Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch–Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich–Wilhelms–Universität Bonn

vorgelegt von

Markus Holtz

aus

Bonn

Bonn 2008

Angefertigt mit Genehmigung der Mathematisch–Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachter: Prof. Dr. Michael Griebel

2. Gutachter: Priv.-Doz. Dr. Thomas Gerstner

Tag der Promotion: 19.12.2008

Zusammenfassung

Wir beschäftigen uns in der vorliegenden Arbeit mit der effizienten numerischen Berechnung hochdimensionaler Integrale, die in Anwendungen aus dem Finanz- und Versicherungsbereich auftreten. Klassische Produkt-Quadraturverfahren können wegen ihres Aufwands, der exponentiell mit der Dimension steigt, für solche Probleme nicht mehr eingesetzt werden. Die Berechnung solcher Integrale erfolgt daher meistens mit Monte Carlo Methoden, deren Konvergenz unabhängig von der Dimension aber relativ langsam ist.

In dieser Arbeit entwickeln wir eine neue allgemeine Klasse von Quadraturverfahren zur Berechnung hochdimensionaler Integrale. Unsere Methoden profitieren direkt von vorhandener Glattheit und niedriger effektiver Dimension des Integranden. Zudem erhält unsere allgemeine Verfahrensklasse die Klasse der bekannten Dünngitter-Quadraturverfahren (DG-Verfahren) als Spezialfall.

Zur Konstruktion unserer Verfahren bestimmen wir die wichtigsten Terme der Anker-ANOVA Zerlegung des Integranden mit Hilfe eines dimensions-adaptiven Fehlerschätzers. Dann integrieren wir die niederdimensionalen Terme unter Verwendung klassischer Quadraturverfahren. Wir analysieren die auftretenden Modellierungs- und Diskretisierungsfehler und zeigen, dass für Funktionen mit niedriger effektiver Dimension auf diese Weise der Fluch der Dimension gebrochen werden kann. Weiterhin bestimmen wir optimale DG-Verfahren für Funktionen aus gewichteten Sobolevräumen und leiten eine neue Schranke für die ε-Kostenkomplexität dieser Verfahren her.

Außerdem verbessern wir existierende DG-Verfahren durch die Wahl der zugrundeliegenden eindimensionalen Quadraturformel und durch orthogonale Koordinatentransformationen. Auf diese Weise können wir vorhandene Glattheit und niedrige effektive Dimension auf optimale Weise nutzen. Anhand verschiedener Anwendungsprobleme aus dem Finanzbereich, die zu glatten Integranden führen, zeigen wir, dass diese neuen Verfahren selbst in hunderten von Dimensionen Monte Carlo und anderen existierenden Ansätzen um mehrere Größenordnungen überlegen sind.

Desweiteren diskutieren wir verschiedene Lösungsansätze für die Problematik, dass viele Anwendungen nicht die hohen Glattheitsvoraussetzungen von DG-Verfahren erfüllen. Wir betrachten Gebietszerlegungsmethoden und präsentieren neue dimensions-adaptive Quadraturverfahren, welche geringe Regularität mit Hilfe von lokaler Adaptivität in den niederdimensionalen Anker-ANOVA Termen behandeln. Wir demonstrieren die Effizienz dieser Ansätze anhand der Bewertung von Asiatischen Optionen, Barrier Optionen und insbesondere von performance-abhängigen Optionen.

Schließlich betrachten wir die Simulation stochastischer Asset-Liability Management Modelle im Lebensversicherungsbereich als ein Anwendungsgebiet, das in den letzten Jahren zunehmend an Bedeutung gewonnen hat und zu besonders komplexen

Integralen führt, die bisher nur mit unbefriedigend langen Rechenzeiten lösbar sind. Unsere numerischen Tests demonstrieren den Vorteil unserer Ansätze im Vergleich zu Monte Carlo Methoden auch für diese Anwendung.

Abschließend analysieren wir die effektive Dimension von verschiedenen Integralen aus dem Finanz- und Versicherungsbereich und erklären, wie das Konvergenzverhalten von DG- und anderen numerischen Verfahren von diesen Dimensionen abhängt. Insgesamt liefern wir auf diese Weise neue Erkenntnisse zum Zusammenspiel von DG-Verfahren, Koordinatentransformationen und effektiven Dimensionen.

Danksagung

An dieser Stelle möchte ich allen danken, die mich bei dieser Arbeit unterstützt haben; allen voran meinem Betreuer Prof. Dr. Michael Griebel. Ihm danke ich für die interessante und praxisrelevante Promotionsthematik, viele wertvolle Anregungen, Ratschläge und Diskussionen, aber auch für das exzellente Arbeitsumfeld. Weiter bedanke ich mich herzlich bei Priv.-Doz. Dr. Thomas Gerstner für die vielen fachlichen Gespräche, hilfreichen Kommentare und für die Übernahme des Zweitgutachtens. Für die gute Zusammenarbeit bedanke ich mich außerdem bei meinen Co-Authoren Ralf Goschnick und Marcus Haep von der Zürich Gruppe Deutschland und bei Prof. Dr. Ralf Korn von der Universität Kaiserslautern. Mein Dank gilt auch den Studenten Niclas Doll, Alexander Hullmann, Christina Kürten, Jens Oettershagen, Tim Osadnik und Eva-Maria Rau, die zu einigen Teilen dieser Arbeit beigetragen haben. Abschließend möchte ich meinen Kollegen am Institut für Numerische Simulation, insbesondere meinem ehemaligen Zimmerkollegen Bram Metsch, für das angenehme and kollegiale Arbeitsklima danken.

Diese Arbeit wurde finanziell durch das Bundesministerium für Bildung und Forschung (BMBF) im Rahmen des Forschungsprojektes „Numerische Simulation für Asset/Liability Management im Versicherungswesen“ gefördert, wofür ich mich ebenfalls bedanke.

Bonn, im November 2008 *Markus Holtz*

Contents

Chapter 1
Introduction

Financial institutions have to understand the risks that their financial instruments create as precisely as possible. To this end, quantitative, mathematical models are developed which are usually based on tools from stochastic calculus. Most of the models are too complex to be analytically tractable and are hence analysed with the help of computer simulations which rely on efficient algorithms from scientific computing.

An important example is the quantitative, i.e. model-based, pricing of financial derivatives. Derivatives are financial instruments whose values are derived from the value of one or several underlying assets such as stocks, interest rates or commodities. A fundamental result from mathematical finance is that, under certain model assumptions, the prices of derivatives can be represented as expected values which in turn correspond to high-dimensional integrals

$$I_d = \int_{\mathbb{R}^d} g(\mathbf{z})\, \varphi(\mathbf{z})\; d\mathbf{z} \tag{1.1}$$

over the d-dimensional Euclidean space with the Gaussian weight function φ and $\mathbf{z} := (z_1, \ldots, z_d)$ or, after a suitable transformation, to high-dimensional integrals

$$I_d = \int_{[0,1]^d} f(\mathbf{x})\; d\mathbf{x} \tag{1.2}$$

over the unit cube. The dimension d depends on the number of sources of uncertainty respected by the model assumptions and is the larger the more random variables are involved. This way, high-dimensional integrals in hundreds of variables appear in many applications from finance. Since the integrals can in most cases not be calculated analytically, they have to be computed numerically up to a prescribed accuracy ε. Today, almost one fifth of the most powerful commercially available computer systems worldwide is owned by financial institutions and used for such purposes [111]. For comparison note that this share has doubled in the last two years and that there are by now more supercomputers working for finance than for, e.g., climate research, defense or geophysics.

For an efficient computation of high-dimensional integrals, one of the key prerequisites is that the *curse of dimension* [9] can be avoided at least to some extent. The curse of dimension states that the cost to compute an approximation with a prescribed accuracy ε depends exponentially on the dimension d of the problem. This is one of the main obstacles for a conventional numerical treatment of high

dimensional problems. Classical product quadrature methods for the computation of multivariate integrals [28] achieve with n evaluations of the integrand an accuracy of

$$\varepsilon(n) = O(n^{-r/d})$$

for functions with bounded derivatives up to order r. For fixed r, their convergence rates r/d thus deteriorate with increasing dimension and are already in moderate dimensions so small that high accuracies can no longer be obtained in practise. On the positive side, the case $r \sim d$ indicates that the problem of a high dimension can sometimes be compensated by, e.g., a high degree of smoothness. Also other aspects such as the concentration of measure phenomenon[1] or the superposition theorem of Kolmogorov[2] show that there is some chance to treat high-dimensional problems despite the curse of dimension. Furthermore, it is known from numerical complexity theory [157] that some algorithm classes can break the curse of dimension for certain function classes.

Randomised methods, so called *Monte Carlo methods*, are the most well-known examples of such classes of algorithms. Here, the integrand is approximated by the average of n function values at random points. Monte Carlo methods were first introduced to derivative pricing by Boyle [12] and are today the workhorses in the financial industry in particular for complex problems which depend on many variables. For square integrable functions f, the expected mean square error of the Monte Carlo method with n samples is

$$\varepsilon(n) = O(n^{-1/2}). \tag{1.3}$$

The convergence rate is thus independent of the dimension d, but quite low and a high accuracy is only achievable with a tremendously high number n of function evaluations. This slow convergence of the Monte Carlo method is one of the main reasons for the enormous need of the financial industry for computer resources.

Under more restrictive assumptions on the smoothness of the integrand it can be shown that faster rates of convergence can be attained by deterministic integration methods such as quasi-Monte Carlo methods [59, 119] and sparse grid methods [18, 150, 170]. *Quasi-Monte Carlo methods* are number theoretic algorithms which approximate the integral by the average of n function values at deterministic, uniformly distributed points. For integrands f of bounded variation, their ε-cost complexity [119] is

$$\varepsilon(n) = O(n^{-1}(\log n)^d). \tag{1.4}$$

They thus converge with a rate of almost one, almost independent of the dimension and almost half an order faster than the Monte Carlo method. *Sparse grid methods*

[1] The concentration of measure phenomenon [102] says that every Lipschitz function on a sufficiently high dimensional domain is well approximated by a *constant* function.

[2] The theorem of Kolmogorov [95] shows that every continuous function of several variables can be represented by the superposition of continuous functions that depend on only *one* variable.

are deterministic methods based on polynomial exactness, which are constructed using certain combinations of tensor products of one-dimensional quadrature rules.[3] In their simplest form [150], they have the ε-cost complexity

$$\varepsilon(n) = O(n^{-r}(\log n)^{(d-1)(r+1)}) \tag{1.5}$$

for all integrands which have bounded mixed partial derivatives of order r. Their convergence rate is almost independent of the dimension and increases with higher smoothness r of the integrand. For analytic functions even spectral convergence is observed.

A difficulty in higher dimensions is that quasi-Monte Carlo and sparse grid methods still depend on the dimension d through the logarithmic terms in (1.4) and (1.5). Furthermore, the implicit constants in (1.4) and (1.5) depend on d and often grow exponentially with d. Moreover, it is known from numerical complexity theory that many classes of integration problems are intractable [157] with respect to these deterministic methods meaning that even the best quasi-Monte Carlo or the best sparse grid algorithm can not completely avoid the curse of dimension. For a large dimension d and a small or moderate number n of sample points, the asymptotic advantage of the deterministic numerical methods over the Monte Carlo method might thus not pay off.

Nevertheless, integrals from practise are often in different or smaller problem classes and thus may be tractable. Paskov and Traub [134] indeed observed in 1995 that quasi-Monte Carlo methods converge nearly independent of the dimension and faster than Monte Carlo for a 360-dimensional integration problem which was given to them by the investment bank Goldman Sachs. This empirical observation indicated that the long computing times required by the Monte Carlo method may be avoided by deterministic integration methods even in high dimensions. It initiated intensive research to generalise the result of Paskov and Traub to wider classes of problems and to explain the success of the quasi-Monte Carlo method despite the high dimension. In the following years also sparse grid methods were successfully applied to this particular integration problem [48, 49, 123, 135]. These methods were also clearly superior to Monte Carlo, showing a similar efficiency as quasi-Monte Carlo.

One explanation for the success of the deterministic quadrature methods, which is by now widely accepted, is based on the *analysis of variance* (ANOVA) representation of the integrand [19]. There, the function f on $\mathbb{R}^d$ is decomposed into a sum of 2^d terms, where each term describes the relative importance of a subset of variables with respect to the total variance of f. It turned out that for most inte-

[3] We refer with sparse grid methods to the generalised sparse grid approach [48, 71, 137, 165]. While the classical sparse grid method [150] is uniquely determined by the choice of the underlying univariate quadrature rule, generalised sparse grid methods leave in addition the choice of an underlying index set open.

grands from finance the importance of each variable is naturally weighted by certain hidden weights. With increasing dimension, the lower-order terms of the ANOVA decomposition continue to play a significant role, whereas the higher-order terms tend to be negligible [19, 161]. Moreover, often coordinate transformations (usually interpreted as path generating methods), such as the Brownian bridge [117], can be used to exploit the underlying special structure of the problems from finance and to enforce the importance of the leading dimensions in this way. It is known by now that most high-dimensional integrands from finance are of *low effective dimension*, or can be transformed to be of low effective dimension, in the sense that they can be well approximated by a sum of functions which depend on only few variables.

The relation between the effective dimension and the performance of quasi-Monte Carlo methods has been investigated intensively in recent years. While the effective dimension has no impact on Monte Carlo methods, quasi-Monte Carlo methods profit from low effective dimensions, since low dimensional projections of their points are especially well-distributed and since their points are usually better uniformly distributed in smaller dimensions than in higher ones.

Classical sparse grid methods can not utilize low effective dimensions. However, *dimension-adaptive sparse grid methods*, as recently introduced by Gerstner and Griebel [49], take advantage of low effective dimensions in a very general and automatic way by a dimension-adaptive grid refinement.

Sloan and Woźniakowski introduced in [148] *weighted* Sobolev spaces of functions with bounded mixed regularity and proved that there exist quasi-Monte Carlo methods which can avoid the curse of dimension in such spaces, provided the function space weights decay sufficiently fast with growing dimension. Their results were generalised, also to sparse grid methods [124], and complemented by constructive approaches, e.g. the CBC-construction of lattice rules [149], in a series of papers and are still in the focus of active research.

Today many different settings are known in which integration is tractable with respect to quasi-Monte Carlo and sparse grid methods [124]. However, these settings usually do not apply to applications, since most of them do not satisfy the smoothness assumptions on bounded mixed regularity. This issue was recently addressed in [64, 106] with the argument that the lower order terms in the ANOVA decomposition are in certain cases smoother than the original function. Since the higher order terms are small because of low effective dimension, this may explain the fast convergence of the deterministic methods despite the low regularity of the application problems.

This thesis deals with the numerical analysis and with the efficient numerical treatment of high-dimensional integration problems from finance and insurance. It departs from most of the existing literature in the following two ways:

- We mostly address the arising integrals directly on $\mathbb{R}^d$ and avoid transformations[4] to the unit cube.
- We base our numerical analysis and our numerical methods mainly on the *anchored-ANOVA decomposition*[5] instead of the (classical) ANOVA.

This way, we gain new insights into the interplay between coordinate transformations, effective dimensions and the error of sparse grid methods. We present a novel general class of multivariate quadrature methods, which is based on the anchored-ANOVA. Our approach exploits low effective dimension and, surprisingly, it includes the class of sparse grid methods as a special case. We derive new error bounds for sparse grid methods in weighted spaces and improve existing sparse grid methods in several ways. Our theoretical considerations indicate that these methods can profit from low effective dimension and from smoothness of the integrand much more efficiently than existing approaches. In fact many numerical experiments clearly demonstrate that our novel numerical methods are superior to Monte Carlo, quasi-Monte Carlo and existing sparse grid methods for various applications from finance even in hundreds of dimensions by several orders of magnitude.

We next describe the contributions of this thesis in more detail.

- Based on the anchored-ANOVA decomposition we define new notions of effective dimension, which we refer to as *effective dimensions in the anchored case*, and derive new error bounds which relate these dimensions to integration errors. We determine the effective dimensions in the anchored and in the classical case for several applications from finance with hundreds of dimensions and indicate by theoretical arguments and by numerical experiments that the performance of sparse grid methods can be better explained with the help of our new notion of effective dimension than with the classical one.
- We furthermore present a new general class of quadrature methods for the computation of high-dimensional integrals, which we refer to as *dimension-wise quadrature methods*. These new quadrature methods are developed in two

[4]Transformation to the unit cube introduce singularities which deteriorate the efficiency of methods that take advantage of higher smoothness, such as sparse grids.

[5]The anchored-ANOVA decomposition expresses a multivariate function as superposition of its values on lines, faces, hyperplanes, etc., which intersect a certain anchor point and are parallel to the coordinate axes [140]. Only a finite number of function values is required for its calculation. The computation of the classical ANOVA decomposition is significantly more expensive, since here 2^d many high-dimensional integrals have to be computed.

steps: First, the anchored-ANOVA decomposition is truncated either, a priori, based on function space weights or, a posteriori, in a dimension-adaptive fashion where important terms of the decomposition are automatically detected. This truncation introduces a modeling error which is controlled by the effective dimension in the anchored case. Then, the remaining terms are integrated using appropriate low-dimensional quadrature rules which may be different from term to term and may refine the approximation in a locally-adaptive way. This introduces a discretization error which only depends on the maximum order of the kept terms in the decomposition, but not on the nominal dimension d. We present numerical results using the CUHRE algorithm [10] for the integration of the low-order anchored-ANOVA terms and quasi-Monte Carlo methods for the higher-order ones. This way, we obtain mixed CUHRE/QMC methods which are to our knowledge the first numerical quadrature methods that can profit from low effective dimension by dimension-adaptivity and can at the same time deal with low regularity by local adaptivity. A correct balancing of modeling and discretization error is more difficult with our new methods than with sparse grid methods. However, numerical experiments with an 16-dimensional sample function from finance with discontinuous first derivatives demonstrate that this disadvantage is more than compensated by the benefits of the local adaptivity.

- We show that our new general class of methods includes the class of *sparse grid methods* as a special case if we use particular tensor product methods for the integration of the subterms. We explain that sparse grid methods can be interpreted as a refinement of the anchored-ANOVA decomposition by first expanding each term of the decomposition into an infinite basis and then truncating this expansion appropriately. This allows to intertwine the truncation of the anchored-ANOVA series and the subsequent discretization and allows to balance modeling and discretization error in an optimal way in the sense of [18] through the choice of the underlying index set. Such optimal index sets can be found in a dimension-adaptive fashion as in [49] or by using a priori information on the function space weights similar as in [165]. We determine optimal index sets for integrands from weighted tensor products of Sobolev spaces and derive new cost and error bounds, which take into account the function space weights and recover known results in case of equal weights.

- Moreover, we present two new variants of the sparse grid method which can treat the integral (1.1) directly on $\mathbb{R}^d$ avoiding the singular transformation to the unit cube. For moderate high-dimensional integrals with Gaussian weight and equally important dimensions, we define sparse grid methods based on the *delayed Genz-Keister sequence* using the recent approach from Petras [135]. For integrals with Gaussian weight, which have a high nominal, but a low

effective dimension, we propose dimension-adaptive sparse grid methods based on the *slowly increasing Gauss-Hermite sequence* combining ideas from [49, 123, 135]. We apply the latter method to several applications from finance and observe that this method is superior to Monte Carlo, quasi-Monte Carlo and existing sparse grid methods [49, 121, 135, 150] by several orders of magnitude even in hundreds of dimensions.

- To further improve the performance of dimension-adaptive sparse grid methods for problems from finance, we combine these methods for the first time with the *linear transformation method* from Imai and Tan [82]. Here, the main idea is that the integral (1.1) is invariant with respect to orthonormal transformations, such as rotations, and hence equals to

$$I_d = \int_{\mathbb{R}^d} g(\mathbf{Q}\mathbf{z})\varphi(\mathbf{z})d\mathbf{z}$$

for all orthogonal matrices $\mathbf{Q} \in \mathbb{R}^{d\times d}$. The linear transformation method identifies the matrix $\mathbf{Q}$ which minimizes the effective dimension of the integrand for certain function classes and can in this way maximize the performance of dimension-adaptive sparse grid methods. We provide numerical experiments with several application problems from finance which illustrate the efficiency of this new approach.

- We also address the difficulty that integrands from finance often have kinks or even jumps and do therefore not satisfy the smoothness requirement of sparse grid methods. To overcome this obstacle we here investigate the approach first to identify all kinks and jumps and then to decompose the integration domain $\mathbb{R}^d$ into subdomains Ω_i in which the integrand g is smooth. We thus shift the integration of one discontinuous function to the computation of

$$I_d = \sum_i \int_{\Omega_i} g(\mathbf{Q}\mathbf{z})\varphi(\mathbf{z})d\mathbf{z},$$

i.e., to the integration of several smooth functions. This way, we regain the fast convergence of sparse grid methods with costs that depend on the number of terms in the sum and on the complexity of the decomposition. We show that this approach is superior to standard methods for the pricing problems of barrier options and performance-dependent options.[6] In the first case, we efficiently decompose the integration domain with the help of *conditional sampling* [59]. In the second case, the decomposition is performed using novel

[6]Barrier options are financial derivatives which become worthless if the underlying asset crosses a specified barrier. Performance-dependent options are financial derivatives whose payoff depends on the performance of one asset in comparison to a set of benchmark assets.

tools from *computational geometry* for the enumeration and decomposition of hyperplane arrangements in high dimensions.

- One of the most complex applications from finance and insurance is the simulation of *stochastic asset-liability management (ALM) models in life insurance*. In such models the development of the capital markets, the behaviour of the policyholders and the decisions of the company's management have simultaneously be taken into account as well as guarantees and option-like features of the insurance products. New regulations, stronger competitions and more volatile capital markets have increased the demand for such simulations in recent years. The numerical simulation of such models is usually performed by Monte Carlo methods, which, however, often lead to unsatisfactorily long computing times of several days even on a supercomputer. In this thesis, we successfully apply, to our knowledge for the first time, quasi-Monte Carlo and sparse grid methods to these problems. To this end, we rewrite the ALM simulation problem as a multivariate integration problem and then apply the deterministic methods in combination with adaptivity and dimension reduction techniques. We provide various numerical experiments with a general ALM model framework, which incorporates the most important features of ALM simulations in life insurance such as the surrender of contracts, a reserve-dependent surplus declaration, a dynamic asset allocation and a two-factor stochastic capital market. The results demonstrate that the deterministic methods often converge faster, less erratic and produce more accurate results than Monte Carlo simulation even for small sample sizes n and complex models with many variables. We determine the effective dimensions and the important variables of different ALM models and show that even our most complex ALM model problems are of very low effective dimension, or can be transformed to be of low effective dimension by coordinate transformations. This way, we also provide a theoretical explanation for the success of the deterministic quadrature methods.

Several parts of this thesis are already published in the journal articles [52, 56], the book contribution [53] and the conference proceedings contributions [50, 54, 57] or are in process of being published [51, 62].

The remainder of this thesis is organized as follows. In Chapter 2, we introduce the classical ANOVA and the anchored-ANOVA decomposition of a multivariate function f. Based on these decompositions, we then define our new notions of effective dimensions and derive error bounds for approximation and integration.

In Chapter 3, we start with a short survey of classical numerical methods for the computation of high-dimensional integrals. Then, we define our new general class of multivariate quadrature methods. These methods proceed dimension-wise and are constructed by truncation of the anchored-ANOVA decomposition and by

integration of the remaining terms using one or several of the classical numerical methods. We derive cost and error bounds for the new methods and discuss a priori and a posteriori approaches to exploit low effective dimension.

We specify our general approach in Chapter 4, which leads us to the class of sparse grid methods. We then consider these methods in more detail. We first define new sparse grid methods based on delayed Genz-Keister and on slowly increasing Gauss-Hermite sequences. Then, we derive optimal index sets of sparse grid constructions for integrands from weighted Sobolev spaces.

The scope of Chapter 5 are approaches which can be used to improve the performance of sparse grid methods by dimension reduction and by the smoothing of the integrands. Here, we consider different path generating methods to reduce the dimension. Furthermore, we discuss domain decompositions and conditional sampling to regain smoothness.

In Chapter 6, we finally present several applications from finance which can efficiently be treated by our numerical methods. Using the pricing problems of different interest rate derivatives we study the effects of coordinate transformations and compare the performance of different sparse grid methods. Then, we consider path-dependent options, which lead to integrands with kinks or jumps. To overcome this obstacle we apply local adaptivity in the low-order anchored-ANOVA terms using the CUHRE algorithm and consider smoothing by conditional sampling. Moreover, we discuss the efficient pricing of performance-dependent options using domain decompositions to regain smoothness. Finally, we consider the simulation of stochastic asset-liability management models in life insurance using our deterministic integration methods.

We conclude with a summary of the presented results and some remarks on areas of future research in Chapter 7.

We give complementary information in the appendix. In Appendix A, we formally define the notion of tractability and summarize some related results. In Appendix B, we provide more details on performance-dependent options and their valuation. Finally, in Appendix C, we describe our general asset-liability management model and present complementary numerical results which illustrate the impact of the parameters of the model on the convergence behaviour of our numerical methods.

Chapter 2
Dimension-wise Decompositions

In this chapter, we introduce the classical ANOVA and the anchored-ANOVA decomposition of a multivariate function f. Based on these decompositions, we then define different notions of effective dimensions of f and derive error bounds for approximation and integration.

We start with the introduction of general dimension-wise decompositions. To this end, let $\Omega \subseteq \mathbb{R}$ be a domain and let

$$d\mu(\mathbf{x}) = \prod_{j=1}^{d} d\mu_j(x_j) \tag{2.1}$$

denote a d-dimensional product measure defined on Borel subsets of Ω^d. Here, $\mathbf{x} = (x_1, \ldots, x_d)$ and μ_j, $j = 1, \ldots, d$, are probability measures on Borel subsets of Ω. Let $V^{(d)}$ denote the Hilbert space of all functions $f : \Omega^d \to \mathbb{R}$ with the inner product

$$(f, g) := \int_{\Omega^d} f(\mathbf{x}) g(\mathbf{x})\, d\mu(\mathbf{x}).$$

For a given set $\mathbf{u} \subseteq \mathcal{D}$, where $\mathcal{D} := \{1, \ldots, d\}$ denotes the set of coordinate indices, the measure μ induces projections $P_{\mathbf{u}} : V^{(d)} \to V^{(|\mathbf{u}|)}$ by

$$P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) := \int_{\Omega^{d-|\mathbf{u}|}} f(\mathbf{x}) d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{x}). \tag{2.2}$$

Thereby, $\mathbf{x}_{\mathbf{u}}$ denotes the $|\mathbf{u}|$-dimensional vector containing those components of $\mathbf{x}$ whose indices belong to the set $\mathbf{u}$ and $d\mu_{\mathcal{D}\setminus\mathbf{u}}(\mathbf{x}) := \prod_{j\notin\mathbf{u}} d\mu_j(x_j)$. The projections define a decomposition of $f \in V^{(d)}$ into a finite sum according to

$$f(x_1, \ldots, x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i,j=1}^{d} f_{i,j}(x_i, x_j) + \ldots + f_{1,\ldots,d}(x_1, \ldots, x_d)$$

which is often written in the more compact notation

$$f(\mathbf{x}) = \sum_{\mathbf{u}\subseteq\mathcal{D}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}). \tag{2.3}$$

The 2^d terms $f_{\mathbf{u}}$ describe the dependence of the function f on the dimensions $j \in \mathbf{u}$ with respect to the measure μ. They are recursively defined by

$$f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) := P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) - \sum_{\mathbf{v}\subset\mathbf{u}} f_{\mathbf{v}}(\mathbf{x}_{\mathbf{v}}) \tag{2.4}$$

and can also be given explicitly by

$$f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} P_{\mathbf{v}} f(\mathbf{x}_{\mathbf{v}}), \tag{2.5}$$

see [97]. The resulting decomposition (2.3) is unique for a fixed measure μ and orthogonal in the sense that

$$(f_{\mathbf{u}}, f_{\mathbf{v}}) = 0 \tag{2.6}$$

for $\mathbf{u} \neq \mathbf{v}$, see, e.g., [61, 140].

2.1 Classical ANOVA Decomposition

For $\Omega = [0,1]$ and the example of the Lebesgue measure $d\mu(\mathbf{x}) = d\mathbf{x}$ in (2.1), the space $V^{(d)}$ is the space of square integrable functions and the projections are given by

$$P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) = \int_{[0,1]^{d-|\mathbf{u}|}} f(\mathbf{x})\, d\mathbf{x}_{\mathcal{D}\setminus\mathbf{u}}.$$

The decomposition (2.3) then corresponds to the well-known analysis of variance (ANOVA) decomposition which is used in statistics to identify important variables and important interactions between variables in high-dimensional models. It goes back to [78] and has been studied in many different contexts and applications, e.g., [36,71,85,158]. Recently, it has extensively been used for the analysis of quasi-Monte Carlo methods, see, e.g., [19, 103, 106, 152] and the references cited therein.

In the ANOVA the orthogonality (2.6) implies that the variance of the function f can be written as

$$\sigma^2(f) = \sum_{\substack{\mathbf{u}\subseteq\mathcal{D}\\ \mathbf{u}\neq\emptyset}} \sigma^2(f_{\mathbf{u}}), \tag{2.7}$$

where $\sigma^2(f_{\mathbf{u}})$ denotes the variance of the term $f_{\mathbf{u}}$. The values $\sigma^2(f_{\mathbf{u}})/\sigma^2(f)$, called global sensitivity indices in [151, 152], can then be used to measure the relative importance of the term $f_{\mathbf{u}}$ with respect to the function f.

Example 2.1. For given univariate functions $g_j \in L^2([0,1])$, $j = 1, \ldots, d$, let

$$Ig_j = \int_{[0,1]^d} g_j(x)dx \quad \text{and} \quad \sigma^2(g_j) = \int_{[0,1]^d} (g_j(x) - Ig_j)^2\, dx.$$

For the classes of purely additive or multiplicative functions

$$f^+(\mathbf{x}) := \sum_{j=1}^{d} g_j(x_j) \quad \text{and} \quad f^*(\mathbf{x}) := \prod_{j=1}^{d} g_j(x_j)$$

the ANOVA decomposition can easily be derived analytical, see also [129]. We obtain

$$If^+ = \sum_{j=1}^{d} Ig_j \quad \text{and} \quad \sigma^2(f^+) = \sum_{j=1}^{d} \sigma^2(g_j).$$

Furthermore,

$$If^* = \prod_{j=1}^{d} Ig_j \quad \text{and} \quad \sigma^2(f^*) = \prod_{j=1}^{d} (I^2 g_j + \sigma^2(g_j)) - If^*.$$

The ANOVA terms of f^+ and their variances are given by

$$f^+_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = \begin{cases} If^+ & \text{if } \mathbf{u} = \emptyset \\ g_j(x_j) - Ig_j & \text{if } \mathbf{u} = \{j\} \\ 0 & \text{if } |\mathbf{u}| > 1 \end{cases} \quad \text{and} \quad \sigma^2(f^+_{\mathbf{u}}) = \begin{cases} 0 & \text{if } \mathbf{u} = \emptyset \\ \sigma^2(g_j) & \text{if } \mathbf{u} = \{j\} \\ 0 & \text{if } |\mathbf{u}| > 1. \end{cases}$$

For the function f^*, a simple computation yields

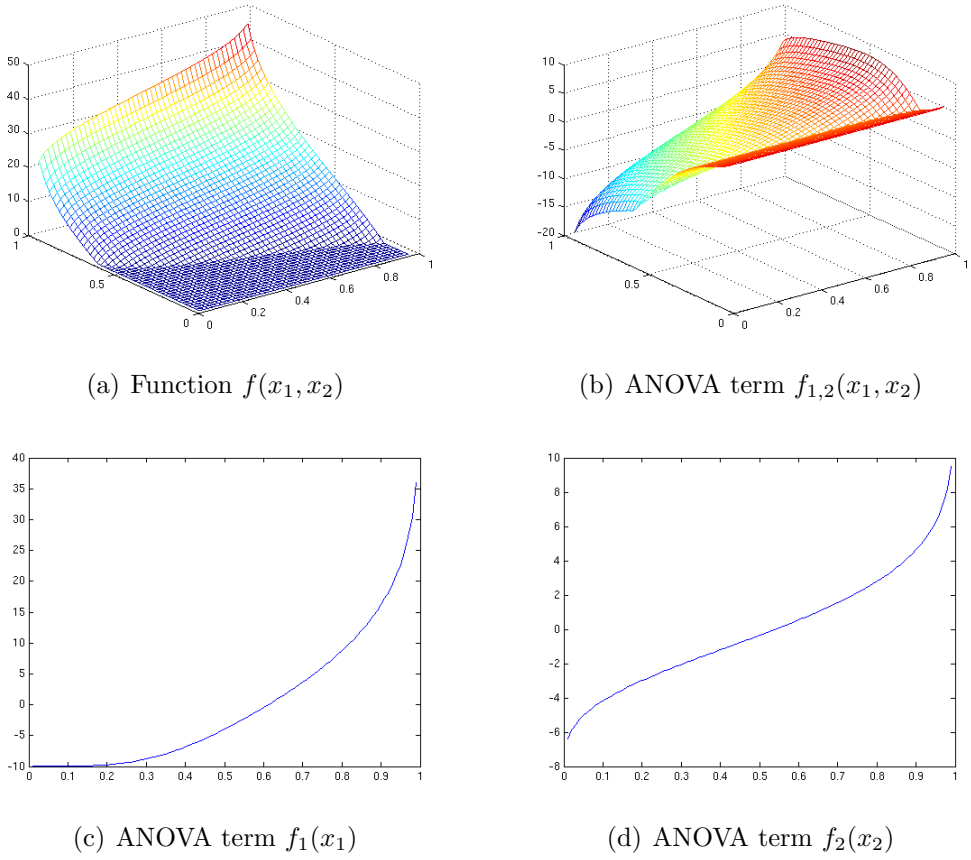
$$f^*_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = \prod_{j \in \mathbf{u}} (g_j(x_j) - Ig_j) \prod_{j \notin \mathbf{u}} Ig_j \quad \text{and} \quad \sigma^2(f^*_{\mathbf{u}}) = \prod_{j \in \mathbf{u}} \sigma^2(g_j) \prod_{j \notin \mathbf{u}} I^2 g_j.$$

For illustration, we show in Figure 2.1 a two-dimensional sample function f and its three ANOVA terms f_1, f_2 and $f_{1,2}$. The displayed function appears in the problem to price Asian options using the Brownian bridge path constructions.[1] Observe that variance of the ANOVA term f_2 is significantly smaller than the variance of f_1. Such a concentration of the variance in the first variables leads to low effective dimension and can be used for the efficient numerical treatment of high-dimensional integration problems as we will explain in Chapter 3 and Chapter 4. Note further that the first order ANOVA terms are smooth functions despite the fact that f is not differentiable. This smoothing effect of the ANOVA decomposition indicates that also functions of low regularity may be integrated efficiently in high dimensions if the higher-order ANOVA terms are sufficiently small. This smoothing effect was first observed in [106] and is further investigated in [64].

