text{ for } \ k = 0, 1, \ldots. \tag{3.4.17}$$

The *hierarchical basis* for V_k is given by

$$\cup_{l=0}^{k} \{\phi_{l,j}(y)\}_{j \in B_l}. \tag{3.4.18}$$

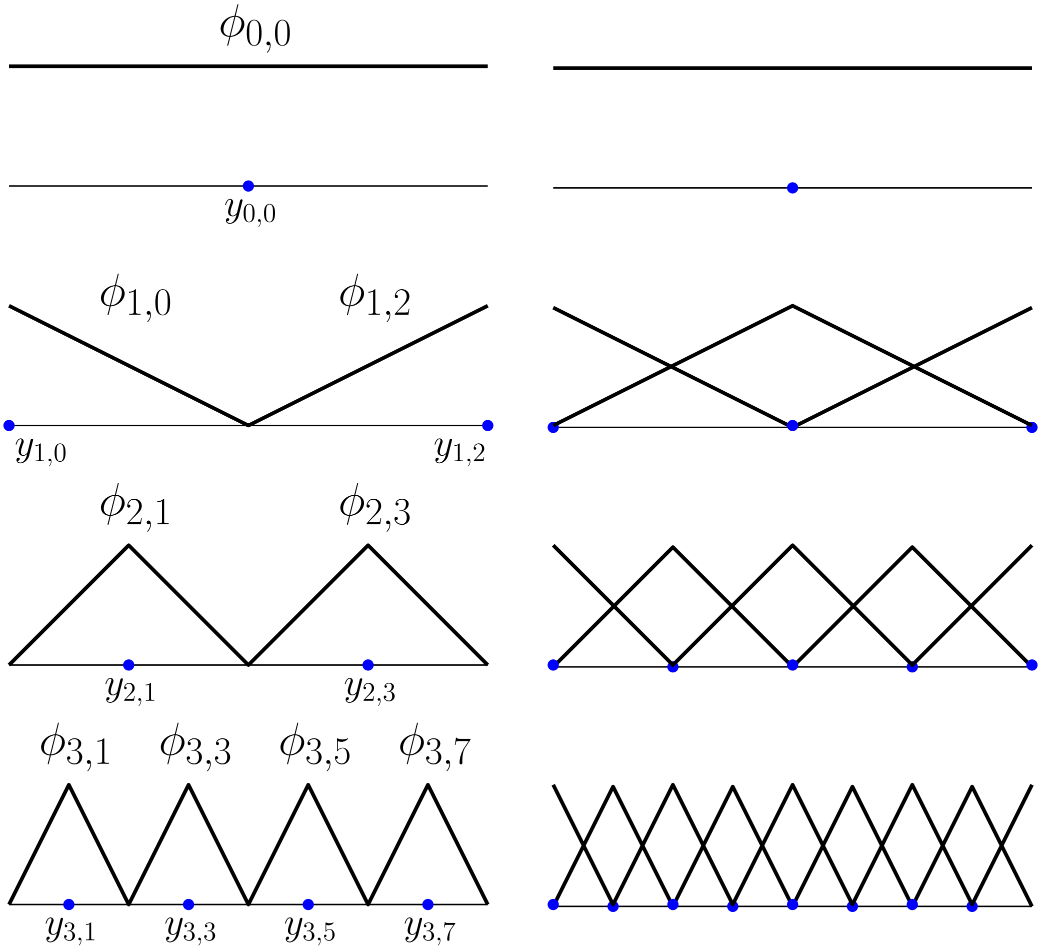


Figure 3.2: Hierarchical basis functions (left) vs. nodal basis functions (right). The functions on the left represent subspaces W_0, W_1, W_2, W_3, while those on the right correspond to V_0, V_1, V_2, V_3.

It can be easily verified that for each k the subspaces spanned by the nodal and the hierarchical bases are the same. The hierarchical functions for the first few levels and their nodal counterparts are depicted in Figure 3.2.

Recall that the nodal basis $\{\phi_{k,j}\}_{j=0}^{2^k}$ posesses the delta property, that is, $\phi_{k,j}(y_{k,i}) = \delta_{j,i}$ for $i, j \in \{0, \dots, 2^k\}$. The hierarchical basis (3.4.18) for V_k posesses only a partial delta property - the basis functions corresponding to a specific level l posses delta property with respect to their own level and coarser levels, but not with respect to finer levels. That is, for $l = 0, 1, \dots, k$ and $j \in B_l$ we have

$$\begin{aligned} &\text{for } 0 \le l' < l, && \phi_{l,j}(y_{l',i}) = 0 && \text{for all } i \in B_{l'}, \\ &\text{for } l' = l, && \phi_{l,j}(y_{l',i}) = \delta_{j,i} && \text{for all } i \in B_{l'}, \\ &\text{for } l < l' \le k && \phi_{l,j}(y_{l',i}) \neq 0 && \text{for all } i \in B_{l'}. \end{aligned} \tag{3.4.19}$$

The delta property of the nodal basis implies that interpolation coefficients are just the values of the function at the nodes as seen in (3.4.6). For the hierarchical basis the coefficients are hierarchical surpluses. Due to the recursive form of $\mathcal{I}_k(f)$ in (3.4.15) interpolant at each level k can be constructed using the procedure in Algorithm 1.

Algorithm 1: Computing $\mathcal{I}_k[f]$

```
$\mathcal{I}_{-1}[f](y) \equiv 0$;
for $l \leftarrow 0$ to $k$ do
    $\Delta_l[f](y) = 0$;
    for $j \in B_l$ do
        $s_{l,j} = f(y_{l,j}) - \mathcal{I}_{l-1}[f](y_{l,j})$;
        $\Delta_l[f](y) = \Delta_l[f](y) + s_{l,j}\phi_{l,j}(y)$;
    end
    $\mathcal{I}_l[f](y) = \mathcal{I}_{l-1}[f](y) + \Delta_l[f](y)$;
end
```

Hierarchical surpluses $s_{k,j}$ can in fact be computed only using function values at the points $y_{k,j} - h_k$, $y_{k,j}$ and $y_{k,j} + h_k$. This is due to the fact that the supports of the basis functions spanning W_k are mutually disjoint and do not contain coarse grid points $y_{l,j}$ for $l < k$. One can write the surplus $s_{k,j}$ as follows:

$$s_{k,j} = f(y_{k,j}) - \frac{1}{2}(f(y_{k,j} - h_k) + f(y_{k,j} + h_k)). \tag{3.4.20}$$

Adopting multigrid stencil notation (see for example [31]) the above expression can be rewritten as [8]

$$s_{k,j} = [\ -\frac{1}{2} \quad 1 \quad -\frac{1}{2}\ ]_{y_{k,j},k} f \tag{3.4.21}$$

3.4.2 Multi-dimensional hierarchical interpolation

I now generalize the construction of Section 3.4.1 to the problem of interpolating a function $f(\mathbf{y})$ defined over a d-dimensional hypercube $\Gamma = \bigotimes_{i=1}^{d} \Gamma_i = [-1,1]^d$. The one-dimensional hierarchical basis (3.4.18) can be generalized to the d-dimensional basis using tensor product construction. That is for each point $\mathbf{y}_{\mathbf{k},\mathbf{j}} = (y^{(1)}_{k_1,j_1}, \ldots, y^{(d)}_{k_d,j_d})$ the associated d-variate basis function $\phi_{\mathbf{k},\mathbf{j}}(\mathbf{y})$ is defined as

$$\phi_{\mathbf{k},\mathbf{j}}(\mathbf{y}) = \prod_{i=1}^{d} \phi_{k_i,j_i}(y^{(i)}), \tag{3.4.22}$$

where $\phi_{k_i,j_i}(y^{(i)})$ is a one-dimensional hat function associated with the point $y^{(i)}_{k_i,j_i} \in \Gamma_i$, $j_i = 0, \ldots, n_{k_i} - 1$, $i = 1, \ldots, d$. Here $\mathbf{k} = (k_1, \ldots, k_d)$ is a multi-index denoting level of refinement along each dimension i. The examples of two-dimensional basis function are depicted in Figure 3.3.

The collection of d-dimensional functions $\{\phi_{\mathbf{k},\mathbf{j}}(\mathbf{y})\}$ for $\mathbf{0} \leq \mathbf{j} \leq 2^{\mathbf{k}}$ form a basis of a discrete space

$$V_{\mathbf{k}} = \operatorname{span}\{\phi_{\mathbf{k},\mathbf{j}}(\mathbf{y}) \ \mid \ \mathbf{0} \leq \mathbf{j} \leq 2^{\mathbf{k}}\}. \tag{3.4.23}$$

Here $\mathbf{0} = (0, \ldots, 0)$ is a multi-index with d zeros and all operations on multi-indices are performed componentwise, i.e., $\boldsymbol{\alpha} \leq \boldsymbol{\beta}$ means $\alpha_i \leq \beta_i$ for all $i = 1, \ldots, d$, and $2^{\boldsymbol{\alpha}} = (2^{\alpha_1}, \ldots, 2^{\alpha_d})$.

As in the one-dimensional case define the d-dimensional incremental subspace $W_{\mathbf{k}}$ by

$$W_{\mathbf{k}} = \bigotimes_{i=1}^{d} W_{k_i} = \operatorname{span}\{\phi_{\mathbf{k},\mathbf{j}}(\mathbf{y}) \ \mid \ \mathbf{j} \in B_{\mathbf{k}}\} \tag{3.4.24}$$

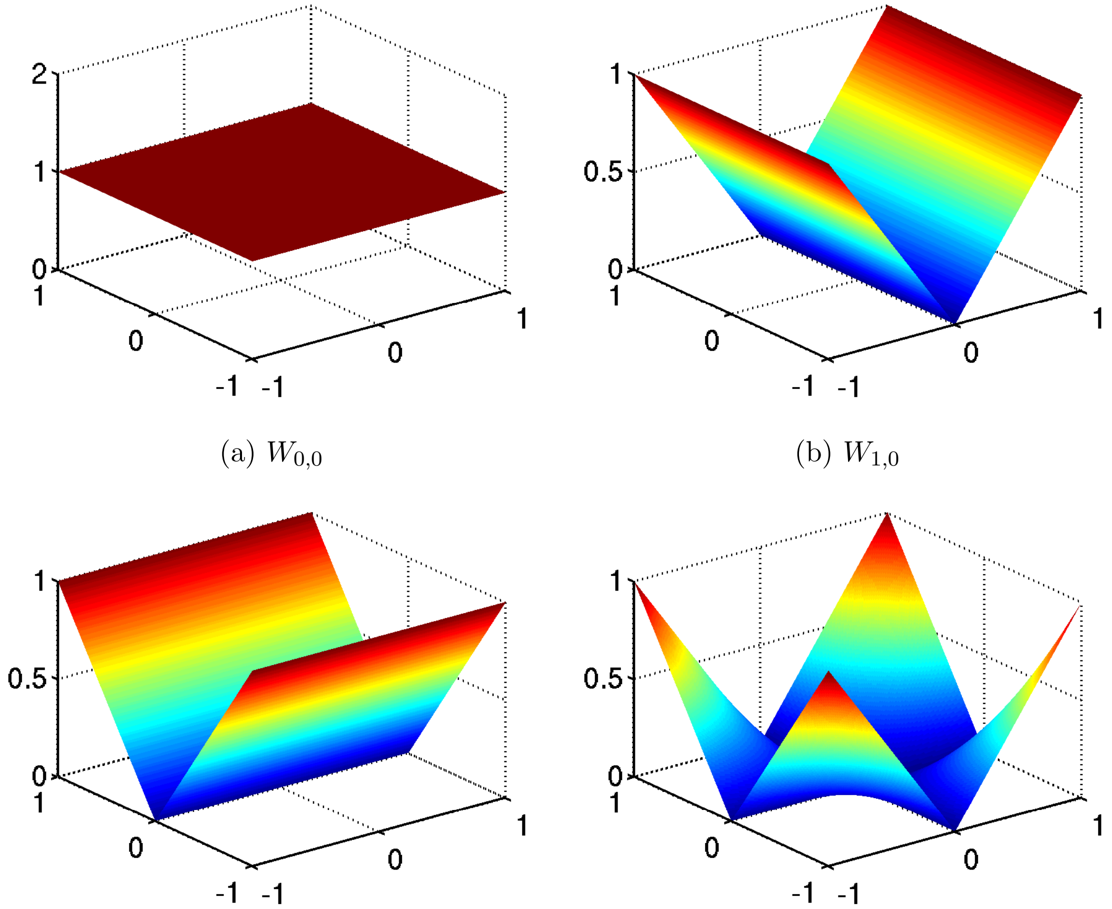
where the multi-index set $B_{\mathbf{k}}$ is given by

$$B_{\mathbf{k}} = \{\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^d_+ \ \mid \ j_i \in B_{l_i} \text{ for } i = 1, \ldots, d\}. \tag{3.4.25}$$

The schematics of subspaces $W_{\mathbf{k}}$ for $d = 2$ is depicted in Figure 3.4.

Similar to the one-dimensional case we obtain

$$V_{\mathbf{k}} = \bigoplus_{\mathbf{l} \leq \mathbf{k}} W_{\mathbf{l}} \tag{3.4.26}$$



(a) $W_{0,0}$

(b) $W_{1,0}$

(c) $W_{0,1}$

(d) $W_{1,1}$

Figure 3.3: 2D hierarchical basis functions.

Note again that the supports of the basis functions $\phi_{\mathbf{k},\mathbf{j}}$ spanning $W_{\mathbf{k}}$ are mutually disjoint. Thus, we obtain a hierarchical basis for $V_{\mathbf{k}}$:

$$\{\phi_{\mathbf{l},\mathbf{j}} \;:\; \mathbf{j} \in B_{\mathbf{l}},\; \mathbf{l} \leq \mathbf{k}\}. \tag{3.4.27}$$

I will mostly be interested in the finite-dimensional subspaces constructed from $W_{\mathbf{l}}$ as follows:

$$V_k^{(\alpha)} = \bigoplus_{\alpha(\mathbf{l}) \leq k} W_{\mathbf{l}} \tag{3.4.28}$$

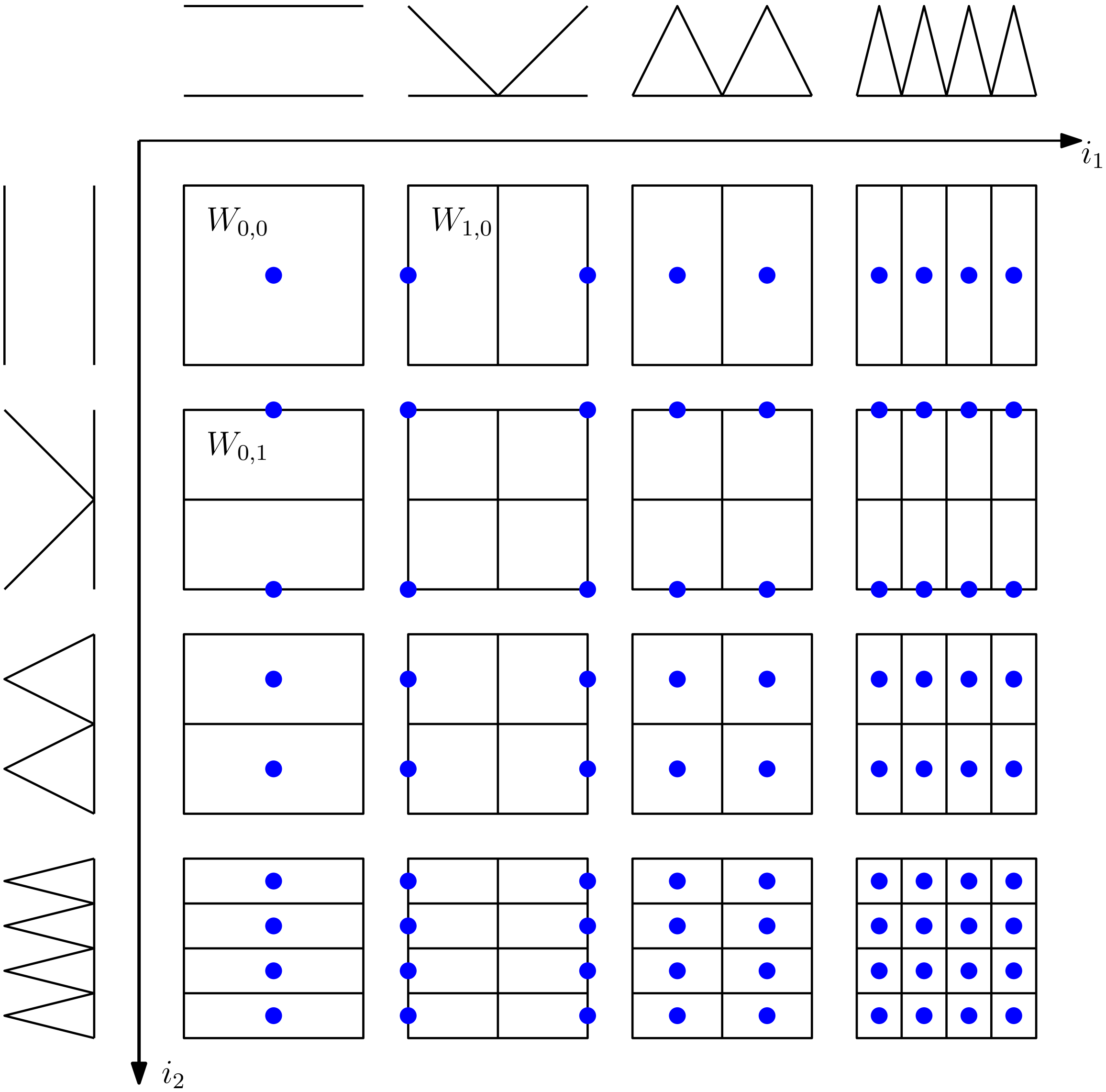


Figure 3.4: Scheme of subspaces $W_{\mathbf{k}}$ for $d = 2$. Squares represent subspaces with their grid points. The basis functions belonging to a particular subspace are given as tensor products of corresponding 1D basis functions of each dimension. The supports of basis functions are indicated by the lines dividing the squares.