2.1.1 Effective Dimensions

Based on the ANOVA decomposition, different notions of effective dimensions have been introduced in [19]. For the proportion $\alpha \in (0, 1]$, the effective dimension in

[1] This application problem is described in detail in Section 6.2.1. Path constructions are the topic of Section 5.1.



(a) Function $f(x_1, x_2)$

(b) ANOVA term $f_{1,2}(x_1, x_2)$

(c) ANOVA term $f_1(x_1)$

(d) ANOVA term $f_2(x_2)$

Figure 2.1. *Sample function $f(x_1, x_2)$ and its ANOVA terms.*

the truncation sense (the *truncation dimension*) of the function f is defined as the smallest integer d_t, such that

$$\sum_{\substack{\mathbf{u}\subseteq\{1,\ldots,d_t\}\\ \mathbf{u}\neq\emptyset}} \sigma^2(f_{\mathbf{u}}) \geq \alpha\ \sigma^2(f). \tag{2.8}$$

Here, often the proportion $\alpha = 0.99$ is used. The effective dimension in the superposition sense (the *superposition dimension*) is defined as the smallest integer d_s, such that

$$\sum_{\substack{|\mathbf{u}|\leq d_s\\ \mathbf{u}\neq\emptyset}} \sigma^2(f_{\mathbf{u}}) \geq \alpha\ \sigma^2(f). \tag{2.9}$$

If the variables are ordered according to their importance, the truncation dimension d_t roughly describes the number of important variables of the function f. The superposition d_s dimension roughly describes the highest order of important interactions between variables.

For instance, for the function $f(x_1, x_2, x_3) = \exp\{x_1\} + x_2$ with $d = 3$, we obtain (independently of α) that $d_t = 2$ and $d_s = 1$ since the third variable as well as the interaction between two and more variables are unimportant.

For large d, it is no longer possible to compute all 2^d ANOVA terms, but the effective truncation dimension can still be computed in many cases, see [151,160]. The algorithm requires the computation of several integrals with up to $2d-1$ dimensions. Without additional effort, the algorithm also yields the values

$$T_j := \frac{1}{\sigma^2(f)} \sum_{\mathbf{u} \not\subseteq \{1,\ldots,j\}} \sigma^2(f_{\mathbf{u}}) \tag{2.10}$$

for $j = 0, \ldots, d$. Note that $T_0 = 1$ and $T_d = 0$. The decay of these values often more clearly illustrate the importance of the first dimensions than the truncation dimension since the dependence on the proportion α is avoided.

For the more difficult problem to compute the superposition dimension or the values

$$S_j := \frac{1}{\sigma^2(f)} \sum_{|\mathbf{u}|>j} \sigma^2(f_{\mathbf{u}}). \tag{2.11}$$

the recursive method described in [161] can be used. Because of cancellation problems and costs which grow exponential in d_s, the computation of the superposition dimension is only feasible for moderately high-dimensional function or for functions with very low superposition, however.

2.1.2 Error Bounds

The following two lemmas relate the effective dimensions to approximation errors. The second lemma is taken from [151].

Lemma 2.2. *Let d_t denote the truncation dimension of the function f with proportion α and let $f_{d_t}(\mathbf{x}) := \sum_{\mathbf{u}\subseteq\{1,\ldots,d_t\}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$. Then*

$$\|f - f_{d_t}\|_{L_2}^2 \leq (1-\alpha)\sigma^2(f).$$

Proof. Note that $\sigma^2(f_{\mathbf{u}}) = \|f_{\mathbf{u}}\|_2^2$ for $\mathbf{u} \neq \emptyset$ since $\int_{[0,1]^{|\mathbf{u}|}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})\, d\mathbf{x}_{\mathbf{u}} = 0$ for $\mathbf{u} \neq \emptyset$. From (2.3), one obtains

$$\begin{aligned} \|f - f_{d_t}\|_{L_2}^2 &= \Big\| \sum_{\mathbf{u} \not\subseteq \{1,\ldots,d_t\}} f_{\mathbf{u}} \Big\|_{L_2}^2 = \sum_{\mathbf{u} \not\subseteq \{1,\ldots,d_t\}} \|f_{\mathbf{u}}\|_{L_2}^2 \\ &= \sum_{\mathbf{u}\subseteq\mathcal{D}} \sigma^2(f_{\mathbf{u}}) - \sum_{\mathbf{u}\subseteq\{1,\ldots,d_t\}} \sigma^2(f_{\mathbf{u}}) \leq (1-\alpha)\sigma^2(f), \end{aligned}$$

where the second equality holds by orthogonality and where the inequality follows from (2.7) and (2.8). □

Lemma 2.3. *Let d_s denote the superposition dimension of the function f with proportion α and let $f_{d_s}(\mathbf{x}) := \sum_{|\mathbf{u}|\leq d_s} f_{\mathbf{u}}(\mathbf{x}_\mathbf{u})$. Then*

$$\|f - f_{d_s}\|_{L_2}^2 \leq (1-\alpha)\sigma^2(f).$$

Proof. Similar as in Lemma 2.2 we compute

$$\|f - f_{d_s}\|_{L_2}^2 = \|\sum_{|\mathbf{u}|>d_s} f_{\mathbf{u}}\|_{L_2}^2 = \sum_{|\mathbf{u}|>d_s} \|f_{\mathbf{u}}\|_{L_2}^2 = \sum_{|\mathbf{u}|>d_s} \sigma^2(f_{\mathbf{u}}) \leq (1-\alpha)\,\sigma^2(f)$$

using orthogonality, (2.3), (2.7) and (2.9). □

For integration, we immediately obtain as corollary the error bound

$$|If - If_{\mathrm{tr}}| \leq \|f - f_{\mathrm{tr}}\|_{L_1} \leq \|f - f_{\mathrm{tr}}\|_{L_2} \leq \sqrt{1-\alpha}\,\sigma(f) \tag{2.12}$$

either if $f_{\mathrm{tr}} := f_{d_t}$ (as in Lemma 2.2) or if $f_{\mathrm{tr}} := f_{d_s}$ (as in Lemma 2.3) and if α is the proportion corresponding to d_t and d_s, respectively. One can see that quadrature methods produce small errors if α is close to one and if the methods can compute If_{tr} efficiently.

Quasi-random points[2] are usually more uniformly distributed in smaller dimensions than in higher ones such that we can expect that If_{d_t} is well approximated for small d_t. Moreover, quasi-random points usually have very well distributed low dimensional projections such that we can expect that If_{d_s} is efficiently computed for small d_s. Hence, the bound (2.12) partly explains the success of quasi-Monte Carlo methods for high-dimensional integrals with functions of low truncation dimension or low superposition dimension.[3]

The bound (2.12) also partly explains the success of sparse grid methods[4] for high-dimensional integrals with functions of low effective dimension since these methods can compute If_{tr} very efficiently for small d_s or small d_t with the help

[2]Quasi-Monte Carlo methods are the topic of Section 3.1.2.

[3]Note that low effective dimension is not necessary for the situation that quasi-Monte Carlo methods are more efficient than Monte Carlo methods for large d and small sample sizes n. This is shown in [156]. Note also that the combination of low effective dimension and of good low dimensional projections is not sufficient to imply that quasi-Monte Carlo is more efficient than Monte Carlo.

[4]We discuss sparse grid methods and their combination with dimension-adaptivity in Chapter 4.

of a dimension-adaptive grid refinement.

Remark 2.4. We can also choose $\Omega = \mathbb{R}$ and the Gaussian measure $d\mu(\mathbf{x}) = \varphi_d(\mathbf{x})d\mathbf{x}$ in (2.1) where

$$\varphi_d(\mathbf{x}) := e^{-\mathbf{x}^T\mathbf{x}/2}/(2\pi)^{d/2} \tag{2.13}$$

denotes the standard Gaussian density in d dimensions. This induces projections

$$P_{\mathbf{u}}f(\mathbf{x}_{\mathbf{u}}) = \int_{\mathbb{R}^{d-|\mathbf{u}|}} f(\mathbf{x})\varphi_{d-|\mathbf{u}|}(\mathbf{x}_{\mathbf{u}})\, d\mathbf{x}_{\mathcal{D}\setminus\mathbf{u}}.$$

Then, by (2.3), a corresponding decomposition of functions $f : \mathbb{R}^d \to \mathbb{R}$ results, which we refer to as *ANOVA decomposition with Gaussian weight*. Based on this decomposition, effective dimensions for the ANOVA decomposition with Gaussian weight can be defined as in (2.8) and (2.9).

2.2 Anchored-ANOVA Decomposition

For $\Omega = [0,1]$ and the example of the Dirac measure located at a fixed anchor point $\mathbf{a} \in [0,1]^d$, i.e. $d\mu(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{a})d\mathbf{x}$, we obtain from (2.2) the projections

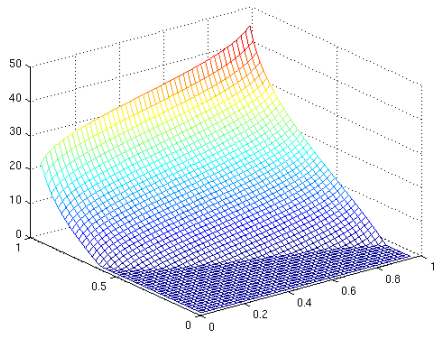
$$P_{\mathbf{u}}f(\mathbf{x}_{\mathbf{u}}) = f(\mathbf{x})|_{\mathbf{x}=\mathbf{a}\setminus\mathbf{x}_{\mathbf{u}}}$$

where we use the notation $f(\mathbf{x})|_{\mathbf{x}=\mathbf{a}\setminus x_i} = f(a_1,\dots,a_{i-1},x_i,a_{i+1},\dots,a_d)$ with its obvious generalisation to $\mathbf{a}\setminus\mathbf{x}_{\mathbf{u}}$. The terms of the anchored-ANOVA decomposition are thus related to the terms of the classical ANOVA decomposition in the sense that all integrals are replaced by point evaluations at a fixed anchor point $\mathbf{a} \in [0,1]^d$. This approach is considered in [140] under the name CUT-HDMR. The decomposition expresses f as superposition of its values on lines, faces, hyperplanes, etc., which intersect the anchor point $\mathbf{a}$ and are parallel to the coordinate axes. It is closely related to the multivariate Taylor expansion [61], and to the anchor spaces considered, e.g., in [32, 75, 147, 166]. There are various generalisations of the anchored-ANOVA decomposition such as the multi-CUT-HDMR, the mp-CUT-HDMR and the lp-RS, see [61] and the references cited therein.

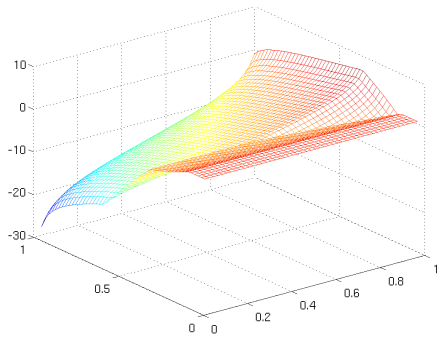
While the classical ANOVA decomposition is very useful to analyse the importance of different dimensions and of their interactions it can not be used as a tool for the design of new integration schemes since already the constant term in the classical ANOVA decomposition requires to compute the integral. The anchored-ANOVA decomposition has the advantage that its sub-terms are much cheaper to compute since instead of integrals only point evaluations at the anchor point $\mathbf{a} \in [0,1]^d$ are required. We will use this property in Section 3.2 to design new quadrature methods for high-dimensional functions.

For illustration, we consider, as in Section 2.1, the two-dimensional function f which arises in the problem to price Asian options with two time steps. The

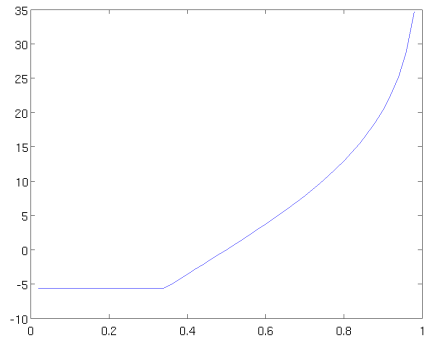
resulting anchored-ANOVA terms are displayed in Figure 2.1. One can see that the concentration of the variance in the first variables is maintained in the anchored case whereas the smoothing effect of the classical ANOVA is lost.



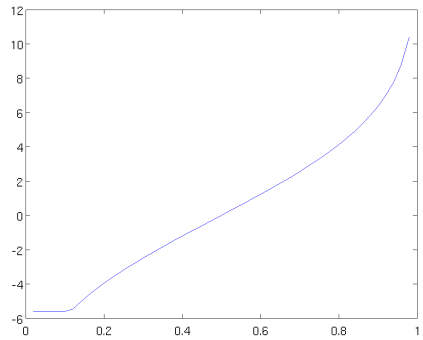
(a) Function $f(x_1, x_2)$



(b) ANOVA term $f_{12}(x_1, x_2)$



(c) ANOVA term $f_1(x_1)$



(d) ANOVA term $f_2(x_2)$

Figure 2.2. *Sample function $f(x_1, x_2)$ and its anchored-ANOVA terms.*

Example 2.5. For purely additive functions f^+ or multiplicative functions f^*, see Example 2.1, the anchored-ANOVA terms and their integrals can be computed analytically. We obtain

$$f_{\mathbf{u}}^+(\mathbf{x_u}) = \begin{cases} \sum_{j=1}^d g_j(a_j) & \text{if } \mathbf{u} = \emptyset \\ g_j(x_j) - g_j(a_j) & \text{if } \mathbf{u} = \{j\} \\ 0 & \text{if } |\mathbf{u}| > 1, \end{cases} \quad \text{and } If_{\mathbf{u}}^+ = \begin{cases} 0 & \text{if } \mathbf{u} = \emptyset \\ Ig_j - g_j(a_j) & \text{if } \mathbf{u} = \{j\} \\ 0 & \text{if } |\mathbf{u}| > 1. \end{cases}$$

Furthermore,

$$f_{\mathbf{u}}^*(\mathbf{x_u}) = \prod_{j \in \mathbf{u}} (g_j(x_j) - g_j(a_j)) \prod_{j \notin \mathbf{u}} g_j(a_j) \text{ and } If_{\mathbf{u}}^* = \prod_{j \in \mathbf{u}} (Ig_j - g_j(a_j)) \prod_{j \notin \mathbf{u}} g_j(a_j).$$

2.2.1 Effective Dimensions

We next define a new notion of effective dimension which is based on the anchored-ANOVA decomposition. While the effective dimensions in the classical case are based on the L_2-norm, we now introduce effective dimensions for the anchored case, which are based on the operator $|I(\cdot)|$ and, since $|I(f)| = |\int_{[0,1]^d} f(\mathbf{x})\,d\mathbf{x}| \leq \|f\|_{L_1}$, which are thus related to the L_1-norm. While the effective dimensions in the classical case directly lead to error bounds for approximation (see Lemma 2.2 and Lemma 2.3), we will use the effective dimensions in the anchored case to derive error bounds for integration (see Lemma 2.7 and Lemma 2.8). To this end, let

$$\hat{\sigma}(f) := \sum_{\substack{\mathbf{u}\subseteq\mathcal{D}\\ \mathbf{u}\neq\emptyset}} |If_{\mathbf{u}}| \leq \sum_{\substack{\mathbf{u}\subseteq\mathcal{D}\\ \mathbf{u}\neq\emptyset}} \|f_{\mathbf{u}}\|_{L_1} \tag{2.14}$$

be the sum of all contributions to the value of the integral. Then, analogous to (2.8), for the proportion $\alpha \in (0,1]$, the *truncation dimension in the anchored case* is defined as the smallest integer d_t, such that

$$\sum_{\substack{\mathbf{u}\subseteq\{1,\ldots,d_t\}\\ \mathbf{u}\neq\emptyset}} |If_{\mathbf{u}}| \geq \alpha\ \hat{\sigma}(f), \tag{2.15}$$

whereas, analogous to (2.9), the *superposition dimension in the anchored case* is defined as the smallest integer d_s, such that

$$\sum_{\substack{|\mathbf{u}|\leq d_s\\ \mathbf{u}\neq\emptyset}} |If_{\mathbf{u}}| \geq \alpha\ \hat{\sigma}(f). \tag{2.16}$$

As in the classical case, these notions describe roughly the number of important dimensions and the order of important interactions, respectively. Compared to the classical case, the effective dimensions in the anchored case have the following advantages: They are directly related to integration errors as we show below and they can easily be determined by dimension-wise integration methods as we will explain in Section 3.2 in more detail. We also have a direct relation of the effective dimensions in the anchored case to sparse grid methods as we will show in Section 4.3. As $|I(\cdot)|$ is not a norm, it may happen, however, that the effective dimensions in the anchored case fail to detect some important dimensions and interactions.[5] For instance, for the function $f(x_1,x_2) = e^{x_1} - e^{0.5} + x_2 - \frac{1}{2}$ we obtain $f_2(x_2) = x_2 - \frac{1}{2}$ such that $\sigma^2(f_2) > 0$ but $|If_2| = 0$ which misleadingly indicates independence of

[5]It is also possible to base the definition on the L_1-norm. Then this drawback disappears. Nevertheless we here stick to the operator $|I(\cdot)|$ in order to exploit a more direct relation to dimension-adaptive sparse grid methods.

x_2. This effect which we have not yet observed in practical applications from finance, see also Chapter 6, is closely related to the early determination problem of dimension-adaptive sparse grid methods which is discussed in [49].

To compute the effective dimensions in the anchored case, we use multivariate quadrature methods to compute $If_{\mathbf{u}}$ for all $\mathbf{u} \subseteq \mathcal{D}$. We denote the resulting approximations by $q_{\mathbf{u}} \approx If_{\mathbf{u}}$. By summation of the computed values $q_{\mathbf{u}}$, we estimate

$$\hat{\sigma}(f) = \sum_{\substack{\mathbf{u} \subseteq \mathcal{D} \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}|$$

according to (2.14) and determine the effective dimensions by the smallest integers d_t and d_s, such that

$$\sum_{\substack{\mathbf{u} \subseteq \{1,\dots,d_t\} \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}| \geq \alpha\ \hat{\sigma}(f) \quad \text{and} \quad \sum_{\substack{|\mathbf{u}| \leq d_s \\ \mathbf{u} \neq \emptyset}} |q_{\mathbf{u}}| \geq \alpha\ \hat{\sigma}(f),$$

respectively, see (2.15) and (2.16). Without additional effort, the values $q_{\mathbf{u}}$ can also be used to compute the values

$$T_j := \frac{1}{\hat{\sigma}(f)} \sum_{\mathbf{u} \not\subseteq \{1,\dots,j\}} |q_{\mathbf{u}}| \quad \text{and} \quad S_j := \frac{1}{\hat{\sigma}(f)} \sum_{|\mathbf{u}| > j} |q_{\mathbf{u}}| \tag{2.17}$$

for $j = 0, \dots, d$. Note that $T_0 = S_0 = 1$ and $T_d = S_d = 0$. The values T_j and S_j approximate the relative modeling error which results if we truncate all terms of the anchored-ANOVA decomposition that satisfy $\mathbf{u} \not\subseteq \{1, \dots, j\}$ and $|\mathbf{u}| > j$, respectively. While T_j describes the importance of the first j dimensions, S_j illustrate the importance of the interactions of up to j many variables.

In Chapter 6 we will compute these values for various high-dimensional functions from finance and relate the results to the effective dimensions in the classical case. We will see that the effective dimensions in the anchored case provide similar information on the importance of dimensions and interactions as the effective dimensions in the classical case but have the advantage that they are significantly cheaper to compute. While the computation of the effective dimensions in the classical case requires the computation of *many* high-dimensional integrals with, e.g., quasi-Monte Carlo methods,[6] the computation of the effective dimensions in the anchored case requires only the computation of *one* high-dimensional integral with, e.g., sparse grids or other dimension-wise quadrature methods that we introduce in Chapter 3 and Chapter 4.

Remark 2.6. We can also choose $\Omega = \mathbb{R}$ and the measure $d\mu(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{a})\varphi_d(\mathbf{x})d\mathbf{x}$ with a fixed anchor point $\mathbf{a} \in \mathbb{R}^d$, where φ_d is the Gaussian density (2.13). This

[6] The computation of the superposition dimension in the classical case in addition suffers from cancellation problems and costs which grow exponential in the superposition dimension.

example induces projections

$$P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) = \big(f(\mathbf{x})\varphi_{d-|\mathbf{u}|}(\mathbf{x}_{\mathbf{u}})\big)\,|_{\mathbf{x}=\mathbf{a}\setminus\mathbf{x}_{\mathbf{u}}}$$

and, by (2.3), a corresponding decomposition of functions $f : \mathbb{R}^d \to \mathbb{R}$ which we refer to as *anchored-ANOVA decomposition with Gaussian weight*. Based on this decomposition, effective dimensions for the anchored-ANOVA decomposition with Gaussian weight can be defined as in (2.15) and (2.16).

2.2.2 Error Bounds

The following two estimates relate effective dimensions in the anchored case and integration errors.

Lemma 2.7. *Let d_t denote the truncation dimension of the function f in the anchored case with proportion α and let $f_{d_t}(\mathbf{x}) := \sum_{\mathbf{u}\subseteq\{1,\dots,d_t\}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$. Then*

$$|If - If_{d_t}| \le (1-\alpha)\,\hat{\sigma}(f).$$

Proof. We obtain

$$|If - If_{d_t}| = |\sum_{\mathbf{u}\not\subseteq\{1,\dots,d_t\}} If_{\mathbf{u}}| \le \sum_{\mathbf{u}\not\subseteq\{1,\dots,d_t\}} |If_{\mathbf{u}}| = \sum_{\mathbf{u}\subseteq\mathcal{D}} |If_{\mathbf{u}}| - \sum_{\mathbf{u}\subseteq\{1,\dots,d_t\}} |If_{\mathbf{u}}| \le (1-\alpha)\,\hat{\sigma}(f)$$

where the first equality results from (2.3) and from the definition of the function f_{d_t}. The last inequality follows from (2.14) and (2.15). ∎

Lemma 2.8. *Let d_s denote the superposition dimension of the function f in the anchored case with proportion α and let $f_{d_s}(\mathbf{x}) := \sum_{|\mathbf{u}|\le d_s} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$. Then*

$$|If - If_{d_s}| \le (1-\alpha)\,\hat{\sigma}(f).$$

Proof. Similar as in Lemma 2.7 we obtain

$$|If - If_{d_s}| = |\sum_{|\mathbf{u}|>d_s} If_{\mathbf{u}}| \le \sum_{|\mathbf{u}|>d_s} |If_{\mathbf{u}}| \le (1-\alpha)\,\hat{\sigma}(f)$$

using (2.14) and (2.16) for the last inequality. ∎

We conclude this chapter with a short summary. We here considered the classical ANOVA and the anchored-ANOVA decomposition of high-dimensional functions.

With respect to these dimension-wise decompositions, we then defined the effective dimension in the classical and in the anchored case and derived corresponding error bounds. While the effective dimension in the classical case is based on the L_2-norm and related to approximation, the effective dimension in the anchored case is related to the L_1-norm and to integration. We will return to the anchored-ANOVA decomposition in Chapter 3 and Chapter 4 to define a new general class of quadrature methods and to provide a new interpretation of sparse grid methods, respectively. In Chapter 6, we will finally compute the effective dimensions of several application problems from fiance and insurance in the classical and in the anchored case and relate the results to the performance of different numerical quadrature methods.

Chapter 3
Dimension-wise Quadrature

High-dimensional integrals appear in various mathematical models from physics, chemistry or finance. The large number of dimensions arises, e.g., from small time steps in time discretizations and/or a large number of state variables. In many cases the arising integrals can not be calculated analytically and numerical methods must be applied. Here, one of the key prerequisites for a successful application is that the *curse of dimension* [9] can be avoided at least to some extent. The curse of dimension states that the cost to compute an approximation with a prescribed accuracy ε depends exponentially on the dimension d of the problem. It is one of the main obstacles for the numerical treatment of high dimensional problems, see, e.g., [61].

In this chapter, we first, in Section 3.1, survey classical numerical methods for the computation of high-dimensional integrals. Then, in Section 3.2, we define a new general class of multivariate quadrature methods. The methods are constructed by truncation of the anchored-ANOVA decomposition of the integrand. The kept terms are then integrated using one or several of the classical numerical methods.

3.1 Classical Multivariate Quadrature Methods

In this section, we survey numerical methods for the computation of high-dimensional integrals

$$If := \int_{[0,1]^d} f(\mathbf{x})\ d\mathbf{x} \tag{3.1}$$

over the unit cube. Note that rectangular domains $[a_1, b_1] \times \ldots \times [a_d, b_d]$ can easily be mapped to the unit cube by a linear transformation. We also consider numerical methods for high-dimensional integrals

$$I_\varphi f := \int_{\mathbb{R}^d} f(\mathbf{z})\, \varphi_d(\mathbf{z})\ d\mathbf{z} \tag{3.2}$$

over the d-dimensional Euclidean space with the Gaussian weight function φ_d from (2.13). The two domains unit cube and $\mathbb{R}^d$ typically appear in high-dimensional applications.

For integration over non-tensor product domains, which are important for $d = 2$ and $d = 3$, and for non-uniform or non-Gaussian integration, we refer to [20–22, 153] where a comprehensive listing of polynomial-based methods is given. General reference to multivariate numerical integration are [28, 153]. Applications of multivariate numerical integration in finance and economics are discussed, e.g., in [47, 59, 89].

All numerical methods, which we discuss here, approximate the d-dimensional integral If by a weighted sum of n function evaluations

$$Q_n(f) := \sum_{i=1}^{n} w_i f(\mathbf{x}^i) \tag{3.3}$$

with weights $w_i \in \mathbb{R}$ and nodes $\mathbf{x}^i \in \Omega^d$. Here, either $\Omega = [0,1]$ or $\Omega = \mathbb{R}$. Depending on the choice of the weights and nodes, different classes of methods with varying properties are obtained, which are shortly reviewed in the next sections. Here, one can distinguish between statistical methods (*Monte Carlo*), number theoretic methods (*quasi-Monte Carlo*) and polynomial-based methods (*product rules*, *sparse grids* and other interpolatory formulas). While the methods of the first two classes are based on uniformly distributed point sets, the rules of the third class are designed to be exact for a certain set of polynomials. We next survey Monte Carlo, quasi-Monte Carlo and product rules in more detail. Sparse grid methods are separately discussed in Chapter 4.

Note that the integral (3.2) can be transformed to an integral on the unit cube $[0,1]^d$, e.g., by the standard, component-wise substitution with the inverse of the cumulative normal distribution function.[1] This is important since most multivariate numerical quadrature methods, e.g., (quasi-) Monte Carlo methods, compute integrals of the form (3.1).

Remark 3.1. The resulting transformed integrand is unbounded on the boundary of the unit cube, which is undesirable from a numerical as well as theoretical point of view. Nevertheless, in combination with (quasi-) Monte Carlo methods this transformation turns out to be very effective because it cancels the Gaussian weight, see also [162]. However, the singularities which are introduced by the transformation deteriorates the efficiency of methods which take advantage of higher smoothness, such as product methods or sparse grids. Here, it is often better to avoid the transformation and the corresponding loss of regularity and to address the integral directly on $\mathbb{R}^d$. This is possible if one bases the product or sparse grid construction on univariate quadrature formulas for integrals on $\mathbb{R}$, e.g. Gauss-Hermite rules, which we will discuss in Section 3.1.3.

3.1.1 Monte Carlo

High dimensional integrals on the unit cube are mostly approximated with the Monte Carlo (MC) method. Here, all weights equal $w_i = 1/n$ and uniformly distributed sequences of pseudo-random numbers $\mathbf{x}^i \in [0,1]^d$ are used as nodes.[2] The law of

[1] The standard univariate normal distribution function and its inverse can efficiently and up to high accuracy be computed by a Moro [114] scheme.

[2] Algorithms which generate uniform pseudo-random numbers are surveyed, e.g., in [59,83,119].

large numbers then ensures that the estimate

$$\mathrm{MC}_n(f) := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}^i) \tag{3.4}$$

converges to If for $n \to \infty$ if f has finite variance $\sigma^2(f)$. The MC method is very easy to implement but suffers from a relative low probabilistic convergence rate of $1/2$, which is, on the positive side, independent of the dimension. The error $\varepsilon(n)$ of a MC estimate with n samples is approximately normally distributed with mean zero and standard deviation $\sigma^2(f)/\sqrt{n}$. In this way, the MC method also provides a useful statistical error estimate because $\sigma^2(f)$ can be estimated in each sampling step from the available function values.

Variance reduction methods

Several variance reduction techniques, such as antithetic variates, control variates, stratified sampling and importance sampling, exist to enhance the performance of the MC method, see, e.g., [13, 59, 90]. Such approaches are designed either by using a priori knowledge of the integrand or in an automatic way as realised in (locally) adaptive MC algorithms such as Vegas, Miser, Suave or Divonne, see, e.g., [68, 139]. The adaptive approaches exhibit costs which increase exponentially fast with the dimension, though, and are thus limited to the computation of at most moderately high dimensional integrals. Variance reduction techniques do not overcome the slow and erratic convergence but can improve the implicit constant of the method. Decreasing the variance $\sigma^2(f)$ by a factor c does roughly as much for error reduction as increasing the number n of samples by a factor of c^2. A drawback of some variance reduction techniques is that the construction of statistical error estimators is often no longer straightforward.

We next survey some of the most important variance reduction techniques. Several extensions, variants and combinations of these techniques are possible, see [59]. There also a comparison of the different methods with respect to their complexity and effectiveness is given.

Antithetic Variates The simplest variance reduction technique is to use antithetic variates. This technique requires no a priori information about the integrand, is simple to implement, but rarely provides much variance reduction. The MC estimate with antithetic variates is defined by

$$\mathrm{MC}_n^{\mathrm{AV}}(f) := \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{x}^i)$$

with $g(\mathbf{x}^i) = \frac{1}{2}(f(\mathbf{x}^i) + f(\mathbf{1} - \mathbf{x}^i))$.[3] Since $1 - x$ is uniformly distributed in the unit interval if x is uniformly distributed in the unit interval, this estimate converges to $I(f)$ with the same rate as the MC method (3.4). As $\mathrm{MC}_n^{\mathrm{AV}}$ requires $2n$ evaluations of the function f, antithetic sampling provides a variance reduction if

$$\sigma^2(g) \leq \frac{1}{2}\sigma^2(f).$$

This condition is satisfied for many functions from mathematical finance. It is always satisfied if f is monotonic in $\mathbf{x}$. The method of antithetic variates matches the first moment of the underlying uniform probability distribution since $\mathrm{MC}_n^{\mathrm{AV}}(f)$ has by construction the correct mean of $1/2$. Also the second and higher order moments can be matched which is known as the method of *moment matching*.

Control Variates The method of control variates is a very effective variance reduction technique which can be applied if a function g is known which is similar to f and for which Ig can be evaluated in closed form. Then,

$$\mathrm{MC}_n^{\mathrm{CV}}(f) := \frac{1}{n}\sum_{i=1}^{n}\left(f(\mathbf{x}^i) + \beta(Ig - g(\mathbf{x}^i)\right)$$

can be used as MC estimator. The parameter β can either be set to one or it can be estimated during the simulation such that the variance of $\mathrm{MC}_n^{\mathrm{CV}}(f)$ is minimized. The optimal value is thereby given by $\beta = \mathrm{Cov}(f, g)/\sigma^2(g)$. The valuation of options based on arithmetic averages are an important application of this approach. Here, their counterparts based on geometric averages turn out to be very effective control variates as they have a similar payoff and can usually be priced in closed form. More generic but less effective alternatives are to use the underlying asset or the moments of $\mathbf{x}^i$ as control variates. The method of control variates can also be interpreted as *weighted MC method*, see, e.g., [59].

Importance Sampling Importance sampling techniques try to reduce the variance of the integrand by a change the underlying probability measure. In this way, one exploits knowledge about the integrand and the MC method can focus on those regions of the integration domain which contribute most to the integral value. If the importance sampling distribution is carefully chosen then this approach can lead to an enormous improvement of the constant of the MC approximation. Important applications are barrier or out-of-the-money options. Here, sometimes the probability distributions of the underlying asset conditional on the asset being greater as the barrier or greater as the strike can be determined. The importance sampling

[3] Note that if the vector $\mathbf{x}^i$ is used to construct the path of a Brownian motion, then $\mathbf{1} - \mathbf{x}^i$ corresponds to the reflection of this path about the origin.

method then samples from this conditional probability distribution which ensures that all drawn samples are in regions where the payoff is non-zero. We will use a similar approach in Section 6.2.2 for the smoothing of some integrands from finance problems.

Stratified Sampling and Latin Hypercube Sampling Stratified and Latin hypercube sampling are methods which seek to distribute the MC samples more regular. They lie in between the plain MC method (completely random) and QMC methods (completely deterministic).[4] In the stratified sampling method, the space $[0,1]^d$ is divided into n^d disjoint subsets. From each subset then a fixed number of ℓ samples is drawn.

Since the total number of ℓn^d function evaluations grows exponentially with the dimension, stratified sampling easily becomes inapplicable in higher dimensions. Latin hypercube sampling avoids this exponential growth of costs by using permutations of suitable one dimensional stratified samples. To this end, random points $\mathbf{x}^i$ are generated for $i = 1, \ldots, n$ such that $\mathbf{x}^i$ is uniformly distributed in the hypercube $[(i-1)/n, i/n]^d$ which intersects the diagonal of the unit cube $[0,1]^d$. The coordinates of these points are then randomly permuted in order to distribute the points uniformly over the unit cube. Their coordinates thereby stay perfectly stratified[5] in the sense that any one-dimensional projection of the point set covers all intervals $[(i-1)/n, i/n]$ for $i = 1, \ldots, n$. This also indicates that Latin hypercube sampling is most efficient for integrands which are nearly additive, i.e. the sum of univariate functions. In option pricing problems, it is often efficient to stratify the underlying Brownian motion along those values which contribute most to the variability of the payoff of the option.