The key for constructing such subspaces is to specify the mapping $\alpha(\mathbf{l})$. For example, taking $\alpha(\mathbf{l}) = \max_{i=1,\ldots,d}\{l_i\}$ gives the full tensor product space, which I will denote by $V_k^{(\infty)}$ (see Figure(3.5)). Note, that the sequence of subspaces $V_k^{(\infty)}$ is nested, that is $V_k^{(\infty)} \subset V_{k+1}^{(\infty)}$. Furthermore, this sequence is dense in $L^2(\Gamma)$.

The choice of $\alpha(\mathbf{l}) = |\mathbf{l}|_1 = \sum_{i=1}^d l_i$ leads to a sparse polynomial space corresponding to the sparse grids introduced earlier. I will denote it by $V_k^{(1)}$ (see Figure (3.6)). It is shown in [8] that this choice of $\alpha(\mathbf{l})$ is optimal with respect to L^∞ and L^2 norms.

The multi-linear interpolant corresponding to $V_k^{(\alpha)}$ is given by

$$\mathcal{I}_k^{(\alpha)}[f] = \sum_{\alpha(\mathbf{l}) \le k} \sum_{\mathbf{j} \in B_\mathbf{l}} s_{\mathbf{l},\mathbf{j}} \phi_{\mathbf{l},\mathbf{j}}(\mathbf{y}), \tag{3.4.29}$$

where $s_{\mathbf{l},\mathbf{j}}$ are multi-dimensional hierarchical surpluses. To find the expression for $s_{\mathbf{l},\mathbf{j}}$ consider just as in the 1D case the incremental interpolation operators Δ_{l_i} for $i = 1, \ldots, d$ and their tensor products.

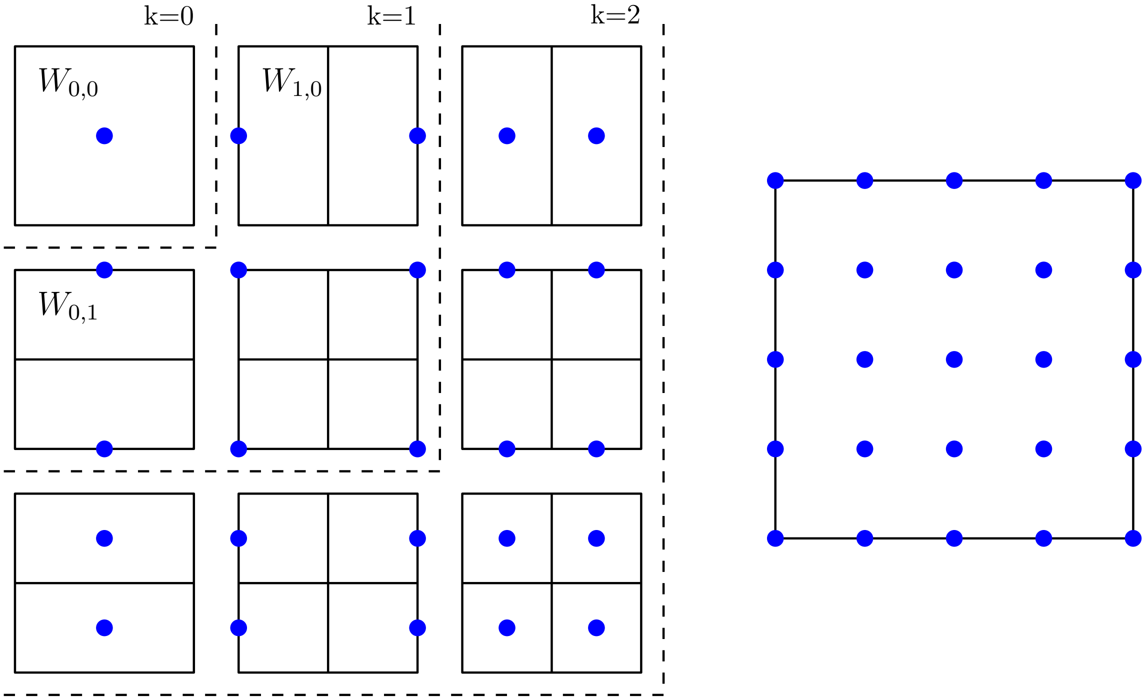


Figure 3.5: The subspaces comprising $V_2^{(\infty)}$ and the corresponding grid (d=2).

Recall that $\Delta_{l_i} \in W_{l_i}$ for $i = 1, \ldots, d$, and, therefore, $\bigotimes_{i=1}^d \Delta_{l_i} \in W_\mathbf{l}$. Further notice that

$$\mathcal{I}_k^{(\alpha)}[f] = \mathcal{I}_{k-1}^{(\alpha)}[f] + \sum_{\alpha(\mathbf{l})=k} \bigotimes_{i=1}^d \Delta_{l_i}[f]. \tag{3.4.30}$$

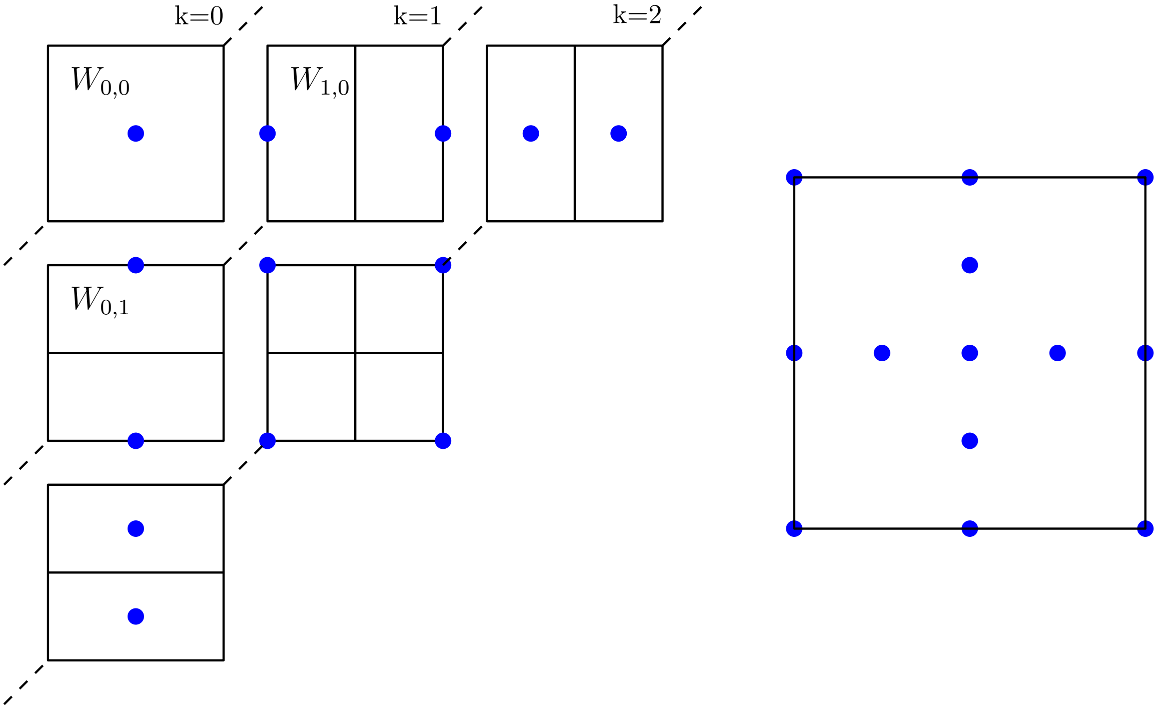


Figure 3.6: The subspaces comprising $V_2^{(1)}$ and the corresponding grid (d=2).

Moreover,

$$\bigotimes_{i=1}^{d} \Delta_{l_i}[f](\mathbf{y}) = (\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d})[f](\mathbf{y}) = \\ = \sum_{\substack{\boldsymbol{\gamma}\in\{0,1\}^d \\ \mathbf{l}-\boldsymbol{\gamma}\geq\mathbf{0}}} (-1)^{|\boldsymbol{\gamma}|_1} (\bigotimes_{i=1}^{d} \mathcal{I}_{l_i-\gamma_i})[f](\mathbf{y}). \tag{3.4.31}$$

Here the tensor product of the one-dimensional interpolation formulas is given by

$$(\bigotimes_{i=1}^{d} \mathcal{I}_{l_i})[f](\mathbf{y}) = \sum_{j_1=0}^{n_{l_1}-1} \cdots \sum_{j_d=0}^{n_{l_d}-1} f(y_{j_1}^{(1)}, \ldots, y_{j_d}^{(d)}) \cdot (\phi_{l_1,j_1} \otimes \cdots \otimes \phi_{l_d,j_d})(\mathbf{y}) \\ = \sum_{\mathbf{j}=0}^{2^{\mathbf{l}}} f(\mathbf{y}_{\mathbf{l},\mathbf{j}})\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y}). \tag{3.4.32}$$

Plugging-in (3.4.32) into (3.4.31):

$$\begin{aligned}\bigotimes_{i=1}^{d}\Delta_{l_i}[f](\mathbf{y}) &= \sum_{\substack{\boldsymbol{\gamma}\in\{0,1\}^d\\ \mathbf{l}-\boldsymbol{\gamma}\geq\mathbf{0}}}(-1)^{|\boldsymbol{\gamma}|_1}\sum_{\mathbf{j}=\mathbf{0}}^{2^{\mathbf{l}-\boldsymbol{\gamma}}} f(\mathbf{y}_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}})\phi_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}}(\mathbf{y})\\ &= \sum_{\mathbf{j}=\mathbf{0}}^{2^{\mathbf{l}}} f(\mathbf{y}_{\mathbf{l},\mathbf{j}})\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y}) - \sum_{\substack{\boldsymbol{\gamma}\in\{0,1\}^d\\ \boldsymbol{\gamma}\neq\mathbf{0}\\ \mathbf{l}-\boldsymbol{\gamma}\geq\mathbf{0}}}(-1)^{|\boldsymbol{\gamma}|_1}\sum_{\mathbf{j}=\mathbf{0}}^{2^{\mathbf{l}-\boldsymbol{\gamma}}} f(\mathbf{y}_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}})\phi_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}}(\mathbf{y}).\end{aligned}$$

Now just as in the 1D case we can expand everything in the basis corresponding to the level $\mathbf{l}$:

$$\bigotimes_{i=1}^{d}\Delta_{l_i}[f](\mathbf{y}) = \sum_{\mathbf{i}=\mathbf{0}}^{2^{\mathbf{l}}}\left(f(\mathbf{y}_{\mathbf{l},\mathbf{i}}) - \sum_{\substack{\boldsymbol{\gamma}\in\{0,1\}^d\\ \boldsymbol{\gamma}\neq\mathbf{0}\\ \mathbf{l}-\boldsymbol{\gamma}\geq\mathbf{0}}}(-1)^{|\boldsymbol{\gamma}|_1}\sum_{\mathbf{j}=\mathbf{0}}^{2^{\mathbf{l}-\boldsymbol{\gamma}}} f(\mathbf{y}_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}})\phi_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}}(\mathbf{y}_{\mathbf{l},\mathbf{i}})\right)\phi_{\mathbf{l},\mathbf{i}}(\mathbf{y}).$$

Finally due to the nestedness of the grids as was the case in the 1D setting we get the cancellation of the coefficients in front of the functions $\phi_{\mathbf{l},\mathbf{i}}(\mathbf{y})$ corresponding to the points that already appeared at the levels previous to $\mathbf{l}$. Hence,

$$\bigotimes_{i=1}^{d}\Delta_{l_i}[f](\mathbf{y}) = \sum_{\mathbf{i}\in B_{\mathbf{l}}}\left(f(\mathbf{y}_{\mathbf{l},\mathbf{i}}) - \sum_{\substack{\boldsymbol{\gamma}\in\{0,1\}^d\\ \boldsymbol{\gamma}\neq\mathbf{0}\\ \mathbf{l}-\boldsymbol{\gamma}\geq\mathbf{0}}}(-1)^{|\boldsymbol{\gamma}|_1}\sum_{\mathbf{j}=\mathbf{0}}^{2^{\mathbf{l}-\boldsymbol{\gamma}}} f(\mathbf{y}_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}})\phi_{\mathbf{l}-\boldsymbol{\gamma},\mathbf{j}}(\mathbf{y}_{\mathbf{l},\mathbf{i}})\right)\phi_{\mathbf{l},\mathbf{i}}(\mathbf{y}).$$

The expression inside the parenthesis is the multi-dimensional hierarchical surplus $s_{\mathbf{l},\mathbf{i}}$. In short, it can be computed as a difference between the function value at the point $\mathbf{y}_{\mathbf{l},\mathbf{i}}$ and the value of the interpolant of f constructed using basis functions from $V_{\mathbf{l}}\ominus W_{\mathbf{l}}$ at the point $\mathbf{y}_{\mathbf{l},\mathbf{i}}$. Since the subspaces $W_{\mathbf{l}}$ were constructed so that their basis functions have mutually disjoint supports and do not contain coarse grid points $\mathbf{y}_{\mathbf{k},\mathbf{i}}$, $\mathbf{k} < \mathbf{l}$, the surpluses can be computed from function values $f(\mathbf{y}_{\mathbf{l},\mathbf{i}})$ in the following way:

$$s_{\mathbf{l},\mathbf{i}} = \left(\prod_{j=1}^{d}[\,-\frac{1}{2}\ \ 1\ \ -\frac{1}{2}\,]_{y_{l_i,i_j},l_j}\right)[f] =: \mathbf{I}_{\mathbf{y}_{\mathbf{l},\mathbf{i}},\mathbf{l}}[f], \qquad (3.4.33)$$

where $\mathbf{I}_{\mathbf{y}_{\mathbf{l},\mathbf{i}},\mathbf{l}}$ denotes a d-dimensional stencil which gives the coefficients for a linear combination of nodal values of the argument f (compare with (3.4.21)).

3.4.3 Error estimation

Here I follow the analysis presented in [8]. The slight modification to the construction presented there is the choice of the basis functions on the first two levels. In [8] the functions under consideration were vanishing on the boundary of the domain, therefore, the only basis functions needed were those that corresponded to the interior nodes. Specifically, the basis functions of the 0-th and 1-st levels in our construction were not considered, instead the only basis function of the first level was the the hat function centered at 0 with support on the whole interval $[-1, 1]$. Thus, with the exception of the basis functions on the 0-th and 1-st levels in the construction presented here the following analysis is still applicable.

In the following assume that $f : \Gamma \to \mathbb{R}$ has (in some sense) bounded weak mixed derivatives

$$D^{\boldsymbol{\alpha}} f := \frac{\partial^{|\boldsymbol{\alpha}|_1} f}{(\partial y^{(1)})^{\alpha_1} \cdots (\partial y^{(d)})^{\alpha_d}}$$

up to a given order. Specifically, for the case of piecewise linear approximations that I used up to this point assume that $f \in X_0^{q,2}(\Gamma)$, where

$$X^{q,r}(\Gamma) := \{f : \Gamma \to \mathbb{R} \ : \ D^{\boldsymbol{\alpha}} f \in L^q(\Gamma), |\boldsymbol{\alpha}|_\infty \leq r\} \tag{3.4.34a}$$

$$X_0^{q,r}(\Gamma) := \{f \in X^{q,r}(\Gamma) \ : \ f\Big|_{\partial\Gamma} = 0\} \tag{3.4.34b}$$

For the functions $f \in X_0^{q,r}$ introduce the following semi-norms:

$$|f|_{\boldsymbol{\alpha},\infty} := \|D^{\boldsymbol{\alpha}} f\|_{L^\infty}, \quad |f|_{\boldsymbol{\alpha},2} := \|D^{\boldsymbol{\alpha}} f\|_{L^2} \tag{3.4.35}$$

The starting point for the analysis is the following integral representation of the surpluses $s_{\mathbf{l},\mathbf{j}}$:

$$s_{\mathbf{l},\mathbf{j}} = \int_\Gamma \psi_{\mathbf{l},\mathbf{j}}(\mathbf{y}) \cdot D^{\mathbf{2}} f(\mathbf{y}) d\mathbf{y}, \tag{3.4.36}$$

where $\psi_{\mathbf{l},\mathbf{j}}(\mathbf{y}) = \prod_{i=1}^{d} \psi_{l_i,j_i}(y^{(i)})$ with $\psi_{l_i,j_i}(y^{(i)}) = -2^{-(l_i+1)} \phi_{l_i,j_i}(y^{(i)})$.

Next, the following equations hold for the norms of the hierarchical basis functions $\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y})$:

$$\|\phi_{\mathbf{l},\mathbf{j}}\|_\infty = 1 \tag{3.4.37a}$$

$$\|\phi_{\mathbf{l},\mathbf{j}}\|_p = \left(\frac{2}{p+1}\right)^{d/p} \cdot 2^{-|\mathbf{l}|_1/p}, \quad p \geq 1 \tag{3.4.37b}$$

The integral representation (3.4.36) combined with the values of the norms (3.4.37) gives the following bounds on $|s_{\mathbf{l},\mathbf{j}}|$.

Lemma 3.4.1 *For any $f \in X_0^{q,r}(\Gamma)$ given in its hierarchical representation $f(\mathbf{y}) = \sum\limits_{\mathbf{l}} \sum\limits_{\mathbf{j}\in B_{\mathbf{l}}} s_{\mathbf{l},\mathbf{j}}\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y})$ the following estimates on $|s_{\mathbf{l},\mathbf{j}}|$ hold:*

$$|s_{\mathbf{l},\mathbf{j}}| \leq 2^{-d} \cdot 2^{-2|\mathbf{l}|_1} \cdot |f|_{\mathbf{2},\infty} \tag{3.4.38a}$$

$$|s_{\mathbf{l},\mathbf{j}}| \leq 2^{-d} \cdot (\frac{2}{3})^{d/2} \cdot 2^{-3/2\cdot|\mathbf{l}|_1} \cdot |f\Big|_{\mathrm{supp}(\phi_{\mathbf{l},\mathbf{j}})}|_{\mathbf{2},2} \tag{3.4.38b}$$

where $\mathrm{supp}(\phi_{\mathbf{l},\mathbf{j}})$ *denotes the support of $\phi_{\mathbf{l},\mathbf{j}}$.*

The next lemma provides bounds on the contributions of incremental subspaces $W_{\mathbf{l}}$, which I denote by $f_{\mathbf{l}}$, i.e., $f_{\mathbf{l}}(\mathbf{y}) = \sum\limits_{\mathbf{j}\in B_{\mathbf{l}}} s_{\mathbf{l},\mathbf{j}}\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y}) \in W_{\mathbf{l}}$.

Lemma 3.4.2 *Let $f \in X_0^{q,r}(\Gamma)$ be given by its hierarchical representation $f(\mathbf{y}) = \sum\limits_{\mathbf{l}} f_{\mathbf{l}}$ with $f_{\mathbf{l}} \in W_{\mathbf{l}}$. Then the contributions $f_{\mathbf{l}}$ can be bounded as follows:*

$$\|f_{\mathbf{l}}\|_\infty \leq 2^{-d}2^{-2|\mathbf{l}|_1}|f|_{\mathbf{2},\infty} \tag{3.4.39a}$$

$$\|f_{\mathbf{l}}\|_2 \leq 3^{-d}2^{-2|\mathbf{l}|_1}|f|_{\mathbf{2},2} \tag{3.4.39b}$$

Having bounds on the contributions of $f_{\mathbf{l}}$ to the infinite expansion $f = \sum\limits_{\mathbf{l}} f_{\mathbf{l}}$ of a function $f \in X_0^{q,2}$ one can estimate the error between f and its interpolants $\mathcal{I}_k^{(\infty)}$ and $\mathcal{I}_k^{(1)}$ (see (3.4.29) and discussion above).

Lemma 3.4.3 *(Interpolation error of the full grids) For any $f \in X_0^{q,2}$ the following estimates hold:*

$$\|f - \mathcal{I}_k^{(\infty)}\|_\infty \leq \frac{d}{6^d} \cdot 2^{-2k} \cdot |f|_{\mathbf{2},\infty} \qquad = \mathcal{O}(h_k^2), \tag{3.4.40a}$$

$$\|f - \mathcal{I}_k^{(\infty)}\|_2 \leq \frac{d}{9^d} \cdot 2^{-2k} \cdot |f|_{\mathbf{2},2} \qquad = \mathcal{O}(h_k^2). \tag{3.4.40b}$$

(Here h_k stands for the mesh size at level k 1D grid).

This accuracy comes at a cost of $(2^k + 1)^d = \mathcal{O}(2^{k \cdot d}) = \mathcal{O}(h_k^{-d})$ function evaluations, since this is the number of points and basis functions in $V_k^{(\infty)}$. We can clearly see here the manifestation of the curse of dimensionality: the number of function evaluations necessary to achieve the accuracy of order $\mathcal{O}(h^2)$ grows exponentially with dimension d.

Lemma 3.4.4 *(Interpolation error of regular sparse grids) The following error bounds hold for the interpolant of the function $f \in X_0^{q,2}$ in the sparse grid space $V_k^{(1)}$:*

$$\|f - \mathcal{I}_k^{(1)}\|_\infty \leq \frac{2 \cdot |f|_{\mathbf{2},\infty}}{8^d} \cdot 2^{-2k} \cdot A(d,k) \qquad = \mathcal{O}(h_k^2 \cdot |\log_2 h_k^{-1}|^{d-1}), \tag{3.4.41a}$$

$$\|f - \mathcal{I}_k^{(1)}\|_2 \leq \frac{2 \cdot |f|_{\mathbf{2},2}}{12^d} \cdot 2^{-2k} \cdot A(d,k) \qquad = \mathcal{O}(h_k^2 \cdot |\log_2 h_k^{-1}|^{d-1}), \tag{3.4.41b}$$

where $A(d,k) := \sum_{n=0}^{d-1} \binom{k+d-1}{n} = \frac{k^{d-1}}{(d-1)!} + \mathcal{O}(k^{d-2})$.

As far as the cost of sparse grid, it can be shown that the number of points in $V_k^{(1)}$ is given by $2^k \cdot \left(\frac{k^{d-1}}{(d-1)!} + \mathcal{O}(k^{d-2})\right) = \mathcal{O}(h_k^{-1} \cdot |\log_2 h_k|^{d-1})$.

3.4.4 Adaptivity

In constructing interpolant (3.4.29) one can think of an adaptive way of selecting the subspaces $W_{\mathbf{l}}$ based on some error criterion that estimates their contribution to the interpolant. This corresponds to the dimension-adaptive approach mentioned earlier in the section on generalized sparse grids and leads to algorithm by Gerstner and Griebel [21].

Alternatively, one could perform adaptation and refinement on the level of the single basis function $\phi_{\mathbf{l},\mathbf{j}}$.

As we see from Lemma 3.4.1 for functions with bounded second order weak derivatives the surpluses $s_{\mathbf{l},\mathbf{j}}$ tend to zero as interpolation level increases. This provides an avenue for constructing adaptive sparse grids using the magnitude of the surplus as a local error indicator. This approach should allow us to better capture possible irregularities of the function of interest, such as steep slopes or jump discontinuities.

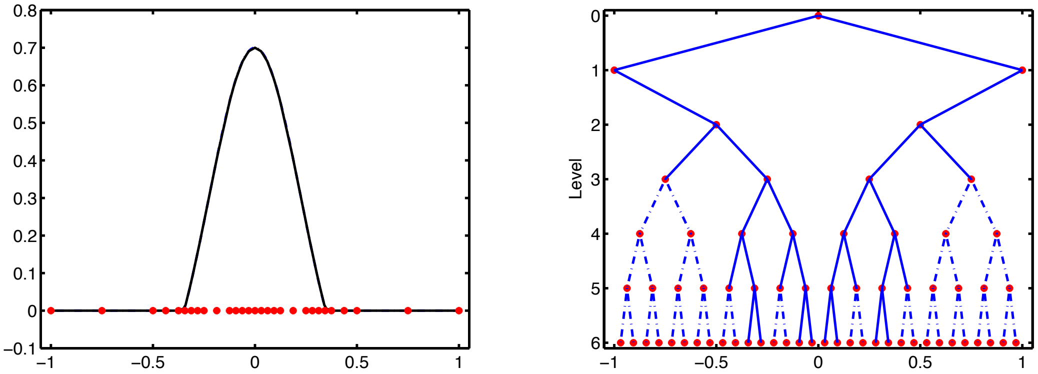
I first review adaptive procedure for one-dimensional grids and then generalize it to multi-dimensional grids.

One-dimensional hierarchical grids have a tree-like structure. In general, a grid point $y_{k,j}$ on level k has two children $y_{k+1,2j-1}$ and $y_{k+1,2j+1}$ on level $k+1$, except for the points of level 1, which only have 1 child point each. The basic idea of adaptivity is to refine the grid only by adding the children of the points with relatively large contributions given by $|s_{k,j}| \cdot \|\phi_{k,j}\|$. Note, that we can use different norms for estimating the error. The common and most obvious choice is taking the L^∞ norm, which leads to estimating local variation by the means of $|s_{k,j}|$. The resulting interpolant can be represented as

$$\mathcal{I}_l[f](y) = \sum_{k=0}^{l} \sum_{j \in B_k^\epsilon} s_{k,j} \phi_{k,j}(y) \tag{3.4.42}$$

where $B_k^\epsilon = \{j \in B_k \ : \ |s_{k,j}| \geq \epsilon\}$.

An example of adaptive interpolation of the function $f(y) = \max\{\exp\left(-10(y^2)\right) - 0.3, 0\}$ on the interval $[-1,1]$ using tolerance $\epsilon = 0.01$ is presented in Figure 3.7. Note that I limit the maximum level of the tree to guarantee termination.



(a) The function of interest and the points of the adaptive grid (in red).

(b) Adaptivity tree. The lines represent parent-child relation with dotted lines indicating that the endpoints were not added to the grid.

Figure 3.7: A six-level adaptive sparse grid for interpolating the one-dimensional function $f(y) = \max\{\exp(-10(y^2)) - 0.3, 0\}$ using error tolerance 0.01. The resulting sparse grid has 29 points, whereas the full grid has 65 points.

Note that in some cases the magnitude of the hierarchical surplus $s_{l,j}$ might not be a good error indicator. The extreme case is a function

$$f(y) = \begin{cases} 1, & y \in [-1, 0), \\ 0, & y \in [0, 1] \end{cases} \tag{3.4.43}$$

For this function hierarchical surpluses to the left of the origin are always 1/2 and, thus, the refinement process will never stop unless given a maximum level stopping criterion. A better error indicator in this case would be $|s_{l,j}| \cdot \|\phi_{l,j}\|_{L^2}$, which would take into account the sizes of the support of basis functions.

Another source of problem could be early termination due to, for example, vanishing of hierarchical surpluses corresponding to the inflection points of the function (as is seen from (3.4.36) surpluses are related to the second derivatives). To circumvent this problem look-ahead strategies might be employed, i.e., looking at the node's children even if the node is “irrelevant” according to the error indicator [17].

In general, any refinement criterion might fail in some special cases. In practice, one might develop error criteria specific for a given problem to enhance convergence (see for example [63], [64]).

Adaptive approach can be easily extended to multi-dimensional setting. In general, in d dimensions a grid point has $2 \cdot d$ children which are also its neighboring points. Given a tolerance ϵ we can construct an interpolant

$$\mathcal{I}_k^{(1)}[f] = \sum_{|\mathbf{l}|_1 \leq k} \sum_{\mathbf{j} \in B_{\mathbf{l}}^{\epsilon}} s_{\mathbf{l},\mathbf{j}} \phi_{\mathbf{l},\mathbf{j}}(\mathbf{y}) \tag{3.4.44}$$

where $B_{\mathbf{l}}^{\epsilon} = \{\mathbf{j} \in B_{\mathbf{l}} \ : \ |s_{\mathbf{l},\mathbf{j}}| \cdot \|\phi_{\mathbf{l},\mathbf{j}}(\mathbf{y})\| \geq \epsilon\}$.

Since in practice this interpolant is built recursively the choice of refinement set associated with a given point $\mathbf{y}_{\mathbf{l},\mathbf{j}}$ is not completely trivial. One might consider several strategies. Here we present a short overview of refinement strategies presented in [79] and implemented in TASMANIAN [78].

- “Classic refinement”. Isotropic refinement that adds only the children of the points with large coefficients (surpluses). It is noted that this approach might result in unstable interpolant, i.e., algorithm might fail to converge.

- “Family selective refinement”. Improves stability of the algorithm by adding the parents of the node associated with large surplus if these parents were not yet included in the grid. The parents are always considered before children.

- “Direction selective refinement”. Besides considering the coefficients associated with the multi-dimensional interpolant, it takes into account the coefficients of one-directional interpolants that are constructed using points lying on the same line in a given direction. Has the effect of reducing the clustering of the points along the lines, but includes additional cost associated with forming one-directional interpolants.

- “Family direction selective refinement”. Combines the family selective and direction selective strategies.

3.4.5 Other types of bases

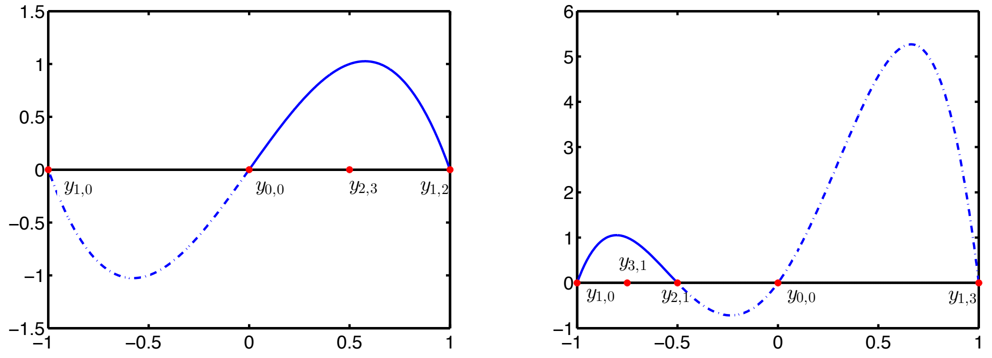
Besides the piecewise d-linear basis functions that were introduced earlier other types of basis functions are commonly employed. Some of these bases are introduced in the following.

Piecewise d-polynomial basis functions. I outline here the main idea and refer to [5], [6] for detailed description.

Polynomial basis functions of arbitrary degree p are based on Lagrange polynomials. In order to construct a polynomial of degree p associated with $y_{l,j}$ one needs $p+1$ supporting nodes. If we want to maintain the sparse structure of the grid, we need to avoid introducing any new degrees of freedom. The solution is then to use the hierarchical ancestors of the given point $y_{l,j}$ that lie outside of $[y_{l,j} - h_l, y_{l,j} + h_l]$ to help build the Lagrange polynomial. Once constructed, this polynomial is restricted to $[y_{l,j} - h_l, y_{l,j} + h_l]$, i.e., set to zero outside of the interval. In this way a higher order basis can be constructed that still retains the hierarchical structure. See figure 3.8 for examples of constructing cubic basis function $\phi_{2,3}(y)$ and quartic basis function $\phi_{3,1}(y)$. In each case hierarchical ancestors are used to construct the Lagrange polynomial. The dotted line represents those parts of the functions that get omitted.

Note that a polynomial of order l cannot be used before level $l-1$ due to insuffcient number of ancestors.

The d-dimensional basis functions are derived by tensor product of one-dimensional ones. The polynomial degree can be different in each dimension. The piecewise d-polynomial basis functions have the advantage that higher order convergence can be reached than in piecewise d-linear case: for a function f with (in some sense) bounded weak mixed derivatives of the order less than or equal to $p+1$ in each dimension it



(a) Cubic basis function $\phi_{2,3}$.

(b) Quartic basis function $\phi_{3,1}$.

Figure 3.8: Construction of cubic and quartic basis functions. Dotted part of the line in each case gets omitted at the final step of the constuction.

can be shown [8] that the order of approximation is increased to

$$\mathcal{O}(h_k^{(p+1)} \cdot |\log_2 h_k^{-1}|^{(p+2)(d-1)}) \tag{3.4.45}$$

(compare with 3.4.4).

In order to provide comparison to the rates of convergence obtained using Monte Carlo-like methods I provide the ϵ-complexities of the approximations using piecewise d-polynomial basis of degree p.

Theorem 3.4.5 *Given N grid points the following accuracies can be obtained for the problem of computing the interpolant $\mathcal{I}_k^{(p,1)}[f] \in V_k^{(p,1)}$, i.e. maximum degree p in each direction, with respect to L^∞ and L^2 norms:*

$$\epsilon_\infty^{(p)}(N) = \mathcal{O}(N^{-(p+1)} \cdot |\log_2 N|^{(p+2)(d-1)}) \tag{3.4.46}$$

$$\epsilon_2^{(p)}(N) = \mathcal{O}(N^{-(p+1)} \cdot |\log_2 N|^{(p+2)(d-1)}) \tag{3.4.47}$$

Wavelet basis. One of the major drawbacks of utilizing multi-polynomial basis in the framework of the hierarchical adaptive sparse grid method introduced above is the lack of the error estimate from below with constants independent of the number of hierarchical levels. This means that the magnitude of the hierarchical coefficient only provides a local error indicator and not a true error estimator. In order to overcome this drawback authors in [30] propose the use of multi-resolution wavelet basis that “guarantees optimality”. In particular, they build a locally supported hierarchical basis consisting of second-generation wavelets constructed using lifting scheme ([80]). Presented numerical results speak in favor of the proposed approach, however, improvements appear to be marginal.