3.1.2 Quasi-Monte Carlo

Quasi-Monte Carlo (QMC) methods are equal-weight rules like MC. Instead of pseudo-random numbers, however, deterministic point sequences are used as nodes. These sequences are designed to yield a better uniformity than random samples. For a two-dimensional example, this property is illustrated by the grids in Figure 4.1 where 128 MC points and 128 QMC points are displayed. One can see that the pseudo-random points cluster, exhibit gaps and are not as evenly distributed as the quasi-random ones. To measure the quality of QMC points, i.e., the irregularity of their distribution in the unit cube, the notion of *discrepancy* is used. Many different discrepancy measures are possible and have been studied in the literature, see, e.g., [76, 125, 126]. Some are based on geometric properties of the point set

[4] An other class of methods that are combining random and deterministic components is the class of randomized QMC methods, see, e.g., [101].

[5] Note that many QMC methods, e.g. Sobol point sets, have the same property build in.

whereas others are based on the worst case errors for integration in certain function spaces. We give a survey in the Appendix A.2. Several different approaches to construct low-discrepancy sequences exist. Most of them are either based on *digital nets* (see, e.g., [119]) or on *lattices* (see, e.g., [98,146]). Some popular QMC methods are Halton, Faure, Sobol and Niederreiter-Xing sequences and lattice rules based on Korobov or fast component-by-component constructions [127, 149]. Their application to problems from mathematical finance is considered in, e.g., [59, 167]. The classical QMC theory is based on the Koksma-Hlawka inequality (see, e.g., [119]) which shows that the worst case error of a QMC method with n samples is of the order

$$\varepsilon(n) = O(n^{-1}(\log n)^d) \tag{3.5}$$

for integrands of bounded variation which is asymptotically better than the probabilistic $O(n^{-1/2})$ error bound of MC.[6] For periodic integrands, lattice rules can achieve convergence of higher order depending on the decay of the Fourier coefficients of f, see [146]. Using novel digital net constructions [31], QMC with higher order convergence can also be obtained for non-periodic integrands depending on the number of mixed bounded partial derivatives of f.[7] QMC methods can be randomised, e.g., by random shifts or random permutations of digits, see, e.g., [101]. In this way, MC like simple statistical error estimates can be derived while the higher QMC convergence rate is maintained.

The construction and theory of QMC point sets is beyond the scope of this thesis. We thus only briefly survey the construction of digital nets and lattice rules.

Digital nets

An important family of QMC methods are (t, m, d)–nets and (t, d)–sequences.[8] The idea of (t, m, d)–nets is to find point sets which have a perfect uniform distribution for all elementary intervals in the unit cube. An *elementary interval* in base $b \geq 2$ is a sub-interval of $[0, 1)^d$ of the form

$$\prod_{i=1}^{d} \left[\frac{a_i}{b^{c_i}}, \frac{a_i + 1}{b^{c_i}} \right)$$

with integers c_i and a_i satisfying $c_i \geq 0$ and $0 \leq a_i < b^{c_i}$. A set of b^m points in $[0, 1]^d$ is called a (t, m, d)–net in base b if every elementary interval in base b with volume b^{t-m} contains exactly b^t points, where t is an integer with $0 \leq t \leq m$. A sequence of

[6] The bound (3.5) shows that QMC points exhibit almost (up to a logarithmic factor) the smallest possible discrepancy, see Appendix A.3 for details.

[7] Efficient implementations and the practical use of these methods is not yet clear, though.

[8] For consistency reasons, we use the term (t, m, d)–net instead of the more common notation (t, m, s)–net.

points $\mathbf{x}^0, \mathbf{x}^1, \ldots$ in $[0,1]^d$ is called a (t,d)–sequence in base b if for all integers $k \geq 0$ and $m > t$ the point set

$$\left\{\mathbf{x}^{kb^m}, \ldots, \mathbf{x}^{(k+1)b^m-1}\right\}$$

is a (t,m,d)–net in base b.

For a fixed dimension d, one aims to find (t,d)–sequences with t as small as possible because the smaller t the more intervals have perfect uniform distribution. If t is small compared to m then one can show that a (t,m,d)–net defines a low discrepancy point set, which in turn means that the resulting QMC method satisfies the bound (3.5).

The digital nets provide a general framework for the construction of (t,m,d)–nets, which includes almost all of the known constructions. Next, we describe the digital net construction in the simplest case which requires a base b which is a prime number.[9] Let $\mathbb{F}_b = \{1, \ldots, b-1\}$ be a finite field with b elements and let $\mathbf{C}^i \in \mathbb{F}_b^{m \times m}$ for $i = 1, \ldots, d$. The matrices $\mathbf{C}^i$ are used to define a mapping of vectors $\mathbf{a} \in \mathbb{F}_b^m$ to points $\mathbf{x}^{\mathbf{a}} \in [0,1]^d$. Thereby, the i-th coordinate of $\mathbf{x}^{\mathbf{a}}$ is given by

$$x_i^{\mathbf{a}} := T(\mathbf{C}^i \mathbf{a}) \quad \text{with } T(\mathbf{e}) := \sum_{j=1}^{m} e_j b^{-j} \text{ for } \mathbf{e} \in \mathbb{F}_b^m.$$

Then, one can show that the point set $\{\mathbf{x}^{\mathbf{a}} : \mathbf{a} \in \mathbb{F}_b^m\}$ defines a (t,m,d)–net. The value of t, i.e., the quality of the point set, depends on the choice of the matrices $\mathbf{C}^1, \ldots, \mathbf{C}^d$. If the system of their row vectors has rank ρ over $\mathbb{F}_b$, then $t = m - \rho$ for $m \geq d$, see [119].

Sobol sequences are (t,d)–sequences in base 2 with $t = O(d \log d)$. *Faure sequences* are $(0,d)$–sequences which can be constructed in any prime base $b \geq d$. *Niederreiter sequences* are (t,d)–sequences which can be constructed for any d and any b. For fixed b, their quality parameter t can be shown to be of order $O(d \log d)$. The use of algebraic curves over finite fields with many $\mathbb{F}_b$–rational points leads to *Niederreiter-Xing sequences*. They are (t,d)–sequences which can be constructed for any d and any b and $t = O(d)$ for fixed b, which is asymptotically optimal. The *van der Corput sequence* is a $(0,1)$–sequence. It arises by the digital net construction if $C_{ij}^1 = 1$ for $j = i-1$ and $C_{ij}^1 = 0$ otherwise. Its multidimensional extension, the *Halton sequence*, is not a (t,d)–sequence, though, as its constructions uses van der Corput sequences with different prime numbers for each dimension.

Lattice rules

Lattice rules have originally been developed for the numerical integration of periodic functions. In recent years their use has been extended to the non-periodic case by the introduction of shifts that lead to the class of shifted lattice rules.

[9] The construction is easily generalised to the case where b is a prime power.

For periodic functions f, lattice rules have the advantage that the regularity of the integrand is reflected in the error bound similar as it is known for one-dimensional quadrature rules, for product methods and sparse grids, see Section 3.1.3 and Chapter 4. The regularity is thereby measured by the rate of decay of the Fourier coefficients of f. If the function f is non-periodic, periodization techniques can be used. However, the full power of lattice rules can only be preserved in special cases using this approach. Lattice rules have optimal one-dimensional projections, like many other QMC methods and like Latin hypercube sampling. They are usually not easily scalable with respect to the number of integration points and also not extensible in the dimension.[10] This drawback results since they are usually constructed for a fixed dimension d and define fixed point sets, rather than infinite sequences.

There are many different kinds of lattice rules. They can be classified according to their rank or according to their underlying construction. The family of *rank-1 lattice rules* is particular simple and interesting. Here, the lattice points are defined by

$$\mathbf{x}^i := \left\{ i\frac{\mathbf{z}}{n} \right\}$$

for $i = 1, \ldots, n$ where $\{u\} := u - \lfloor u \rfloor$ denotes the fractional part of u. The vector $\mathbf{z} \in \{1, 2, \ldots, n-1\}^d$ is the generating vector of the lattice rule. The resulting sequence $\mathbf{x}^1, \mathbf{x}^2, \ldots$ then has a period length of n, i.e., $\mathbf{x}^i = \mathbf{x}^{i+n}$.[11] It can be shown that there exist generating vectors $\mathbf{z}$ such that the point set $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ is of low discrepancy. Such points are called good lattice points. For their identification, usually computer searches are used. A full search of this kind is exponentially expensive and thus usually infeasible. Instead, either Korobov or fast component-by-component constructions [149] are used. In the first case, the search space is reduced by assuming that the generating vector is of the special form

$$\mathbf{z} = (1, a, a^2, \ldots, a^{d-1})$$

modulo n with some integer $1 \leq a \leq n-1$. In the latter approach, the generating vector is constructed component-by-component which is possible in only $O(d \log d)$ operations using the fast Fourier transformation, see [127]. For non-periodic f, shifted lattice rules are used. Here, the lattice points are defined by

$$\mathbf{x}^i := \left\{ i\frac{\mathbf{z}}{n} + \mathbf{\Delta} \right\}$$

where $\mathbf{\Delta} \in [0, 1)^d$ is the shift which is often assumed to be a random vector. For more information on shifted and non-shifted lattice rules we refer to the survey article [98] and to the references cited therein.

[10] Lattice rules which are extensible in the number of points and Korobov-form lattice rules which are extensible in the dimension are investigated in [167].

[11] A vector $\mathbf{z}$ with irrational components would lead to sequences with infinite period.

3.1.3 Product Methods

(Q)MC methods use uniformly distributed point sets and integrate constant functions exactly. We next consider product methods. Such methods are exact for polynomials of higher degree, can profit from higher smoothness of the integrand and also obtain convergence rates larger than one. Furthermore, product methods can address integrals on $\mathbb{R}^d$ directly.

Let $\mathcal{P}_l^d$ denote the space of all multivariate polynomials in d dimensions which have the maximum degree l. A quadrature rule Q_n has the *degree l of polynomially exactness* if

$$Q_n f = I f$$

for all $f \in \mathcal{P}_l^d$ and if $Q_n f \neq I f$ for at least one polynomial of degree $l+1$. In the following we denote the degree of Q_n by $\deg(Q_n)$.

Product methods are constructed by tensor products of the weights and nodes of one-dimensional quadrature rules, e.g., Gauss rules, see, e.g. [28, 153]. Let

$$U_n(f) := \sum_{i=1}^{n} w_k \, f(x_k) \tag{3.6}$$

denote a univariate quadrature rule with n points $x_k \in \Omega$, where $\Omega = [0,1]$ or $\Omega = \mathbb{R}$, and weights $w_k \in \mathbb{R}$. The d-dimensional tensor product

$$P_n(f) = U_n(f) \otimes \ldots \otimes U_n(f)$$

of such formulas is given by

$$P_n(f) := \sum_{i_1=1}^{n} \ldots \sum_{i_d=1}^{n} w_{i_1} \ldots w_{i_d} \, f(x_{i_1}, \ldots, x_{i_d}). \tag{3.7}$$

This way, a d-dimensional quadrature rule is defined which requires n^d function evaluations. We show the grid points of a two-dimensional product rule in Figure 4.1. Product constructions preserve the degree of polynomially exactness of the underlying univariate quadrature rule, i.e.

$$\deg(P_n) = \deg(U_n).$$

If the polynomial degree of the one-dimensional quadrature rules is sufficiently[12] large then the error of a product method decreases with the number of points n proportional to

$$\varepsilon(n) = O(n^{-r/d}) \tag{3.8}$$

[12] It suffice if interpolatory rules, e.g. Gauss rules, are used which are defined below.

for all functions f from the space

$$C^r([0,1]^d) = \left\{ f : [0,1]^d \to \mathbb{R} : \max_{|\mathbf{k}|_1 \leq r} \left\| \frac{\partial^{|\mathbf{k}|_1} f}{\partial {x_1}^{k_1} \cdots \partial {x_d}^{k_d}} \right\|_\infty < \infty \right\}$$

of functions with bounded derivatives up to order r, see, e.g. [28]. The error bound (3.8) indicates the positive impact of the smoothness r as well as the negative impact of the dimension d on the convergence rate. In particular one can see that product methods suffer from the *curse of dimension*, which prevents their efficient application for high-dimensional (roughly $d > 8$) applications.

Nevertheless, for low-dimensional integrals standard multivariate numerical integration methods, like ADAPT [46] or DCUHRE [10], which combine product methods with globally adaptive subdivision schemes, are among the most reliable and efficient approaches.

Univariate Quadrature Formulas

We next briefly summarise different univariate quadrature rules, either for uniform integration in the unit interval $[0,1]$ or for Gaussian integration on the real axis $\mathbb{R}$. Each of these rules corresponds via (3.7) to a different product method, either for the computation of integrals on the unit cube $[0,1]^d$ or for integrals on $\mathbb{R}^d$ with Gaussian weight. We will see in the Chapter 4 that each rule also corresponds to a different sparse grid method.

Univariate quadrature rules can be classified in various ways: For integers $n_1 \leq n_2 \leq \ldots$, a sequence $U_{n_1}, U_{n_2}, \ldots$ of quadrature rules is called *nested* if the nodes of U_{n_i} are a subset of the nodes of U_{n_j} if $i < j$. This property is of advantage if the rules are used to construct sparse grid methods. It is also important for (locally) adaptive quadrature methods, like ADAPT or DCUHRE.

An integration formula on $[0,1]$ is called of *closed type* if the function is evaluated at the end points $x = 0$ or $x = 1$ of the interval. Here, we only consider *open type* formulas since integrands which come from finance are often singular on the boundary of the unit cube.

Two important classes of univariate quadrature rules are *Newton-Cotes* and *Gauss formulas*. While Newton-Cotes formulas use arbitrary, mostly equidistant, points x_i and choose the weights w_i such all piecewise polynomials of a certain degree are integrated exactly, Gauss formulas are constructed by choosing both the points and the weights with the goal to exactly integrate as many polynomials as possible. The dimension of the space $\mathcal{P}_l^1$ of polynomials with maximum degree l is $l+1$. Hence, if there are n specific abscissas $x_1, \ldots, x_n$ given, then there exists a unique set of weights $w_1, \ldots, w_n$ such that the corresponding quadrature rule U_n is exact for all polynomials in $\mathcal{P}_{n-1}^1$. Such quadrature rules are called *interpolatory formulas* and satisfy $n - 1 \leq \deg(U_n) \leq 2n - 1$. Gauss formulas attain the optimal result $\deg(U_n) = 2n - 1$.

We start with quadrature formulas for uniform integration in the interval $[0, 1]$. Then, we proceed with quadrature rules for univariate integrals with Gaussian weight function.

Newton-Cotes formulas We here only use iterated Newton-Cotes formulas since the non-iterated versions get numerically instable for large numbers of points. Iterated Newton-Cotes formulas result from a piecewise polynomial approximation of the integrand f. This approximation is then integrated exactly. The Newton-Cotes formula which corresponds to piecewise constant approximation is the *midpoint rule*. The *trapezoidal rule* is based on a piecewise linear approximation of f. With n equidistant points it is defined as

$$U_n(f) = \frac{1}{n+1}\left(\frac{3}{2}\cdot f\left(\frac{1}{n+1}\right) + \sum_{i=2}^{n-1} f\left(\frac{i}{n+1}\right) + \frac{3}{2}\cdot f\left(\frac{n}{n+1}\right)\right).$$

For integrands $f \in C^2([0,1])$, its convergence rate is given by

$$|U_n f - If| = O(n^{-2}).$$

The trapezoidal rule is particular powerful for smooth periodic functions. In this case the error bound

$$|U_n f - If| = O(n^{-r})$$

can be derived for $f \in C^r([0,1])$ from the Euler-Maclaurin formula. The Newton-Cotes formula based on piecewise quadratic approximations finally results in the *Simpson rule*.

Clenshaw-Curtis formula The Clenshaw-Curtis formula (also known as Filippi formula) is an interpolatory integration formula whose abscissas are the extreme points of Tchebyscheff polynomials of first kind. The abscissas are nested and simple to compute. They are given by

$$x_i = \frac{1}{2}\left(1 - \cos\left(\frac{\pi i}{n+1}\right)\right)$$

for $i = 1, \ldots, n$. The weights are determined by the interpolatory conditions. For odd n one obtains

$$w_i = \frac{2}{n+1}\sin\left(\frac{\pi i}{n+1}\right)\sum_{j=1}^{(n+1)/2}\frac{1}{2j-1}\sin\left(\frac{(2j-1)\pi i}{n+1}\right).$$

This resulting quadrature formula U_n satisfies $\deg(U_n) = n - 1$ and

$$|U_n f - If| = O(n^{-r})$$

for $f \in C^r([0,1])$.

Gauss formulas *Gauss-Legendre formulas* achieve the maximum possible polynomial degree of exactness which is $2n-1$. The abscissas are the zeros of the Legendre polynomials and the weights are computed by integrating the associated Lagrange polynomials. The Gauss-Legendre formulas are not nested.

An n-point Gauss-Legendre formula can be extended by $n+1$ points to a quadrature formula with $2n+1$ points which has the polynomial degree of exactness of $3n+2$. If this extension (called Konrod extension) is iterated recursively, the *Gauss-Patterson formulas* result. These sequences are nested by construction. They achieve the polynomial degree $3/2n+1/2$ for odd $n > 1$ which is lower than the degree of Gauss-Legendre rules but considerably higher than the degree of Clenshaw-Curtis formulas. Gauss-Patterson formulas do not exist for all $n \in \mathbb{N}$, but their construction is known for $n = 1, 3, 7, 15, 63, 127$ which is usually sufficient for product and sparse grid constructions in moderate and high dimensions as shown in [48].

Gaussian-Integration Next, we consider rules for Gaussian integration. *Gauss-Hermite formulas* use nodes that are defined on the real axis $\mathbb{R}$. Their construction is known for all $n \in \mathbb{N}$, but for large n the construction easily gets unstable. Gauss-Hermite formulas achieve the optimal polynomial degree $2n-1$, but are not nested.

In [45], a sequence of quadrature rules with $n = 1, 3, 9, 19, 35$ points is presented which is nested and satisfies $\deg(U_n) \approx 3/2n$. We refer to these formulas as *Genz-Keister formulas*. They can be seen as the analog of the Gauss-Patterson rules for Gaussian integration.

We will use different univariate quadrature rules in Chapter 4 to construct *sparse grid methods*. These methods maintain the high polynomial degree of exactness of product methods to a large extent but avoid that the cost grows exponentially with increasing dimension.

Note that there exist several other interpolatory methods for multivariate integration which are neither of product- nor of sparse grid form. Like sparse grids, some of these methods can avoid an exponential increase of the grid points with the dimension. We refer to [21, 153] and for Gaussian integration in particular to [107]. The methods are often classified by the number of function evaluations which are required to exactly integrate all polynomials up to a given degree. For uniform and Gaussian integration the rules in [42, 77, 135] and [45, 77] are currently the best known constructions in the sense that they require the fewest number of points for a given polynomial degree. Note that all of these rules can be written as generalised sparse grid formulas, see, e.g., [77]. We will discuss generalised sparse grid formulas in Section 4.1.3.

3.2 Dimension-wise Quadrature Methods

Next, we use the anchored-ANOVA decomposition from Section 2.2 to define a new class of methods for the computation of high-dimensional integrals

$$If = \int_{[0,1]^d} f(\mathbf{x})\, d\mathbf{x} \tag{3.9}$$

over the unit cube and for integrals

$$I_\varphi f = \int_{\mathbb{R}^d} f(\mathbf{z})\, \varphi_d(\mathbf{z})\, d\mathbf{z} \tag{3.10}$$

over the d-dimensional Euclidean space with Gaussian weight.

3.2.1 Truncation and Discretization

In the following, we develop our new class of quadrature methods. The methods result by truncation and discretization of the anchored-ANOVA decomposition

$$f(\mathbf{x}) = \sum_{\mathbf{u}\subseteq\mathcal{D}} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$$

of the integrand f, see Section 2.2. We start with $\Omega = [0,1]$ and take μ as the Dirac measure with anchor point $\mathbf{a} \in [0,1]^d$. Then, (2.2) and (2.4) imply

$$f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) - \sum_{\mathbf{v}\subset\mathbf{u}} f_{\mathbf{v}}(\mathbf{x}_{\mathbf{v}}) \quad \text{where } P_{\mathbf{u}} f(\mathbf{x}_{\mathbf{u}}) = f(\mathbf{x})|_{\mathbf{x}=\mathbf{a}\setminus\mathbf{x}_{\mathbf{u}}}. \tag{3.11}$$

Applying the integral operator to the anchored-ANOVA decomposition (2.3), the d-dimensional integral is decomposed, by linearity, into the finite sum

$$If = \sum_{\mathbf{u}\subseteq\mathcal{D}} If_{\mathbf{u}} = f(\mathbf{a}) + \sum_{i=1}^{d} If_i + \sum_{i,j=1}^{d} If_{i,j} + \ldots + If_{1,\ldots,d} \tag{3.12}$$

which contains $\binom{d}{j}$ many j-dimensional integrals for $j = 0,\ldots,d$. Starting with the decomposition (3.12) we now define a general class of quadrature methods for the approximation of If. We proceed as follows:

1. *Truncation:* We take only a subset $\mathcal{S}$ of all indices $\mathbf{u} \subseteq \mathcal{D}$ and thus truncate the sum in (3.12). Here, we assume that the set $\mathcal{S}$ satisfies the property[13]

$$\mathbf{u} \in \mathcal{S} \text{ and } \mathbf{v} \subset \mathbf{u} \quad \Longrightarrow \quad \mathbf{v} \in \mathcal{S}. \tag{3.13}$$

[13] Note that this condition is closely related to the admissibility condition (4.11) in Section 4.1.3.

For example, the set $\mathcal{S}_{d_s} := \{\mathbf{u} \subseteq \mathcal{D} : |\mathbf{u}| \leq d_s\}$ or the set $\mathcal{S}_{d_t} := \{\mathbf{u} \subseteq \{1, \ldots, d_t\}\}$ could be used to take into account the superposition and the truncation dimension of the function f, respectively. Alternatively, dimension-adaptive methods can be applied to build up an appropriate index set $\mathcal{S}$. This will be later discussed in more detail.

2. *Discretization:* For each $\mathbf{u} \in \mathcal{S}$, we compute approximations to $If_{\mathbf{u}}$. To this end, we choose $|\mathbf{u}|$-dimensional quadrature rules $Q_{\mathbf{u}}$. Starting with $q_{\emptyset} = f(\mathbf{a})$ we recursively compute

$$q_{\mathbf{u}} := Q_{\mathbf{u}}(P_{\mathbf{u}}f) - \sum_{\mathbf{v} \subset \mathbf{u}} q_{\mathbf{v}}. \tag{3.14}$$

Then, $q_{\mathbf{u}}$ is an approximation to $If_{\mathbf{u}}$ due to the recursive representation (3.11) of $f_{\mathbf{u}}$. Observe that we avoid to compute and integrate the functions $f_{\mathbf{u}}$ explicitly. Instead we numerically integrate $P_{\mathbf{u}}f$ and correct the resulting value by the (previously computed) values $q_{\mathbf{v}}$ using (3.11). The admissibility condition thereby ensures that we can run over the set $\mathcal{S}$ by starting with $\mathbf{u} = \emptyset$ and proceeding with indices $\mathbf{u}$ for which the values $q_{\mathbf{v}}$, $\mathbf{v} \subset \mathbf{u}$, have already be computed in previous steps. Note that we allow for arbitrary quadrature methods $Q_{\mathbf{u}}$ in (3.14) which can be different for each $\mathbf{u}$. Specific choices for $Q_{\mathbf{u}}$ will be discussed later.

Altogether, this defines a quadrature formula $A_{\mathcal{S}}f$ for the approximation of If which is given by

$$A_{\mathcal{S}}f := \sum_{\mathbf{u} \in \mathcal{S}} q_{\mathbf{u}} \tag{3.15}$$

and which we refer to as *dimension-wise quadrature method* in the following. Note that the method $A_{\mathcal{S}}f$ requires

$$n = \sum_{\mathbf{u} \in \mathcal{S}} n_{\mathbf{u}}$$

evaluations of the function f, where $n_{\mathbf{u}}$ denotes the number of function evaluations of $Q_{\mathbf{u}}$.

Remark 3.2. Dimension-wise quadrature methods for integrals over $\mathbb{R}^d$ with Gaussian weight can be constructed analogously to (3.15). To this end, we set $\Omega = \mathbb{R}$ and use the measure $d\mu(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{a})\varphi_d(\mathbf{x})d\mathbf{x}$ such that the anchored-ANOVA decomposition with Gaussian weighted results. Then, we select as above a suitable index set $\mathcal{S}$ and appropriate quadrature rules $Q_{\mathbf{u}}$ to integrate the resulting functions $P_{\mathbf{u}}f$. Since now $f : \mathbb{R}^d \to \mathbb{R}$, either quadrature rules for unbounded domains, e.g. Gauss-Hermite rules, or transformations of the resulting integrals to the unit cube must be used.

3.2.2 Error and Costs

We first consider the case of arbitrary quadrature methods $Q_{\mathbf{u}}$. By construction, we then have the error bound

$$|If - A_{\mathcal{S}}f| = |\sum_{\mathbf{u}\subseteq\mathcal{D}} If_{\mathbf{u}} - \sum_{\mathbf{u}\in\mathcal{S}} q_{\mathbf{u}}| \leq \sum_{\mathbf{u}\in\mathcal{S}} |If_{\mathbf{u}} - q_{\mathbf{u}}| + \sum_{\mathbf{u}\notin\mathcal{S}} |If_{\mathbf{u}}|. \tag{3.16}$$

This shows how the error of the method (3.15) depends on the quadrature rules $Q_{\mathbf{u}}$ (which determine $q_{\mathbf{u}}$) and on the choice of the index set $\mathcal{S}$. Here, the second term describes the *modeling error* which is introduced by the truncation of the anchored-ANOVA series whereas the first term describes the *discretization error* which results from the subsequent discretization of the remaining subspaces.

In the following, we aim to balance costs and accuracies by relating the cost of the quadrature method $Q_{\mathbf{u}}$ to the importance of the anchored-ANOVA term $f_{\mathbf{u}}$. We first relate the accuracy of the methods $Q_{\mathbf{u}}$ to the accuracy of the method $A_{\mathcal{S}}f$.

To this end, we fix $\alpha \in (0, 1]$ and assume that d_s and d_t, the corresponding superposition and truncation dimensions in the anchored case, are known. With help of these effective dimensions we define the index set

$$\mathcal{S}_{d_t,d_s} := \{\mathbf{u} \subseteq \{1, \dots, d_t\} : |\mathbf{u}| \leq d_s\}. \tag{3.17}$$

We now have the following lemma.

Lemma 3.3. *Let $\mathcal{S} = \mathcal{S}_{d_t,d_s}$. For $\varepsilon > 0$, let furthermore $Q_{\mathbf{u}}$ be such that $|I(P_{\mathbf{u}}f) - Q_{\mathbf{u}}(P_{\mathbf{u}}f)| \leq \varepsilon(|\mathbf{u}|)$ with $\varepsilon(j) := \varepsilon/(2\, d_t^{d_s} \binom{d_t}{j})$ for all $\mathbf{u} \in \mathcal{S}$. Then, it holds*

$$|If - A_{\mathcal{S}}f| \leq \varepsilon + 2(1-\alpha)\,\hat{\sigma}(f).$$

Proof. Starting with $|If - A_{\mathcal{S}}f| \leq |If - If_{d_t,d_s}| + |If_{d_t,d_s} - A_{\mathcal{S}}f|$ with the function $f_{d_t,d_s} := \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} f_{\mathbf{u}}$, we observe that the modelling error is bounded by

$$|If - If_{d_t,d_s}| = |\sum_{\mathbf{u}\notin\mathcal{S}_{d_t,d_s}} If_{\mathbf{u}}| \leq \sum_{|\mathbf{u}|>d_s} |If_{\mathbf{u}}| + \sum_{\mathbf{u}\not\subset\{1,\dots,d_t\}} |If_{\mathbf{u}}| \leq 2(1-\alpha)\,\hat{\sigma}(f),$$

see the proofs of Lemma 2.7 and 2.8. From (2.5) and (3.14), we have the explicit representation

$$If_{\mathbf{u}} - q_{\mathbf{u}} = \sum_{\mathbf{v}\subseteq\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \left(I(P_{\mathbf{v}}f) - Q_{\mathbf{v}}(P_{\mathbf{v}}f)\right). \tag{3.18}$$

Since $|I(P_{\mathbf{v}}f) - Q_{\mathbf{v}}(P_{\mathbf{v}}f)| \leq \varepsilon(|\mathbf{v}|)$ for all $\mathbf{v} \subseteq \mathbf{u}$, we obtain for all $\mathbf{u}$ with $|\mathbf{u}| \leq d_t$ that

$$|If_{\mathbf{u}} - q_{\mathbf{u}}| \leq \sum_{\mathbf{v}\subseteq\mathbf{u}} \varepsilon(|\mathbf{v}|) = \sum_{j=1}^{|\mathbf{u}|} \binom{|\mathbf{u}|}{j} \varepsilon(j) \leq \sum_{j=1}^{|\mathbf{u}|} \binom{d_t}{j} \varepsilon(j),$$

where we used that there are $\binom{|\mathbf{u}|}{j}$ many sets $\mathbf{v} \subseteq \mathbf{u}$ which satisfy $|\mathbf{v}| = j$. Using the definition of $\varepsilon(j)$, we can bound the discretization error by

$$\begin{aligned} |If_{d_t,d_s} - A_{\mathcal{S}_{d_t,d_s}} f| &\leq \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} |If_{\mathbf{u}} - q_{\mathbf{u}}| \leq \sum_{k=1}^{d_s} \binom{d_t}{k} \sum_{j=1}^{k} \binom{d_t}{j} \varepsilon(j) \leq \sum_{k=1}^{d_s} \binom{d_t}{k} \sum_{j=1}^{k} \frac{\varepsilon}{2\, d_t^{d_s}} \\ &= \frac{\varepsilon}{2\, d_t^{d_s}} \sum_{k=1}^{d_s} \binom{d_t}{k} k \leq \frac{\varepsilon}{2\, d_t^{d_s}} \sum_{k=1}^{d_s} \frac{d_t^{d_s}}{k!} k = \frac{\varepsilon}{2} \sum_{k=1}^{d_s} \frac{1}{(k-1)!} \leq \varepsilon, \end{aligned}$$

which concludes the proof. ∎

We next relate the error $|If - A_{\mathcal{S}} f|$ to the cost $n = \sum_{\mathbf{u}\in\mathcal{S}} n_{\mathbf{u}}$ of the method $A_{\mathcal{S}} f$. Furthermore, we aim to balance the cost $n_{\mathbf{u}}$ of the methods $Q_{\mathbf{u}}$ with their accuracy. Here, we restrict ourselves to the case that all employed methods $Q_{\mathbf{u}}$ are based on a univariate quadrature formula U_m with m points, which converges for $f \in C^r([0,1])$ with rate m^{-r}. An examples for such a univariate formula with $r = 1$ is the trapezoidal rule. For arbitrary r, Gauss rules can be used.

Theorem 3.4 (ε-complexity). *If we choose $\mathcal{S} = \mathcal{S}_{d_t,d_s}$ and $Q_{\mathbf{u}}$ to be the $|\mathbf{u}|$-dimensional tensor product of the rule U_m with $m := \lfloor n^{1/d_s} \rfloor$ then*

$$|If - A_{\mathcal{S}} f| \leq c(d_t, d_s)\, n^{-r/d_s} + 2(1-\alpha)\, \hat{\sigma}(f)$$

for all $f \in C^r([0,1]^d)$. Here, the constant $c(d_t, d_s)$ depends on the effective dimensions d_t and d_s in the anchored case, but not on the nominal dimension d.

Proof. We have the same modelling error as in Lemma 3.3, i.e.,

$$|If - A_{\mathcal{S}} f| \leq |If_{d_t,d_s} - A_{\mathcal{S}} f| + 2(1-\alpha)\, \hat{\sigma}(f)$$

with the function $f_{d_t,d_s} := \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} f_{\mathbf{u}}$. Since $f \in C^r([0,1]^d)$ also $f_{\mathbf{u}} \in C^r([0,1]^{|\mathbf{u}|})$ for all $\mathbf{u} \subseteq \mathcal{D}$. Consequently, $Q_{\mathbf{u}}$ converge with rate $r/|\mathbf{u}|$. By definition, $Q_{\mathbf{u}}$ requires $n_{\mathbf{u}} = \lfloor n^{|\mathbf{u}|/d_s} \rfloor$ function evaluations such that

$$|I(P_{\mathbf{u}} f) - Q_{\mathbf{u}}(P_{\mathbf{u}} f)| \leq c(|\mathbf{u}|)\, n_{\mathbf{u}}^{-r/|\mathbf{u}|} \leq c(|\mathbf{u}|)\, n^{-r/d_s}$$

for a constant $c(|\mathbf{u}|) > 0$ which depends on the order $|\mathbf{u}|$. By (3.18) we have

$$\begin{aligned} |If_{d_t,d_s} - A_{\mathcal{S}_{d_t,d_s}} f| &\leq \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} |If_{\mathbf{u}} - q_{\mathbf{u}}| \leq \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} \sum_{\mathbf{v}\subseteq\mathbf{u}} |I(P_{\mathbf{v}} f) - Q_{\mathbf{v}}(P_{\mathbf{v}} f)| \\ &\leq \sum_{\mathbf{u}\in\mathcal{S}_{d_t,d_s}} \sum_{\mathbf{v}\subseteq\mathbf{u}} c(|\mathbf{v}|)\, n^{-r/d_s} = \sum_{k=1}^{d_s} \binom{d_t}{k} \sum_{j=1}^{k} \binom{k}{j} c(j)\, n^{-r/d_s} \end{aligned}$$

which is bounded by $c(d_t, d_s)\, n^{-r/d_s}$ with the constant

$$c(d_t, d_s) := \sum_{k=1}^{d_s} \binom{d_t}{k} \sum_{j=1}^{k} \binom{k}{j} c(j) \leq \max_{j=1,\ldots,d_s} c(j) \sum_{k=1}^{d_s} \binom{d_t}{k} 2^k \leq 2 \max_{j=1,\ldots,d_s} c(j)\, 2^{d_s} d_t^{d_s}.$$

This completes the proof. □

Note that the first term in the error bound describes the discretization error which depends on n whereas the second term corresponds to the modeling error which depends on the proportion α. Note furthermore that the cost to obtain a prescribed discretization error does not exponentially depend on the nominal dimension d, but only on the superposition dimension d_s in the anchored case.

3.2.3 A priori Construction

In applications, the effective dimensions of f are usually unknown. These dimensions can also not be computed since this would be at least as expensive as the integration of f. In general it is thus difficult to determine the set $\mathcal{S}_{d_t,d_s}$ in (3.17).

To overcome this obstacle, we here assume that the integrand is contained in some function class that is defined by certain *function space weights* $\gamma_{\mathbf{u}} \geq 0$, which describe the importance of the term $f_{\mathbf{u}}$ of the anchored-ANOVA decomposition. Using this a priori information, we then determine the index set $\mathcal{S}$ by including those indices $\mathbf{u}$ which correspond to the largest weights $\gamma_{\mathbf{u}}$. In the following, we use the set

$$\mathcal{S}_{\boldsymbol{\gamma}} := \{\mathbf{u} \subseteq \mathcal{D} : \gamma_{\mathbf{u}} > \varepsilon\}$$

which includes all indices $\mathbf{u}$ for which $\gamma_{\mathbf{u}}$ is larger than some threshold ε. We consider two different approaches to define the weights: *order-dependent weights* for functions f with low superposition dimension and *product weights* for functions f with low truncation dimension.

- *Order-dependent Weights:* We define the order-dependent weights $\gamma_{\mathbf{u}} = 1/|\mathbf{u}|$. Then, the indices are added according to their order $|\mathbf{u}|$ and, by construction, the admissibility condition (3.13) is always satisfied. In this way the resulting index set $\mathcal{S}_{\boldsymbol{\gamma}}$ is tailored to the superposition dimension d_s of the function f.

- *Product Weights:* As in [148], we assume that the dimensions are ordered according to their importance which is modulated by a sequence of weights

  $$\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_d \geq 0.$$

  Using the weights γ_i, we then assign the product weight

  $$\gamma_{\mathbf{u}} := \prod_{j \in \mathbf{u}} \gamma_j \tag{3.19}$$

to the index $\mathbf{u} \subseteq \mathcal{D}$. The weights γ_i can here either be input parameters of the algorithm similar as in the CBC construction of lattice rules [149] or they can be derived from the first order terms f_j of the anchored-ANOVA decomposition. In the latter case they are defined by

$$\gamma_j := \frac{|q_j|}{|q_\emptyset|} = \frac{|Q_j(P_j f) - f(\mathbf{a})|}{|f(\mathbf{a})|}$$

for $j = 1, \ldots, d$, see (3.14). If the size of the product weights corresponds to the importance of the anchored-ANOVA terms then the resulting index set $\mathcal{S}_{\boldsymbol{\gamma}}$ includes the most important contributions to the integral value.

We will use these weights and the resulting index sets $\mathcal{S}_{\boldsymbol{\gamma}}$ in our numerical experiments in Section 6.2. Note that also more general weights can be used in our construction as long as the admissibility condition (3.13) is satisfied. Nevertheless, we here only shifted the problem of the choice of $\mathcal{S}$ to the problem of determining the weights $\gamma_{\mathbf{u}}$. In Section 3.2.4, we will consider a different approach. There, we will determine the index set $\mathcal{S}$ a posteriori in a dimension-adaptive way.

3.2.4 Dimension-adaptive Construction

In Section 3.2.3, we shifted the problem of the choice of the index set $\mathcal{S}$ to the problem of determining the weights $\gamma_{\mathbf{u}}$. In practise, however, the weights are usually unknown and can also not be computed as this would be more expensive than computing the integral.

In these cases adaptive algorithms are required which can estimate the weights a posteriori during the actual calculation of the integral. This way appropriate index sets $\mathcal{S}$ can be constructed automatically for a given function f without any a priori information on the dimension structure of the integrand being required. Such algorithms were presented in [49] to determine optimal index sets of generalised sparse grid methods, see Section 4.1.4.

We next consider the same approach to determine the index set $\mathcal{S}$ in a dimension-adaptive fashion. To estimate the importance of the term $I(f_{\mathbf{u}})$ we define $\gamma_{\mathbf{u}} := |q_{\mathbf{u}}| \approx |I(f_{\mathbf{u}})|$ with $q_{\mathbf{u}}$ from (3.14). Furthermore, we denote by $\mathcal{A}$ the subset of all indices $\mathbf{u} \in \mathcal{D} \setminus \mathcal{S}$ which satisfy the admissibility condition (3.13) with respect to the set $\mathcal{S}$. With help of the weights $\gamma_{\mathbf{u}}$, we then heuristically build up the index set $\mathcal{S}$ in a general and automatic way by the following Greedy-approach: We start with the initial set $\mathcal{S} = \{\emptyset\}$ and add step by step the index $\mathbf{u} \in \mathcal{A}$ with the largest weight $\gamma_{\mathbf{u}}$ to $\mathcal{S}$ until the largest weight is below some threshold ε, see Algorithm 3.1.

Note that in the first step of Algorithm 3.1 the values $q_{\mathbf{u}}$ have to be computed for all $\mathbf{u} \in \mathcal{A}$ for which $q_{\mathbf{u}}$ has not yet been computed in previous steps. In high

Algorithm 3.1: Dimension-adaptive constructions of the index set $\mathcal{S}$.

Initialise: Let $\mathcal{S} = \{\emptyset\}$, $q_\emptyset := f(\mathbf{a})$ and $s = f(\mathbf{a})$.

repeat

1) Compute the values $q_\mathbf{u}$ from (3.14) for all $\mathbf{u} \in \mathcal{A}$ for which $q_\mathbf{u}$ has not yet been computed and set $s = s + q_\mathbf{u}$.

2) Move the index $\mathbf{u} \in \mathcal{A}$ with the largest weight $\gamma_\mathbf{u} = |q_\mathbf{u}|$ from $\mathcal{A}$ to $\mathcal{S}$.

until $\gamma_\mathbf{u} \leq \varepsilon$;

Set $A_\mathcal{S} f = s$.

dimensions d, this may result in a certain overhead since not all of these values significantly contribute to the integral value.

Example 3.5. We next provide specific examples of dimension-wise quadrature methods of the form (3.15) which can refine the approximation of the integral in both a locally adaptive and a dimension-adaptive way. To this end, recall that we still have to specify the quadrature rules $Q_\mathbf{u}$ and the index set $\mathcal{S}$ to finalize the construction of the method (3.15). Here we choose the $Q_\mathbf{u}$'s as follows: If $|\mathbf{u}| < 4$, we use the locally-adaptive product method CUHRE, see [10, 68], to address low regularity of the terms $P_\mathbf{u} f$. If $|\mathbf{u}| \geq 4$, we use a randomised quasi-Monte Carlo method based on Sobol point sets to lift the dependence on the dimension. We always use the anchor point $(1/2, \ldots, 1/2)$. For the construction of the index set $\mathcal{S}$ (i.e. for finding the most important terms of the anchored-ANOVA decomposition) we employ the a priori constructions from the previous section and the a posteriori construction from this section. This defines the following three new quadrature methods:

- mixed CUHRE/QMC method with order-dependent weights (COW),
- mixed CUHRE/QMC method with product weights (CPW),
- mixed CUHRE/QMC method with dimension-adaptivity (CAD).

We refer to COW, CPW and CAD as *mixed CUHRE/QMC methods*. To our knowledge these methods are the first numerical quadrature methods which can profit from low effective dimension (by the selection of appropriate function space weights or by dimension-adaptivity) and which can at the same time resolve low regularity to some extent by local adaptivity. We will study the performance of these methods in Section 6.2 for high-dimensional integrals as they arise in option pricing problems.

We conclude this chapter with a short summary. We here defined dimension-wise quadrature methods which resulted by truncation and discretization of the anchored-ANOVA decomposition. We discussed a priori and a posteriori approaches for the truncation and derived bounds for the resulting modeling and discretization errors. The results indicated that the curse of dimension can be avoided for the class of functions which is of low effective dimension in the anchored case. We will return to dimension-wise quadrature methods in the next chapter and show that sparse grid methods can be regraded as a special case of this general approach.

Chapter 4
Sparse Grid Quadrature

This chapter is concerned with sparse grid (SG) quadrature methods. These methods are constructed using certain combinations of tensor products of one-dimensional quadrature rules. They can exploit the smoothness of f, overcome the curse of dimension to a certain extent and profit from low effective dimensions, see, e.g., [18, 47, 48, 61, 121, 150].

The remainder of this chapter is structured as follows: Section 4.1 is concerned with the construction of several variants (classical, delayed, generalised, dimension-adaptive) of the general sparse grid approach. Then, in Section 4.2 we derive optimal generalised sparse grid constructions for integrands from weighted Sobolev spaces. Finally, in Section 4.3, we relate sparse grids to the anchored-ANOVA decomposition and show that sparse grids methods can be regarded as special cases of our general class (3.15) of dimension-wise quadrature methods.

4.1 Sparse Grid Methods

The sparse grid approach is based on a decomposition of the d-dimensional integral If into a infinite telescope sum and on a truncation of this sum which balances work and accuracy. Different ways to truncate the sum thereby correspond to different sparse grid constructions such as the *classical construction* [150], the *delayed construction* [135], the *generalised construction* [48, 71, 137, 165] and the *dimension-adaptive construction* [49] which we will summarise in the next sections.

4.1.1 Classical Construction

Sparse grids (SG) can be defined for general tensor product domains $\Omega^d \subseteq \mathbb{R}^d$. We here consider the case $\Omega = [0,1]$, but most results can be generalised in straightforward way to other domains. We will give remarks on the case $\Omega = \mathbb{R}$.

For a univariate function $f : [0,1] \to \mathbb{R}$ and a sequence of non-decreasing integers m_k, $k \in \mathbb{N}$, let

$$U_{m_k} f := \sum_{i=1}^{m_k} w_{i,k}\, f(x_{i,k}) \tag{4.1}$$

denote a sequence of univariate quadrature rules with m_k points $x_{i,k}$ and weights $w_{i,k}$, which converges pointwise to If for $k \to \infty$. We assume $m_1 = 1$ and $U_{m_1} f = f(1/2)$

and define the difference quadrature formulae

$$\Delta_k := U_{m_k} - U_{m_{k-1}} \quad \text{with } U_{m_0} := 0 \tag{4.2}$$

for $k \geq 1$.

Now let $f : [0,1]^d \to \mathbb{R}$ be a multivariate function. Then, the d-dimensional integral If can be represented by the infinite telescoping sum

$$If = \sum_{\mathbf{k} \in \mathbb{N}^d} \boldsymbol{\Delta}_{\mathbf{k}} f \tag{4.3}$$

which collects the products of each possible combination of the univariate difference formulae. Here, $\mathbf{k} \in \mathbb{N}^d$ denotes a multi-index with $k_j > 0$ and

$$\boldsymbol{\Delta}_{\mathbf{k}} f := (\Delta_{k_1} \otimes \ldots \otimes \Delta_{k_d})\, f. \tag{4.4}$$

For a given level $\ell \in \mathbb{N}$, the *classical sparse grid method*, often also denoted as Smolyak method, see, e.g., [48, 121, 150], is then defined by

$$\mathrm{SG}_\ell f := \sum_{|\mathbf{k}|_1 \leq \ell + d - 1} \boldsymbol{\Delta}_{\mathbf{k}} f \tag{4.5}$$

where $|\mathbf{k}|_1 := \sum_{j=1}^d k_j$.

From the set of all possible indices $\mathbf{k} \in \mathbb{N}^d$ thus only those are considered whose $|\cdot|_1$-norm is smaller than a constant. Note that the product approach is recovered if instead of the $|\cdot|_1$-norm in (4.5) the norm $|\cdot|_\infty := \max\{k_1, \ldots, k_d\}$ is used for the selection of indices.

The SG method (4.5) can exploit the smoothness of f and in this way also obtain convergence rates larger than one. In addition, it can overcome the curse of dimension to a certain extent. The function classes for which this is possible are the spaces

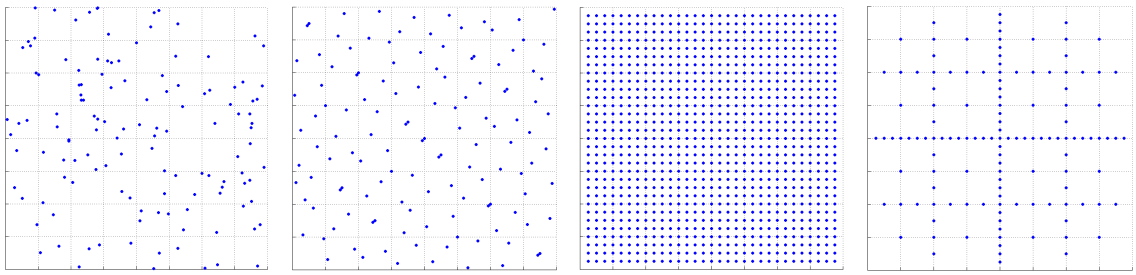
$$H^r([0,1]^d) := \left\{ f :\ [0,1]^d \to \mathbb{R} :\ \max_{|\mathbf{k}|_\infty \leq r} \left\| \frac{\partial^{|\mathbf{k}|_1} f}{\partial {x_1}^{k_1} \ldots \partial {x_d}^{k_d}} \right\|_\infty < \infty \right\}$$

of functions which have bounded mixed partial derivatives of order r. The error of the classical SG quadrature formula is bounded by

$$\varepsilon(n) = O(n^{-r} (\log n)^{(d-1)(r-1)}) \tag{4.6}$$

for all $f \in H^r([0,1]^d)$.[1] The convergence rate is thus independent of the dimension up to a logarithmic factor. For analytic functions even spectral convergence is

[1] We will give a proof of this result for the case $r = 1$ in Section 4.2.



(a) Monte Carlo (b) Quasi-MC (c) Product grid (d) Sparse grid

Figure 4.1. *Sample grids of four different classes of methods for multivariate numerical integration. The QMC points come from the Sobol low discrepancy sequence. The product grid and the sparse grid are based on the trapezoidal rule with* 31 *nodes.*

observed. In addition, the SG method (4.5) achieves (up to logarithmic factors) optimal rates of convergence for the classical spaces $C^r([0,1]^d)$ which we defined in Section 3.1.3. These rates depend exponentially on the dimension, however, see [121].

SG quadrature formulas come in various types depending on the one-dimensional basis integration routine. SG methods with *Clenshaw-Curtis* rules are used in [121]. *Gauss-Patterson* and *Gauss-Legendre* formulas are investigated in [48]. For Gaussian integration, SG methods with *Gauss-Hermite* and *Genz-Keister* rules are used in [118, 123] and [73], respectively. In many cases, the performance of the classical sparse grid method can be enhanced by the use of *delayed sequences* of one-dimensional quadrature rules (see Section 4.1.2), by *spatial adaptivity* [11, 16] or by a *dimension-adaptive refinement* (see Section 4.1.4).

In Figure 4.1, the two-dimensional product grid and the corresponding sparse grid are displayed which result if the trapezoidal rule with 31 nodes is used as univariate quadrature rule. One can see that the sparse grid contains much fewer nodes than the corresponding product grid. This effect is even more pronounced in higher dimensions.

Remark 4.1. Sparse grid methods can directly be applied to the numerical computation of integrals on $\mathbb{R}^d$ with Gaussian weight. To this end, only the sequence of univariate quadrature rules U_{m_k} must be replaced by quadrature formulas for functions $f : \mathbb{R} \to \mathbb{R}$ on unbounded domains, such as Gauss-Hermite or Genz-Keister rules, see, e.g., [73, 118, 123].

4.1.2 Delayed Basis Sequences

This section is concerned with an optimization of the classical sparse grid construction with respect to its polynomial degree of exactness.

To this end, we denote by $\mathcal{P}_\ell^d$ the space of all multivariate polynomials in d dimensions which have the maximum degree ℓ. We assume that the univariate formulas U_{m_ℓ} are exact for the space $\mathcal{P}_\ell^1$. Then the classical sparse grid quadrature rule SG_ℓ is exact for the non-classical space of polynomials

$$\tilde{\mathcal{P}}_\ell^d := \left\{ \mathcal{P}_{k_1}^1 \otimes \ldots \otimes \mathcal{P}_{k_d}^1 : |\mathbf{k}|_1 = \ell + d - 1 \right\},$$

see [121]. The method also preserves the classical polynomial exactness (based on the space $\mathcal{P}_\ell^d$) of the univariate quadrature rules to a certain extend as the following lemma shows. The lemma is taken from [122].

Lemma 4.2. *Let m_i, $i \in \mathbb{N}$, be a non-decreasing sequence of integers. Let further U_{m_i} denote a univariate quadrature rule with m_i points and*

$$\deg(U_{m_i}) \geq 2i - 1. \tag{4.7}$$

Then,

$$\deg(SG_\ell) \geq 2\ell - 1, \tag{4.8}$$

where SG_ℓ is the sparse grid method (4.5) based on the sequence U_{m_i}.

Proof. The proof proceeds via induction over d using dimension recursive formulations [165] of the method SG_ℓ. We refer to [122] and [135] for details. □

The classical SG quadrature rule thus integrates all polynomials up to a certain degree exactly. The number of points thereby only increases polynomially with the dimension. Nevertheless, the number of points increases still rather fast. In [135], modifications of the classical SG construction are presented which have the same degree of polynomial exactness with a fewer number of points. To construct these methods, *delayed basis sequences* are used which result from a repetition of quadrature formulas in the univariate basis sequences.

Delayed Gauss-Patterson sequences In the classical SG construction the sequence $m_i = 2^i - 1$ is used. If U_{m_i} is the Gauss-Patterson rule then $\deg(U_{m_i}) = 3/2m_i + 1/2$ for $i > 1$. The degree of exactness thus increases considerably faster than $2i - 1$, see Table 4.1(a). To avoid this overhead, it is proposed in [135] to base the SG construction on a different sequence $\tilde{m}_i$ which increases slower with i than the sequence m_i. The sequence $\tilde{m}_i$ is thereby determined as

$$\tilde{m}_i = 1, 3, 3, 7, 7, 7, 15, \ldots \quad \text{for } i = 1, 2, \ldots \tag{4.9}$$

in such a way that $\deg(U_{\tilde{m}_i})$ increases as slow as possible but is always larger than $2i-1$. This way, the number of points of SG_ℓ increases much slower with the level ℓ while (4.8) holds as in the classical SG construction. The degrees of polynomial exactness of U_{m_i} and $U_{\tilde{m}_i}$ are shown in Table 4.1(a) for $i = 1, \ldots, 7$.

Table 4.1. *Degrees of polynomial exactness of different univariate quadrature rules with the classical sequence m_i and with the delayed sequence $\tilde{m}_i$. The lower bound is $2i-1$.*

(a) Gauss-Patterson

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	11	23	47	95	191
$\deg(U_{\tilde{m}_i})$	1	5	5	11	11	11	13
$2i-1$	1	3	5	7	9	11	13

(b) Gauss-Legendre

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	13	29	61	125	252
$\deg(U_{\tilde{m}_i})$	1	3	5	7	9	11	13
$2i-1$	1	3	5	7	9	11	13

(c) Gauss-Hermite

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	13	29	61	125	252
$\deg(U_{\tilde{m}_i})$	1	3	5	7	9	11	13
$2i-1$	1	3	5	7	9	11	13

(d) Genz-Keister

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	15	29	51	-	-
$\deg(U_{\tilde{m}_i})$	1	5	5	15	15	15	15
$2i-1$	1	3	5	7	9	11	13

It is shown in [135] that the SG method with the sequence $U_{\tilde{m}_i}$ is almost asymptotically optimal and that there are only minor improvements possible. Also the numerical experiments presented in [135] indicate that the delayed constructions are more efficient than their non-delayed counterparts provided smooth functions are integrated, since such functions are well approximated by polynomials.

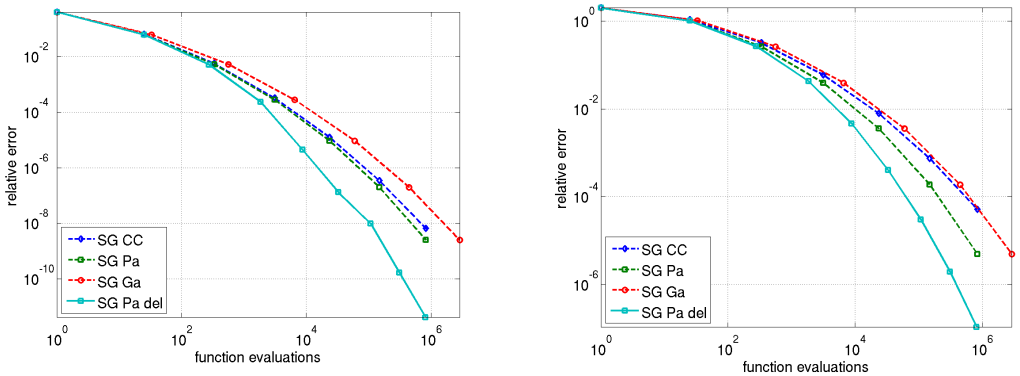
We observe the same effect in Figure 4.2 where we compare several SG variants for the integration of the quadrature test functions

$$f_1(\mathbf{x}) := \exp\left(-\sum_{j=1}^{d} x_i^2\right) \quad \text{and} \quad f_2(\mathbf{x}) := \prod_{j=1}^{d}\left(1 + \left(\frac{3}{4}\right)^j \left(\frac{\pi}{4}\sin(\pi x_j) - \frac{1}{2}\right)\right)$$

with $d = 8$. The function f_1 is from the testing package of Genz [43]. The function f_2 is a slight modification[2] of a test function from [161].

Slowly increasing Gauss-Legendre sequences The approach from [135] is not restricted to SG methods based on Gauss-Patterson formulas. It can be used to determine improved sequences $\tilde{m}_i$ for SG methods based on arbitrary univariate

[2] In [161] specific polynomials are used to study the performance of QMC methods for integrands from weighted spaces. Since SG methods can integrate these polynomials exactly we replaced all monomials x_j by $\frac{\pi}{4}\sin(\pi x_j)$.



(a) *Function* f_1 (b) *Function* f_2

Figure 4.2. *Errors and required number of function evaluations of the SG method with Clenshaw-Curtis (SG CC), Gauss-Patterson (SG Pa), Gauss-Legendre (SG Ga) and delayed Gauss-Patterson (SG Pa del).*

quadrature rules. To this end, we determine the slowest increasing sequence $\tilde{m}_i$, $i = 1, 2, \ldots$, which fulfils (4.7). The sparse grid method corresponding to this sequence then satisfies the condition (4.8) as in the classical case but requires fewer points.

For the Gauss-Legendre rule, which has a higher polynomial degree than the Gauss-Patterson rule, but is not nested, we obtain

$$\tilde{m}_i = i \quad \text{for } i = 1, 2, \ldots.$$

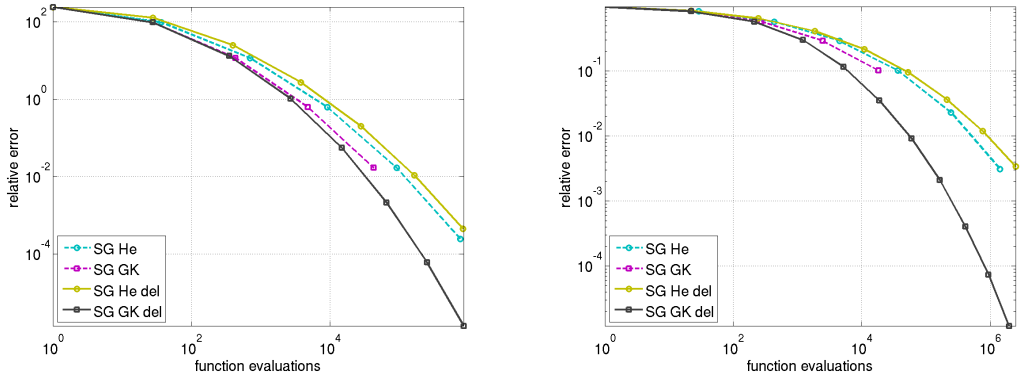
In this way, the degree of exactness $\deg(U_{\tilde{m}_i}) = 2i - 1$ exactly equals the minimum requirement, see Table 4.1(b).

Slowly increasing Gauss-Hermite sequences Next, we improve the classical construction of SG methods for integrals on $\mathbb{R}^d$ with Gaussian weight with respect to its polynomial degree.

SG methods with Gauss-Hermite rules and the sequence $m_i = 2^i - 1$ have been used in [118,123]. The degree of polynomial exactness of the m_i-point Gauss-Hermite rule U_{m_i} is $2m_i - 1$. Hence, its degree increases much faster than $2i - 1$ in the classical construction where $m_i = 2^i - 1$. For the Gauss-Hermite rule, the slowest increasing sequence subject to (4.7) is given by

$$\tilde{m}_i = i \quad \text{for } i = 1, 2, \ldots.$$

It exactly satisfies the minimum requirement $\deg(U_{\tilde{m}_i}) = 2i - 1$, see Table 4.1(c). SG methods with such slowly increasing Gauss-Hermite sequences maintain the condition (4.8) from the classical case but require fewer points on level ℓ.



(a) $Function f_3$ (b) $Function f_4$

Figure 4.3. *Errors and required number of function evaluations of the SG method with Gauss-Hermite (SG He), Genz-Keister (SG GK), slowly increasing Gauss-Hermite (SG He del) and delayed Genz-Keister (SG GK del).*

Delayed Genz-Keister sequences Finally, we consider Genz-Keister rules. They have a lower degree as Gauss-Hermite rules but have the advantage that they are nested. SG methods with these rules have been used in [73] with the sequence $m_i = 1, 3, 9, 19, 35$.

In order to reduce the number of points on level ℓ while keeping at the same time the property (4.8), we again determine the slowest increasing sequence subject to (4.7). One obtains the delayed version

$$\tilde{m}_i = 1, 3, 3, 9, 9, 9, 9, 19, \dots \qquad \text{for } i = 1, 2, \dots. \tag{4.10}$$

The degrees of exactness of U_{m_i} and $U_{\tilde{m}_i}$ are shown in Table 4.1(d).

To provide numerical examples which illustrate the improvement that results from the delay of basis sequences, we consider the quadrature test functions

$$f_3(\mathbf{x}) := \cos\left(\sum_{i=1}^{d} x_i^2\right)\varphi_d(\mathbf{x}) \quad \text{and} \quad f_4(\mathbf{x}) := \exp\left(\sum_{i=1}^{d} x_i\right)\varphi_d(\mathbf{x})$$

with $d = 9$ and $d = 7$, respectively. Here, φ_d denotes the standard Gaussian density (2.13) in d dimensions. The function f_3 is also used in [129, 132] to show that QMC methods can be superior to MC methods also for isotropic problems. The function f_4 corresponds to a simplified version of the problem to price Asian options, see [161].

One can see in Figure 4.3 that the SG method with delayed Genz-Keister sequences is in both examples more efficient than its classical counterpart. While SG methods with non-delayed Genz-Keister rules can only be constructed until level five [45], higher levels and also higher accuracies can be achieved in the delayed

case. In the two tests, the slowly increasing Gauss-Hermite sequences turn out to be slightly less effective than the classical Gauss-Hermite sequences. Nevertheless, we will see in Section 4.1.4 that dimension-adaptive SG methods can benefit from such sequences as then the grid refinement can be performed more fine granular.

4.1.3 Generalised Sparse Grids

The sparse grid construction can be tailored to certain classes of integrands if a priori information about the importance of the dimensions or the importance of the interactions between the dimensions is available. This is achieved by choosing appropriate index sets $\mathcal{I} \subset \mathbb{N}^d$ in the representation (4.3).

To ensures the validity of the telescoping sum expansion, the index set $\mathcal{I}$ has to satisfy the admissibility condition

$$\mathbf{k} \in \mathcal{I} \text{ and } \mathbf{l} \leq \mathbf{k} \quad \Longrightarrow \quad \mathbf{l} \in \mathcal{I}, \tag{4.11}$$

where $\mathbf{l} \leq \mathbf{k}$ is defined by $l_j \leq k_j$ for $j = 1, \ldots, d$.

In this way, the *generalised sparse grid method*

$$\mathrm{SG}_{\mathcal{I}} f := \sum_{\mathbf{k} \in \mathcal{I}} \boldsymbol{\Delta}_{\mathbf{k}} f \tag{4.12}$$

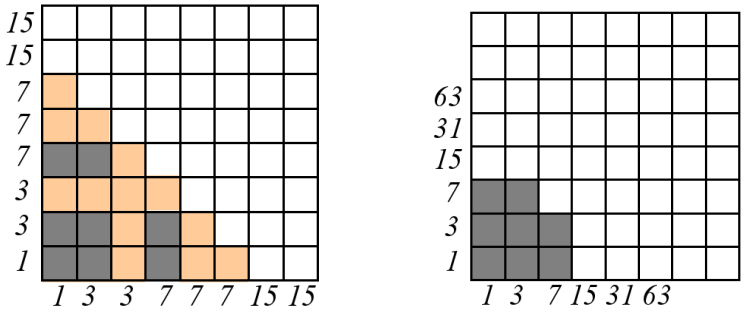
is defined, see, e.g., [48, 71, 137, 165].

Different ways to truncate the sum (4.3) then correspond to different quadrature methods. Examples are the product approach (3.7), the classical SG construction (4.5) and SG methods with delayed basis sequences. Moreover, SG methods based on weighted norms $|\mathbf{k}|_{1,\mathbf{a}} := \sum_{j=1}^{d} a_j k_j$, where $\mathbf{a} \in \mathbb{R}_+^d$ is a weight vector for the different dimensions (see, e.g., [48]), or SG methods with finite-order weights, where the norm $|\mathbf{k}|_0 := \sum_{j=1}^{d} 1_{k_j > 1}$ is used for the selection of indices, are special cases of this approach.

Using the example of the Gauss-Patterson sequence, we illustrate in Figure 4.4 the fact that SG methods with delayed basis sequences are special cases of the generalised SG method. Due to the delay of the basis sequence it holds the equality $U_{m_{k-1}} = U_{m_k}$ and thus $\Delta_k = U_{m_k} - U_{m_{k-1}} = 0$ for several $k \in \mathbb{N}$. Therefore, also several product difference formulas $\Delta_{\mathbf{k}} = 0$ and do not contribute to the integral value. In Figure 4.4(a) these indices $\mathbf{k}$ are colored orange. If these zero quadrature rules are omitted, one obtains a generalised index set as indicated in Figure 4.4(b).

4.1.4 Dimension-adaptive Sparse Grids

In practise, usually no a priori information about the dimension structure of the integrand is available. In these cases algorithms are required which can construct appropriate index sets $\mathcal{I}$ automatically during the actual computation. Such algorithms



(a) Sample index set of the SG method with the delayed Gauss-Patterson sequence

(b) Index set of the corresponding generalised SG method

Figure 4.4. *Illustration of the relation between SG methods based on delayed sequences and generalized SG methods.*

were presented in [49, 71] where the index sets are found in a dimension-adaptive way by the use of suitable error indicators.

The adaptive methods start with the smallest index set $\mathcal{I} = \{(1, \ldots, 1)\}$. Then, step-by-step from the set of all admissible indices, the index $\mathbf{k}$ is added to $\mathcal{I}$ which has the largest $|\boldsymbol{\Delta}_{\mathbf{k}} f|$ value and is therefore expected to provide an as large as possible error reduction, see [47, 49, 118] for details.

The resulting dimension-adaptive construction of the index set[3] is shown in Algorithm 4.1. There, $\mathcal{A}$ denotes the subset of all indices $\mathbf{k} \in \mathbb{N}^d \setminus \mathcal{I}$ which satisfy the admissibility condition (4.11) with respect to the set $\mathcal{I}$.

Algorithm 4.1: Dimension-adaptive sparse grid method.

Initialise: Let $\mathcal{I} = \{(1, \ldots, 1)\}$ and $s = \boldsymbol{\Delta}_{(1,\ldots,1)} f$.

repeat

1) Compute the values $\boldsymbol{\Delta}_{\mathbf{k}} f$ from (4.4) for all $\mathbf{k} \in \mathcal{A}$ for which $\boldsymbol{\Delta}_{\mathbf{k}} f$ has not yet been computed and set $s = s + \boldsymbol{\Delta}_{\mathbf{k}} f$.

2) Move the index $\mathbf{k} \in \mathcal{A}$ with the largest weight $|\boldsymbol{\Delta}_{\mathbf{k}} f|$ from $\mathcal{A}$ to $\mathcal{I}$.

until $|\boldsymbol{\Delta}_{\mathbf{k}} f| \leq \varepsilon$;

Set $SG_{\mathcal{I}} f = s$.

Altogether, the algorithm allows for an adaptive detection of the important dimensions and heuristically constructs optimal index sets $\mathcal{I}$ in the sense of [17, 63]. Note that this is closely related to best N-term approximation [30].

[3] Note that in [49] more sophisticated stop criterions are used than $|\boldsymbol{\Delta}_{\mathbf{k}} f| \leq \varepsilon$.

Remember that we already presented a similar algorithm, Algorithm 3.1, in Section 3.2.4. We next compare these two algorithms. Recall that Algorithm 3.1 can be based on two *separate* types of adaptivity. The important anchored-ANOVA terms $If_{\mathbf{u}}$ are found in a dimension-adaptive fashion with help of the weights $\gamma_{\mathbf{u}}$ and are approximated by $q_{\mathbf{u}}$ using possibly locally adaptive methods. In Algorithm 4.1 the calculation of the contributions $\Delta_{\mathbf{k}}$ is more restrictive since the telescoping sum expansion has to hold. The algorithm is already completely determined by the choice of the univariate quadrature rule U_{m_k}. While Algorithm 3.1 has the advantage that low regularity of low order anchored-ANOVA terms can be resolved by local adaptivity, Algorithm 4.1 has the advantage that modeling and discretization errors are *simultaneously* taken into account and can thus be balanced in an optimal way.