B-splines, mexican hat functions, etc. Successful results have been reported of the use of other types of bases tailored for specific applications: B-spline basis functions for regression problems and Mexican hat wavelet for the problems in finance ([63]).

Chapter 4

Numerical experiments

In this chapter I use the following contractions for different integration methods: MC for Monte-Carlo, QMC for quasi-Monte Carlo, SG for sparse grids (Smolyak rule) and hASG for hierarchical adaptive sparse grids using piecewise polynomial basis.

The numerical experiments presented here were performed in MATLAB2013b. The following codes and software packages were used: for quasi-Monte Carlo sequences - SOBOL_DATASET and FAURE_DATASET by John Burkardt ([10, 9]), for sparse grids and locally adaptive sparse grids - TASMANIAN by Miroslav Stoyanov et al at ORNL ([78]).

4.1 Simple integration problems

I study the performance of the methods introduced in Chapter 3 on several simple test problems. For all the problems the exact value of the integral is known either by direct computation or highly accurate approximation.

Problem 1 Gaussians

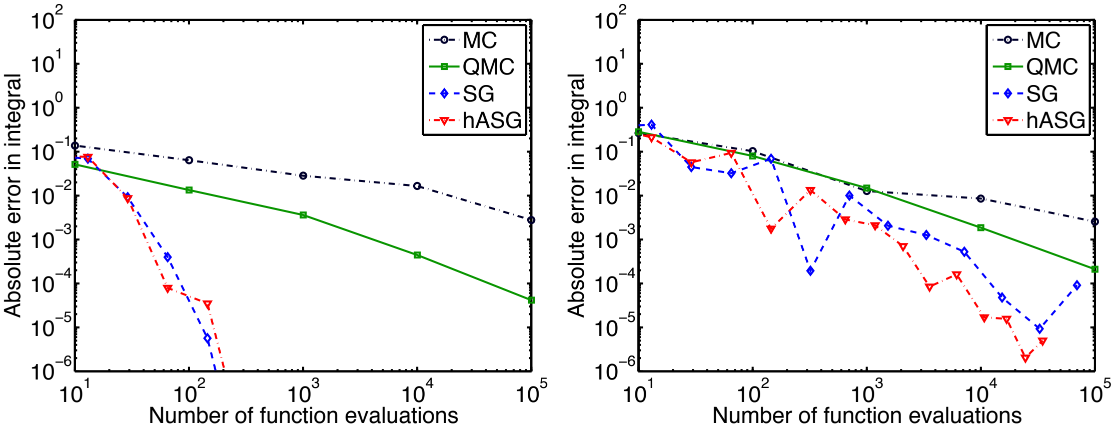
Consider a multi-dimensional Gaussian defined on $[-1, 1]^d$:

$$f_1(\mathbf{y}) = \exp\,(-\sum_{i=1}^{d}(y^{(i)})^2) \tag{4.1.1}$$

with exact integral given by $\pi^{d/2} \cdot \mathrm{erf}(1)^d$. This function is infinitely smooth and has a product structure with each dimension being equally important.

As well as comparing different methods for f_1 with varying dimension, I compare them also on the problem with non-smoothness introduced by $[\cdot]_+$ function:

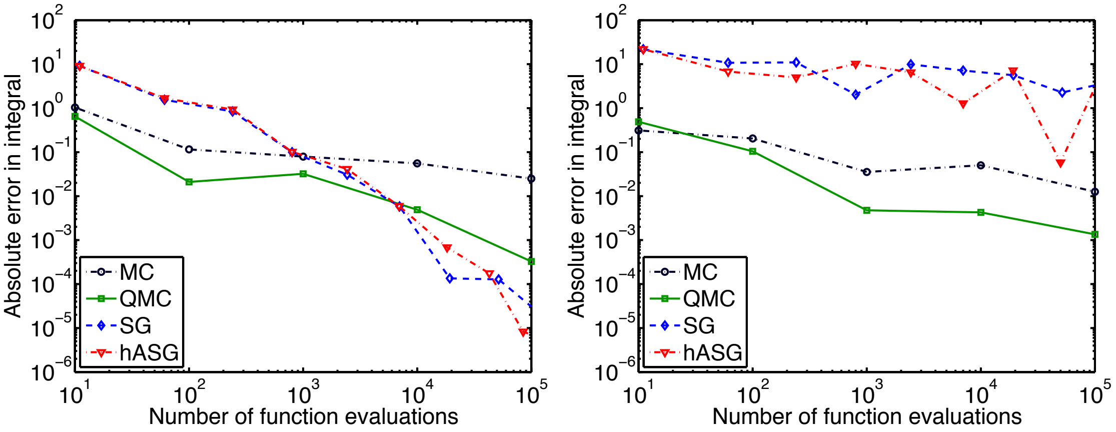
$$f_2(\mathbf{y}) = 2 \cdot [f_1(\mathbf{y}) - \frac{1}{2}]_+ = 2 \cdot \max\{\exp\left(-\sum_{i=1}^{d} (y^{(i)})^2\right) - \frac{1}{2}, 0\} \tag{4.1.2}$$



(a) Integration errors in f_1.

(b) Integration errors in f_2.

Figure 4.1: Convergence comparison for $d = 2$.



(a) Integration errors in f_1.

(b) Integration errors in f_2.

Figure 4.2: Convergence comparison for $d = 5$.

The results for dimensions $d = 2$ and $d = 5$ are plotted in Figures 4.1 and 4.2. In both cases the Faure sequence was used as a base for QMC integration, and the Clenshaw-Curtis rule as a base for sparse grid approximation (SG). Furthermore, the hierarchical adaptive sparse grid method (hASG) was used with local polynomial basis of maximum possible order (i.e., the order of basis functions increased from level to level), "classic" type of refinement and tolerance $\epsilon = 10^{-5}$.

Observe that for the smooth function f_1 sparse grids based on global or local polynomials outperform MC type methods dramatically. However, the convergence rate of sparse grids deteriorates with dimension faster. Although this is expected from the theory (see (3.3.30) and (3.4.46)), it is still surprising that already for dimension $d = 5$ sparse grids provide not much of an improvement over MC type methods. This, however, might be a particular case of approximating Gaussians (see [63] for a detailed discussion).

For the non-smooth function f_2, I observe that sparse grids methods behave much worse even for dimension 2 with convergence establishing rather late. Going to dimension 5 it is clear that QMC is the method of choice for this particular problem. Of the sparse grid methods, the hierarchical adaptive method is marginally better than the regular sparse grids, but improvement is much less than one would hope for. It might be partially due to the fact that the region of discontinuity in the derivative for function f_2 is a circle and, thus, it is completely misalligned with the grid, which is a worst case scenario for the method (see Figure 4.3). Therefore, a lot of points are needed are spent around the jump in the derivative. To prove this theory, I also present the results for a function with discontinuity aligned along the axes (Example 2).

Now, consider smoothing the function f_2 in order to improve convergence rates of sparse grids. Specifically I replace the $[\cdot]_+$ function with a C^∞ approximation. I choose the smooth approximation to be

$$\mathcal{P}(y, \gamma) = y + \frac{1}{\gamma} \log(1 + \exp^{-\gamma y}) \tag{4.1.3}$$

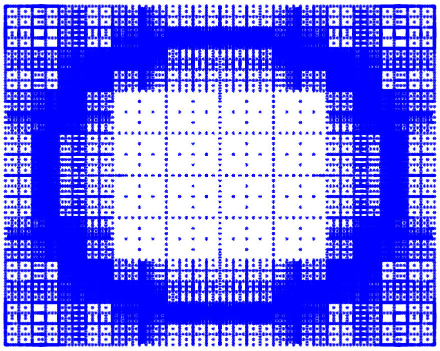


Figure 4.3: Adaptive sparse grid for the function $f_2(y)$.

(see [12] for details). This function has the following properties:

1. $\mathcal{P}(y,\gamma) \to [y]_+$ as $\gamma \to \infty$;

2. $\mathcal{P}(y,\gamma) > [y]_+$;

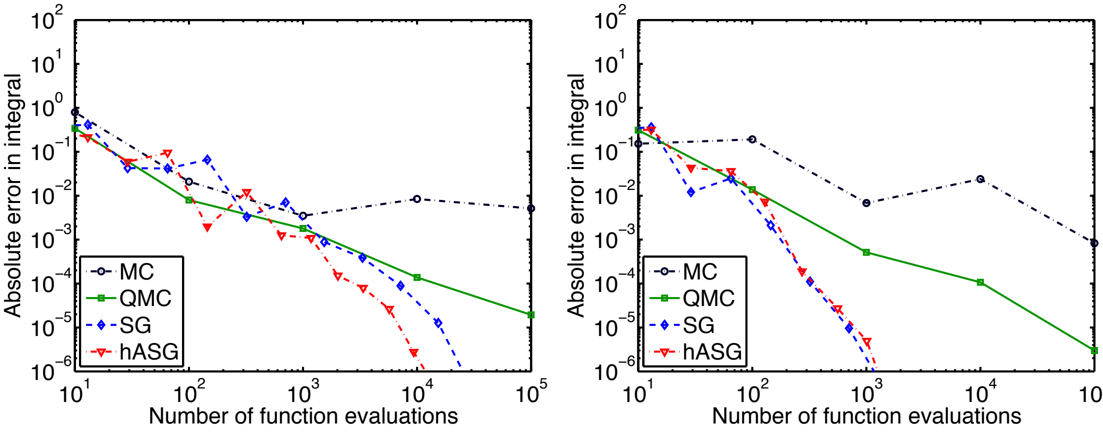
3. $|\mathcal{P}'(y,\gamma)| < 1$;

4. $|\mathcal{P}''(y,\gamma)| \leq \frac{\gamma}{4}$.

I let the parameter γ take values 100 and 10 and compare the performance of methods on the smoothed version of function f_2, that I denote by f_3:

$$f_3(\mathbf{y}) = 2 \cdot \mathcal{P}(f_2(\mathbf{y}) - \frac{1}{2}, \gamma) \tag{4.1.4}$$

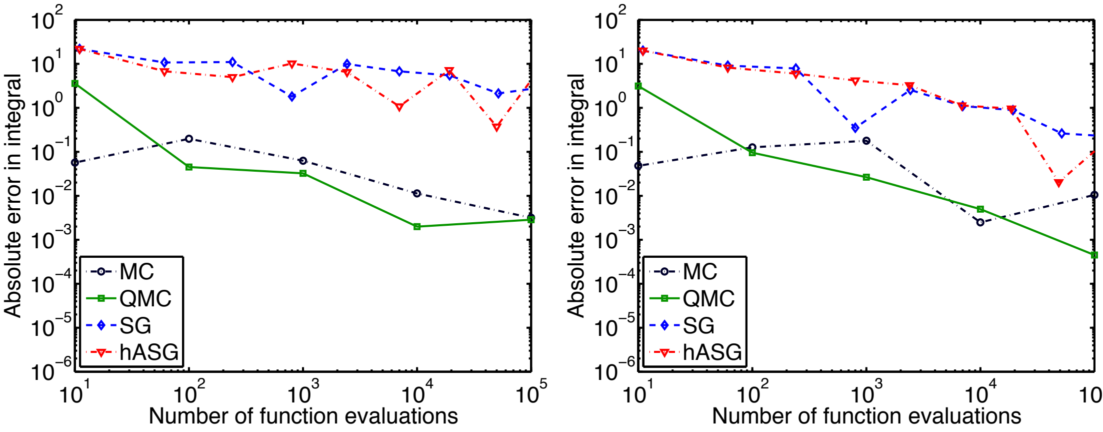
The smaller value of γ corresponds to the smoother version of the function, while the higher value provides closer approximation to the original function. As we see from Figures 4.4 and 4.5, the smoothing of the integrand reflects favorably on the convergence properties of sparse grids. These figures are to be compared with 4.1b and 4.2b respectively.



(a) $\gamma = 100$.

(b) $\gamma = 10$.

Figure 4.4: Convergence comparison for f_3 with $d = 2$.



(a) $\gamma = 100$.

(b) $\gamma = 10$.

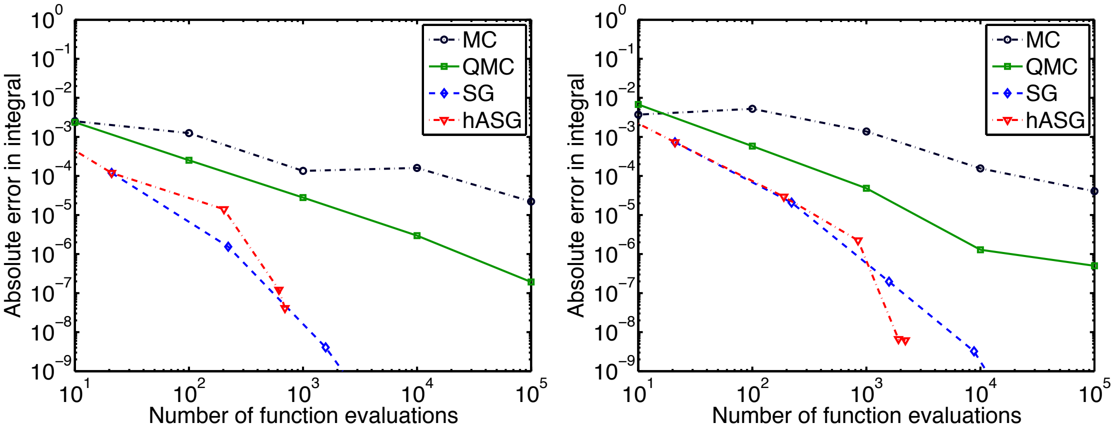
Figure 4.5: Convergence comparison for f_3 with $d = 5$.

Finally, consider the version of the function f_1 with weights assigned to different dimensions:

$$f_4(\mathbf{y}) = \exp\left(-\sum_{i=1}^{d} \frac{(y^{(i)} - c_i)^2}{w_i^2}\right) \tag{4.1.5}$$

Let $c_i = \frac{1}{2}$ for all i and integrate f_4 over $[0, 1]^d$. I consider two choices of weights w_i. First, take $w_i = 2^i$ for $i = 1, \ldots, d$. With this choice the function becomes more and more constant with increasing dimension. Next, let $w_i = 2^{r_i}$, where r

is a random vector scaled so that $\sum_{i=1}^{d} r_i = \sum_{i=1}^{d} i$. The results are plotted in Figures 4.6 for $d = 10$. Observe that performance of sparse grids is much better now that the function has only few "important" dimensions. Furthermore, when the weights assigned to different dimensions are random, the adaptive sparse grids do a better job by automatically finding dimensions with greater variability.



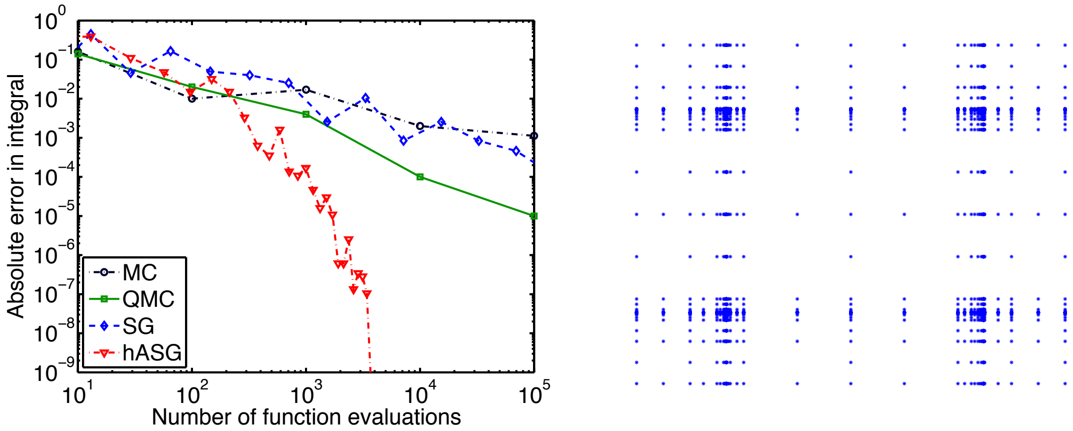
(a) $w_i = 2^i$.

(b) $w_i = 2^{r_i}$.

Figure 4.6: Function f_4 with different choice of weights for $d = 10$.

Problem 2 Jump along hypercube

Consider integrating a function $f_5(\mathbf{y}) = \chi_{[0.21,0.81]^d}$ over $[0, 1]^d$. This simple test allows to see the benefit of axis-aligned structure for the performance of sparse grids based on local polynomials. For this test I take standard hat basis functions and set the refinement tolerance to $\epsilon = 10^{-3}$. The results are depicted in Figure 4.7.



(a) Integration errors in f_5.

(b) Adaptive sparse grid.

Figure 4.7: Convergence comparison for the function f_5 with $d = 2$.