Remark 4.3. Note finally that dimension-adaptive SG methods do not profit from the *delay of basis sequences* since dimension-adaptive SG methods operate on the class of generalised SG methods. This class includes all SG methods based on delayed sequences as special cases as we explained in the previous section. Therefore, dimension-adaptive SG methods based, e.g., on the Gauss-Patterson sequence and dimension-adaptive SG methods based on the delayed Gauss-Patterson sequence provide the same results. Their grids equal to the grid of the classical SG method based on the delayed Gauss-Patterson sequence if this grid is optimal with respect to the error indicator used for the dimension-adaptive grid refinement.

Dimension-adaptive SG methods can however take advantage of *slowly increasing sequences* like Gauss-Hermite rules (see Section 4.1.2) as here the grid refinement can be performed more fine granular. In this way, the constant of the approximation can be reduced in particular for anisotropic integrands in high dimensions. We will present various numerical results in Chapter 6 which illustrate the efficiency of this particular SG method.

4.2 Optimal Sparse Grids in Weighted Spaces

This section we consider the generalised SG method (4.12) in more detail. We here determine the index set $\mathcal{I}$ of this method, which balances error and cost in an optimal way for integrands from weighted tensor product Sobolev spaces [148, 165]. To this end, we partly proceed as in [18, 165].

For reasons of simplicity, we restrict ourselves to the case that the univariate quadrature rules U_{m_k} in (4.1) are given by the trapezoidal rule with $m_1 = 1$, $U_1 f = f(0)$ and $m_i = 1 + 2^{i-2}$ points for $i \geq 2$. Our analysis is based on the univariate function space

$$H^1_\gamma([0,1]) := \{f : [0,1] \to \mathbb{R} : \|f\|_{1,\gamma} < \infty\}$$

with the norm

$$\|f\|^2_{1,\gamma} := f(0)^2 + \gamma^{-1}\|f'\|^2_{L_2}, \tag{4.13}$$

where $\gamma \in (0,1]$ denotes a weight. In the multivariate case we consider a given sequence of weights

$$1 = \gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_d \geq 0$$

and assign to each set $\mathbf{u} \subseteq \mathcal{D}$ the product weight $\gamma_{\mathbf{u}}$ from (3.19). We then define the tensor product space[4]

$$H^{1,\mathrm{mix}}_{\boldsymbol{\gamma}}([0,1]^d) := \bigotimes_{j=1}^{d} H^1_{\gamma_j}([0,1])$$

with the norm

$$\|f\|^2_{1,\boldsymbol{\gamma}} := \sum_{\mathbf{u}\subseteq\mathcal{D}} \gamma_{\mathbf{u}}^{-1} \|f_{\mathbf{u}}\|^2_{1,\mathrm{mix}}$$

with

$$\|f_{\mathbf{u}}\|^2_{1,\mathrm{mix}} := \int_{[0,1]^{|\mathbf{u}|}} \left| \frac{\partial^{|\mathbf{u}|}}{\partial \mathbf{x}_{\mathbf{u}}} f(\mathbf{x}_{\mathbf{u}}, \mathbf{0}) \right|^2 d\mathbf{x}_{\mathbf{u}},$$

where $f_{\mathbf{u}}$ denote the sub-terms of the anchored-ANOVA decomposition anchored at the origin.[5]

4.2.1 Cost-Benefit Ratio

For the space $H^{\mathrm{mix}}_{\boldsymbol{\gamma}}([0,1]^d)$ we next determine the index set $\mathcal{I}$ of the generalised sparse grid method $\mathrm{SG}_{\mathcal{I}}$, which has the best possible cost-benefit ratio. To this end, we first associate each index $\mathbf{k} \in \mathbb{N}^d$ with a local cost value, namely the number of function evaluations $n_{\mathbf{k}}$ required by $\boldsymbol{\Delta}_{\mathbf{k}} f$. Since the methods U_{m_i} are nested and since $m_i \leq 2^{i-1}$, we have

$$n_{\mathbf{k}} = \prod_{j=1}^{d} m_{k_j} \leq 2^{|\mathbf{k}-\mathbf{1}|_1} =: c_{\mathbf{k}}.$$

For the global costs of (4.12) we thus have the bound

$$\sum_{\mathbf{k}\in\mathcal{I}} n_{\mathbf{k}} \leq \sum_{\mathbf{k}\in\mathcal{I}} c_{\mathbf{k}} =: n_{\mathcal{I}}. \tag{4.14}$$

We now consider the error of the method $\mathrm{SG}_{\mathcal{I}}$. To this end, note that

$$|If - \mathrm{SG}_{\mathcal{I}} f| = |\sum_{\mathbf{k}\in\mathbb{N}^d} \boldsymbol{\Delta}_{\mathbf{k}} f - \sum_{\mathbf{k}\in\mathcal{I}} \boldsymbol{\Delta}_{\mathbf{k}} f| \leq \sum_{\mathbf{k}\in\mathbb{N}^d\setminus\mathcal{I}} |\boldsymbol{\Delta}_{\mathbf{k}} f|. \tag{4.15}$$

[4]Note that $H^{1,\mathrm{mix}}_{\boldsymbol{\gamma}}([0,1]^d)$ is the reproducing kernel Hilbert space to the product kernel $K(\mathbf{x},\mathbf{y}) = \prod_{j=1}^{d} k(x_j, y_j)$, where $k(x,y) := 1 + \gamma \min\{x,y\}$ is the reproducing kernel of the space $H^1_\gamma([0,1])$. We provide more detail on reproducing kernel Hilbert spaces in Appendix A.1.

[5]It is also possible to anchor the space $H^1_\gamma([0,1])$ at the point $1/2$.

To derive bounds for $\mathbf{\Delta}_{\mathbf{k}} f$, we associate to each index $\mathbf{k} \in \mathbb{N}^d$ the product weight

$$\gamma_{\mathbf{k}} := \prod_{\substack{j=1,\ldots,d \\ k_j>1}} \gamma_j,$$

where the product is taken over all j for which $k_j > 1$ holds.

Lemma 4.4. *It holds*

$$|\mathbf{\Delta}_{\mathbf{k}} f| \leq b_{\mathbf{k}} \|f\|_{1,\boldsymbol{\gamma}} \tag{4.16}$$

where

$$b_{\mathbf{k}} := 2^{-|\mathbf{k}-\mathbf{1}|_1} \gamma_{\mathbf{k}}^{1/2}. \tag{4.17}$$

Proof. We first consider the univariate case and show that

$$|\Delta_i f| \leq \gamma^{1/2} 2^{-i+1} \|f\|_{1,\gamma} \tag{4.18}$$

for $i \geq 2$. In fact, by (4.13) we have

$$\|f'\|_{L_2}^2 = \sqrt{\gamma(\|f\|_{1,\gamma}^2 - f(0)^2)} \leq \gamma^{1/2} \|f\|_{1,\gamma}.$$

Therefore,

$$|\Delta_i f| = |U_{m_i} f - U_{m_i-1} f| \leq 2^{-i+1} \|f'\|_{L_2}^2 \leq \gamma^{1/2} 2^{-i+1} \|f\|_{1,\gamma}$$

for $i \geq 2$, where a proof of the first inequality can be found in [165]. For $i = 1$, we have $|\Delta_i f| = |U_{m_1} f| = |f(0)| \leq \|f\|_{1,\gamma}$. Using the tensor product structure $\mathbf{\Delta}_{\mathbf{k}} = \bigotimes_{i=1}^d \Delta_{k_i}$ we obtain the assertion. ∎

Motivated by (4.16), we refer to $b_{\mathbf{k}}$ in the following as the local benefit associated with the index $\mathbf{k} \in \mathbb{N}^d$. The global benefit of the method (4.12) is then given by

$$B_{\mathcal{I}} := \sum_{\mathbf{k} \in \mathcal{I}} b_{\mathbf{k}}. \tag{4.19}$$

This leads to the restricted optimization problem

$$\max_{n_{\mathcal{I}}=w} B_{\mathcal{I}}, \quad w \in \mathbb{N}$$

to maximize the global benefit $B_{\mathcal{I}}$ for fixed global costs $n_{\mathcal{I}}$. Using the argumentation from [18], this *global* optimization problem can be reduced to the problem to order the *local* cost-benefit ratios

$$cbr_{\mathbf{k}} := b_{\mathbf{k}} / c_{\mathbf{k}} = 2^{-2|\mathbf{k}-\mathbf{1}|_1} \gamma_{\mathbf{k}}^{1/2} \tag{4.20}$$

associated with the index $\mathbf{k}$ according to their size. The optimal index set $\mathcal{I}$ then contains all indices whose local cost-benefit ratios are larger than or equal to some constant value. Here, we use the value

$$cbr_{\bar{\mathbf{k}}} := 2^{-2(\ell-1)} \tag{4.21}$$

as threshold, which is associated with the index $\bar{\mathbf{k}} = (\ell, 1, \ldots, 1)$.

Theorem 4.5 (Optimal index sets in the weighted case). *The optimal index set in the weighted case is given by*

$$\mathcal{I}_{\ell,\boldsymbol{\gamma}} := \{\mathbf{k} \in \mathbb{N}^d : \ |\mathbf{k}|_1 + \sigma_{\mathbf{k}} \leq \ell + d - 1\} \tag{4.22}$$

where

$$\sigma_{\mathbf{k}} := \sum_{\substack{j=1,\ldots,d \\ k_j>1}} \sigma_j \ \textit{with}\ \sigma_j := -\log(\gamma_j)/4.$$

Proof. Using

$$\gamma_{\mathbf{k}}^{1/2} = 2^{\sum_{j\in\mathcal{D}_{\mathbf{k}}} \log(\gamma_j)/2} = 2^{-2\sigma_{\mathbf{k}}} \ \text{ with } \mathcal{D}_{\mathbf{k}} := \{j \in \mathcal{D} : \ k_j > 1\}$$

we obtain from (4.20) that

$$cbr_{\mathbf{k}} = 2^{-2(|\mathbf{k}-\mathbf{1}|_1+\sigma_{\mathbf{k}})}.$$

The comparison with (4.21) shows that $cbr_{\mathbf{k}} \geq cbr_{\bar{\mathbf{k}}}$ if and only if $-2(|\mathbf{k}-\mathbf{1}|_1+\sigma_{\mathbf{k}}) \geq -2(\ell-1)$, i.e., if $|\mathbf{k}| - d + \sigma_{\mathbf{k}} \leq \ell - 1$, which proves the assertion. ∎

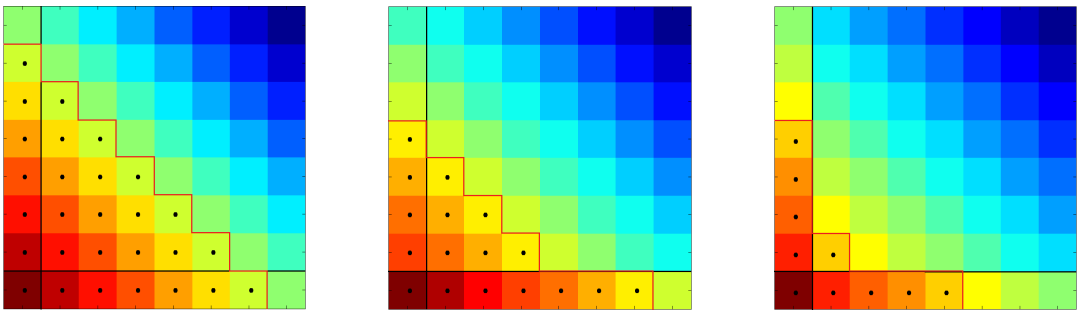
The resulting sparse grid method with the index set $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ is then given by

$$\mathrm{SG}_{\ell,\boldsymbol{\gamma}} f := \sum_{\mathbf{k}\in\mathcal{I}_{\ell,\gamma}} \boldsymbol{\Delta}_{\mathbf{k}} f. \tag{4.23}$$

Note that the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ is the classical sparse grid approach (4.5) in the unweighted case $\gamma_j = 1$, $j = 1, \ldots, d$. For illustration, the resulting optimal index sets $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ on level $\ell = 7$ are shown in Figure 4.5 for $d = 2$ and different choices of the weights $\boldsymbol{\gamma} = (\gamma_1, \gamma_2)$. There, the local cost-benefit ratios of the indices $\mathbf{k} = (k_1, k_2)$, $k_i \in \{1, \ldots, 8\}$, $i = 1, 2$, are color-coded. In addition the indices $\mathbf{k}$ which belong to $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ with $\ell = 7$ are marked with a dot.

One can see that the index sets $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ can be represented by the disjoint union of the four subsets

$$\begin{aligned} \mathcal{I}_{\emptyset} &= \{(1,1)\}, \\ \mathcal{I}_1 &= \{(k_1,1) : \ k_1 > 1 \text{ and } k_1 < b_1\}, \\ \mathcal{I}_2 &= \{(1,k_2) : \ k_2 > 1 \text{ and } k_2 < b_2\}, \\ \mathcal{I}_{12} &= \{(k_1,k_2) : \ k_1, k_2 > 1 \text{ and } |\mathbf{k}|_1 < b_{12}\}, \end{aligned} \tag{4.24}$$



(a) $\boldsymbol{\gamma} = (1, 1)$ (b) $\boldsymbol{\gamma} = (1, 2^{-8})$ (c) $\boldsymbol{\gamma} = (2^{-8}, 2^{-8})$

Figure 4.5. *Optimal index sets $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ on the level $\ell = 7$.*

with integers b_1, b_2 and b_{12} which depend on the weights γ_1 and γ_2, the dimension d and on the level ℓ. Note that all four subsets correspond to index sets of classical sparse grid methods. In general 2^d subsets are required (one for each ANOVA subterm). We will use this decomposition of the index set in the next two sections to derive cost and error bounds for the generalized sparse grid method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ from the known cost and error bounds for the classical method SG_ℓ.

4.2.2 Cost Analysis

In the following, we use $n(d, \ell, \boldsymbol{\gamma})$ to denote the number of function evaluations of the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$. To analyse these costs we first recall the well-known cost bound for classical sparse grids, see, e.g., [18, 163]. In this case, we omit the index $\boldsymbol{\gamma}$ and write $n(d, \ell) := n(d, \ell, \mathbf{1})$.

Lemma 4.6 (Costs of classical sparse grids). *Let $m_i \leq 2^{i-1}$. Then*

$$n(d, \ell) \leq 2^\ell \binom{\ell + d - 2}{d - 1}.$$

Proof. We compute

$$\begin{aligned} n(d, \ell) &\leq \sum_{|\mathbf{k}|_1 \leq \ell+d-1} 2^{|\mathbf{k}-\mathbf{1}|} = 2^{-d} \sum_{j=d}^{\ell+d-1} \binom{j-1}{d-1} 2^j \leq \binom{l+d-2}{d-1} 2^{-d} \sum_{j=d}^{\ell+d-1} 2^j \\ &= \binom{l+d-2}{d-1} 2^{-d} (2^{\ell+d} - 2^d) = (2^\ell - 1) \binom{l+d-2}{d-1}. \end{aligned}$$

□

We now present a generalised cost bound for the weighted case. The cost bound results from the insight that the sparse grid method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ can be represented by the

combination of 2^d many classical sparse grids (one for each anchored-ANOVA term). We already gave an example of this general relation in (4.24) for the case $d = 2$. Here and in the following, we define that $\binom{n}{d} := 0$ for $n < d$ and that $\binom{x}{d} := \binom{\lfloor x \rfloor}{d}$ for $x \in \mathbb{R}$.

Theorem 4.7 (Costs of weighted sparse grids). *Let $m_i \leq 2^{i-1}$. Then*

$$n(d, \ell, \boldsymbol{\gamma}) \leq 2^\ell \sum_{\mathbf{u} \subseteq \mathcal{D}} {\gamma_\mathbf{u}}^{1/4} \binom{\ell + \log(\gamma_\mathbf{u})/4 - 2}{|\mathbf{u}| - 1}.$$

Proof. We start with

$$n(d, \ell, \boldsymbol{\gamma}) \leq \sum_{\mathbf{k} \in \mathcal{I}_{\ell,\boldsymbol{\gamma}}} c_\mathbf{k} \tag{4.25}$$

and remark that the index set $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ from (4.22) can be represented by $\mathcal{I}_{\ell,\boldsymbol{\gamma}} = \bigcup_{\mathbf{u} \subset \mathcal{D}} \mathcal{I}_\mathbf{u}$ as the disjoint union of the sets

$$\mathcal{I}_\mathbf{u} = \{\mathbf{k} \in \mathcal{I}_{\ell,\boldsymbol{\gamma}} : k_j > 1 \text{ if and only if } j \in \mathbf{u}\}.$$

Thus,

$$\sum_{\mathbf{k} \in \mathcal{I}_{\ell,\boldsymbol{\gamma}}} c_\mathbf{k} = \sum_{\mathbf{u} \subseteq \mathcal{D}} \sum_{\mathbf{k} \in \mathcal{I}_\mathbf{u}} 2^{|\mathbf{k}-1|_1} \tag{4.26}$$

for $\mathbf{u} \subseteq \mathcal{D}$. Here, the inner sum bounds the costs of a $|\mathbf{u}|$-dimensional classical sparse grid starting with $m_1 = 2$ points and with level $\ell - \sigma_\mathbf{u} - |\mathbf{u}|$, where $\sigma_\mathbf{u} = \sum_{j \in \mathbf{u}} \sigma_j$. From Lemma 4.6 we see that

$$2^{d+\ell} \binom{\ell + d - 2}{d - 1}$$

bounds the costs of a classical sparse grid algorithm on level ℓ with $m_i = 2^i$. Therefore,

$$\sum_{\mathbf{k} \in \mathcal{I}_\mathbf{u}} 2^{|\mathbf{k}-1|_1} \leq 2^{\ell - \sigma_\mathbf{u}} \binom{\ell - \sigma_\mathbf{u} - 2}{|\mathbf{u}| - 1}. \tag{4.27}$$

Using $\sigma_\mathbf{u} = -\log(\gamma_\mathbf{u})/4$ and $2^{-\sigma_\mathbf{u}} = \gamma_\mathbf{u}^{1/4}$ we obtain the assertion by combining (4.25) - (4.27). ∎

Note that Theorem 4.7 recovers Lemma 4.6 in the unweighted case $\gamma_j = 1$, $j = 1, \ldots, d$. This can be seen as follows: We have

$$n(d, \ell, \mathbf{1}) \leq 2^\ell \sum_{\mathbf{u} \subseteq \mathcal{D}} \binom{\ell - 2}{|\mathbf{u}| - 1} = 2^\ell \sum_{j=1}^{d} \binom{d}{j} \binom{\ell - 2}{j - 1}$$

and use

$$\sum_{j=1}^{d}\binom{d}{j}\binom{\ell-2}{j-1}=\sum_{j=1}^{d}\binom{d}{d-j}\binom{\ell-2}{j-1}=\sum_{j=0}^{d-1}\binom{d}{d-1-j}\binom{\ell-2}{j}=\binom{d+\ell-2}{d-1}$$

where we applied the Vandermonde's identity for the last equality.

4.2.3 Error Analysis

We now consider the error of the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$. To this end, we start with an error bound for classical sparse grids, see, e.g., [18]. In the unweighted case, we again omit the index $\boldsymbol{\gamma}$ and write $\|f\|_1 := \|f\|_{1,\mathbf{1}}$.

Lemma 4.8 (Error of classical sparse grids). *Let SG_ℓ denote the classical sparse grid method (4.5) and let $\ell \geq d-1$. Then*

$$|If - SG_\ell f| \leq 2^{-\ell} A(d,\ell)\|f\|_1$$

where

$$A(d,\ell) := 2d\binom{\ell+d-1}{d-1}.$$

Proof. Using (4.15) and Lemma 4.4 with $\gamma_j = 1$ for all j, we conclude

$$|If - \mathrm{SG}_{\mathcal{I}} f| \leq \sum_{|\mathbf{k}|_1 > \ell+d-1} |\Delta_{\mathbf{k}} f| \leq \sum_{|\mathbf{k}|_1 > \ell+d-1} 2^{-|\mathbf{k}-1|_1}\|f\|_1.$$

A simple computation with combinatorial identities and the geometric series lead to

$$\sum_{|\mathbf{k}|_1 > \ell+d-1} 2^{-|\mathbf{k}-1|_1} = 2^{-\ell+1}\sum_{j=0}^{d-1}\binom{\ell+d-1}{j},$$

see Lemma 3.7 in [18]. Since

$$\binom{\ell+d-1}{j} \leq \binom{\ell+d-1}{d-1}$$

for $j = 0,\ldots,d-1$ if $\ell \geq d-1$ we have proved the assertion. ∎

We now present a corresponding error bound for the weighted case. To derive this bound, we in particular use the fact that the error of the sparse grid method

$\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ can be bounded by the sum of the errors of 2^d many classical sparse grids (one for each anchored-ANOVA term).

Theorem 4.9 (Error of weighted sparse grids). *Let* $\ell \geq d - \log(\gamma_{\{1,\dots,d\}})/4 - 1$. *Then*

$$|If - SG_{\ell,\boldsymbol{\gamma}} f| \leq 2^{-\ell} \sum_{\mathbf{u}\subseteq\mathcal{D}} {\gamma_{\mathbf{u}}}^{1/4}\ 2^{|\mathbf{u}|} \binom{\ell + \log(\gamma_{\mathbf{u}})/4 - 1}{|\mathbf{u}| - 1} \|f\|_{1,\boldsymbol{\gamma}}.$$

Proof. We start with

$$|If - \mathrm{SG}_{\ell,\boldsymbol{\gamma}} f| \leq \sum_{\mathbf{k}\in\mathbb{N}^d\setminus\mathcal{I}_{\ell,\boldsymbol{\gamma}}} b_{\mathbf{k}} \|f\|_{1,\boldsymbol{\gamma}} \tag{4.28}$$

which follows from (4.15) and (4.16). Note that

$$\mathbb{N}^d \setminus \mathcal{I}_{\ell,\boldsymbol{\gamma}} = \bigcup_{\mathbf{u}\subseteq\mathcal{D}} (\mathcal{N}_{\mathbf{u}} \setminus \mathcal{I}_{\mathbf{u}})$$

with $\mathcal{N}_{\mathbf{u}}$ as in (4.37) and $\mathcal{I}_{\mathbf{u}}$ as in the proof of Theorem 4.7. By (4.17), we thus have

$$\sum_{\mathbf{k}\in\mathbb{N}^d\setminus\mathcal{I}_{\ell,\boldsymbol{\gamma}}} b_{\mathbf{k}} \|f\|_{\boldsymbol{\gamma}} = \sum_{\mathbf{u}\subseteq\mathcal{D}} \gamma_{\mathbf{u}}^{1/2} \sum_{\mathbf{k}\in\mathcal{N}_{\mathbf{u}}\setminus\mathcal{I}_{\mathbf{u}}} 2^{-|\mathbf{k}-1|_1} \|f\|_{1,\boldsymbol{\gamma}} \tag{4.29}$$

with $\gamma_{\mathbf{u}} = \prod_{j\in\mathbf{u}} \gamma_j = 2^{-4\sigma_{\mathbf{u}}}$. Here, the inner sum bounds the error of a $|\mathbf{u}|$-dimensional classical sparse grid starting with $m_1 = 2$ points and with level $\ell - \lfloor\sigma_{\mathbf{u}}\rfloor - |\mathbf{u}|$. From Lemma 4.8 we see that

$$2^{-d-\ell} A(d,\ell) \|f\|_1$$

bounds the error of a classical sparse grid algorithm with $m_i = 2^i$ and level ℓ. Therefore,

$$\sum_{\mathbf{k}\in\mathcal{N}_{\mathbf{u}}\setminus\mathcal{I}_{\mathbf{u}}} 2^{-|\mathbf{k}-1|_1} \leq 2^{-\ell+\sigma_{\mathbf{u}}} A(|\mathbf{u}|, \ell - \sigma_{\mathbf{u}} - |\mathbf{u}|). \tag{4.30}$$

If $\ell \geq d - \log(\gamma_{\{1,\dots,d\}})/4 - 1$, it holds $\ell - \sigma_{\mathbf{u}} \geq |\mathbf{u}| - 1$ for all $\mathbf{u} \subseteq \mathcal{D}$ and thus

$$A(|\mathbf{u}|, \ell - \sigma_{\mathbf{u}} - |\mathbf{u}|) = 2^{|\mathbf{u}|} \binom{\ell - \sigma_{\mathbf{u}} - 1}{|\mathbf{u}| - 1} \tag{4.31}$$

by the definition of $A(d,\ell)$, see Lemma 4.8. Using $\sigma_{\mathbf{u}} = -\log(\gamma_{\mathbf{u}})/4$ and $\gamma_{\mathbf{u}}^{1/2} 2^{\sigma_{\mathbf{u}}} = \gamma_{\mathbf{u}}^{1/4}$, we finally obtain the assertion by combining (4.28) - (4.31). ∎

Note that Theorem 4.9 recovers Lemma 4.8 in the unweighted case $\gamma_j = 1$, $j = 1, \dots, d$. We see this from

$$\begin{aligned} |If - \mathrm{SG}_{\ell,\mathbf{1}} f| &\leq 2^{-\ell} \sum_{\mathbf{u} \subseteq \mathcal{D}} 2|\mathbf{u}| \binom{\ell - 1}{|\mathbf{u}| - 1} \|f\|_{1,\mathbf{1}} = 2^{-\ell}\, 2d \sum_{j=1}^{d} \binom{d}{j} \binom{\ell - 1}{j - 1} \|f\|_{1,\mathbf{1}} \\ &= 2^{-\ell}\, 2d \binom{\ell + d - 1}{d - 1} \|f\|_{1,\mathbf{1}} = 2^{-\ell} A(d, \ell) \|f\|_1. \end{aligned}$$

Here, the second equality follows from the Vandermonde's identity.

4.2.4 ε-Cost Analysis

Using the results of Section 4.2.2 and Section 4.2.3, we now represent the error of the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ as a function of its costs $n = n(d, \ell, \boldsymbol{\gamma})$. We again start with the classical case.

Lemma 4.10 (ε-cost of classical sparse grids). *For $f \in H_{\mathbf{1}}^{mix}([0,1]^d)$ and $\ell \geq d - 1$ it holds*

$$|If - SG_\ell f| = O(n^{-1} (\log n)^{2(d-1)})$$

where n denotes the number of points used by the method SG_ℓ.

Proof. Note that $A(d, \ell) = O(\ell^{d-1})$. By Lemma 4.8 we can thus estimate

$$|If - \mathrm{SG}_\ell f| = O\left(2^{-\ell} \ell^{d-1}\right) = O\left(\frac{\ell^{2(d-1)}}{2^\ell \ell^{d-1}}\right) = O\left(\frac{(\log n)^{2(d-1)}}{n}\right).$$

Here, we used $\ell \leq \log(n)$ and $n = O(2^\ell \ell^{d-1})$ where the latter bound can be derived with the help of Lemma 4.6. ∎

We now generalise this result such that also the weighted case is covered.

Theorem 4.11 (ε-cost of weighted sparse grids). *Let $\ell \geq d - \log(\gamma_{\{1,\dots,d\}})/4 - 1$. Then*

$$|If - SG_{\ell,\boldsymbol{\gamma}} f| \leq n^{-1}\, 2d\, B(d, \ell, \boldsymbol{\gamma})^2\, \|f\|_{1,\boldsymbol{\gamma}} \tag{4.32}$$

where

$$B(d, \ell, \boldsymbol{\gamma}) := \sum_{j=1}^{d} \gamma_{\{1,\dots,j\}}^{1/4} \binom{d}{j} \binom{\ell + \log(\gamma_{\{1,\dots,j\}})/4 - 1}{j - 1}.$$

Proof. We first show that

$$n(d, \ell, \boldsymbol{\gamma}) \leq 2^\ell B(d, \ell, \boldsymbol{\gamma}). \tag{4.33}$$

To this end, note that $\gamma_{\{1,\dots,j\}} = \prod_{i=1}^{j} \gamma_i \geq \gamma_{\mathbf{u}}$ for all $\mathbf{u}$ with $|\mathbf{u}| = j$ since the weights are ordered according to their size. Thus, by Lemma 4.7,

$$\begin{aligned} n(d,\ell,\boldsymbol{\gamma}) &\leq 2^{\ell} \sum_{j=1}^{d} \sum_{|\mathbf{u}|=j} {\gamma_{\mathbf{u}}}^{1/4} \binom{\ell + \log(\gamma_{\mathbf{u}})/4 - 2}{|\mathbf{u}| - 1} \\ &\leq 2^{\ell} \sum_{j=1}^{d} \binom{d}{j} \gamma_{\{1,\dots,j\}}^{1/4} \binom{\ell + \log(\gamma_{\{1,\dots,j\}})/4 - 1}{j-1} \end{aligned}$$

where we use the fact that $\binom{n}{d}$ is monotone increasing in n. Similarly, we derive

$$|If - \mathrm{SG}_{\ell,\boldsymbol{\gamma}} f| \leq 2^{-\ell}\, 2d\, B(d,\ell,\boldsymbol{\gamma})\, \|f\|_{1,\boldsymbol{\gamma}} \tag{4.34}$$

from Lemma 4.9. From (4.34) and (4.33) with $n = n(d,\ell,\boldsymbol{\gamma})$ we conclude that

$$|If - \mathrm{SG}_{\ell,\boldsymbol{\gamma}} f| \leq \frac{2d\, B(d,\ell,\boldsymbol{\gamma})^2}{2^{\ell}\, B(d,\ell,\boldsymbol{\gamma})} \|f\|_{1,\boldsymbol{\gamma}} \leq 2d\, B(d,\ell,\boldsymbol{\gamma})^2\, n^{-1}\, \|f\|_{1,\boldsymbol{\gamma}}$$

which proves the theorem. □

We now comment on Theorem 4.11:

- In the unweighted case $\gamma_j = 1$, $j = 1,\dots,d$, we obtain $B(d,\ell,\boldsymbol{\gamma}) = A(d,\ell) = O((\log n)^{d-1})$. Theorem 4.11 is thus a generalisation of the classical case in Lemma 4.10.

- Theorem 4.11 shows that the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ converges with rate n^{-1} which is independent of the dimension. The error bound still depends on the value $B(d,\ell,\boldsymbol{\gamma})$, however. In general, we see that $B(d,\ell,\boldsymbol{\gamma}) = O(\ell^{d-1})$. It thus introduces a logarithmic dependence on the dimension.

- The value $B(d,\ell,\boldsymbol{\gamma})$ is decreasing with the size of the weights γ_j, $j = 1,\dots,d$, though. Moreover, the level

  $$\ell^* := d - \log(\gamma_{\{1,\dots,j\}})/4 \tag{4.35}$$

  grows with decreasing weights, see also Example 4.12. It describes the level where the asymptotic regime in the error bound of Theorem 4.11 starts and thus gives the point where the logarithmic factor ℓ^{d-1} appears in the complexity.

- If the weights decay sufficiently fast such that

  $$\sup_{d} \sum_{j=1}^{d} \gamma_j^{1/2} < \infty \tag{4.36}$$

  then the general results of [165] indicate that $B(d,\ell,\boldsymbol{\gamma})$ and hence also the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ depends only polynomially on the dimension.

- Note that the error bound (4.32) also depends on the norm $\|f\|_{1,\boldsymbol{\gamma}}$ of the integrand. This norm may grow exponentially fast for increasing d which can cause problems in higher dimensions. Note that this effect is not included in the notion of tractability in [148,165] since only functions with norm $\|f\|_{1,\boldsymbol{\gamma}} \leq 1$ are addressed there.

- Note finally that the error estimator used in the dimension-adaptive sparse grid method is based on the values $\boldsymbol{\Delta}_{\mathbf{k}} f$ on which also the above analysis is based. We can thus expect that these methods correctly identify the optimal index sets $\mathcal{I}_{\ell,\boldsymbol{\gamma}}$ provided no early determination problems occur. In this case, the results of this section can also be used to show that dimension-adaptive sparse grid methods can avoid the curse of dimension in weighted function spaces whose weights decay sufficiently fast.

Example 4.12. As in [165], we consider the family of weights

$$\gamma_j = j^{-\alpha}, \quad \text{with } \alpha \geq 0.$$

This example with $\alpha = 2$ is motivated by the fact that in many application problems from finance the high nominal dimension arises from the discretization of an underlying continuous time process. The corresponding integrals can thus be written as an approximation to some infinite-dimensional integrals with respect to the Wiener measure.[6] In these cases, the integrands are contained in some weighted function spaces whose weights are related to the eigenvalues of the covariance operator of the Wiener measure. These eigenvalues, sorted by their magnitude, are decaying proportionally to j^{-2}, where j is the number of the eigenvalue.

For $\gamma_j = j^{-\alpha}$, we obtain $\gamma_{\{1,\dots,j\}} = (j!)^{-\alpha}$ and $\ell^* = d + \alpha/4 \log(d!)$. We thus can compute the level ℓ^* in (4.35) for different exponents α and different dimensions d. The results are shown in Table 4.2. For instance, let $d = 360$ and $\alpha = 2$. Then, one can see that the asymptotic log-factor ℓ^{d-1} in the error bound from Theorem 4.11 does not appear as long as $\ell < \ell^* = 1633$. In the unweighted case, the level $\ell^* = 361$ is significantly smaller.

Table 4.2. *The values $\ell^* := d + \alpha/4 \log(d!)$ for $\alpha \in \{0, 1, 2, 3\}$ and different nominal dimensions d.*

$\alpha \setminus d$	1	3	5	10	50	100	360	1024
0	1	4	6	11	51	101	361	1025
1	1	4	8	16	105	232	997	3217
2	1	5	9	22	158	363	1633	5409
3	1	6	11	27	212	495	2268	7601

[6] See, e.g., [105] and the references listed there.

If $\alpha > 2$ holds, then the condition (4.36) is satisfied and we can use the general results of [165] to see that the ε-cost of the method $\mathrm{SG}_{\ell,\boldsymbol{\gamma}}$ is independent of the dimension. In this case the number of function evaluations $n(\varepsilon)$ to obtain an accuracy of ε can be bounded by

$$n(\varepsilon) \leq c\, \varepsilon^{\max\{1, \frac{2}{\alpha-1}\}}$$

for integrands from the unit ball $\|f\|_{\boldsymbol{\gamma}} \leq 1$, where the constant c is independent of d and ε. It is known, see [138], that the ε-exponent in this bound can not be improved using generalised sparse grid methods. It is optimal for $\alpha \geq 3$ but far from optimal for $\alpha \approx 1$.