Problem 3 Absorption problem

Consider a simple transport problem described by the integral equation

$$f(x) = \int_x^1 \gamma f(t)dt + x. \tag{4.1.6}$$

This equation describes particles traveling through a one-dimensional slab of length one [53]. In each step the particle covers a distance which is uniformly distributed in $[0, 1]$. This may cause it to exit the slab; otherwise, it may be absorbed with probability $1 - \gamma$ before the next step. The variable x corresponds to the current position of the particle, while the value of $f(x)$ gives the probability that the particle will eventually leave the slab given that it has already made it to x. The quantity of interest is $f(0)$, i.e., the probability that a particle entering the slab will leave the slab.

The exact solution to this problem is given by

$$f(x) = \frac{1}{\gamma}(1 - (1 - \gamma)\exp(\gamma(1 - x))) \tag{4.1.7}$$

and can also be represented as an infinite-dimensional integral over the unit cube

$$f(x) = \int_{[0,1]^\infty} \sum_{i=1}^{\infty} F_i(x, \mathbf{y})d\mathbf{y}, \tag{4.1.8}$$

where

$$F_i(x, \mathbf{y}) = \gamma^i \theta \left(1 - x - \sum_{j=1}^{i} y^{(j)} \right) \cdot \theta \left(\sum_{j=1}^{i+1} y^{(j)} - (1 - x) \right). \tag{4.1.9}$$

Here function $\theta(t)$ is the Heaviside function:

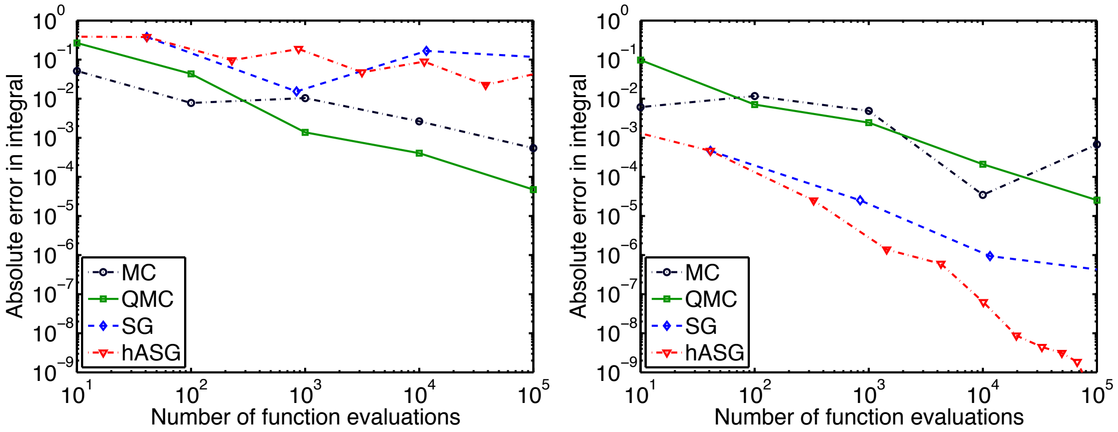
$$\theta(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{4.1.10}$$

This corresponds to a particle doing a forward random walk with jump size uniformly distributed on $[0, 1]$. If the particle leaves on its $(i + 1)$-st jump, it contributes γ^i to the approximation of $f(x)$. Since higher dimensions correspond to particles that undergo many collisions before leaving the slab, and the likelihood that a particle can do more than a few collisions before leaving or being absorbed is small, $f(x)$ can be approximated sufficiently accurately by truncating the infinite sum to a finite value d. For this experiment I took $d = 20$. Also, the survival probability γ was chosen to be $\frac{1}{2}$. See results in Figure 4.8a.

The integrand above is non-smooth with discontinuity along the diagonal, which is a worse case for sparse grids. Much better results can be obtained by changing the formulation to the one with smooth integrand. The idea behind it is that each jump that a particle makes should contribute something to the evaluation of the integral. Therefore, the discontinuous integrand $F_i(x, \mathbf{y})$ can be replaced by a polynomial $F_i^*(x, \mathbf{y})$:

$$F_i^*(x, \mathbf{y}) = \gamma^i (1 - x)^i \left(\prod_{j=1}^{i-1} (y^{(j)})^{i-j} \right) \left(1 - (1 - x) \prod_{j=1}^{i} y^{(j)} \right). \tag{4.1.11}$$

The results for smooth problem are presented in Figure 4.8b. As expected, sparse grids now perform much better. Adaptive sparse grids give an additional improvement by automatically detecting more important dimensions.



(a) Integration errors for discontinuous absorption problem.

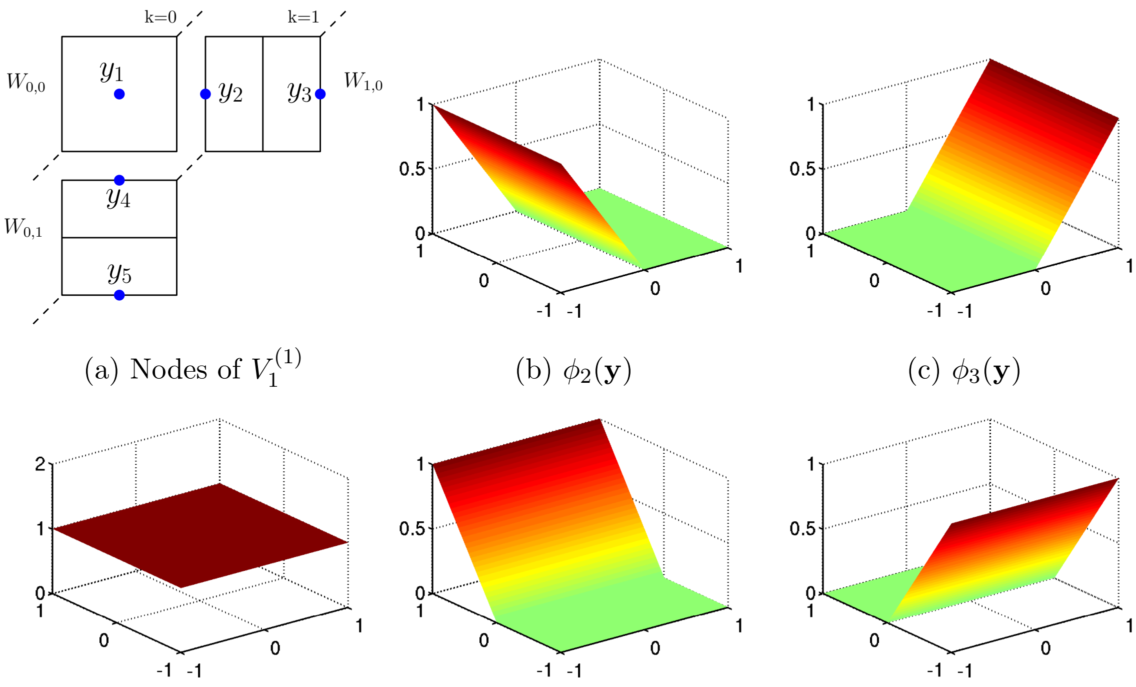
(b) Integration errors for smooth absorption problem.

Figure 4.8: Convergence results for absorption problem with d=20.

4.1.1 Note on weights

One of the major drawbacks of sparse grids in the context of optimization is the fact that they lead to the appearance of negative weights, as was mentioned earlier in the subsection on Smolyak quadrature. This poses a problem for optimization methods (see, e.g., [41]). To demonstrate the appearance of negative weights I consider a problem of integrating a two-dimensional function using hierarchical bilinear basis introduced in Section 3.4.2.

Recall that when building the interpolant using a hierarchical basis we add the basis functions from hierarchical subspaces (see Figure 3.6). In particular, the functions comprising the subspaces $W_{0,0}$, $W_{1,0}$ and $W_{0,1}$ that are used to construct sparse grid interpolant corresponding to $V_1^{(1)}$ are depicted in Figure 4.9. Here in order to simplify notation I relabeled the points of the grid and ordered them in a sequence of N points (4.9a). In this basis the coefficients of the interpolant are hierarchical surpluses.



(a) Nodes of $V_1^{(1)}$ (b) $\phi_2(\mathbf{y})$ (c) $\phi_3(\mathbf{y})$

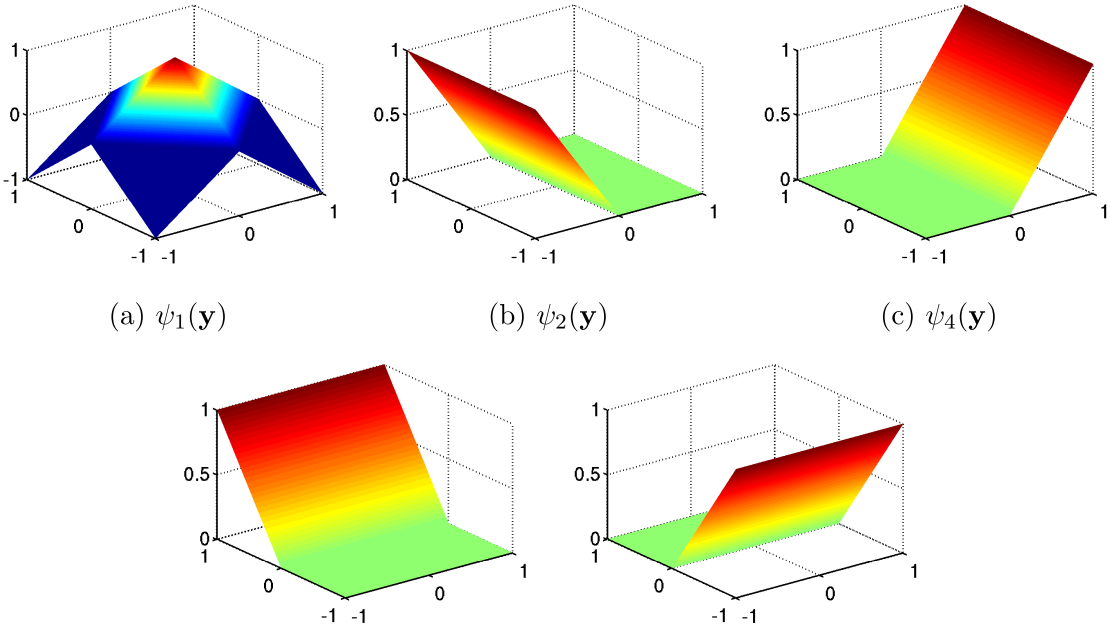
(d) $\phi_1(\mathbf{y})$ (e) $\phi_4(\mathbf{y})$ (f) $\phi_5(\mathbf{y})$

Figure 4.9: 2D hierarchical basis functions.

Now I wish to switch from this hierarchical basis to its dual set in which the coefficients of the interpolant are the function values at grid points, i.e., I want to represent

$$f(\mathbf{y}) \approx \sum_{i=1}^{N} f(\mathbf{y}_i)\psi_i(\mathbf{y}). \tag{4.1.12}$$

The functions $\psi_i(\mathbf{y})$ are constructed by applying the dual of the linear transformation to the hierarchical basis functions $\phi_i(\mathbf{y})$ (Figure 4.10). Since hierarchical surpluses corresponding to $\phi_2(\mathbf{y})$-$\phi_5(\mathbf{y})$ at this level interpolant are just the function values at $\mathbf{y}_2$-$\mathbf{y}_5$, the functions $\psi_2(\mathbf{y})$-$\psi_5(\mathbf{y})$ coincide with $\phi_2(\mathbf{y})$-$\phi_5(\mathbf{y})$.



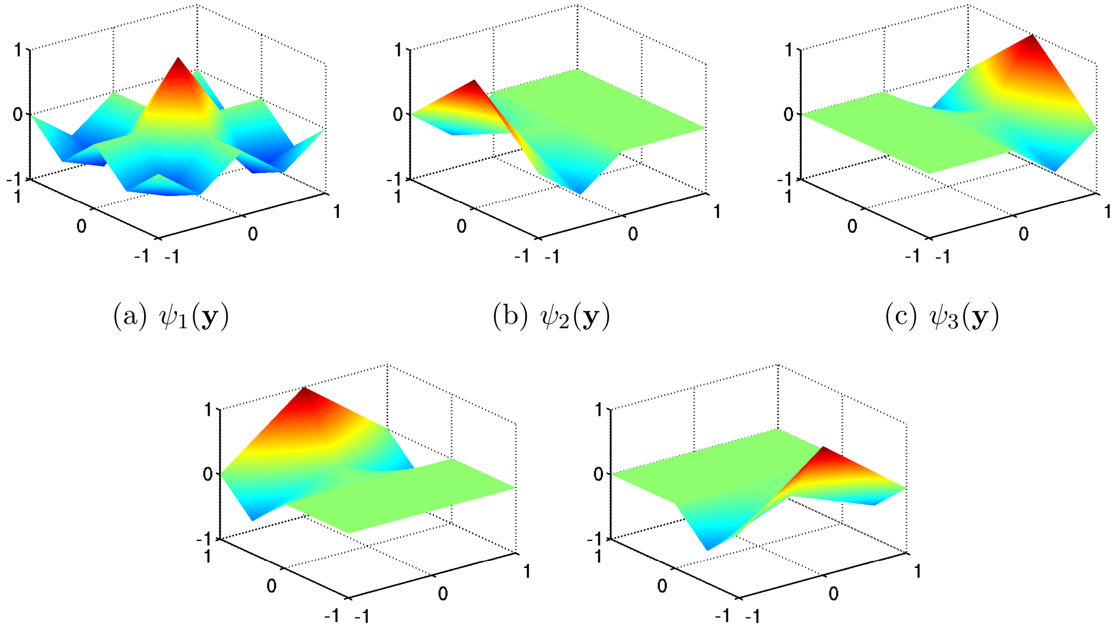
(a) $\psi_1(\mathbf{y})$ (b) $\psi_2(\mathbf{y})$ (c) $\psi_4(\mathbf{y})$

(d) $\psi_4(\mathbf{y})$ (e) $\psi_5(\mathbf{y})$

Figure 4.10: Functions $\psi_i(\mathbf{y})$ corresponding to the sparse grid of level 1.

The function $\psi_1(\mathbf{y})$ corresponding to the point $\mathbf{y}_1 = (0,0)$, however, changes and becomes partially negative. Its integral is zero.

When going to the next level of the sparse grid, i.e., adding the subspaces needed to get $V_2^{(1)}$, the functions $\psi_i(\mathbf{y})$ corresponding to the old points of the grid change (see Figure 4.11). The integral of function $\psi_1(\mathbf{y})$ now becomes -1, while the other functions shown on the figure integrate to 0.



(a) $\psi_1(\mathbf{y})$

(b) $\psi_2(\mathbf{y})$

(c) $\psi_3(\mathbf{y})$

(d) $\psi_4(\mathbf{y})$

(e) $\psi_5(\mathbf{y})$

Figure 4.11: Functions $\psi_i(\mathbf{y})$ corresponding to the sparse grid of level 2.

Thus, the sparse structure of the grid leads to the negative weights in the quadrature formula for $f(\mathbf{y})$. This is independent of the type of the basis considered. By contrast the full grids corresponding to subspaces $V_k^{(\infty)}$ will always have positive weights.

4.2 Linear elliptic problem with random inputs

Consider the problem described in Example 2.2.1. Assuming that the control is $z(x) \equiv 0$ we have:

$$-\nabla \cdot (a(\omega, x)\nabla u(\omega, x)) = f(\omega, x) \qquad (\omega, x) \in \Omega \times D, \tag{4.2.1a}$$

$$u(\omega, x) = 0 \qquad (\omega, x) \in \Omega \times \partial D \tag{4.2.1b}$$

with $D = [0, 1]^2$. I will denote the components of $x \in D$ by x_1 and x_2.

Here I want to investigate the convergence of sparse grids and MC type methods when applied to the problem of computing the expected value and semi-deviation of the solution of (4.2.1). I adopt the description of the problem from [57].

Consider deterministic source $f(\omega, x) \equiv f(x) = \cos(x_1)\sin(x_2)$. The approximation of the random diffusion coefficient $a(\omega, x)$ is constructed in the following manner. First, construct the Karhunen-Loève (KL) expansion of a one-dimensional random field $b(\omega, x_1)$ with covariance function

$$\mathbb{COV}[b(\cdot, x_1), b(\cdot, x_1')] = \exp\left(\frac{-(x_1 - x_1')^2}{L_c^2}\right). \tag{4.2.2}$$

Here parameter L_c corresponds to the physical correlation length for the random field $b(\omega, x_1)$, meaning that random variables $b(\cdot, x_1)$ and $b(\cdot, x_1')$ become essentially uncorrelated if $|x_1 - x_1'| \gg L_c$.

The eigenvalues and eigenfunctions of the covariance function (4.2.2) are given by the following analytic expressions [57]:

$$\lambda_1 = \left(\frac{\sqrt{\pi}L}{2}\right)^{1/2}$$

$$\lambda_n = (\sqrt{\pi}L)^{1/2} \exp\left(\frac{-(\lfloor \frac{n}{2} \rfloor \pi L)^2}{8}\right), \quad \text{for } n > 1$$

and

$$\phi_n(x_1) = \begin{cases} \sin\left(\frac{\lfloor \frac{n}{2} \rfloor \pi x_1}{L_p}\right), & \text{if } n \text{ even,} \\ \cos\left(\frac{\lfloor \frac{n}{2} \rfloor \pi x_1}{L_p}\right), & \text{if } n \text{ odd} \end{cases}$$

where parameters L_p and L are given by $L_p = \max\{1, 2L_c\}$ and $L = L_c/L_p$.