4.3 Relation to Dimension-wise Quadrature

There is a close relation of the sparse grid approach and the anchored-ANOVA decomposition. The sparse grid approach (4.12) can indeed be interpreted as a refinement of each anchored-ANOVA term by first expanding it into an infinite basis and then truncating this expansion appropriately.[7] It can thus be regarded as special case of the dimension-wise quadrature method (3.15) where the set $\mathcal{S}$ and the rules $Q_{\mathbf{u}}$ are chosen in a systematic way to exploit smoothness of the integrand. We now show this in more detail. To this end, we always use the anchor $\mathbf{a} = (1/2, \ldots, 1/2)$. We start with the following lemma.

Lemma 4.13. *Let $f_{\mathbf{u}}$ and $P_{\mathbf{u}} f$ as in (3.11). Let further*

$$\mathcal{N}_{\mathbf{u}} := \left\{ \mathbf{k} \in \mathbb{N}^d : \, k_j > 1 \text{ if and only if } j \in \mathbf{u} \right\}. \tag{4.37}$$

Then, $\boldsymbol{\Delta}_{\mathbf{k}} f = \boldsymbol{\Delta}_{\mathbf{k}}(P_{\mathbf{u}} f)$ if $\mathbf{k} \in \mathcal{N}_{\mathbf{v}}$ with $\mathbf{v} \subseteq \mathbf{u}$. Moreover, $\boldsymbol{\Delta}_{\mathbf{k}} f = \boldsymbol{\Delta}_{\mathbf{k}} f_{\mathbf{u}}$ if $\mathbf{k} \in \mathcal{N}_{\mathbf{u}}$.

Proof. The proof follows from the fact that the projections $P_{\mathbf{u}}$ fix dimensions of f at the same point as the operator Δ_1. Indeed, if $\mathbf{k} \in \mathcal{N}_{\mathbf{v}}$ and $\mathbf{v} \subseteq \mathbf{u}$ then $k_j = 1$ for all $j \notin \mathbf{u}$ and thus $\boldsymbol{\Delta}_{\mathbf{k}} f = \boldsymbol{\Delta}_{\mathbf{k}}(P_{\mathbf{u}} f)$ since $\boldsymbol{\Delta}_{\mathbf{k}} = \Delta_{k_1} \otimes \ldots \otimes \Delta_{k_d}$ and $\Delta_1 f = P_{\emptyset} f = f(1/2)$ for all univariate functions f. To show the second assertion, let $\mathbf{k} \in \mathcal{N}_{\mathbf{u}}$. Then, we obtain from (2.4) that

$$\boldsymbol{\Delta}_{\mathbf{k}}(P_{\mathbf{u}} f) = \boldsymbol{\Delta}_{\mathbf{k}} f_{\mathbf{u}} + \sum_{\mathbf{v} \subset \mathbf{u}} \boldsymbol{\Delta}_{\mathbf{k}} f_{\mathbf{v}}.$$

Since $f_{\mathbf{v}}(\mathbf{x}_{\mathbf{v}})|_{x_j = 1/2} = 0$ for all $j \in \mathbf{v}$, which is a direct consequence of the orthogonality (2.6), we conclude $\boldsymbol{\Delta}_{\mathbf{k}} f_{\mathbf{v}} = 0$ for all $\mathbf{v} \subset \mathbf{u}$ and $\mathbf{k} \in \mathcal{N}_{\mathbf{u}}$. This proves $\boldsymbol{\Delta}_{\mathbf{k}} f = \boldsymbol{\Delta}_{\mathbf{k}}(P_{\mathbf{u}} f) = \boldsymbol{\Delta}_{\mathbf{k}} f_{\mathbf{u}}$ for all $\mathbf{k} \in \mathcal{N}_{\mathbf{u}}$. ∎

[7] Note the close relation to [103].

By Lemma 4.13 and (4.3) we obtain

$$If = \sum_{\mathbf{u}\subseteq\mathcal{D}} \sum_{\mathbf{k}\in\mathcal{N}_\mathbf{u}} \boldsymbol{\Delta}_\mathbf{k} f$$

since $\mathbb{N}^d$ is the disjoint union of the sets $\mathcal{N}_\mathbf{u}$, $\mathbf{u} \subseteq \mathcal{D}$. By (3.12), we also have $If = \sum_{\mathbf{u}\subseteq\mathcal{D}} If_\mathbf{u}$, which yield a decomposition

$$If_\mathbf{u} = \sum_{\mathbf{k}\in\mathcal{N}_\mathbf{u}} \boldsymbol{\Delta}_\mathbf{k} f$$

of the integrals of the anchored-ANOVA terms into an infinite sum. Next, we truncate this sum. To this end, we select index sets $\mathcal{I}_\mathbf{u} \subset \mathcal{N}_\mathbf{u}$ for all $\mathbf{u} \subseteq \mathcal{D}$, which satisfy the condition (4.11), and use

$$q_\mathbf{u} := \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{u}} \boldsymbol{\Delta}_\mathbf{k} f \tag{4.38}$$

as approximation to $If_\mathbf{u}$. The corresponding method (3.15) with $\mathcal{S} = \mathcal{D}$ can then be represented as

$$A_\mathcal{S} f = \sum_{\mathbf{u}\subseteq\mathcal{D}} q_\mathbf{u} = \sum_{\mathbf{k}\in\mathcal{I}} \boldsymbol{\Delta}_\mathbf{k} f$$

with the index set $\mathcal{I} = \bigcup_{\mathbf{u}\subseteq\mathcal{D}} \mathcal{I}_\mathbf{u}$. We see that in this way both, the modeling and the discretization error is expressed[8] in terms of the values $\boldsymbol{\Delta}_\mathbf{k} f$. We further see that the resulting method $A_\mathcal{S} f$ coincides with the generalised sparse grid approach (4.12). To this end, we define

$$\mathcal{I}_\mathbf{u} := \{\mathbf{k} \in \mathcal{I} : \, k_j > 1 \text{ if and only if } j \in \mathbf{u}\} = \mathcal{I} \cap \mathcal{N}_\mathbf{u}. \tag{4.39}$$

Theorem 4.14. *The dimension-wise quadrature method (3.15) with anchor* $\mathbf{a} = (\frac{1}{2}, \ldots, \frac{1}{2})$*, the index set* $\mathcal{S} = \mathcal{D}$ *and the quadrature methods*

$$Q_\mathbf{u} f := \sum_{\mathbf{v}\subseteq\mathbf{u}} \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{v}} \boldsymbol{\Delta}_\mathbf{k} f \tag{4.40}$$

coincides with the generalised sparse grid method (4.12).

Proof. We have to show that (3.14) holds with $q_\mathbf{u}$ as in (4.38) and $Q_\mathbf{u}$ as in (4.40). In fact,

$$\begin{aligned} Q_\mathbf{u}(P_\mathbf{u} f) - \textstyle\sum_{\mathbf{v}\subset\mathbf{u}} q_\mathbf{v} &= \sum_{\mathbf{v}\subseteq\mathbf{u}} \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{v}} \boldsymbol{\Delta}_\mathbf{k}(P_\mathbf{u} f) - \sum_{\mathbf{v}\subset\mathbf{u}} \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{v}} \boldsymbol{\Delta}_\mathbf{k} f \\ &= \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{u}} \boldsymbol{\Delta}_\mathbf{k}(P_\mathbf{u} f) + \sum_{\mathbf{v}\subset\mathbf{u}} \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{v}} (\boldsymbol{\Delta}_\mathbf{k}(P_\mathbf{u} f) - \boldsymbol{\Delta}_\mathbf{k} f) = \sum_{\mathbf{k}\in\mathcal{I}_\mathbf{u}} \boldsymbol{\Delta}_\mathbf{k} f = q_\mathbf{u} \end{aligned}$$

[8] Modeling errors are here represented by the case $\mathcal{I}_\mathbf{u} = \emptyset$ for any $\mathbf{u} \subseteq \mathcal{D}$.

where we twice used Lemma 4.13. □

Remark 4.15. Similar as in Theorem 4.14, we see that generalised sparse grid methods for integrals on $\mathbb{R}^d$ with Gaussian weight (e.g. sparse grids based on Gauss-Hermite rules) are special cases of the dimension-wise quadrature method for integrals with Gaussian weight, see Remark 3.2. Both methods result from a discretization of the terms of the anchored-ANOVA decomposition with Gaussian weighted using the anchor $\mathbf{a} = (0, \dots, 0)$, see Remark 2.6.

We conclude this chapter with a short summary. In this chapter, we first discussed different sparse grid constructions which resulted from different choices of the underlying index set. Then, we determined optimal index sets for integrands from weighted Sobolev spaces. For the respective sparse grid methods, we derived cost and error bounds, which indicate that the curse of dimension can be avoided in such spaces if their function space weights decay sufficiently fast. Finally we showed that SG methods can be regarded as special cases of dimension-wise quadrature methods. In the next chapter, we discuss several approaches which can be used to enhance the performance of SG methods for special classes of integrals.

Chapter 5
Dimension Reduction and Smoothing

Numerical quadrature methods can profit from smoothness and low effective dimension as we showed in Chapter 3 and Chapter 4. Integrals that arise in applications from finance often have kinks or even jumps, however. Moreover, they often have a high truncation dimension. We here focus on several different approaches which aim to reduce the effective dimension or to smooth the resulting integrands. In this way, the efficiency of the numerical quadrature methods can be improved in many cases as we finally show in Chapter 6 by numerical experiments.

5.1 Orthogonal Transformations

This section is concerned with transformations that can reduce the effective dimension of high-dimensional integrals of the form

$$I_\varphi g = \int_{\mathbb{R}^d} g(\mathbf{z})\varphi_d(\mathbf{z})d\mathbf{z} \tag{5.1}$$

with Gaussian density φ_d in d dimensions. Such integrals usually appear in application problems from finance that are based on Brownian motions. The main idea is that such integrals are invariant with respect to orthonormal transformations such as rotations or reflections. This yields

$$I_\varphi g = \int_{\mathbb{R}^d} g(\mathbf{Qz})\varphi_d(\mathbf{Qz})d\mathbf{z} = \int_{\mathbb{R}^d} g(\mathbf{Qz})\varphi_d(\mathbf{z})d\mathbf{z}$$

for all matrices $\mathbf{Q} \in \mathbb{R}^{d\times d}$ that satisfy

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{I},$$

where $\mathbf{I}$ is the identity. If we want to compute integrals of the form (5.1), we thus have the degree of freedom to choose the matrix $\mathbf{Q}$ from the $d(d-1)/2$-dimensional group of orthonormal matrices. In the following, we aim to determine $\mathbf{Q}$ such that the effective dimension of the integral (5.1) is minimized.

An equivalent situation arises in all application problems which require the computation of an underlying Brownian motion $W(t)$ on a given simulation grid. To this end, mostly the random walk construction is used. However, since the result of

such a path construction is invariant with respect to orthonormal transformations $\mathbf{Q}$ as we show below, we again have the degree of freedom to choose the matrix $\mathbf{Q}$ which leads to the lowest effective dimension and to the best performance of the deterministic integration methods.

In this section, we present four different approaches that construct a Brownian motion until time T at the equidistant points in time $t_k = k\,\Delta t$ with $k = 0, \ldots, d$ and $\Delta t = T/d$. They all require an input vector $\mathbf{y} = (y_1, \ldots, y_d)$ of d independent standard normally distributed random numbers y_k and generate a vector $\mathbf{W} = (W_1, \ldots, W_d)$ whose joint distribution exactly coincides with the joint distribution of the continuous time Brownian motion $W(t)$ at the discrete time points t_k. All constructions can be written in the form

$$\mathbf{W} = \mathbf{A}\mathbf{y} \tag{5.2}$$

and differ only in the choice of the matrix $\mathbf{A}$ which has to satisfy $\mathbf{A}^T\mathbf{A} = \mathbf{C}$ where $\mathbf{C} \in \mathbb{R}^{d\times d}$ with entries $c_{ij} = \mathrm{Cov}(W(t_i), W(t_j)) = \min(t_i, t_j)$ denotes the covariance matrix of the Brownian motion.

To show the relation to orthonormal matrices, note that if $\mathbf{A}^T\mathbf{A} = \mathbf{C}$ then this also holds for $\mathbf{QA}$ if $\mathbf{Q}$ is orthogonal since

$$(\mathbf{QA})^T\mathbf{QA} = \mathbf{A}^T\mathbf{Q}^T\mathbf{QA} = \mathbf{A}^T\mathbf{A} = \mathbf{C}.$$

Hence, $\mathbf{Ay} \in \mathbb{R}^d$ and $\mathbf{AQy} \in \mathbb{R}^d$ have the same distribution. For a given path generating matrix $\mathbf{A}$, we can thus choose an arbitrary orthogonal matrix $\mathbf{Q}$ to improve the properties of the path construction without changing the distribution of $\mathbf{W}$. This is also the starting point of the LT-construction, which we discussed in more detail in Section 5.1.2.

5.1.1 Random Walk, Brownian Bridge, PCA

We next follow [59] and describe the random walk, the Brownian bridge [117] and the principal component [1] construction of the path $\mathbf{W} = (W_1, \ldots, W_d)$ in the univariate as well as in the multivariate case.

Random Walk

We start with the random walk (RW) construction which is most commonly used and is most simple to implement. It proceeds incrementally and computes

$$W_k = W_{k-1} + \sqrt{\Delta t}\, y_k$$

with $W_0 = 0$. This approach corresponds to (5.2) if $\mathbf{A}$ is the Cholesky matrix of $\mathbf{C}$, i.e.,

$$\mathbf{A} = \sqrt{\Delta t} \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & \dots & \dots & 1 \end{pmatrix}.$$

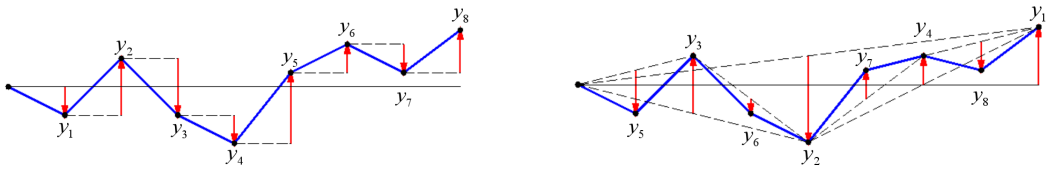
The RW-construction is illustrated in Figure 5.1(a) for a sample path with eight time steps.

Brownian Bridge

The Brownian bridge (BB) construction differs from the RW construction in that rather than constructing the increments sequentially, the path is constructed in a hierarchical way which has the effect that more importance is placed on the earlier variables than on the later ones. It has been proposed in [117] for option pricing problems. If d is a power of two, i.e. $d = 2^m$ with $m \in \mathbb{N}$, then the BB-construction is defined by

$$\begin{aligned}
W_d &= \sqrt{T}\, y_1, \\
W_{d/2} &= (W_0 + W_d)/2 + \sqrt{T/4}\, y_2, \\
W_{d/4} &= (W_0 + W_{d/2})/2 + \sqrt{T/8}\, y_3, \\
W_{3d/4} &= (W_{d/2} + W_d)/2 + \sqrt{T/8}\, y_4, \\
&\vdots \\
W_1 &= (W_0 + W_2)/2 + \sqrt{T/2^{m+1}}\, y_{2^{m-1}+1}, \\
&\vdots \\
W_{d-1} &= (W_{d-2} + W_d)/2 + \sqrt{T/2^{m+1}}\, y_d.
\end{aligned}$$

The construction thus proceeds level-wise where the ℓ-th level corresponds to the first 2^ℓ dimensions. On each level the previous path is refined by intermediate values $W(t_k)$ which are generated conditional on the neighbouring values $W(t_j)$ and $W(t_l)$ which have already been constructed on previous levels. If d is not a power of two, the above procedure is first applied to a subset of the points t_k, whose number is a power of two. The remaining points are then filled in at the end. The BB-construction can also be written in the form (5.2). The explicit formula for the matrix $\mathbf{A}$ can be found in [159]. For illustration, the construction is displayed in Figure 5.1(b) for a sample path with eight time steps.



(a) Random walk (b) Brownian bridge

Figure 5.1. *Construction of the sample path* $W = (W_1, \ldots, W_8)$ *based on the input vector* $\mathbf{y} = (y_1, \ldots, y_8)$ *of standard normally distributed random numbers.*

Principal Component

The principal component (PCA) construction [1] is based on an eigenvalues decomposition $\mathbf{C} = \mathbf{V}^T \mathbf{\Lambda} \mathbf{V}$ of the covariance matrix $\mathbf{C}$. It maximises the concentration of the total variance of the Brownian path in the first few dimensions. Here, the path is obtained via (5.2) with

$$\mathbf{A} = \sqrt{\mathbf{\Lambda}}\mathbf{V} = \begin{pmatrix} \sqrt{\lambda_1} v_{11} & \cdots & \sqrt{\lambda_d} v_{1d} \\ \vdots & & \vdots \\ \sqrt{\lambda_1} v_{d1} & \cdots & \sqrt{\lambda_d} v_{dd} \end{pmatrix}$$

where v_{ij} denotes the j-th coordinate of the i-th eigenvector and λ_i denotes the i-th eigenvalue of the covariance matrix $\mathbf{C}$. The eigenvectors and -values are given by

$$v_{ij} = \frac{2}{\sqrt{2d+1}} \sin\left(\frac{2i-1}{2d+1} j\pi\right)$$

and

$$\lambda_i = \frac{\Delta t}{4} \sin^{-2}\left(\frac{2i-1}{4d+2}\pi\right),$$

see, e.g., [59]. Since $\mathbf{A}$ is now a full matrix, the PCA construction requires $O(d^2)$ operations for the generation of one path instead of $O(d)$ operations which are needed for the random walk or for the Brownian bridge construction. For large d, this often increases the run times of the simulation and limits the practical use of the PCA construction.

The sample paths which result from the RW, BB and PCA-construction after $k = 1, 2, 4$ and 8 intermediate steps are shown in Figure 5.2 for a grid with $d = 8$ time points. After few intermediate steps ($k = 4$) one can see, that BB and PCA have already captured the main movements of the total random path, whereas RW has only focused on the first part of the path. These effects are the more pronounced the larger the dimension d and result since BB and PCA add increasingly fine detail to the path at each step and do not proceed incrementally as RW. The resulting

concentration of variance in the first dimensions is thereby related to the decay of the factors $\sqrt{T/j}$, $j = 1, 4, 8, \ldots$, and $\sqrt{\lambda_i}$, $i = 1, 2, \ldots$ in the BB and PCA construction and often results in integrands which are of low truncation dimension.

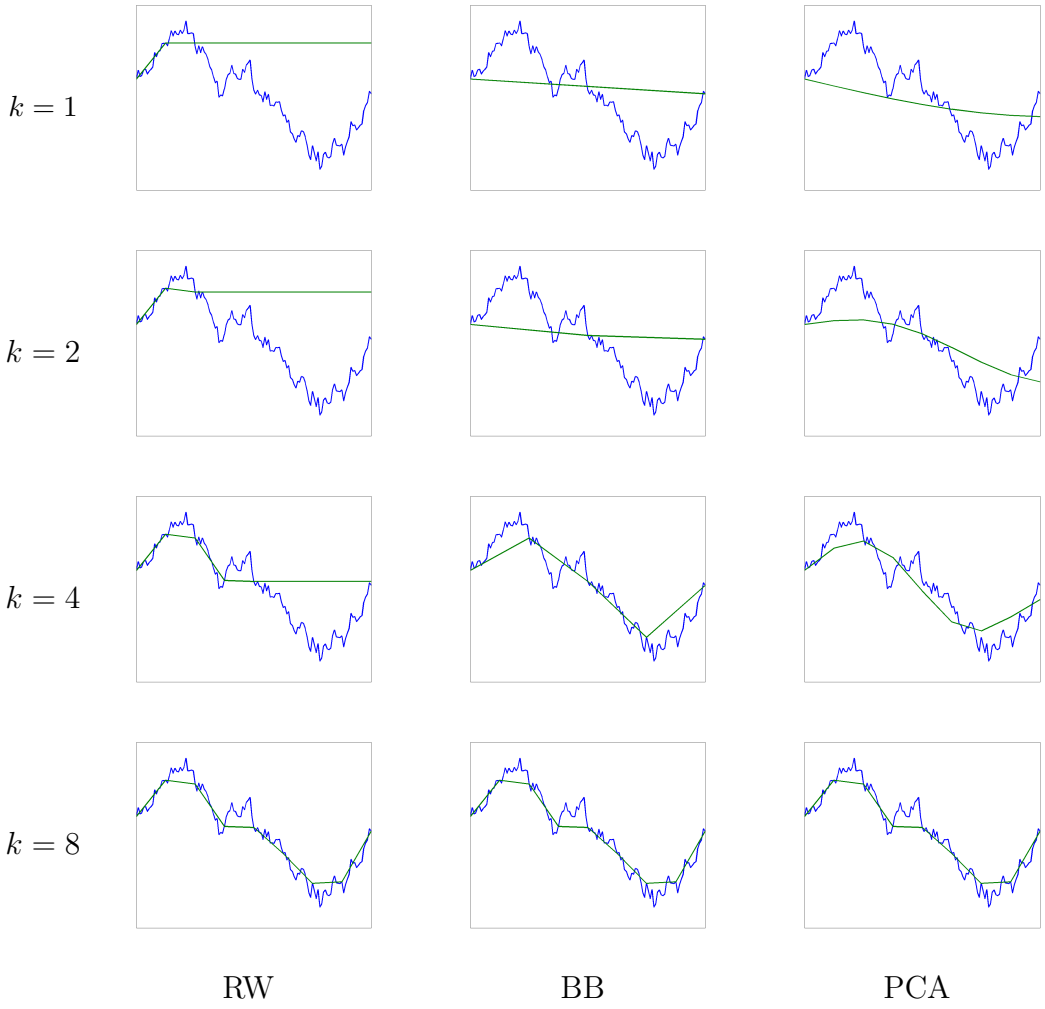


Figure 5.2. *Intermediate results of the different path constructions with* $d = 8$.

Multivariate case

The RW, BB and PCA construction can also be used in the multivariate case when a D-dimensional Brownian motion $\mathbf{W}(t) = (W_1(t), \ldots, W_D(t))$ with correlation matrix $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$ has to be generated on the simulation grid. To this end, we first generate a discrete D-dimensional uncorrelated Brownian motion $\tilde{\mathbf{W}} = (\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_d) \in \mathbb{R}^{d \times D}$ at the time points t_k, $k = 1, \ldots, d$, by applying either RW, BB or PCA separately to each component of the Brownian motion. Then, we respect

the correlation in a second step by computing

$$\mathbf{W}_k = \mathbf{\Lambda}\tilde{\mathbf{W}}_k$$

for $k = 1, \ldots, d$ using a matrix $\mathbf{\Lambda} \in \mathbb{R}^{D\times D}$ which satisfies $\mathbf{\Lambda}^T\mathbf{\Lambda} = \mathbf{\Sigma}$. Here, either the Cholesky matrix or an eigenvalue decomposition can be used which require $O(D^2)$ and $O(D^3)$ operations, respectively. The optimal construction with the highest possible concentration of variance in the first variables in the multivariate case can be achieved in $O(D^3 + d^3)$ operations by an eigenvalue decomposition of the covariance matrix of $(W_1(t_1), \ldots, W_D(t_1), W_1(t_2), \ldots, W_D(t_2), \ldots, W_1(t_d), \ldots, W_D(t_d))$, see [59] for details. We refer to this approach as eigenvalue decomposition (EVD).

5.1.2 Linear transformations

In this section, we discuss the *LT-construction*. It is further approach to generate the path of an underlying Brownian motion, that was first presented by Imai and Tan in [81, 82]. The construction aims to identify the optimal linear transformation $\mathbf{Q}$ in (5.2) by taking into account not only the underlying Brownian (i.e. the Gaussian weight) but also the linear part of the integrand g. This is an important improvement of other path constructions such as BB and PCA which concentrate the variance of the underlying Brownian motion in the first few dimensions but do not take into account the particular structure of the integrand. A priori it is thus not clear that BB and PCA lead to smaller effective dimensions as e.g. RW. In fact, digital options are an example where BB performs worse than RW, see [130].

A Special Class of Functions

To illustrate the main idea of the LT-construction, we start to discuss dimension reduction for a special class of functions $g : \mathbb{R}^d \to \mathbb{R}$ that can be represented as concatenation $g = f \circ g_L$ of an univariate function $f : \mathbb{R} \to \mathbb{R}$ and an affine function $g_L : \mathbb{R}^d \to \mathbb{R}$. Such functions are known under the names *ridge functions* in the context of computerized tomography and neural networks, *plane waves* in the study of hyperbolic partial differential equations and *projection pursuit* in statistic and approximation theory, see [136] for details. This class of functions is also interesting since some finance problems can be represented in this form as we show below.

For this function class an optimal transformation $\mathbf{Q}$ can be obtained such that $g(\mathbf{Qz})$ is an only one-dimensional function. Interestingly, the optimal matrix can be determined without that information on the special form of the functions f and g_L are required. Only the gradient of g has to be computed at a fixed anchor point

$\mathbf{a} \in \mathbb{R}^d$ as stated in the following theorem.

Theorem 5.1. *For* $\mathbf{w} = (w_1, \ldots, w_d) \in \mathbb{R}^d$ *and* $w_0 \in \mathbb{R}$ *let*

$$g(\mathbf{z}) := f\left(w_0 + \sum_{i=1}^{d} w_i z_i\right) \tag{5.3}$$

with $f : \mathbb{R} \to \mathbb{R}$ *and* $f'(\mathbf{w}^T\mathbf{a}) \neq 0$. *Then, there exist an orthonormal matrix* $\mathbf{Q} \in \mathbb{R}^{d\times d}$ *such that*

$$g(\mathbf{Qz}) = f\left(w_0 \pm \|\mathbf{w}\| z_1\right)$$

where the sign corresponds to the sign of $f'(\mathbf{w}^T\mathbf{a})$. *The matrix* $\mathbf{Q} = (\mathbf{q_1}, \ldots, \mathbf{q}_d)$ *is thereby determined as follows: Its first column vector is defined by*

$$\mathbf{q_1} := \nabla g(\mathbf{a}) / \|\nabla g(\mathbf{a})\|$$

and the remaining column vectors are arbitrary as long as the orthogonality condition

$$(\mathbf{q_j}, \mathbf{q}_k) = 0 \quad \text{for } k = 1, \ldots, j-1$$

holds, where $(\cdot,\cdot)$ *denotes the euclidean inner product in* $\mathbb{R}^d$.

Proof. We compute

$$\begin{aligned} g(\mathbf{Qz}) &= f\left(w_0 + \sum_{i=1}^{d} w_i (\mathbf{Qz})_i\right) \\ &= f\left(w_0 + \sum_{i=1}^{d} w_i \sum_{j=1}^{d} q_{ij} z_j\right) \\ &= f\left(w_0 + \sum_{j=1}^{d} z_j \sum_{i=1}^{d} w_i q_{ij}\right) \\ &= f\left(w_0 + \sum_{j=1}^{d} \alpha_j z_j\right) \quad \text{with } \alpha_j := \mathbf{w}^T\mathbf{q_j}. \end{aligned}$$

Since

$$\frac{\partial g}{\partial z_j}(\mathbf{a}) = w_j\, f'\left(\mathbf{w}^T\mathbf{a}\right)$$

we have $\nabla g(\mathbf{a}) = f'\left(\mathbf{w}^T\mathbf{a}\right)\, \mathbf{w}$. If we choose $\mathbf{q_1} := \nabla g(\mathbf{a}) / \|\nabla g(\mathbf{a})\|$ we see that

$$\alpha_1 = \mathbf{w}^T\mathbf{q_1} = \frac{\mathbf{w}^T \nabla g(\mathbf{a})}{\|\nabla g(\mathbf{a})\|} = \frac{f'\left(\mathbf{w}^T\mathbf{a}\right)\, \mathbf{w}^T\mathbf{w}}{|f'\left(\mathbf{w}^T\mathbf{a}\right)|\|\mathbf{w}\|} = \pm\|\mathbf{w}\|.$$

Since $(\mathbf{q_1}, \mathbf{q_j}) = 0$ for $j = 2, \ldots, d$, it further holds that

$$\alpha_j = \mathbf{w}^T \mathbf{q_j} = \frac{\nabla g(\mathbf{a})^T \mathbf{q_j}}{f'(\mathbf{w}^T \mathbf{a})} = \frac{\|\nabla g(\mathbf{a})\|}{f'(\mathbf{w}^T \mathbf{a})} (\mathbf{q_1}, \mathbf{q_j}) = 0 \quad \text{for } j = 2, \ldots, d.$$

This proves the assertion. □

In this way, an initially d-dimensional integrand g can be reduced to an integrand of only one dimension without changing the value of the integral (5.1). This result is based on the simple fact that the sum of normally distributed random variables is again normally distributed. Nevertheless, it shows that for some non-linear functions very effective dimension reductions can be obtained by orthonormal transformations that only depend on the linear part, i.e. on the gradient of the function at a certain anchor point. We will see in the next section that these optimal transformations are correctly identified by the LT-construction.

Important examples which are included in the function class (5.3) are functions of the form

$$g(\mathbf{z}) = \max\{0, e^{g_L(\mathbf{z})} - K\} \quad \text{with } g_L(\mathbf{z}) := w_0 + \sum_{j=1}^{d} w_j z_j \tag{5.4}$$

where w_j, $j = 0, \ldots, d$, and K are constants. This special class of functions is also considered in [161] to explain why high-dimensional finance problems are often of low effective dimension. Asian options and European basket options with geometric averages are examples which can be represented in the form (5.4) and are hence reduced to one-dimensional problems by the LT-construction.

The general LT-construction

Theorem 5.1 only specifies the first column of the matrix $\mathbf{Q}$ since this is already sufficient to reduce functions g of the form (5.3) to only one dimension. For general integrands g such a large dimension reduction can usually not be expected. The 'optimal' transformation $\mathbf{Q}$ then also depends on the choice of the remaining columns $\mathbf{q_j}$ for $j = 2, \ldots, d$. In special cases the contribution of the j-th dimension to the total variance of the integrand is known or can be approximated analytically. Then, the matrix $\mathbf{Q}$ can be constructed inductively, column by column, such that the explained variance due to the first dimensions is sequentially maximized with the side condition such that all columns are orthonormal. Similarly, it is possible in special cases to find the matrix $\mathbf{Q}$ which minimizes the truncation dimension. These approaches, which are denoted by LT-I and LT-III in [81] can be used to price Asian basket options with arithmetic averages as it is shown in this reference.

In general the contribution of the j-th dimension to the total variance of the integrand is unknown, though. It can also not be computed as this would be more

expensive than to compute the integral. To circumvent this problem, we here proceed as in the *general LT-construction* that only takes into account the contribution of the linear part of the function g. The construction proceed inductively and determines the matrix $\mathbf{Q}$ column by column. In the j-th step, the j-th column $\mathbf{q_j}$ is constructed using the following approach:

- *Taylor approximation:* First, the linear part g_L of the function g is determined using the first order Taylor expansion.
- *Optimal transformation:* Then, the j-th column $\mathbf{q_j}$ is determined such that the contribution of the j-th dimension of g_L is maximized with the orthogonality side condition $(\mathbf{q_j}, \mathbf{q_k}) = 0$ for $k = 1, \ldots, j-1$.

Next, we describe the Taylor approximation and the construction of $\mathbf{q_j}$ in more detail.

Taylor approximation In order to linearize the integrand $g_{\mathbf{Q}}(\mathbf{z}) := g(\mathbf{Qz})$ we use the first order Taylor approximation of $g_{\mathbf{Q}}$ in a fixed point $\mathbf{a} \in \mathbb{R}^d$ which yields

$$g_L(\mathbf{z}) := g_{\mathbf{Q}}(\mathbf{a}) + \sum_{j=1}^{d} \frac{\partial g_{\mathbf{Q}}}{\partial z_j}(\mathbf{a})(z_j - a_j). \tag{5.5}$$

Note that in each step j of the construction a different anchor point $\mathbf{a}$ has to be used.[1] As in [82] we here use the point $\mathbf{a} := (1, \ldots, 1, 0, \ldots, 0)$ with $j-1$ leading ones in the j-th step of the construction.

Note further that if $\mathbf{z} \in \mathbb{R}^d$ is standard normally distributed, then the contribution of the j-th dimension to the total variance of the linear function g_L is given by

$$\sigma_j := \left(\frac{\partial g_{\mathbf{Q}}}{\partial z_j}(\mathbf{a}) \right)^2 = \left(\nabla g(\mathbf{Qa})^T \mathbf{Q},\, \mathbf{e_j} \right)^2 = \left(\nabla g(\mathbf{Qa}),\, \mathbf{q_j} \right)^2, \tag{5.6}$$

where $\mathbf{e_j}$ denotes the j-th unit vector in $\mathbb{R}^d$. To compute the gradient $\nabla g(\mathbf{Qa})$ we use the finite difference approximation

$$\frac{\partial g}{\partial z_j}(y_j) \approx \frac{g(y_j + h) - g(y_j - h)}{2h}$$

with small $h > 0$.

[1] As noted in [82] a more complex and costly alternative would be to use the Taylor approximation of order j in the j-th step.

Optimal transformation Taking the linear approximation g_L into account, we then aim to determine the j-th column of the matrix $\mathbf{Q}$ such that the contribution σ_j is maximized and such that in addition the orthogonality side conditions are respected. The solution to this optimization problem is given in the following theorem, which we state and proof in a slightly different way as in [82].

Theorem 5.2. *Let $V := span\{\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}\}$, σ_j as in (5.6) and $\mathbf{w} := \nabla g(\mathbf{Qa})$. Assume that $\mathbf{w} \notin V$. Then,*

$$\mathbf{q_j} := \pm \mathbf{w}^\perp / \|\mathbf{w}^\perp\| \quad \text{with} \quad \mathbf{w}^\perp := \mathbf{w} - \sum_{i=1}^{j-1} (\mathbf{w}, \mathbf{q_i})\, \mathbf{q_i}$$

is the (up to the sign) unique solutions of the constrained optimization problem

$$\max_{\mathbf{q_j} \in V^\perp} \sigma_j \text{ with } \|\mathbf{q_j}\| = 1, \tag{5.7}$$

where $V^\perp \subset \mathbb{R}^d$ denotes the orthogonal complement to V.