The truncation of the KL expansion of the field $b(\omega, x_1)$ is then given by

$$b_N(\omega, x_1) = 1 + Y_1(\omega)\left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{n=2}^{N} \lambda_n \phi_n(x_1) Y_n(\omega) \tag{4.2.3}$$

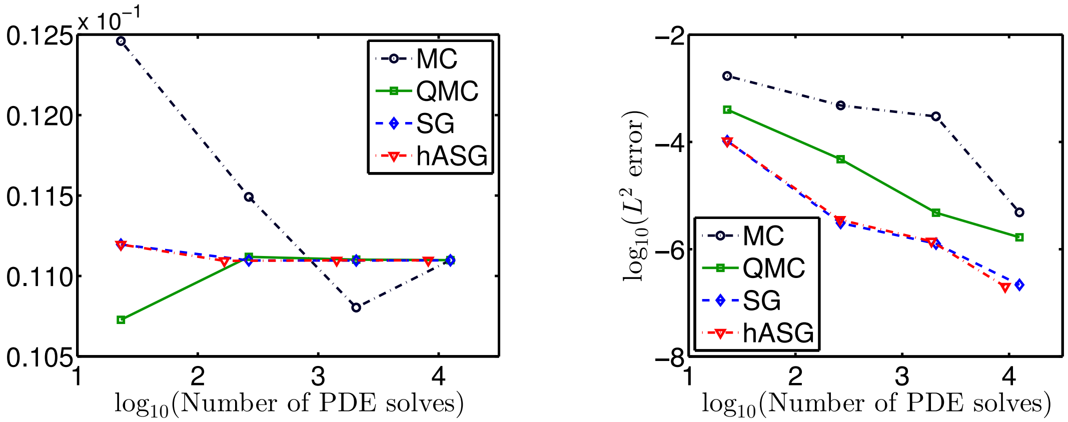
Here $Y_n(\omega)$, $n = 1, \ldots, N$, are random variables that could be determined given the analytical expression for the random field $b(\omega, x_1)$ ([22, Sec. 2.3]). For the purposes of this experiment I take $\{Y_n(\omega)\}_{n=1}^{N}$ to be independent uniformly distributed random variables in the interval $[-\sqrt{3}, \sqrt{3}]$.

Finally, take an approximation of the random field $a(\omega, x)$ to be

$$a_N(\omega, x) = 0.5 + \exp\left(b_N(\omega, x_1)\right) \tag{4.2.4}$$

Small values of the correlation length L_c correspond to the slow decay of eigenvalues λ_n, thus, each random dimension is weighed almost equally. On the other hand, large values of L_c result in the fast decay rates with only few first dimensions being most important. For my numerical experiments I take $N = 11$ and $L_c = 1/2$. For this choice of L_c the decay is fast with only first 4-6 dimensions contributing most to the variability of the coefficient $a_N(\omega, x)$.

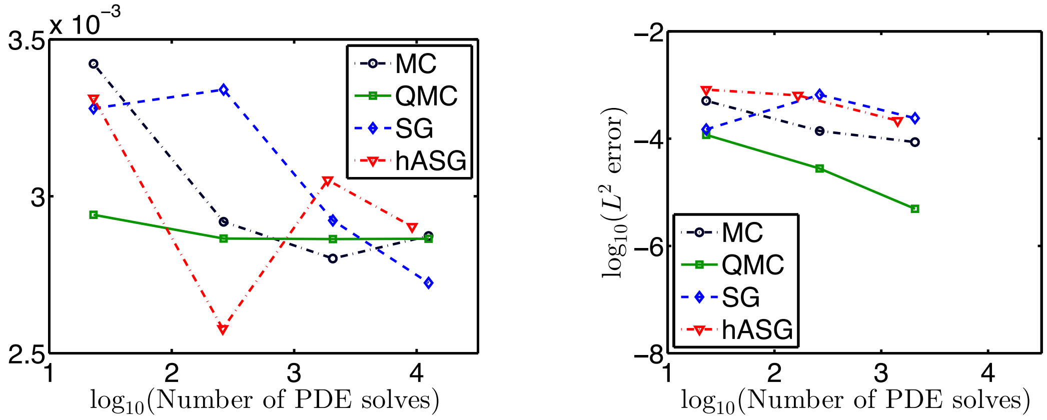
First, I approximate the expected value of the solution $u_N(\omega, x)$. This is done for each point x in the spatial discretization. I then take the $L^2(D)$ norm of the result. The approximations of L^2 norm of the expected value of the solution for different methods are plotted versus the number of points used in the parameter space in Figure 4.12a. Note that the same number of points is used for MC methods as for SG method (based on Clenshaw-Curtis rule). The tolerance for hASG methods is set to 10^{-5} and "family direction selective refinement" is used (see Section 3.4.4). I observe that sparse grid methods and quasi-Monte Carlo method (Sobol' sequence) both converge fast to what should be the true value. In Figure 4.12b I plot the L^2 norms of the difference between expected values obtained using each method and expected value obtained using sparse grids of level 6 (with ≈ 63000 of points). In other words, I plot $\| \mathbb{E}[u_N(\omega,x)] - \mathbb{E}[u^*(\omega,x)] \|_{L^2(D)}$ vs the number of points in parameter space, where $u^*(\omega,x)$ is a "reference" solution obtained sparse grids of level 6. This allows to see the convergence rates of different methods. I observe that sparse grids converge fastest with a little performance gained by adaptive sparse grids over the standard ones.



(a) $\| \mathbb{E}[u_N(\omega,x)] \|_{L^2}$.

(b) Errors wrt the reference solution.

Figure 4.12: Approximation of the expected value of the solution (N=11).



(a) $\| \mathbb{E}[u_N(\omega,x) - \mathbb{E}[u_N(\omega,x)]]_+ \|_{L^2}$.

(b) Errors wrt the next level solution.

Figure 4.13: Approximation of the semi-deviation of the solution (N=11).

Next, I estimate the semi-deviation of the solution $u_N(\omega,x)$. In Figure 4.13a I plot the $L^2(D)$ norm of the semi-deviation values obtained using different methods vs the number of points in the parameter space. Now that the integrand is non-smooth due to the presence of the $[\cdot]_+$ function the sparse grid methods do not seem to converge within the number of PDE solves allowed. Although the true solution is not known, the behavior of MC methods indicates that they converge to what should be a true value, with QMC clearly converging to it much faster. Since the sparse grid solution in this case cannot be trusted, instead of plotting the errors with respect to the finest sparse grid solution, I plot the errors between two consecutive levels of approximation for each of the methods, i.e., I estimate $\|\,\mathbb{E}[u_N^{(l)}(\omega,x)] - \mathbb{E}[u_N^{(l+1)}(\omega,x)]\|_{L^2(D)}$, where l for the sparse grids means the level in the Smolyak's construction and for MC methods means the corresponding level of approximation. The results are presented in Figure 4.13b. I observe that reduction in error is smallest for QMC which indicates faster convergence, while for SG method using more points does not guarantee reduction in error. For hASG from level to level the error reduces but slower than for QMC. Even standard MC looks more preferrable in this setting than any of the sparse grid methods.

The results obtained here for the case of expected value essentially repeat the results in Ma and Zabaras (2009) [52] (where only SG and hASG methods were compared). In their paper the authors also demonstrate the convergence of hASG for higher dimensions (N=25, 50, 75, 100) when the correlation length L_c is large ($L_c = 0.6$). As mentioned above, in this case only few dimensions contribute most to the variability of the solution, hence, adaptive sparse grids perform well by detecting important dimensions and keeping the total number of points relatively low. To the best of my knowledge, comparison of different methods has not been done on this problem for the case of semi-deviation. The findings here and in Section 4.1 indicate that although hierarchical sparse grids perform better on non-smooth problems than standard sparse grids, even for moderately high dimensions ($d = 5 \sim 11$) their performance is inferior to quasi-Monte Carlo and even standard Monte-Carlo methods.

4.3 Optimal control problem

4.3.1 A 1D model problem

As an example problem I use the stationary advection diffusion equation in 1 dimension. Consider the following minimization problem

$$\min_{u,z} \quad J(u,z) := \max_{x\in[0.8,0.9]} [u(x) - v]_+ + \frac{\alpha}{2} \int_{0.5}^{0.6} z(x)dx \tag{4.3.1}$$

with $v \equiv 0.1$ being a given "target" or "goal", subject to

$$-D\frac{\partial^2}{\partial x^2}u(x) + a\frac{\partial}{\partial x}u(x) = f(x) + \chi_{[0.5,0.6]}z(x), \quad x \in [0,1], \tag{4.3.2a}$$

$$u(0) = u_0, \quad u(1) = u_1. \tag{4.3.2b}$$

where D and a are given constants, and $z(x)$ plays the role of the control. Multiplying the control by a characteristic function restricts it to the interval $[0.5, 0.6]$. For the case $u_0 = u_1 = 0.1$ and a source $f(x)$ being a gaussian function, the optimal control $z(x)$ is depicted in Figure 4.14.

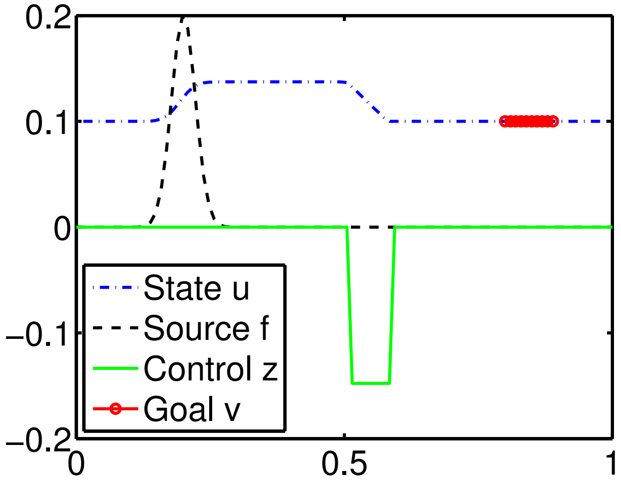


Figure 4.14: Optimal control z for the problem (4.3.1)-(4.3.2). Here control z and source f are scaled by 1/5 in order to better demonstrate the solution u.

To derive the weak form of (4.3.2) I multiply the differential equation in (4.3.2) by a function $\phi \in H_0^1(0,1)$; then I integrate both sides over $[0,1]$, and apply integration by parts. This leads to the weak form formulation: Find $u \in H^1(0,1)$ with $u(0) = u_0$, $u(1) = u_1$ such that

$$\int\limits_0^1 D\frac{\partial}{\partial x}u(x)\frac{d}{dx}\phi(x) + a\frac{\partial}{\partial x}u(x)\phi(x) - \int\limits_{0.5}^{0.6} z(x)\phi(x)dx = \int\limits_0^1 f(x)\phi(x)dx \qquad (4.3.3)$$

for all $\phi \in H_0^1(0,1)$.

I subdivide the interval $[0,1]$ into $N+1$ subintervals of length $h = 1/(N+1)$ and define

$$\varphi_i(x) \quad = \quad \begin{cases} (h)^{-1}(x-(i-1)h) & x \in [(i-1)h, ih], \\ (h)^{-1}(-x+(i+1)h) & x \in [ih, (i+1)h], \\ 0 & \text{else} \end{cases}$$

$i = 0, \dots, N+1$.

I approximate u by a function of the form

$$u_h(x) = \sum_{j=1}^{N} u_j\varphi_j(x) + u_0\varphi_0(x) + u_1\varphi_{N+1}(x) \qquad (4.3.4)$$

and z by

$$z_h(x) = \sum_{j=I_c} z_j\varphi_j(x). \qquad (4.3.5)$$

where $I_c = \{j \ : \ \phi_j(x) \ \text{has support on } [0.5, 0.6]\}$ with $|I_c| = m$.

Set

$$\mathbf{u} = (u_1, \dots, u_N)^T, \quad \mathbf{z} = (z_1, \dots, z_m)^T.$$

Insert these approximations into (4.3.3) and require (4.3.3) to hold for $\phi = \varphi_j$, $j = 1, \dots, N$. This leads to the linear system

$$\mathbf{Au} + \mathbf{Bz} = \mathbf{b}, \qquad (4.3.6)$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times m}$, and $\mathbf{b} \in \mathbb{R}^{N}$ are matrices or vectors with entries

$$\begin{aligned}
\mathbf{A}(i,j) &= \int_0^1 D\frac{d}{dx}\varphi_j(x)\frac{d}{dx}\varphi_i(x) + a\frac{d}{dx}\varphi_j(x)\varphi_i(x), \quad i,j = 1,\dots,N, \\
\mathbf{B}(i,j) &= -\int_{0.5}^{0.6} \varphi_j(x)\varphi_i(x)dx, \qquad i = 1,\dots,N, \quad j \in I_c, \\
b_0 &= \int_0^1 f(x)\varphi_1(x)dx - u_0\int_0^1 D\frac{d}{dx}\varphi_0(x)\frac{d}{dx}\varphi_1(x) + a\frac{d}{dx}\varphi_0(x)\varphi_1(x)dx, \\
b_i &= \int_0^1 f(x)\varphi_i(x)dx, \qquad i = 2,\dots,N-1, \\
b_N &= \int_0^1 f(x)\varphi_N(x)dx - u_1\int_0^1 D\frac{d}{dx}\varphi_{N+1}(x)\frac{d}{dx}\varphi_N(x) + a\frac{d}{dx}\varphi_{N+1}(x)\varphi_N(x)dx
\end{aligned}$$

Since the coefficients are constant, I can compute the integrals. Then the stiffness and the mass matrices are given by

$$\begin{aligned}
\mathbf{A} &= \frac{D}{h}\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix} \\
&\quad +\frac{a}{2}\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix} \in \mathbb{R}^{N\times N},
\end{aligned}$$

$$\mathbf{B} = -\frac{h}{6}\begin{pmatrix} 0 & 0 & 0 & & & & \\ \vdots & \vdots & \vdots & & & & \\ 0 & 0 & 0 & & & & \\ 2 & 1 & 0 & & & & \\ 1 & 4 & 1 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 4 & 1 \\ & & & & 0 & 1 & 2 \\ & & & & 0 & 0 & 0 \\ & & & & \vdots & \vdots & \vdots \\ & & & & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{N\times m},$$

and the right hand side vector (using composite trapezoidal rule) by

$$\mathbf{b} = \begin{pmatrix} hf(h) + u_0(\frac{D}{h} + \frac{a}{2}) \\ hf(2h) \\ \vdots \\ hf(1-2h) \\ hf(1-h) + u_1(\frac{D}{h} - \frac{a}{2}) \end{pmatrix} \in \mathbb{R}^N.$$

Consider now objective $J(u, z)$ from (4.3.1) in discretized form:

$$J(\mathbf{u}, \mathbf{z}) = \|[\mathbf{C}\mathbf{u} - \mathbf{v}]_+\|_\infty + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z} \tag{4.3.7}$$

where the matrix $\mathbf{C} \in \mathbb{R}^{p\times N}$ only picks out components of $\mathbf{u}$ that are in the region of interest ($[0.8, 0.9]$). The matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$ is given by

$$\mathbf{R} = \frac{h}{6}\begin{pmatrix} 2 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 2 \end{pmatrix} \in \mathbb{R}^{m\times m},$$

and the penalty coefficient α is set to 1. Moreover, vector $\mathbf{v} \in \mathbb{R}^p$ has all components equal to 0.1. The boundary conditions u_0 and u_1 are set to 0.1 as well.

In the following, I define

$$\mathbf{u}(\mathbf{z}) = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{B}\mathbf{z}).$$

The deterministic optimization problem is

$$\min_{\mathbf{z}} \; \|[\mathbf{C}\mathbf{u}(\mathbf{z}) - \mathbf{v}]_+\|_\infty + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}. \tag{4.3.8}$$

4.3.2 Specification of random variables and risk measures

I introduce uncertainty in the problem by allowing the source term f to be of the form

$$f(y, x) = y_1 \exp\left(-\frac{(x - y_2)^2}{y_3^2}\right), \tag{4.3.9}$$

where y_1, y_2 and y_3 are random variables uniformly distributed within prescribed intervals.

Furthermore, I substitute the constant advection coefficient a by a random variable y_4 also uniformly distributed.

Let $y = (y_1, y_2, y_3, y_4)$. Then the spatially discretized version of the problem (4.3.1)-(4.3.2) with uncertainty is

$$\min_{\mathbf{u},\mathbf{z}} \; \sigma(J(\mathbf{u}(y), \mathbf{z})) \tag{4.3.10a}$$

$$\text{s.t. } \mathbf{A}(y)\mathbf{u}(y) + \mathbf{B}(y)\mathbf{z} = \mathbf{b}(y), \quad y \in \Gamma, \tag{4.3.10b}$$

where σ is a risk measure to be specified later. Note that in my case $\mathbf{B}(y) \equiv \mathbf{B}$, but in general matrix $\mathbf{B}$ could depend on random parameter as well.

In the following, I define

$$\mathbf{u}(y, \mathbf{z}) = \mathbf{A}(y)^{-1}(\mathbf{b}(y) - \mathbf{B}(y)\mathbf{z}).$$

The random optimization problem (4.3.10) is then equivalent to

$$\min_{\mathbf{z}} \; \sigma(J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})). \tag{4.3.11}$$

with $J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z}) = \|[\mathbf{C}\mathbf{u}(\mathbf{z}) - \mathbf{v}]_+\|_\infty + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}$.

For the purposes of this example I consider the following risk measures:

1. Expected value, $\mathbb{E}[J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z})]$;
2. Worst case, $\sup_{y\in\Gamma}[J(\mathbf{u}(y,\mathbf{z}),\mathbf{z})]$;
3. Conditional value-at-risk, $\min_t\{t + \frac{1}{1-c}\,\mathbb{E}[J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z})) - t]_+\}$, $c \in (0,1)$;
4. Absolute semi-deviation, i.e., mean plus semi-deviation of order 1, $\mathbb{E}[J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z}) + c[J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z}) - \mathbb{E}[J(\mathbf{u}(\cdot,\mathbf{z}),\mathbf{z})]_+]$, $c \in [0,1]$.

The problem (4.3.11) with these risk measures is nonsmooth. Next, I will present reformulations of these problems as constrained quadratic programs. For these reformulations to hold, it is necessary that the weights are positive.

4.3.3 Reformulations

Let S denote the total number of scenarios (samples) and let w_j, $j = 1,\ldots,S$, be the weights corresponding to points $y^{(j)}$, $j = 1,\ldots,S$. I will use lower index for the components of the vector as in $\mathbf{u} = (u_1,\ldots,u_N)$, and upper index for the realizations of scenarios as in $\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(S)}$. Furthermore, the bar will denote a vector indexed by scenario index: $\bar{\epsilon} = (\epsilon^{(1)},\ldots,\epsilon^{(S)})$, $\bar{\mathbf{u}} = (\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(S)})$.

4.3.3.1 The deterministic optimization problem

The objective function in (4.3.8) is nonsmooth. I introduce an auxiliary variable $\delta \in \mathbb{R}$. The deterministic variant of the problem is

$$\min_{\mathbf{z},\delta}\ \delta + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z} \tag{4.3.12a}$$

$$\text{s.t.}\ \mathbf{C}\mathbf{u} - \mathbf{v} \le \delta\mathbf{e} \tag{4.3.12b}$$

$$\delta \ge 0, \tag{4.3.12c}$$

where $\mathbf{e} \in \mathbb{R}^p$ is the vector of all ones.

4.3.3.2 The stochastic optimization problem with expected value

The discretization of the stochastic optimization problem (4.3.11) with $\sigma(J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})) = \mathbb{E}[J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})]$ is given by

$$\min_{\mathbf{z}} \left(\sum_{j=1}^{S} w_j \| [\mathbf{C}\mathbf{u}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+ \|_\infty \right) + \frac{\alpha}{2} \mathbf{z}^T \mathbf{R} \mathbf{z}. \tag{4.3.13}$$

If the weights w_j are positive, I can introduce auxiliary variables $\delta^{(1)}, \ldots, \delta^{(S)} \in \mathbb{R}$ to reformulate (4.3.13) as the following constrained smooth problem

$$\min_{\mathbf{z}, \bar{\delta}} \sum_{j=1}^{S} w_j \delta^{(j)} + \frac{\alpha}{2} \mathbf{z}^T \mathbf{R} \mathbf{z} \tag{4.3.14a}$$

$$\text{s.t. } \mathbf{C}\mathbf{u}(y^{(j)}, \mathbf{z}) - \mathbf{v} \leq \delta^{(j)} \mathbf{e} \qquad \text{for } j = 1, \ldots, S \tag{4.3.14b}$$

$$\delta^{(j)} \geq 0 \qquad \text{for } j = 1, \ldots, S. \tag{4.3.14c}$$

4.3.3.3 The stochastic optimization problem with worst case

The discretization of the stochastic optimization problem (4.3.11) with $\sigma(J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})) = \sup_y J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})$ is given by

$$\min_{\mathbf{z}} \max_{j=1,\ldots,S} \| [\mathbf{C}\mathbf{u}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+ \|_\infty + \frac{\alpha}{2} \mathbf{z}^T \mathbf{R} \mathbf{z}. \tag{4.3.15}$$

It can be reformulated as a smooth constrained problem as follows:

$$\min_{\mathbf{z}, \delta} \delta + \frac{\alpha}{2} \mathbf{z}^T \mathbf{R} \mathbf{z} \tag{4.3.16a}$$

$$\text{s.t. } \mathbf{C}\mathbf{u}(y^{(j)}, \mathbf{z}) - \mathbf{v} \leq \delta \mathbf{e} \qquad \text{for } j = 1, \ldots, S \tag{4.3.16b}$$

$$\delta \geq 0. \tag{4.3.16c}$$

4.3.3.4 The stochastic optimization problem with CVaR

Let $\sigma(J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})) = \min_t \{t + \frac{1}{1-c} \mathbb{E}[J(\mathbf{u}(\cdot, \mathbf{z})) - t]_+\}$ with $c \in (0, 1)$. The discretization of the stochastic problem takes the form

$$\min_{\mathbf{z}, t} t + \frac{1}{1-c} \sum_{j=1}^{S} w_j [\| [\mathbf{C}\mathbf{u}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+ \|_\infty - t]_+ + \frac{\alpha}{2} \mathbf{z}^T \mathbf{R} \mathbf{z}. \tag{4.3.17}$$

If the weights w_j are positive, I can remove the outer $[\cdot]_+$ function above by reformulating the problem with auxiliary variables $\epsilon^{(1)}, \cdots, \epsilon^{(S)}$ and added constraints:

$$\begin{aligned}
&\min_{\mathbf{z},t} \ t + \frac{1}{1-c}\sum_{j=1}^{S} w_j \epsilon^{(j)} + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z} && && \text{(4.3.18a)}\\
&\text{s.t.} \ \|[\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty - t \le \epsilon^{(j)} && \text{for } j = 1, \ldots, S, && \text{(4.3.18b)}\\
&\qquad \epsilon^{(j)} \ge 0 && \text{for } j = 1, \ldots, S, && \text{(4.3.18c)}
\end{aligned}$$

By introducing auxiliary variables $\delta^{(j)}$, for $j = 1, \ldots, S$, I can remove the remaining non-smoothness by adding additiona constraints:

$$\begin{aligned}
&\min_{\mathbf{z},t} \ t + \frac{1}{1-c}\sum_{j=1}^{S} w_j \epsilon^{(j)} + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z} && && \text{(4.3.19a)}\\
&\text{s.t.} \ \delta^{(j)} - t \le \epsilon^{(j)} && \text{for } j = 1, \ldots, S, && \text{(4.3.19b)}\\
&\qquad \mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v} \le \delta^{(j)} && \text{for } j = 1, \ldots, S, && \text{(4.3.19c)}\\
&\qquad \epsilon^{(j)}, \delta^{(j)} \ge 0 && \text{for } j = 1, \ldots, S. && \text{(4.3.19d)}
\end{aligned}$$

4.3.3.5 The stochastic optimization problem with semi-deviation

Finally, let $\sigma(J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})) = \mathbb{E}[J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z}) + c[J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z}) - \mathbb{E}[J(\mathbf{u}(\cdot, \mathbf{z}), \mathbf{z})]_+]$.

The discretization of the stochastic optimization problem (4.3.11) is

$$\begin{aligned}
\min_{\mathbf{z}} \ & \left(\sum_{j=1}^{S} w_j \Big(\|[\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty \right)\\
& + c\left(\sum_{j=1}^{S} w_j \Big[\|[\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty - \sum_{k=1}^{S} w_k \|[\mathbf{Cu}(y^{(k)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty \Big]_+ \right)\\
& + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}. \qquad \text{(4.3.20)}
\end{aligned}$$

If the weights w_j are positive, I can define $\epsilon^{(1)}, \ldots, \epsilon^{(S)}$ to remove the $[\cdot]_+$ corresponding to the semideviation by adding constraints. The problem (4.3.20) is equivalent to

$$\min_{\mathbf{z},\epsilon} \ \Big(\sum_{j=1}^{S} w_j \|[\mathbf{Cu}(y^{(j)},\mathbf{z})-\mathbf{v}]_+\|_\infty\Big) + c\Big(\sum_{j=1}^{S} w_j \epsilon^{(j)}\Big) + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}, \tag{4.3.21a}$$

$$\text{s.t.} \ \ \|[\mathbf{Cu}(y^{(j)},\mathbf{z})-\mathbf{v}]_+\|_\infty - \sum_{k=1}^{S} w_k \|[\mathbf{Cu}(y^{(k)},\mathbf{z})-\mathbf{v}]_+\|_\infty \leq \epsilon^{(j)} \quad \text{for} \ \ j=1,\ldots,S, \tag{4.3.21b}$$

$$\epsilon^{(j)} \geq 0 \qquad \text{for} \ \ j=1,\ldots,S. \tag{4.3.21c}$$

Defining $\delta^{(j)} = \|[\mathbf{Cu}(y^{(j)},\mathbf{z})-\mathbf{v}]_+\|_\infty$, the previous problem can be written as

$$\min_{\mathbf{z},\epsilon,\delta} \ \Big(\sum_{j=1}^{S} w_j \delta^{(j)}\Big) + c\Big(\sum_{j=1}^{S} w_j \epsilon^{(j)}\Big) + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}, \tag{4.3.22a}$$

$$\text{s.t.} \ \ \delta^{(j)} - \sum_{k=1}^{S} w_k \delta_k \leq \epsilon^{(j)} \qquad \text{for} \ \ j=1,\ldots,S, \tag{4.3.22b}$$

$$\|[\mathbf{Cu}(y^{(j)},\mathbf{z})-\mathbf{v}]_+\|_\infty = \delta^{(j)} \qquad \text{for} \ \ j=1,\ldots,S, \tag{4.3.22c}$$

$$\epsilon^{(j)} \geq 0 \qquad \text{for} \ \ j=1,\ldots,S. \tag{4.3.22d}$$

If the equality in (4.3.22c) can be replaced by an inequality, the resulting problem can be formulated as a smooth quadratic programming problem. In fact, the problem

$$\min_{\mathbf{z},\epsilon,\delta} \ \Big(\sum_{j=1}^{S} w_j \delta^{(j)}\Big) + c\Big(\sum_{j=1}^{S} w_j \epsilon^{(j)}\Big) + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}, \tag{4.3.23a}$$

$$\text{s.t.} \ \ \delta^{(j)} - \sum_{k=1}^{S} w_k \delta^{(k)} \leq \epsilon^{(j)} \qquad \text{for} \ \ j=1,\ldots,S, \tag{4.3.23b}$$

$$\|[\mathbf{Cu}(y^{(j)},\mathbf{z})-\mathbf{v}]_+\|_\infty \leq \delta^{(j)} \qquad \text{for} \ \ j=1,\ldots,S, \tag{4.3.23c}$$

$$\epsilon^{(j)} \geq 0 \qquad \text{for} \ \ j=1,\ldots,S, \tag{4.3.23d}$$

is equivalent to

$$\min_{\mathbf{z},\epsilon,\delta} \ \Big(\sum_{j=1}^{S} w_j \delta^{(j)}\Big) + c\Big(\sum_{j=1}^{S} w_j \epsilon^{(j)}\Big) + \frac{\alpha}{2}\mathbf{z}^T\mathbf{R}\mathbf{z}, \tag{4.3.24a}$$

$$\text{s.t. } \delta^{(j)} - \sum_{k=1}^{S} w_k \delta_k \le \epsilon^{(j)} \qquad \text{for } j = 1, \dots, S, \tag{4.3.24b}$$

$$\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v} \le \delta^{(j)}\mathbf{e} \qquad \text{for } j = 1, \dots, S, \tag{4.3.24c}$$

$$\epsilon^{(j)}, \delta^{(j)} \ge 0 \qquad \text{for } j = 1, \dots, S. \tag{4.3.24d}$$

Lemma 4.3.1 *If the weights w_j satisfy $w_j \in (0, 1)$ and $\sum_{j=1}^{S} w_j = 1$, then the optimization problems* (4.3.22) *and* (4.3.23) *are equivalent.*

Proof: The feasible set of (4.3.22) is contained in that of (4.3.23). Therefore, we must show that at a solution of (4.3.23), $\|[\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty = \delta^{(j)}$ for all $j = 1, \dots, S$.

Let $\mathbf{z}, \bar{\epsilon}, \bar{\delta}$ solve (4.3.23) and suppose that $\|[\mathbf{Cu}(y^{(j)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty < \delta^{(j)}$ for some j. Without loss of generality let $j = 1$. Define $\tilde{\delta}^{(1)} = \|[\mathbf{Cu}(y^{(1)}, \mathbf{z}) - \mathbf{v}]_+\|_\infty < \delta^{(1)}$ and $\tilde{\delta}^{(2)} = \delta^{(2)}, \dots, \tilde{\delta}^{(S)} = \delta^{(S)}$. I will generate $\tilde{\epsilon}^{(1)}, \dots, \tilde{\epsilon}^{(S)}$ such that $\mathbf{z}, \bar{\tilde{\epsilon}} = (\tilde{\epsilon}^{(1)}, \dots, \tilde{\epsilon}^{(S)})^T, \bar{\tilde{\delta}} = (\tilde{\delta}^{(1)}, \dots, \tilde{\delta}^{(S)})^T$ are feasible for (4.3.23) and have a lower objective function value than that at $\mathbf{z}, \bar{\epsilon}, \bar{\delta}$, which contradicts that $\mathbf{z}, \bar{\epsilon}, \bar{\delta}$ solves (4.3.23).

Define $\tilde{\epsilon}^{(1)} := \epsilon^{(1)}$. The variables $\bar{\tilde{\delta}}$, $\tilde{\epsilon}^{(1)}$ satisfy the constraint (4.3.23b) with $j = 1$ since

$$\tilde{\delta}^{(1)} - \sum_{k=1}^{S} w_k \tilde{\delta}^{(k)} = (1 - w_1)(\delta^{(1)} - \tilde{\delta}^{(1)}) + \delta^{(1)} - \sum_{k=1}^{S} w_k \delta^{(k)} < \delta^{(1)} - \sum_{k=1}^{S} w_k \delta^{(k)} \le \epsilon^{(1)}$$

due to the assumption $w_1 \in (0, 1)$. Define $\tilde{\epsilon}^{(j)} := \epsilon^{(j)} + w_1(\delta^{(1)} - \tilde{\delta}^{(1)})$ for $j > 1$. The variables $\bar{\tilde{\delta}}$, $\tilde{\epsilon}^{(j)}$ satisfy the constraint (4.3.23b) with $j > 1$ since

$$\tilde{\delta}^{(j)} - \sum_{k=1}^{S} w_k \tilde{\delta}^{(k)} = \delta^{(j)} - \sum_{k=1}^{S} w_k \delta^{(k)} + w_1(\delta^{(1)} - \tilde{\delta}^{(1)}) \le \epsilon^{(j)} + w_1(\delta^{(1)} - \tilde{\delta}^{(1)}) = \tilde{\epsilon}^{(j)}.$$

The variables $\bar{\tilde{\delta}}$, $\bar{\tilde{\epsilon}}$ satisfy the constraint (4.3.23c,d).

Consider the value of the objective at $\bar{\tilde{\delta}}$, $\bar{\tilde{\epsilon}}$:

$$
\begin{aligned}
&\sum_{j=1}^{S} w_j \tilde{\delta}^{(j)} + c \sum_{j=1}^{S} w_j \tilde{\epsilon}^{(j)} \\
&= \sum_{j=1}^{S} w_j \delta^{(j)} - w_1(\delta^{(1)} - \tilde{\delta}^{(1)}) + c \sum_{j=1}^{S} w_j \epsilon^{(j)} + c \sum_{j=2}^{S} w_j w_1 (\delta^{(1)} - \tilde{\delta}^{(1)}) \\
&= \sum_{j=1}^{S} w_j \delta^{(j)} + c \sum_{j=1}^{S} w_j \epsilon^{(j)} - w_1(\delta^{(1)} - \tilde{\delta}^{(1)}) \left[1 - c \sum_{j=2}^{S} w_j \right] \\
&< \sum_{j=1}^{S} w_j \delta^{(j)} + c \sum_{j=1}^{S} w_j \epsilon^{(j)}
\end{aligned}
$$

since $\delta^{(1)} > \tilde{\delta}^{(1)}$ and $c \sum\limits_{j=2}^{S} w_j < 1$ (recall $c \in [0, 1]$).

Hence, the solution $(\mathbf{z}, \bar{\delta}, \bar{\epsilon})$ is not optimal, which leads to a contradiction.