Proof. We start by defining $\mathbf{w}^V \in V$ by

$$\mathbf{w}^V := \sum_{i=1}^{j-1} (\mathbf{w}, \mathbf{q_i})\, \mathbf{q_i}. \tag{5.8}$$

Since $V \oplus V^\perp = \mathbb{R}^d$ there exists $\mathbf{w}^\perp \in V^\perp$ such that

$$\mathbf{w} = \mathbf{w}^V + \mathbf{w}^\perp. \tag{5.9}$$

For $\mathbf{q_j} \in V^\perp$ we then obtain that

$$(\mathbf{q_j}, \mathbf{w}) = (\mathbf{q_j}, \mathbf{w}^V) + (\mathbf{q_j}, \mathbf{w}^\perp) = (\mathbf{q_j}, \mathbf{w}^\perp) = \|\mathbf{q_j}\|\, \|\mathbf{w}^\perp\| \cos(\theta),$$

where θ is the angle between the vectors $\mathbf{q_j}$ and $\mathbf{w}^\perp$. Using (5.6), we see that

$$\sigma_j = (\mathbf{q_j}, \mathbf{w})^2 = \|\mathbf{q_j}\|^2\, \|\mathbf{w}^\perp\|^2 \cos^2(\theta). \tag{5.10}$$

Since $\|\mathbf{q_j}\| = 1$, the maximum in (5.10) is attained if $\cos(\theta) = \pm 1$, i.e., if

$$\mathbf{q_j} = \pm \mathbf{w}^\perp / \|\mathbf{w}^\perp\|.$$

Using (5.8) and (5.9), we finally see that

$$\mathbf{w}^\perp = \mathbf{w} - \sum_{i=1}^{j-1} (\mathbf{w}, \mathbf{q_i})\, \mathbf{q_i}$$

which proves the assertion. □

Combining the Taylor approximation and the result of Theorem 5.2 for $j = 1, \ldots, d$, the matrix $\mathbf{Q}$ is now constructed as summarized in Algorithm 5.1.

Algorithm 5.1: LT-construction of the orthonormal matrix $\mathbf{Q} = (\mathbf{q_1}, \ldots, \mathbf{q_d})$.

for $j = 1, \ldots, d$ **do**

1) Set $\mathbf{a} := (1, \ldots, 1, 0, \ldots, 0)$ with $j-1$ leading ones.

2) Compute $\mathbf{w} := \nabla g(\mathbf{Qa})/\|\nabla g(\mathbf{Qa})\|$.

3) Set $\mathbf{w}^\perp := \mathbf{w} - \sum_{i=1}^{j-1} (\mathbf{w}, \mathbf{q_i})\, \mathbf{q}_i$.

4) Set $\mathbf{q_j} := \mathbf{w}^\perp/\|\mathbf{w}^\perp\|$.

Discussion

We conclude this section with several remarks on the LT-construction. Note first that the Taylor approximation is only used to find suitable columns of the matrix $\mathbf{Q}$ and not to compute the integral (5.1). Since the integral is invariant to orthogonal transformations $\mathbf{Q}$ this approach does hence not involve any approximation errors.

Note further that Algorithm 5.1 yields (for $j = 1$) the first column

$$\mathbf{q_1} = \nabla g(\mathbf{Qa})/\|\nabla g(\mathbf{Qa})\|$$

and hence the same result as Theorem 5.1. This is explained from the fact that the function g_L is of the form (5.3) with f the identity and $w_j = \frac{\partial g_{\mathbf{Q}}}{\partial z_j}(\mathbf{a})$. This shows that the LT-construction reduces integrands g of the special form (5.3) to only one dimension. Note that, e.g., Asian or basket options based on geometric averages are included in this class for which optimal transformations can be derived.

By Theorem 5.2 we see that the solution of the optimization problem (5.7) coincides with the result of the Gram-Schmidt process applied to the system

$$\{\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}, \mathbf{w}\}.$$

As alternatives to the Gram-Schmidt process also other orthogonalisation algorithms such as the Householder transformation or the Givens rotation can be used. We recommend one of the latter methods since we observed numerical instabilities with the Gram-Schmidt process in higher dimensions in particular if gradients appear which are 'nearly' linear dependent.

A further important remark is that the LT-construction can be significantly more time-consuming than other path constructions particularly in higher dimensions. In this case, also sub-optimal matrices $\mathbf{Q}$ can be used, which result if Algorithm 5.1 is only applied to the first $k < d$ columns while the remaining columns are arbitrary but orthonormal. In this way, the computational costs is significantly reduced whereas the loss of efficiency is mostly only minimal as also noted in [82]. For many application problems, the LT-construction already provides very effective

transformations with only $k = 1$ as we motivated by Theorem 5.1 and as we will illustrate in Chapter 6 by numerical experiments.

In applications from finance often integrands of the form

$$g(\mathbf{z}) := \max\{0, f(\mathbf{z})\}$$

appear. In these cases, we apply the LT-construction to the inner function f in order to to avoid the region where the integrand and thus also its gradient is zero.

The LT-construction often provides effective transformations $\mathbf{Q}$ also for integrands g which are not of the special form (5.3) as shown in [82] for Asian basket options with arithmetic averages in the Black-Scholes model and for European options in a stochastic volatility model. For these and several further applications from finance, the LT-construction leads to larger reductions of the effective dimension and to larger advantages for numerical quadrature methods than other path constructions such as BB and PCA. In Chapter 6 we provide several numerical experiments which show that in particular *dimension-adaptive sparse grid methods* can profit from this approach. This important relation results from the fact that the dimension-adaptive methods can automatically identify the important dimensions and then only refine the approximation of the integral with respect to these dimensions.

5.2 Domain Decomposition

We now address the issue that integrands from finance often have kinks or even jumps, which deteriorate the performance of numerical quadrature methods. To circumvent this problem, we here investigate the approach first to identify the areas of discontinuity or non-differentiability. Then, we decompose the total integration domain $\Omega = [0,1]^d$ or $\Omega = \mathbb{R}^d$ into sub-domains Ω_i, $i = 1, \ldots, n$, such that the integrand is smooth in the interior of Ω_i and such that all kinks and jumps are located along the boundary of these areas. Instead of one discontinuous function, in this way several smooth functions are integrated. The total integral is then given as the sum of the separate integrals, i.e,

$$If := \int_{\Omega} f(\mathbf{x})\, d\mathbf{x} = \sum_{i=1}^{n} \int_{\Omega_i} f(\mathbf{x})\, d\mathbf{x}.$$

In this way the fast convergence of the numerical methods (e.g., sparse grids) can be regained whereas the costs only increase by a constant (the number of terms in the sum), provided the cost required for the decomposition is sufficiently small such that it can be neglected.

In general such a decomposition is more expensive, however, than to integrate the function. Nevertheless, for some problem classes, the areas of discontinuity have

a particular simple form, which allows to decompose the integration domain with costs which are much smaller than the benefit which results from the decomposition. We here consider two examples of such problem classes. In Section 5.2.1, we have the information that the kink bounds the part of the integration domain where the integrand is zero and can thus be identified by root finding as proposed in [47]. In Section 5.2.2, we have the information that the discontinuities are located on hyperplanes, which allows a decomposition first into polyhedrons and then into orthants as discussed in [56].

5.2.1 Root Finding

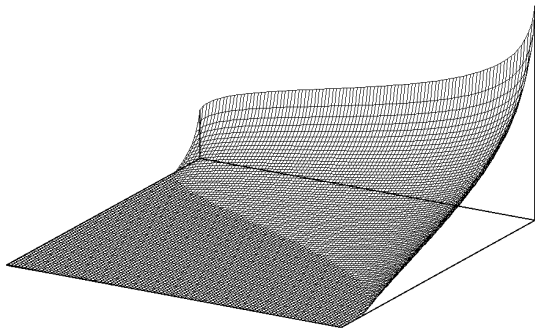
In this section, we closely follows [47] and assume that the integration domain Ω can be divided into two parts Ω_1 and Ω_2. In Ω_1 the integrand f is smooth and positive whereas $f(\mathbf{x}) = 0$ in Ω_2. Hence,

$$If = \int_{\Omega_1} f(\mathbf{x})\, d\mathbf{x}.$$

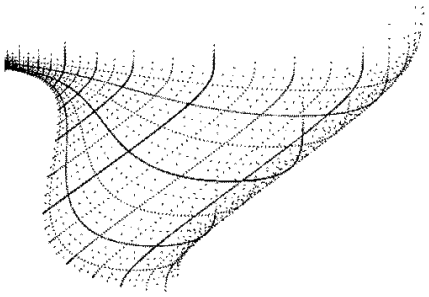
Along the boundary between Ω_1 and Ω_2, the integrand is non-differentiable or non-continuous. This situation arises in several applications from finance, e.g. if Asian options have to be priced.

We now use the idea that for these problems, kinks and jumps can efficiently be identified by a one-dimensional root finding. Then, the kinks and jumps can be transformed to the boundary of integration domain such that they no longer deteriorate the performance of the numerical methods. In order to find the kinks and jumps first the zeros of the integrand are computed. Using iterated integration, the root finding can thereby be restricted to only the last dimension. In this dimension, then, e.g., Newton's method or bisection can be used to identify the point which separates Ω_1 and Ω_2. By a coordinate-wise linear transformation with respect to the last dimension, Ω_1 can be mapped onto the total integration domain Ω. In this way, standard numerical quadrature methods (e.g. sparse grids) can be applied such that they only integrate the smooth part of the function, while they omit the part where the function is zero.

This approach is illustrated in Figure 5.3 for the problem to price an Asian option with two time steps using sparse grids. The application problem is discussed in detail in Section 6.2. Figure 5.3(a) shows the arising integrand defined on the unit cube $\Omega = [0, 1]^2$. One can see that the two parts Ω_1 and Ω_2 are separated by a one-dimensional manifold along which the integrand exhibits a kink. Using coordinate-wise transformations with respect to this manifold sparse grids are obtained as shown in Figure 5.3(b) which only cover the smooth part Ω_1 of the integrand and thus avoid the problem of low regularity. For further details on this approach and for numerical experiments with applications from finance we refer to [47].



(a) Integrand after transformation with the cumulative normal distribution function.



(b) Sparse grids after root finding and coordinate-wise transformation.

Figure 5.3. *Domain decomposition for an Asian options with two time steps. These two figures are taken from [47].*

5.2.2 Hyperplane Arrangements

We here assume that all kinks and jumps of the integrand f are located on lines, faces and hyperplanes which depend on the input parameter of the application problem. In this case, the decomposition reduces to the problem to decompose a given hyperplane arrangement into polyhedrons. The pricing of performance dependent options is an example where this approach can be applied, see [56].

Decomposition into polyhedrons

To formalize this ansatz and to illustrate the main ideas, we use n to denote the number of hyperplanes which is required to describe the location of the discontinuities. We assume $n \geq d$, where d refers to the dimension of the space. Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and a vector $\mathbf{b} \in \mathbb{R}^d$ the i-th hyperplane is defined by

$$H_i := \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a_i} \cdot \mathbf{x} = b_i\}$$

in the space $\mathbb{R}^d$ where $\mathbf{a_i}$ denotes the i-th row of the matrix $\mathbf{A}$. We hence assume that the kinks or jumps of the integrand are contained in the manifold of points $\mathbf{x}$ which satisfy the linear system $\mathbf{Ax} = \mathbf{b}$. The set of hyperplanes H_i induces a dissection of the space into different domains or cells which is called a *hyperplane arrangement* and is denoted by $\mathcal{A}_{n,d}$. Hyperplane arrangements are one of the fundamental concepts in geometry and topology. Their topological properties have been studied thoroughly in many publications, for a summary see, e.g., [34, 128]. Each cell in the hyperplane arrangement $\mathcal{A}_{n,d}$ is a (possibly open) polyhedron P which is uniquely represented by a *position vector* $\mathbf{p} \in \{+, -\}^n$. Each element of the position vector indicates on which side of the corresponding hyperplane the polyhedral cell is located. The position vectors of an example hyperplane arrangement with three planes (lines) in dimension two are shown in Figure 5.4.

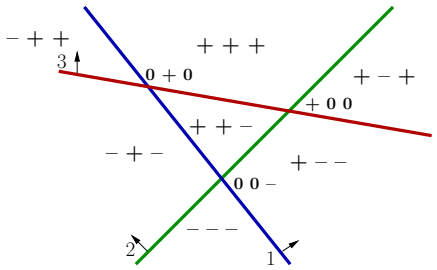


Figure 5.4. *Example hyperplane arrangement $\mathcal{A}_{3,2}$. Shown are the position vectors* $\mathbf{p}$ *of the* 7 *cells and the* 3 *vertices.*

In order to formally define the polyhedral cells, we use a comparison relation $\geq_{\mathbf{p}}$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with respect to the position vector $\mathbf{p}$. It is defined by

$$\mathbf{x} \geq_{\mathbf{p}} \mathbf{y} \;:\Leftrightarrow\; p_i(x_i - y_i) \geq 0 \quad \text{for } 1 \leq i \leq n.$$

Thus, the comparison relation $\geq_{\mathbf{p}}$ is the usual component-wise comparison where the direction depends on the sign of the corresponding entry of the position vector $\mathbf{p}$. This way, each cell in the hyperplane arrangement has the representation

$$P_{\mathbf{p}} = \left\{ \mathbf{x} \in \mathbb{R}^d : \mathbf{A}\mathbf{x} \geq_{\mathbf{p}} \mathbf{b} \right\}.$$

Moreover, each face and each vertex of a hyperplane arrangement can be characterized by a position vector $\mathbf{p} \in \{+, 0, -\}^n$. If the entry p_i is zero, then the corresponding face or vertex is located on hyperplane i. In Figure 5.4, also the three arising vertices are labeled with their position vectors.

A hyperplane arrangement is called *non-degenerate* if any d hyperplanes intersect in a unique vertex and if any $d+1$ hyperplanes possess no common points. In the following, we always assume that the non-degeneracy condition is satisfied.[2] In a non-degenerate hyperplane arrangement there are exactly $\binom{n}{d}$ vertices and

$$c_{n,d} := \sum_{i=0}^{d} \binom{n}{d-i} \tag{5.11}$$

cells, see [34]. Note that non-degenerate arrangements maximize the number of vertices and cells.

In Table 5.1 we show the number of cells in a non-degenerate hyperplane arrangement for various n and d. These complexities have to be taken into account since the number of cells determine the costs of the decomposition. For large n and small d, we have $c_{n,d} \ll 2^n$, i.e., the number of cells in the hyperplane arrangement $\mathcal{A}$ is significantly smaller than the number of different position vectors $\mathbf{p}$. For constant d, we see that the number of cells in a hyperplane arrangement grows like $O(n^d)$.

[2] In the case this condition is not met, it can be ensured by slightly perturbing some entries of the matrix $\mathbf{A}$.

Table 5.1. *Number of cells $c_{n,d}$ in a non-degenerate hyperplane arrangement for varying n and d.*

n	$2^n = c_{n,n}$	$c_{n,20}$	$c_{n,10}$	$c_{n,5}$	$c_{n,3}$
2	4	-	-	-	-
4	16	-	-	-	15
8	256	-	-	219	93
16	65536	-	58651	6885	697
30	1.1e+9	1.0e+9	5.3e+7	1.7e+5	4526

Next, we denote by $\mathcal{P}$ the set of all position vectors $\mathbf{p}$ which correspond to cells in the hyperplane arrangement $\mathcal{A}$. Then, the integral If can be determined by

$$If = \sum_{\mathbf{p}\in\mathcal{P}} \int_{P_\mathbf{p}} f(\mathbf{x})\, d\mathbf{x}, \tag{5.12}$$

provided we can identify $\mathcal{P}$ and integrate f over each cell of the hyperplane arrangement separately. Note that only smooth integrands appear in this approach and that the number of terms is given by $c_{n,d}$.

For illustration, we show in Figure 5.5(a) the integrand which arises in the problem to price a performance-dependent option[3]. One can see that the integrand is divided into several regions which are divided by kinks or jumps. The structure of the discontinuities is also shown in Figure 5.5(b). One can see that the underlying geometry is described by an hyperplane arrangement which contains $n = 5$ hyperplanes in the $d = 2$ dimensional space. Thus, $c_{n,d} = 16$ cells arise of which 13 can be seen in the figure. Each cell corresponds to a different position vector $\mathbf{p}$ as indicated in the figure.

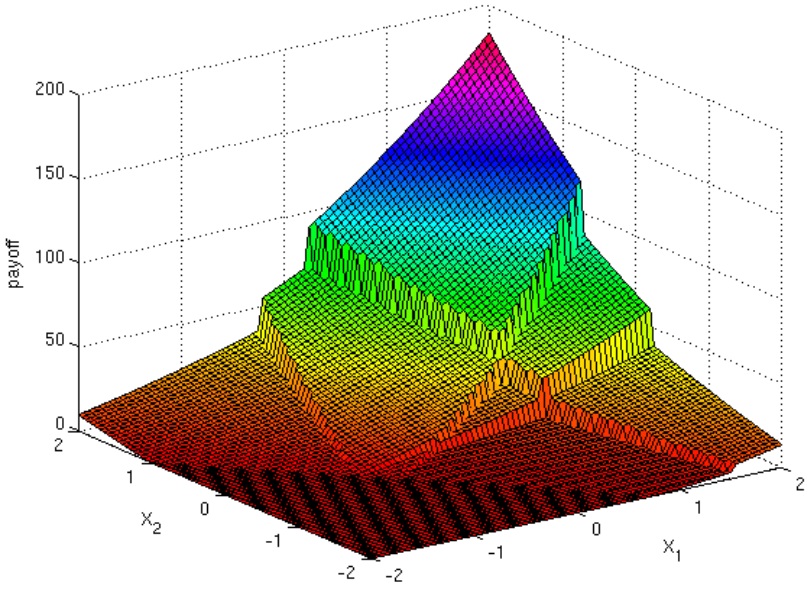
Decomposition into Orthants

Two problems remain with formula (5.12), however. First, it is not easy to see which position vectors $\mathbf{p}$ and corresponding polyhedra $P_\mathbf{p}$ appear in the hyperplane arrangement and which do not. Second, the integration region is now a general polyhedron and, therefore, involved integration rules are required. To resolve these difficulties we need some more utilities from computational geometry summarized in the following two lemmas.

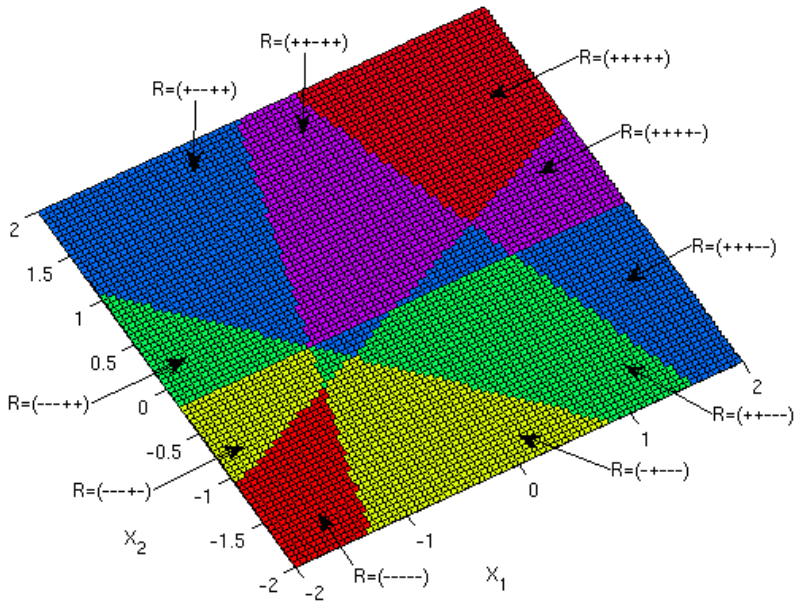
To state the first lemma, we assume[4] here that no row of the matrix $\mathbf{A}$ is a multiple of one of the unit vectors in $\mathbb{R}^d$. The unit vectors impose an order on all

[3]The payoff function of a performance-dependent option is defined in Section 6.3. The integrand in Figure 5.5(a) corresponds to the performance-dependent option in Example 6.5 with $n = 5$ and $d = 2$.

[4]If this condition not holds, it can be ensured by slightly perturbing some of entries of the matrix.

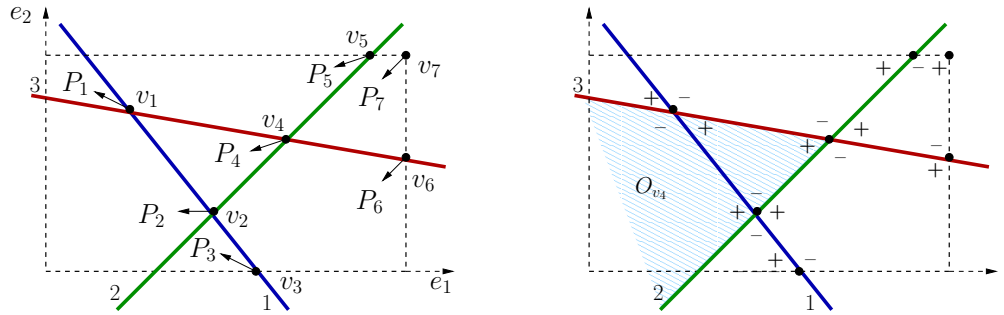


(a) Integrand $f : \mathbb{R}^2 \to \mathbb{R}$ arising in the pricing of the performance-dependent option from Example 6.5 with $n = 5$ and $d = 2$.



(b) Corresponding hyperplane arrangement $\mathcal{A}_{5,2}$ and corresponding position vectors $\mathbf{R} = \mathbf{p} \in \{+, -\}^5$.

Figure 5.5. *Domain decomposition into polyhedral cells.*



(a) Mapping between intersection points $\{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ and polyhedral cells $P_j := P_{\mathbf{v}_j}$.

(b) Reflection signs $s_{\mathbf{v},\mathbf{w}}$ and the orthant $O_{\mathbf{v}_4}$.

Figure 5.6. *Notation for the decomposition of the hyperplane arrangement from Figure 5.4.*

vertices. A vertex $\mathbf{v}$ is said to be smaller than another vertex $\mathbf{w}$ if $v_1 < w_1$. If v_1 and w_1 happen to be equal, v_2 and w_2 are compared, and so on.

The position of each vertex can be computed by solving the corresponding $d \times d$ linear system. By computing the minimum and maximum vertex of the hyperplane arrangement in each direction, an artificial bounding box which encompasses all vertices is defined. This bounding box is only needed for the localization of the polyhedral cells in the following lemma and does not implicate any approximation.

Lemma 5.3. *Let the set $\mathcal{V}$ consist of all vertices of the hyperplane arrangement, of the largest intersection points of the hyperplanes with the bounding box and of the largest corner point of the bounding box. Furthermore, let $P_{\mathbf{v}} \in \mathcal{A}$ be the polyhedron which is adjacent to the vertex $\mathbf{v} \in \mathcal{V}$ and which contains no other vertex which is larger than $\mathbf{v}$. Then the mapping $\mathbf{v} \mapsto P_{\mathbf{v}}$ is one-to-one and onto.*

Proof. The proof of Lemma 5.3 uses a sweep plane argument and induction over d. It can be found in [55]. □

For the two dimensional example with three hyperplanes in Figure 5.4 the mapping between intersection points and polyhedral cells is illustrated in Figure 5.6(a). Each vertex from the set $\mathcal{V} := \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ is mapped to the polyhedral cell indicated by the corresponding arrow. Using Lemma 5.3, an easy to implement optimal order $O(c_{n,d})$ algorithm which enumerates all cells in an hyperplane arrangement can be constructed. Note that by Lemma 5.3 each vertex $\mathbf{v} \in \mathcal{V}$ corresponds to a unique cell $P_{\mathbf{v}} \in \mathcal{A}$ and thus to a unique position vector $\mathbf{p}$.

Next, we assign each vertex $\mathbf{v}$ an associated orthant $O_{\mathbf{v}}$. An *orthant* is defined

as an open region in $\mathbb{R}^d$ which is bounded by at most d hyperplanes. Note that each vertex is the intersection of $0 \leq k \leq d$ hyperplanes of the hyperplane arrangement with $d-k$ boundary hyperplanes of the bounding box. To find the orthant $O_{\mathbf{v}}$ associated with the vertex $\mathbf{v}$, we determine k points which are smaller than $\mathbf{v}$ and which lie on the intersection of $d-1$ of these d hyperplanes. These points are found by solving a $d \times d$ linear system where $d-1$ equations are given by the intersecting hyperplanes and the last equation is $x_1 = v_1 - \varepsilon$ with $\varepsilon > 0$. The unique orthant which contains $\mathbf{v}$ and all smaller points is denoted by $O_{\mathbf{v}}$.

For illustration, the orthant $O_{\mathbf{v}_4}$ is displayed in Figure 5.6(b). Note that vertices which are located on the boundary correspond to orthants with $k < d$ intersecting hyperplanes. For example, $O_{\mathbf{v}_3}$ is defined by all points which are below hyperplane one.

By definition, there exists a $(k \times d)$-submatrix $\mathbf{A}_{\mathbf{v}}$ of $\mathbf{A}$ and a k-subvector $\mathbf{b}_{\mathbf{v}}$ of $\mathbf{b}$ such that the orthant $O_{\mathbf{v}}$ can be characterised as the set

$$O_{\mathbf{v}} = \left\{ \mathbf{x} \in \mathbb{R}^d : \mathbf{A}_{\mathbf{v}}\mathbf{x} \geq_{\mathbf{p}} \mathbf{b}_{\mathbf{v}} \right\}, \tag{5.13}$$

where $\mathbf{p}$ is the position vector which corresponds to $\mathbf{v}$. This way, the submatrix $\mathbf{A}_{\mathbf{v}}$ and the subvector $\mathbf{b}_{\mathbf{v}}$ consist of exactly those rows of $\mathbf{A}$ and $\mathbf{b}$ whose corresponding hyperplanes intersect in $\mathbf{v}$.

Furthermore, given two vertices $\mathbf{v}, \mathbf{w} \in \mathcal{V}$, we define the reflection sign $s_{\mathbf{v},\mathbf{w}} := (-1)^{r_{\mathbf{v},\mathbf{w}}}$ where $r_{\mathbf{v},\mathbf{w}}$ is the number of reflections on hyperplanes needed to map $O_{\mathbf{w}}$ onto $P_{\mathbf{v}}$. The reflection signs $s_{\mathbf{v},\mathbf{w}}$ with $\mathbf{v} \in \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$ and $\mathbf{w} \in P_{\mathbf{v}}$ arising in the two dimensional arrangement in Figure 5.4 are displayed in Figure 5.6(b). For instance, the three reflection signs in the cell $P_{\mathbf{v}_4}$ are given by $s_{\mathbf{v}_4,\mathbf{v}_1} = +$, $s_{\mathbf{v}_4,\mathbf{v}_2} = -$ and $s_{\mathbf{v}_4,\mathbf{v}_4} = +$. Finally, let $\mathcal{V}_{\mathbf{v}}$ denote the set of all vertices of the polyhedron $P_{\mathbf{v}}$.

Lemma 5.4. *It is possible to algebraically decompose any cell of a hyperplane arrangement into a signed sum of orthant cells by*

$$\chi(P_{\mathbf{v}}) = \sum_{\mathbf{w} \in \mathcal{V}_{\mathbf{v}}} s_{\mathbf{v},\mathbf{w}} \chi(O_{\mathbf{w}})$$

where χ is the characteristic function of a set. Moreover, all cells of a hyperplane arrangement can be decomposed into a signed sum of orthants using exactly one orthant per cell.

Proof. The first part of Lemma 5.4 is originally due to Lawrence [100]. The second part follows from the one-to-one correspondence between orthants $O_{\mathbf{v}}$ and cells $P_{\mathbf{v}}$. It can be found in detail in [55]. □

We can now state the main result of this section.

Theorem 5.5. *Let the set $\mathcal{V}$ of vertices, the cell $P_{\mathbf{w}}$, the orthant $O_{\mathbf{v}}$ and the*

reflection sign $s_{\mathbf{v},\mathbf{w}}$ *for* $\mathbf{v}, \mathbf{w} \in \mathcal{V}$ *be defined as above. Then, the decomposition*

$$If = \sum_{\mathbf{v}\in\mathcal{V}} c_{\mathbf{v}} \int_{O_{\mathbf{v}}} f(\mathbf{x})\, d\mathbf{x} \tag{5.14}$$

holds with $c_{\mathbf{v}} \in \{+1, -1\}$ *where*

$$c_{\mathbf{v}} := \sum_{\mathbf{w}\in\mathcal{V}:\ \mathbf{v}\in P_{\mathbf{w}}} s_{\mathbf{v},\mathbf{w}}. \tag{5.15}$$

Proof. Representation (5.12) shows that the integration domain Ω can be decomposed into $c_{n,d}$ many polyhedrons $P_{\mathbf{p}}$. By Lemma 5.3, each position vector $\mathbf{p} \in \mathcal{P}$ can uniquely be mapped onto a vertex $\mathbf{v} \in \mathcal{V}$ which yields

$$If = \sum_{\mathbf{v}\in\mathcal{V}} \int_{P_{\mathbf{v}}} f(\mathbf{x})\, d\mathbf{x}.$$

By Lemma 5.4 we can decompose each polyhedron $P_{\mathbf{v}}$ into a signed sum of orthants and obtain

$$If = \sum_{\mathbf{v}\in\mathcal{V}} \sum_{\mathbf{w}\in\mathcal{V}_{\mathbf{v}}} s_{\mathbf{v},\mathbf{w}} \int_{O_{\mathbf{w}}} f(\mathbf{x}) d\mathbf{x}.$$

By the second part of Lemma 5.4 we know that only $c_{n,d}$ different integrals appear in the above sum. Rearranging the terms leads to the assertion. □

Note that only smooth functions are integrated in (5.14) and that the number of terms is given by $c_{n,d}$. Note further that the complexity of (5.12) to compute $c_{n,d}$ integrals over general polyhedral cells is reduced by Theorem 5.5 to the complexity to compute $c_{n,d}$ integrals over orthant cells. This is a significant simplification since orthants can easily be mapped to the unit cube. This way standard numerical quadrature methods can directly be applied for the computation of the corresponding integrals. Note finally that such an orthant decomposition is not unique. A different decomposition of a polyhedron into a sum of orthants is presented in [35].

Example 5.6. The decomposition of all cells within the hyperplane arrangement from Figure 5.6(a) is given by

$$\begin{aligned}
\chi(P_1) &= \chi(O_1) \\
\chi(P_2) &= \chi(O_2) - \chi(O_1) \\
\chi(P_3) &= \chi(O_3) - \chi(O_2) \\
\chi(P_4) &= \chi(O_4) - \chi(O_2) + \chi(O_1) \\
\chi(P_5) &= \chi(O_5) - \chi(O_4) - \chi(O_1) \\
\chi(P_6) &= \chi(O_6) - \chi(O_4) - \chi(O_3) + \chi(O_2) \\
\chi(P_7) &= \chi(O_7) - \chi(O_6) - \chi(O_5) + \chi(O_4)
\end{aligned}$$

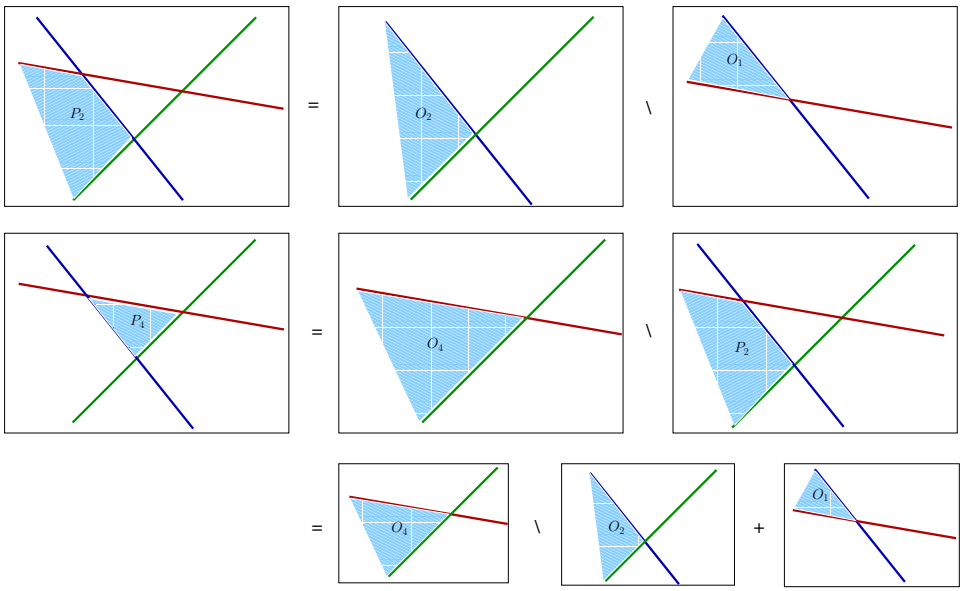


Figure 5.7. *Decomposition of the three cells P_1, P_2 and P_4 using the three orthants O_1, O_2 and O_4.*

where we used the abbreviations $P_j := P_{\mathbf{v}_j}$ and $O_j := O_{\mathbf{v}_j}$. We see that seven orthants are required for the decomposition of seven cells. In Figure 5.7, the decomposition of the three polyhedral cells P_1, P_2, P_4 into the three orthants O_1, O_2, O_4 is illustrated. Note that the orthant O_1 coincides with the cell P_1.

We will return to the presented orthant decomposition in Section 6.3 and apply it there to derive pricing formulas for performance-dependent options.

5.2.3 Conditional Sampling

Most financial derivatives have non-smooth payoff functions. Hence, also the integrands of the corresponding integrals are not smooth. In this section, we consider the approach to smooth these integrands through the use of *conditional sampling* following [59], Section 7.2.3.

We first describe the main idea for the situation that the expected value of a stochastic process has to be computed. For illustration we consider an specific example which corresponds to the problem to price a discretely monitored knock-out barrier option.[5] To this end, typically an expected value of the form

$$P := \mathbb{E}[f(\mathbf{S})] \tag{5.16}$$

[5]Barrier options are the topic of Section 6.2.2. They are financial derivatives which become worthless if the underlying asset crosses a specified barrier.

has to be computed. Here $\mathbf{S} := (S(t_1), \ldots, S(t_d))$ collects the values of a stochastic process $S(t)$, e.g. a geometric Brownian motion in the simplest case, at the discrete points in time $t_k = k\,\Delta t$ for $k = 1, \ldots, d$. The function $f : \mathbb{R}^d \to \mathbb{R}$ corresponds to the payoff function of the barrier option. We assume that $f(\mathbf{S})$ is zero if $S(t) < \underline{B}$ or $S(t) > \overline{B}$ for any $t \in \{t_1, \ldots, t_d\}$. Here $\underline{B} < S(0) < \overline{B}$ are barriers which are specified in the contract of the barrier option. Typically, f is not differentiable and has kinks and jumps which are caused by the knock-out barrier conditions. The function f is smooth, however, on the space of paths $\mathbf{S}$ which not hit one of the barriers.