□

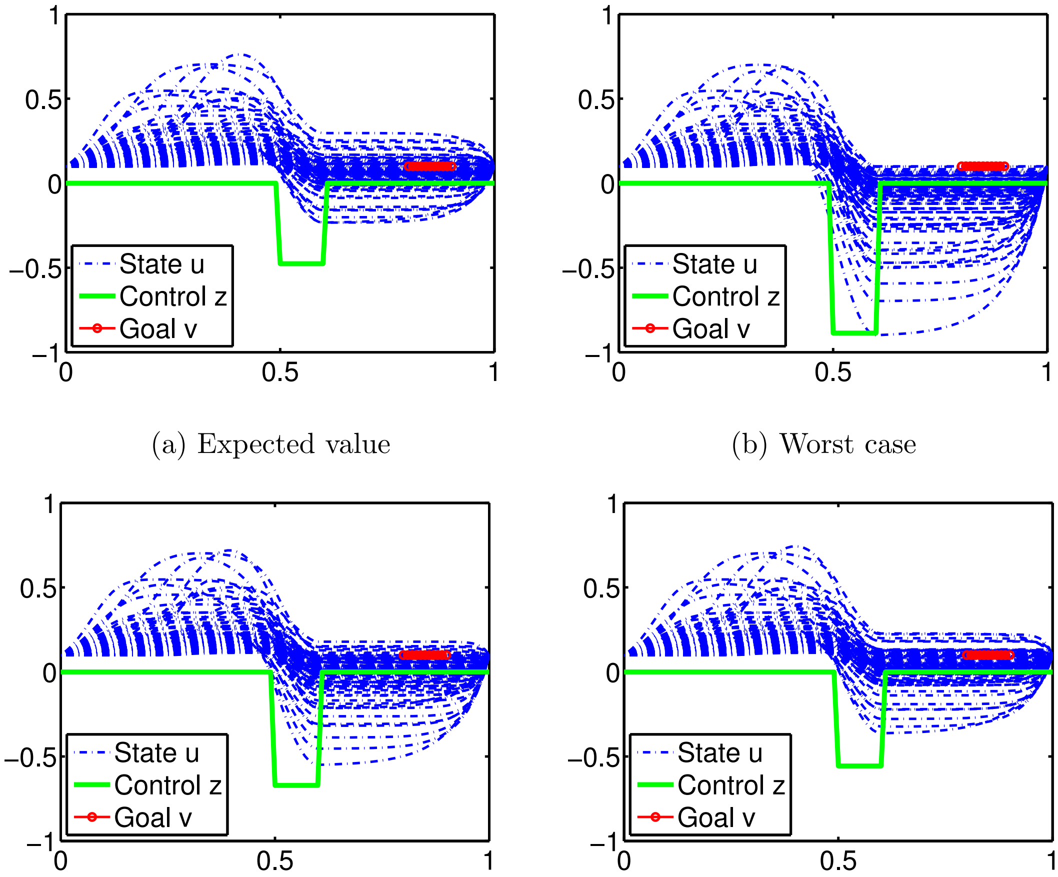
Thus, we proceed with the reformulation given by (4.3.24).

4.3.4 Results

I present here the results of solving problem (4.3.11) with objective given by (4.3.7). These results help demonstrate the differences between various risk measures.

In order to generate realizations of 4-dimensional random variable y I use quasi-Monte Carlo (Sobol' sequence) approach. Since with QMC all the weights $w_j = 1/S$ are positive (where S is the total number of scenarios (realizations)), I am able to use the reformulations that were introduced earlier. I solve the reformulations of problem (4.3.11) with the four risk measures using MATLAB's `quadprog` solver.

For the problems with uncertainty the results are presented in figure 4.15. Here I have used 100 realizations of y in order to approximate the objective of (4.3.11). The results demonstrate the differences between various risk measures. For the expected value the magnitude of the control variable is the smallest with many realizations exceeding the target value $\mathbf{v}$ in the region of interest. For the sup measure, as expected, all of the realizations lie below the target value. The price for it is the larger value of the control $\mathbf{z}$. The other two risk measures produce results in-between these two extreme cases with CVaR being more conservative (recall that for c close to 1 CVaR behaves almost as sup) and mean plus semi-deviation being closer to the expected value.



(a) Expected value

(b) Worst case

(c) CVaR ($c = 0.8$)

(d) Absolute semi-deviation ($c = 1$)

Figure 4.15: Results of solving problem (4.3.11) with uncertainty using reformulations (4.3.14),(4.3.16),(4.3.19) and (4.3.24). Here plotted 100 realizations of the state $\mathbf{u}$, the deterministic optimal control $\mathbf{z}$ and the target $\mathbf{v}$. The control $\mathbf{z}$ is scaled by 1/2.

Chapter 5

Conclusions

In this thesis I have reviewed several topics that are of importance for optimization under uncertainty. The first is the formulation of the optimization problems governed by PDEs with random inputs. Such problems arise in many areas of application, and growing interest in them motivates a lot of research ([73, 35, 42, 69, 43]). I have provided a short overview of the risk measures that can be used in the context of optimization under uncertainty ([65]). The application of these risk measures to the problems with high-dimensional random parameters motivated the core topic of this thesis - multivariate integration methods.

I have provided the overview of the high-dimensional integration methods starting from Monte Carlo and continuing onto sparse grids methods. Paying particular attention to the assumptions on the integrands I have summarized the main convergence properties of the methods considered.

Finally, I have tested the performance of various integration methods on several model problems. The results indicate that among the methods considered MC type methods have an edge over the sparse grids for the problems with high dimensionality and non-smoothness. In particular, quasi-Monte Carlo methods appear to be a promising direction for future research. Although their performance also deteriorates with increasing dimension and loss of smoothness ([11]), the effect is less obvious than for the sparse grids. In addition, MC type methods have an advantage of having all positive weights which is favorable for optimization.

On the other hand, when the integrand is smooth the sparse grids allow to fully exploit its smoothness. In addition, hierarchical sparse grids based on local polynomials provide a natural framework for adaptivity allowing to detect more important dimensions of the integrand and focus on the regions of irregular behavior. Unfortunately, for high-dimensional problems with non-smoothness without careful problem-dependent selection of the refinement criteria hierarchical sparse grids spend still spend too many points making them ineffective and inapplicable. Some attempts to mitigate this drawback have been made recently, such as high-dimensional discontinuity detection [84], which might be worth exploring. Another potential direction for exploration is the relation between sparse grids and tensor representations ([32, 27]).

Bibliography

[1] Ph. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Math. Finance*, 9(3):203–228, 1999.

[2] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Rev.*, 52(2):317–355, 2010.

[3] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.*, 12(4):273–288, 2000. Multivariate polynomial interpolation.

[4] R. E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[5] H.-J. Bungartz. Concepts for higher order finite elements on sparse grids. In A. V. Ilin and L. R. Scott, editors, *Houston Journal of Mathematics: Proceedings of the Third International Conference on Spectral and High Order Methods*, pages 159–170, 1996.

[6] H.-J. Bungartz. *Finite Elements of Higher Order on Sparse Grids.* Shaker Verlag, Aachen, Germany, 1998. Habilitationsschrift, Fakultät für Informatik, Technische Universität München.

[7] H.-J. Bungartz and S. Dirnstorfer. Multivariate quadrature on adaptive sparse grids. *Computing*, 71(1):89–114, 2003.

[8] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[9] J. Burkardt. FAURE_DATASET generate Faure datasets, 2009. http://people.sc.fsu.edu/∼jburkardt/m_src/faure_dataset/faure_dataset.html (accessed Aug. 31, 2015).

[10] J. Burkardt. SOBOL_DATASET generate Sobol datasets, 2009. http://people.sc.fsu.edu/∼jburkardt/m_src/sobol_dataset/sobol_dataset.html (accessed Aug. 31, 2015).

[11] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1 1998.

[12] C. H. Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Math. Programming*, 71(1, Ser. A):51–69, 1995.

[13] C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numer. Math.*, 2:197–205, 1960.

[14] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.*, 14(1):3–15, 2011.

[15] P. J. Davis and D. P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, Boston, San Diego, New York, London, 1984.

[16] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

[17] Chr. Feuersänger. *Sparse Grid Methods for Higher Dimensional Approximation.* Dissertation, Institut für Numerische Simulation, Universität Bonn, September 2010.

[18] P. Frauenfelder, C. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005.

[19] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *J. Comput. Phys.*, 225(1):652–685, 2007.

[20] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, 18(3-4):209–232, 1998.

[21] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71(1):65–87, 2003.

[22] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach.* Springer-Verlag, New York, 1991.

[23] R. G. Ghanem and P. D. Spanos. Spectral techniques for stochastic finite elements. *Arch. Comput. Methods Engrg.*, 4(1):63–100, 1997.

[24] M. B. Giles. Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

[25] M. B. Giles. Multilevel Monte Carlo methods. In *Monte Carlo and quasi-Monte Carlo methods 2012*, volume 65 of *Springer Proc. Math. Stat.*, pages 83–103. Springer, Heidelberg, 2013.

[26] M. B. Giles, F. Y. Kuo, I. H. Sloan, and B. J. Waterhouse. Quasi-Monte Carlo for finance applications. *ANZIAM J.*, 50((C)):C308–C323, 2008.

[27] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitt.*, 36(1):53–78, 2013.

[28] M. Griebel. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing*, 61(2):151–179, 1998.

[29] M. Griebel, F. Y. Kuo, and I. H. Sloan. The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition. *Math. Comp.*, 82(281):383–400, 2013.

[30] M. Gunzburger, C. G. Webster, and G. Zhang. An adaptive wavelet stochastic collocation method for irregular solutions of stochastic partial differential equations. Technical Report ORNL/TM-2012/186, Oak Ridge National Laboratory, 2012.

[31] W. Hackbusch. *Multigrid Methods and Applications.* Computational Mathematics, Vol. 4. Springer–Verlag, Berlin, 1985.

[32] W. Hackbusch. Numerical tensor calculus. *Acta Numerica*, 23:651–742, 5 2014.

[33] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[34] J. M. Hammersley and D. C. Handscomb. *Monte Carlo methods.* Methuen & Co., Ltd., London; Barnes & Noble, Inc., New York, 1965.

[35] M. Heinkenschloss and D. P. Kouri. Stochastic collocation for optimization problems governed by elliptic PDEs with uncertain coefficients. Technical Report TR12–xx, Department of Computational and Applied Mathematics, Rice University, 2012. in preparation.

[36] S. Heinrich. Multilevel Monte Carlo methods. In S. Margenov, J. Waśniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, volume 2179 of *Lecture Notes in Computer Science*, pages 58–67. Springer Berlin Heidelberg, 2001.

[37] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with Partial Differential Equations*, volume 23 of *Mathematical Modelling, Theory and Applications.* Springer Verlag, Heidelberg, New York, Berlin, 2009.

[38] J. D. Jakeman and S. G. Roberts. Local and dimension adaptive stochastic collocation for uncertainty quantification. In J. Garcke and M. Griebel, editors, *Sparse Grids and Applications*, pages 181–203. Springer, 2013.

[39] K. Karhunen. Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, 1947(37):79, 1947.

[40] A. Klimke and B. Wohlmuth. Algorithm 847: spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Trans. Math. Software*, 31(4):561–579, 2005.

[41] D. P. Kouri. Optimization governed by stochastic partial differential equations. Master's thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2010. Available as CAAM TR10-20.

[42] D. P. Kouri. *An Approach for the Adaptive Solution of Optimization Problems Governed by Partial Differential Equations with Uncertain Coefficients.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, May 2012. Available as CAAM TR12-10.

[43] D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders. A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. *SIAM Journal on Scientific Computing*, 35(4):A1847–A1879, 2013.

[44] A. S. Kronrod. *Nodes and weights of quadrature formulas. Sixteen-place tables.* Authorized translation from the Russian. Consultants Bureau, New York, 1965.

[45] L. Kuipers and H. Niederreiter. *Uniform distribution of sequences.* Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974. Pure and Applied Mathematics.

[46] F. Y. Kuo, C. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50(6):3351–3374, 2012.

[47] F. Y. Kuo, C. Schwab, and I. H. Sloan. Multi-level quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *Foundations of Computational Mathematics*, 15(2):411–449, 2015.

[48] P. D. Lax. *Functional Analysis.* John Wiley & Sons, New-York, Chicester, Brisbane, Toronto, 2002.

[49] C. Lemieux. *Monte Carlo and quasi-Monte Carlo sampling.* Springer Series in Statistics. Springer, New York, 2009.

[50] M. Loève. Fonctions aléatoires de second ordre. *Revue Sci.*, 84:195–206, 1946.

[51] G. J. Lord, C. E. Powell, and T. Shardlow. *An introduction to computational stochastic PDEs.* Cambridge Texts in Applied Mathematics. Cambridge University Press, New York, 2014.

[52] X. Ma and N. Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.*, 228(8):3084–3113, 2009.

[53] W. J. Morokoff and R. E. Caflisch. Quasi-Monte Carlo integration. *J. Comput. Phys.*, 122(2):218–230, 1995.

[54] H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bull. Amer. Math. Soc.*, 84(6):957–1041, 1978.

[55] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[56] F. Nobile, R. Tempone, and C. G. Webster. The analysis of a sparse grid stochastic collocation method for partial differential equations with high-dimensional random input data. Technical Report SAND2007–8093, Sandia National Laboratories, 2007.

[57] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[58] E. Novak and K. Ritter. High-dimensional integration of smooth functions over cubes. *Numer. Math.*, 75(1):79–97, 1996.

[59] E. Novak and K. Ritter. The curse of dimension and a universal method for numerical integration. In G. Nürnberger, J. W. Schmidt, and G. Walz, editors, *Multivariate approximation and splines. Papers from the International Conference held in Mannheim, September 7–10, 1996*, volume 125 of *Internat. Ser. Numer. Math.*, pages 177–187. Birkhäuser, Basel, 1997.

[60] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constr. Approx.*, 15(4):499–522, 1999.

[61] B. Øksendal. *Stochastic differential equations.* Universitext. Springer-Verlag, Berlin, sixth edition, 2003. An introduction with applications.

[62] T. N. L. Patterson. The optimum addition of points to quadrature formulae. *Math. Comp. 22 (1968), 847–856; addendum, ibid.*, 22(104, loose microfiche supp.):C1–C11, 1968.

[63] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems.* PhD thesis, Technical University Munich, 2010.

[64] D. Pflüger, B. Peherstorfer, and H.-J. Bungartz. Spatially adaptive sparse grids for high-dimensional data-driven problems. *J. Complexity*, 26(5):508–522, 2010.

[65] R. T. Rockafellar. Coherent approaches to risk in optimization under uncertainty. *Tutorials in Operations Research INFORMS*, pages 38–61, 2007.

[66] R. T. Rockafellar and J. O. Royset. Engineering decisions under risk averseness. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 0(0):04015003, 0.

[67] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(2):21–41, 2000.

[68] R. T. Rockafellar and S. Uryasev. The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys in Operations Research and Management Science*, 18(12):33 – 53, 2013.

[69] E. Rosseel and G. N. Wells. Optimal control with stochastic pde constraints and uncertain controls. *Computer Methods in Applied Mechanics and Engineering*, 213-216(0):152 – 167, 2012.

[70] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Math. Oper. Res.*, 31(3):433–452, 2006. see also [72].

[71] A. Ruszczyński and A. Shapiro. Optimization of risk measures. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design Under Uncertainty*, pages 119–157, London, 2006. Springer Verlag.

[72] A. Ruszczyński and A. Shapiro. Corrigendum to: "Optimization of convex risk functions" [Math. Oper. Res. **31** (2006), no. 3, 433–452]. *Math. Oper. Res.*, 32(2):496, 2007.

[73] V. Schulz and C. Schillings. On the nature and treatment of uncertainties in aerodynamic design. *AIAA Journal*, 47(3):646–654, 2009.

[74] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming. Modeling and theory*, volume 9 of *MPS/SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2009.

[75] R. C. Smith. *Uncertainty quantification. Theory, implementation, and applications*, volume 12 of *Computational Science & Engineering*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[76] S. A. Smoljak. Quadrature and interpolation formulae on tensor products of certain function classes. *Dokl. Akad. Nauk SSSR*, 148:1042–1045, 1963. Russian.

[77] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer Verlag, New York, Berlin, Heidelberg, London, Paris, second edition, 1993.

[78] M. Stoyanov. User manual: Tasmanian sparse grids v2.0. Technical report, Oak Ridge National Laboratory, Computer Science and Mathematics Division, One Bethel Valley Road, P.O. Box 2008, MS-6367, Oak Ridge, TN 37831-6164, August 2013. http://tasmanian.ornl.gov (accessed Decs. 8, 2014).

[79] M. K. Stoyanov. Hierarchy-direction selective approach for locally adaptive sparse grids. Technical Report ORNL/TM-2013/384, Oak Ridge National Laboratory, 2013.

[80] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546 (electronic), 1998.

[81] G. W. Wasilkowski and H. Woźniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *J. Complexity*, 11(1):1–56, 1995.

[82] N. Wiener. The homogeneous chaos. *Amer. J. Math.*, 60:897–938, 1938.

[83] C. Zenger. Sparse grids. In *Parallel algorithms for partial differential equations (Kiel, 1990)*, volume 31 of *Notes Numer. Fluid Mech.*, pages 241–251. Vieweg, Braunschweig, 1991.

[84] G. Zhang, C. G. Webster, M. Gunzburger, and J. Burkardt. A hyper-spherical adaptive sparse-grid method for high-dimensional discontinuity detection. Technical Report ORNL/TM-2014/36, Oak Ridge National Laboratory (ORNL), 2014.