Often, the distribution of the path $\mathbf{S}$ is based on an underlying d-dimensional Gaussian distribution. Then, the expected value (5.16) corresponds to an d-dimensional integral over $\mathbb{R}^d$ with Gaussian weight, see Section 6.2. Since f has kinks and jumps also the corresponding integrand has kinks and jumps, which heavily degrade the performance of numerical quadrature methods.

To smooth the integrand in (5.16), we next construct a modified process $\tilde{S}(t)$ that has the same initial value $\tilde{S}(0) = S(0)$ but never crosses the relevant barriers.

This can be achieved in the following way: If $S(t_k)$ has the distribution F, then $S(t_k)$ can be generated via

$$S(t_k) = F^{-1}(x)$$

by the inverse transformation method from an random number x which is uniformly distributed in the unit interval. Alternatively, we can generate $S(t_k)$ *conditional on* $\underline{B} \le S(t_k) \le \overline{B}$ from x by

$$\tilde{S}(t_k) = F^{-1}\left[F(\underline{B}) + \big(F(\overline{B}) - F(\underline{B})\big)\, x\right],$$

see [59], Example 2.2.5.

Since $f(\mathbf{S}) = 0$ if $S(t_k) < \underline{B}$ or $S(t_k) > \overline{B}$ for any $k = 1, \ldots, d$, we obtain

$$P = \mathbb{E}[f(\tilde{\mathbf{S}})], \tag{5.17}$$

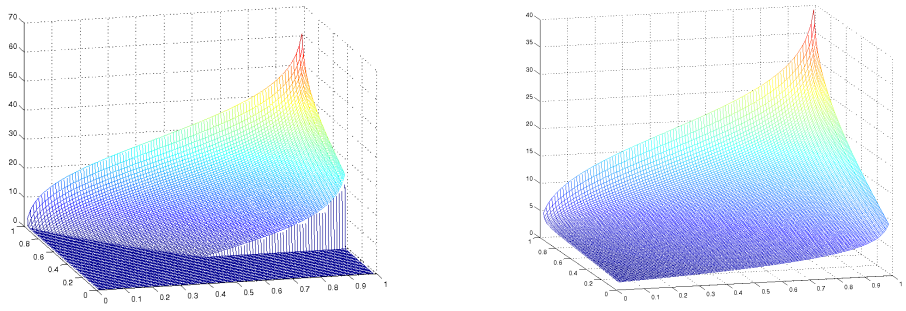
where $\tilde{\mathbf{S}} = (\tilde{S}(t_1), \ldots, \tilde{S}(t_d))$

Since $f(\tilde{\mathbf{S}})$ is a smooth function of $\tilde{\mathbf{S}}$, the value P can now be computed by the integration of a smooth function instead of a function with kinks and jumps as in (5.16).

For illustration, we show in Figure 5.8 the integrands in (5.16) and (5.17) which arise in the pricing of discretely monitored knock-out barrier options with four time steps.[6] The smoothing effect of the conditional sampling approach is clearly visible.

Note that this approach involves no approximation errors. It is closely related to the domain decomposition in Section 5.2.1. While we used there numerical search algorithms to identify the region Ω_2 where the integrand is zero, we here use conditional sampling (and hence a priori information on the underlying stochastic process)

[6] We used the parameters from Section 6.2.2.



(a) Integrands (5.16) (b) Integrand (5.17)

Figure 5.8. *First two dimensions of the integrands which arise in the pricing of discretely monitored knock-out barrier options without and with smoothing.*

to avoid the zero region of the integrand. The domain Ω_1 in Section 5.2.1 thus exactly contains those points $\mathbf{x} \in \mathbb{R}^d$ which correspond to sample paths $\mathbf{S}$ that do not cross the barriers. Note finally that conditional sampling not only smoothness the integrand but also reduces its variance. It can hence also be used to improve the constant of Monte Carlo simulations.

In Section 6.2.2 we will apply conditional sampling in combination with Monte Carlo, quasi-Monte Carlo and sparse grid methods to price discretely monitored knock-out barrier options efficiently.

We conclude this chapter with a short summary. We here discussed several approaches for dimension reduction and smoothing which can be used to enhance the performance of sparse grid and of other dimension-wise quadrature methods. We first considered orthogonal transformations, in particular the linear transformation method of Imai and Tan, which aim to reduce the effective dimension of the integrand. Then, we considered domain decompositions as a possible approach to address low regularity. In the next chapter, we will use applications from finance and insurance to study the impact of these approaches on sparse grid and other numerical quadrature methods.

Chapter 6
Validation and Industrial Applications

In this chapter, we present various numerical experiments with different applications from finance. We study the performance of the different numerical quadrature methods from Chapter 3 and Chapter 4 and investigate the impact of our different approaches for dimension reduction and smoothing from Chapter 5.

Unless states otherwise we compare the following numerical methods:

- *Monte Carlo integration (MC):* The Monte Carlo method is applied without variance reduction techniques since good control variates or importance sampling functions are in general difficult to obtain.

- *Quasi-Monte Carlo integration based on Sobol point sets (QMC):* In preliminary numerical experiments, Sobol point sets turned out to be the most efficient representative of several quasi-Monte Carlo variants. We compared Halton, Faure and Sobol low discrepancy point sets and different lattice rules[1] with and without randomisation.

- *Dimension-adaptive sparse grids based on Gauss-Patterson formulas (SGP):* This method calculates integrals over the unit cube in a dimension-adaptive way, see Section 3.2.4. It was first presented in [49]. The method provided more precise results than sparse grid methods based on the trapezoidal, the Clenshaw-Curtis and the Gauss-Legendre rule in almost all of our experiments. We also tested several other grid refinement strategies [118]. These alternatives turned out to be similar or less effective than the dimension adaptive refinement from [49] and are thus not further investigated here.

- *Dimension-adaptive sparse grids based on Gauss-Hermite formulas (SGH):* This method computes integrals over $\mathbb{R}^d$ with Gaussian weight using the dimension-adaptive sparse grid approach [49] in combination with univariate Gauss-Hermite formulas, see Remark 4.1. We use the slowly increasing Gauss-Hermite sequence from Section 4.1.2. Dimension-adaptive sparse grids based on the Genz-Keister sequence or on the classical Gauss-Hermite sequence led to slightly worse results in most of our experiments.

[1] Lattice rules [149] yield in many cases equal or even more precise results than Sobol points if good weights are used in their CBC construction. But the selection of good weights is a priori not always clear and is thus not further investigated here.

Note that SGP and SGH are both special cases of the dimension-wise quadrature method (3.15), see Theorem 4.14 and Remark 4.15. In our numerical experiments we will consider the following applications from finance and insurance:

- *Interest rates derivatives:* We first consider the pricing of collateralized mortgage obligations and the valuation of zero coupon bonds. These applications lead to the integration of smooth but high-dimensional functions, where the large number of dimensions arises from small time steps in the time discretization. To reduce the effective dimension of the integrand and to enhance the performance of the QMC, SHP and SGH method, we apply the different path generating methods from Section 5.1.

- *Path-dependent options:* We then compute the prices of Asian options and of barrier options. Here, high-dimensional integrands appear which are not differentiable or even discontinuous. To overcome the obstacle of low regularity, we apply local adaptivity in the low order anchored-ANOVA terms using the mixed CUHRE/QMC methods from Example 3.5 in Section 3.2. We furthermore consider the approach from Section 5.2.3 to recover smoothness by conditional sampling.

- *Performance-dependent options:* The payoff of performance-dependent options depends on many underlying assets and is typically discontinuous. The valuation of such options thus requires the integration of high-dimensional, discontinuous functions. Here, we use the domain decomposition approach from Section 5.2.2 to regain smoothness and the principal component analysis to reduce the dimension.

- *Asset-liability management models in life insurance:* Finally, we consider the numerical simulation of stochastic asset-liability management models in life insurance. This is the most complex application considered in this thesis. This problem requires the computation of several complicated, recursively defined integrands with many variables. To reduce the dimension, we here again focus on the different path constructions from Section 5.1.

All of these applications lead to the computation of high-dimensional integrals over $\mathbb{R}^d$ with Gaussian weight. The SGH method can treat such integrals directly on $\mathbb{R}^d$. To apply the MC, QMC and SGP method we need to transform the integrals over $\mathbb{R}^d$ into integrals over the d-dimensional unit cube $[0,1]^d$. To this end, we use the standard substitution with the inverse of the cumulative normal distribution function. Remember that such transformations to the unit cube introduce singularities which deteriorate the efficiency of methods that take advantage of higher smoothness such as sparse grids, compare also Remark 3.1.

Depending on the application problem, we consider different nominal dimensions ranging from $5-512$. We will see that sparse grid or other dimension-wise quadrature methods are superior to Q(MC) in all of our model problems. Nevertheless, we emphasise that all results depend on the input parameters of the problems since they affect their *smoothness* and/or their *effective dimension*. Different choices of the parameters may thus lead to better or worse results. Changes in the parameters often have a rather large impact on SG methods, a moderate one on QMC and a small one on MC.

To gain insight into the importance of dimensions and to explain the convergence behaviour of the numerical methods, we will compute the effective dimension of all model problems in the classical case and in the anchored case using the methods from Section 2.1.1 and Section 2.2.1. We will see that most of the application problems are of low effective dimension or can be transformed to be of low effective dimension, which justifies the use of deterministic numerical methods for their computation.

The remainder of this chapter is organized as follows: In Section 6.1, we consider the pricing of different interest rates derivatives to validate our algorithms and to demonstrate the high efficiency of the SGH method for integrands which are smooth and of low effective dimension. Section 6.2 is about the valuation of path-dependent options and different approaches to overcome the obstacle of low regularity. Section 6.3 is concerned with the pricing of performance-dependent options. Here, we apply the domain decomposition approach from Section 5.2.2 to regain smoothness. Section 6.4 finally deals with the complex application to simulate stochastic asset-liability management models in life insurance. In Section 6.5 we conclude this chapter with a short summary and discussion of the presented results.

6.1 Interest Rates Derivatives

In this section, we consider the pricing of *collateralized mortgage obligations* and the valuation of *zero coupon bonds*. Both applications lead to high-dimensional integrals where the large number of dimensions arises from small time steps in the time discretization. To reduce the effective dimension of these integrals we compare the different methods from Section 5.1 to generate sample paths of the stochastic processes which model the interest rate movements. Since the resulting integrands are smooth functions, no smoothing is required and sparse grid methods are particularly efficient as we will demonstrate by numerical experiments.

6.1.1 Zero Coupon Bonds

We here consider the problem to price zero coupon bonds by simulating the short-term interest rate $r(t)$ using the Vasicek model.[2] The same problem is also consid-

[2]For more information on the Vasicek model and other short rate models we refer to [14].

ered in [115, 120, 159] to analyse the behaviour of QMC methods.

Modeling

In the Vasicek model the movement of the short-term interest rate is given by

$$dr(t) = \kappa(\theta - r(t))dt + \sigma dW(t), \tag{6.1}$$

where $W(t)$ is the standard Brownian motion, $\theta > 0$ denotes the mean reversion level, $\kappa > 0$ denotes the reversion rate and $\sigma \geq 0$ denotes the volatility of the short rate dynamic.

For the solution of the above stochastic differential equation until time $t = T$, we use an Euler-Maruyama discretization with step size $\Delta t := T/d$ on the grid $t_k := \Delta\, k$, $k = 1, \ldots, d$, which yields the time-discrete version

$$r_k = r_{k-1} + \kappa(\theta - r_{k-1})\Delta t + \sigma\, z_k. \tag{6.2}$$

Here $z_k := W(t_k) - W(t_{k-1})$ is normally distributed with mean zero and variance Δt and denotes the increment of the Brownian motion in the k-th time interval.

Based on the short-term interest rates (6.2) the price $P(0, T)$ at time $t = 0$ of a zero coupon bond with maturity date T is given by

$$P(0, T) = \mathbb{E}\left[\exp\left\{-\Delta t \sum_{k=0}^{d} r_k\right\}\right]. \tag{6.3}$$

It can be written as a d-dimensional integral on $\mathbb{R}^d$ with Gaussian density since the expected value is via (6.2) taken over d many normally distributed random variables z_k. The value $P(0, T)$ can also be derived in closed-form, which we use to validate our numerical methods and results.

Lemma 6.1 (Closed-form solution). *The price of the bond (6.3) at time $t = 0$ with the short-term interest rate (6.2) is given by*

$$P(0, T) = \exp\left\{-\frac{(\gamma + \beta_d\, r(0))\; T}{d}\right\} \tag{6.4}$$

where

$$\beta_k := \sum_{j=1}^{k}(1 - \kappa\Delta t)^{j-1},\ k = 1, \ldots, d, \quad \text{and } \gamma := \sum_{k=1}^{d-1} \beta_k \kappa\theta\Delta t - (\beta_k \sigma \Delta t)^2/2.$$

Proof. The proof uses that r_k is normally distributed. The derivation can be found in [154] in more detail. □

In our following numerical experiments, we always use the parameters

$$\kappa = 0.1817303, \theta = 0.0825398957, \sigma = 0.0125901, r(0) = 0.021673 \text{ and } T = 5$$

which are also used in [120]. We consider the case $d = 512$.

Effective Dimensions

We first compute the effective dimensions of the expected value (6.3) to investigate the impact of the different path constructions from Section 5.1. Here, we always compare random walk (RW), Brownian bridge (BB), principal component (PCA) and the LT-construction (LT).[3] The results are shown in Table 6.1.

We first observe that the Vasicek bond pricing problem (6.3) is of very low superposition dimension $d_s \leq 2$ in the anchored case and that d_s is almost independent of the path construction.

One can furthermore see that the truncation dimensions d_t in the classical case almost coincide with the truncation dimensions d_t in the anchored case. For instance, for $\alpha = 0.99$ we obtain $d_t = 420, 7, 1, 1$ using RW, BB, PCA and LT, respectively, for the anchored case and $d_t = 419, 7, 1, 1$ for the classical one. We conjecture that a more precise numerical computation would yield even exact equal results and we believe that this equality holds for a wider class of functions. Note however that this does not hold in general. We will give a counterexample in Section 6.1.2.

Observe in Table 6.1 that the path construction has a significant impact on the truncation dimensions. For α close to one the dimensions d_t are almost as large as the nominal dimension $d = 512$ if we employ the RW approach. The dimensions d_t are significantly smaller if BB, PCA or LT is used instead. The LT-construction even obtains the optimal result $d_t = 1$ for this problem. While it is not surprising that such an optimal transformation exists,[4] it is nevertheless interesting that it is correctly identified by the LT-construction, which takes only the gradient of the integrand at a certain anchor point into account.

This situation is also illustrated in Figure 6.1. There, we show the values T_j, $j = 0, \ldots, d-1$, from (2.17) and (2.10) in the anchored and in the classical case, respectively. The results quantify the impact of the different path constructions on the modeling error. The smaller the values T_j the more important contributions are concentrated in the first j dimensions by the path construction. One can see that BB leads to a significantly faster decay of the importance of the dimensions as RW. We observe in Figure 6.1(a) that the level-wise construction of the Brownian bridge

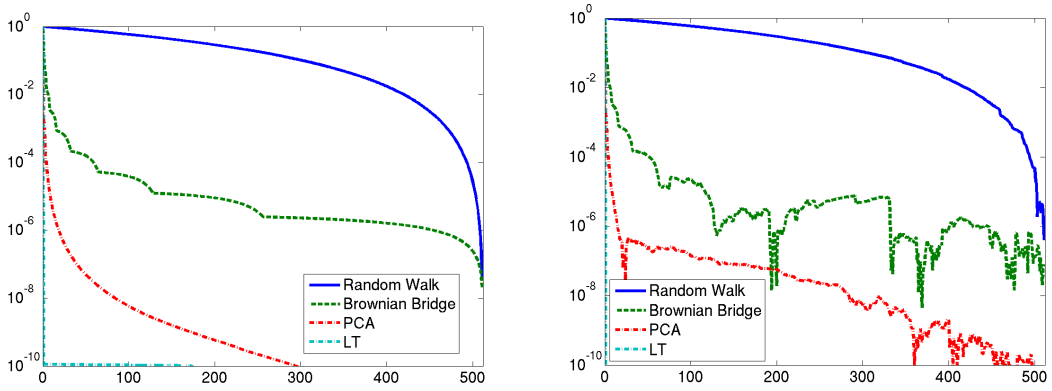
[3] Here and in the following we apply the LT-construction only to the first column of the matrix $\mathbf{A}$. The remaining columns are arbitrary but orthonormal. This way the LT-construction is significantly less expansive than the PCA-construction. The optimization of additional columns increases the computing times but led to no or only minor improvements in our numerical experiments.

[4] This way the closed-form pricing formula is derived.

Table 6.1. *Effective dimensions of the Vasicek bond pricing problem (6.3). The nominal dimension is* $d = 512$.

	anchored-ANOVA								classical ANOVA			
	superposition dim.				truncation dimensions							
$1-\alpha$	RW	BB	PCA	LT	RW	BB	PCA	LT	RW	BB	PCA	LT
1e-1	1	1	1	1	302	2	1	1	305	2	1	1
1e-2	1	1	1	1	420	7	1	1	419	7	1	1
1e-3	1	1	1	1	471	16	2	1	471	16	2	1
1e-4	2	2	1	1	494	59	5	1	494	52	5	1

(the ℓ-th level corresponds to the first 2^ℓ dimensions) is mirrored in the decay of the values T_j. The PCA in turn leads to a significantly faster decay as BB. The LT even reduces the problem to only one effective dimension. In Figure 6.1(b) one can see that the values T_j in the classical case show a similar overall behaviour as in the anchored case but exhibit oscillations in the small T_j-values. These oscillations can be reduced if the algorithm from [160] is used with a sufficiently large number of sampling points which is very expensive, though. One can see that the oscillations are avoided in the anchored case although the results are significantly cheaper to obtain.



(a) Anchored case (2.17) (b) Classical case (2.10)

Figure 6.1. *Decay of the importance of the dimensions of the Vasicek bond pricing problem (6.3) with* $d = 512$. *Note that the results of the LT-construction almost coincide with the y-axis.*

In summary, we see that the Vasicek bond pricing problem (6.3) is of very low superposition dimension d_s. It is also of low truncation dimension if BB, PCA or LT is used. Moreover, we observe that the effective dimensions in the anchored case

provide the same or almost the same information on the importance of dimensions as in the classical case but have the advantage that they are significantly cheaper to compute. While the computation of the effective dimensions in the classical case require the computation of *many* high-dimensional integrals with up to $2d-1$ dimensions,[5] the computation of the effective dimensions in the anchored case require only the computation of *one* d-dimensional integral with a sparse grid method or an other method of the class (3.15).

Quadrature Error and Costs

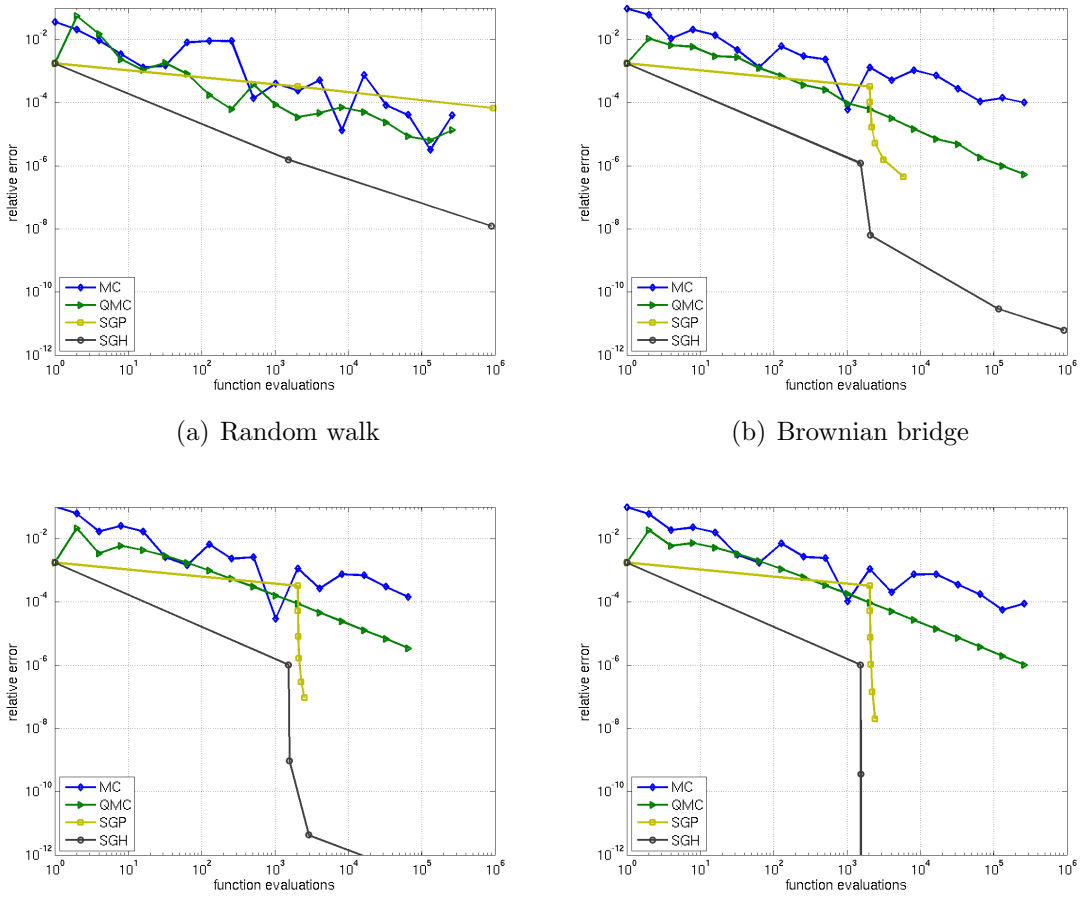
We next compute the integral value (6.3) using the different numerical quadrature and path generating methods. By a comparison of the results with the analytical solution from Lemma 6.1, we determine the relative errors of the different numerical methods. The relative errors are shown in Figure 6.2 for different numbers n of function evaluations.

One can see that the convergence rate of the MC method is always about 0.5 as predicted by the law of large numbers. The rate is not affected by the path construction since the total variance stays unchanged. The convergence rate of the QMC method increases if BB, PCA or LT is used since these path constructions concentrate the total variance in the first few dimensions. This way QMC outperforms MC and achieves higher convergence rates of almost one, smaller relative errors and a less oscillatory convergence behaviour. We furthermore observe in Figure 6.2 that the convergence of SGP and SGH is significantly accelerated by the path constructions BB, PCA and PCA which lead to a low effective dimension.

Note here that two different regimes have to be distinguished to describe the convergence behaviour of these methods, compare, e.g., Figure 6.2(d). In the preasymptotic regime, SGP and SGH first search for the important dimensions and interactions, whereas, in the asymptotic regime, the important dimensions are identified and the grid is then refined only in these directions. Since the LT-construction reduces the problem to only one dimension, its combination with dimension-adaptive methods is particularly efficient. We see from Figure 6.2(d) that SGP and SGH correctly recognize the solely important dimension and then only refine in this respect, which leads to an extremely rapid convergence in the asymptotic regime. This way the dependence on the dimension is completely avoided in the asymptotic regime. There, the convergence of the methods is as fast as it is known for univariate problems despite the high nominal dimension $d = 512$. Overall, SGH is the most efficient method. It is more efficient than SGP since it avoids the singular transformation to the unit cube. It outperforms (Q)MC by several orders of magnitude independent of the employed path construction. By exploiting the low effective dimension and

[5] The computation of the superposition dimension in the classical case in addition suffers from cancellation problems and costs which are exponential in the superposition dimension.

smoothness of the integrand, SGH achieves in combination with PCA or LT almost machine accuracy with only about 1,000 function evaluations.



(a) Random walk

(b) Brownian bridge

(c) Principal components

(d) Linear transformation

Figure 6.2. *Convergence behaviour of the different numerical methods for the Vasicek bond pricing problem (6.3) with* $d = 512$.

6.1.2 Collateralized Mortgage Obligations

The problem to price mortgage-backed securities is a commonly used test problem from finance to study the performance of numerical quadrature algorithms. It was first presented in [134]. There, a collateralized mortgage obligation (CMO) is calculated as 360-dimensional integral which was given to the authors Paskov and Traub by the investment bank Goldman Sachs. They observed that QMC methods outperform MC for this problem, which initiated extensive research to generalise their results to further problems in finance and to explain the success of the QMC despite the high dimension. The CMO problem is also considered in [19, 120] to study the performance of QMC methods and in [48, 49, 135] to demonstrate the efficiency of SG methods.

Modeling

We next describe the CMO problem following [19]. We consider a mortgage-backed security with a maturity of d months. Its holder receives d payments m_k of an underlying pool of mortgages at the times $t_k = \Delta t\, k$, $k = 1, \ldots, d$, where Δt denotes the period length of one month. The present value of the sum of all payments is then given by

$$g(\mathbf{z}) := \sum_{k=1}^{d} u_k m_k$$

where

$$u_k := \prod_{j=0}^{k-1} (1 + i_j)^{-1}$$

is the discount factor for the month k, which depends on the interest rates i_j. The interest rate i_k for month k is modelled by

$$i_k := K_0^k e^{\sigma(z_1 + \ldots + z_k)} i_0$$

using k many standard normally distributed random numbers z_j, $j = 1, \ldots, k$. Here, i_0 is the interest rate at the beginning of the mortgage, σ is a positive constant and $K_0 := e^{-\sigma^2/2}$ such that $\mathbb{E}[i_k] = i_0$. The payments m_k at time t_k are given by

$$m_k := c\, r_k((1 - w_k) + w_k c_k),$$

where c denotes the monthly payment. Moreover,

$$c_k := \sum_{j=0}^{d-k} (1+i_0)^{-j},$$
$$r_k := \prod_{j=1}^{k-1} (1-w_j),$$
$$w_k := K_1 + K_2 \arctan(K_3 i_k + K_4).$$

Here, r_k denotes the fraction of remaining mortgages at month k, which in turn depends on the fraction w_k of mortgages which prepay in month k. The values K_1, K_2, K_3, K_4 are constants of the model for the prepayment rate w_k.

The expected present value of the sum of all payments can then be written as d-dimensional integral

$$PV := \int_{\mathbb{R}^d} g(\mathbf{z}) \varphi_d(\mathbf{z}) d\mathbf{z} \tag{6.5}$$

on $\mathbb{R}^d$ with Gaussian weight φ_d.

We next focus on the efficient numerical computation of this integral. In our numerical experiments we thereby use the parameters

$$i_0 = 0.007, c = 1, K_1 = 0.01, K_2 = -0.005, K_3 = 10, K_4 = 0.5 \text{ and } \sigma = 0.0004$$

which are also used in [19,48,135]. We consider the case $d = 256$ for which we obtain the reference solution $PV = 119.21588257$.

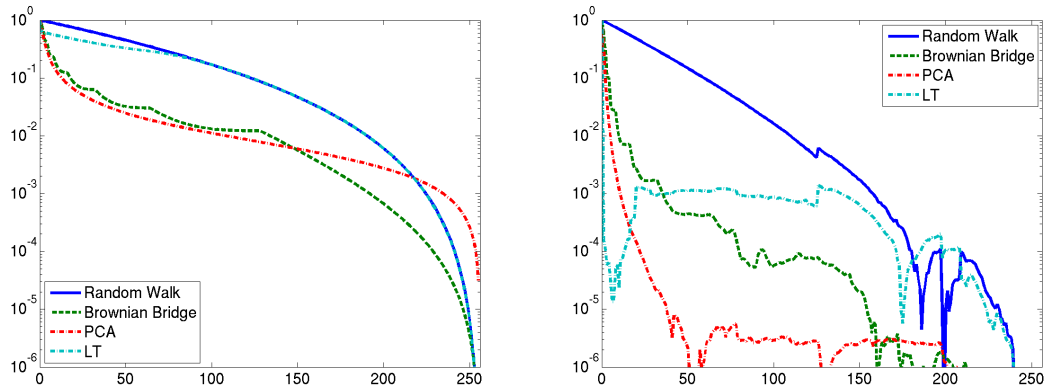
Effective Dimensions

To illustrate the impact of the different path constructions we first consider the effective dimensions of the integral (6.5). From Table 6.2, we see that the CMO

	anchored-ANOVA								classical ANOVA			
	superposition dim.				truncation dimensions							
$1-\alpha$	RW	BB	PCA	LT	RW	BB	PCA	LT	RW	BB	PCA	LT
1e-1	1	1	1	1	123	18	13	123	60	5	2	1
1e-2	2	2	1	2	191	134	108	191	110	10	5	1
1e-3	2	2	2	2	225	192	235	225	158	36	11	1
1e-4	2	2	2	2	242	229	254	242	181	80	23	1

Table 6.2. *Effective dimensions of the CMO pricing problem (6.5). The nominal dimension is* $d = 256$.

problem is of very small superposition dimension $d_s \leq 2$ in the anchored case for all $\alpha \in [0.9, 0.9999]$ and all path constructions.



(a) Anchored case (2.17) (b) Classical case (2.10)

Figure 6.3. *Decay of the importance of the dimensions of the CMO pricing problem (6.5) with* $d = 256$.

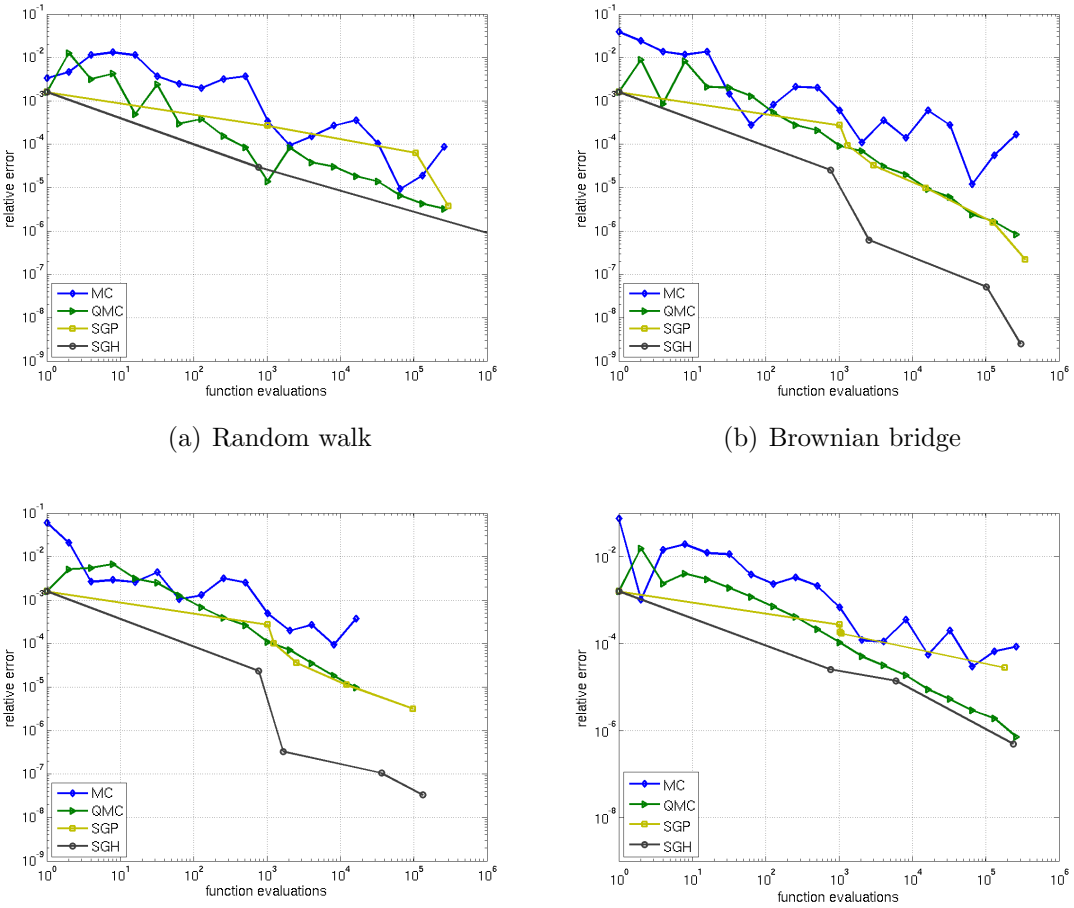
In Table 6.2 also the truncation dimensions d_t of this problem are shown. It is striking that the path construction has only a small impact on the truncation dimension in the anchored case, i.e., the advantage of BB, PCA and LT compared to RW is not so clear for the CMO problem. For $\alpha = 0.9$ we have $d_t = 123$ in case of RW and LT. This truncation dimension is reduced to $d_t = 18$ and $d_t = 13$ if BB and PCA is used, respectively. For higher accuracy requirements, however, i.e. for $\alpha \geq 0.99$, significantly less or even no reduction at all is achieved with these constructions. Note that for the CMO problem the truncation dimensions d_t in the classical case clearly differ from the truncation dimensions in the anchored case. In the classical case, BB, PCA and LT lead to significant dimension reductions. LT even reduces the problem to the truncation dimension one.

This situation is also illustrated by the values T_j from (2.17) and (2.10) which we show in Figure 6.3. The differences between the anchored and the classical case are clearly visible. While LT leads to very similar results as RW in the anchored case, LT leads to a large dimension reduction compared to RW in the classical one. In the anchored case, BB turns out to be the most efficient path construction, whereas LT is most efficient in the classical case.

Quadrature Error and Costs

We next study the convergence behaviour of the different numerical methods for this problem. To measure their accuracy we used the reference solution $PV = 119.21588$. The respective numerical results are illustrated in Figure 6.4. One can see that the QMC method converges faster, less oscillatory and superior to MC if we switch from RW to BB, PCA or LT. SGP performs similar as QMC in case of BB and

PCA and slightly worse in case of RW and LT. SGH combined with BB or PCA is most efficient for the CMO problem. It achieves the highest convergence rate and the most precise results. With 10^4 function evaluations SGH obtains a relative error which is about 100 times smaller than the relative error of the QMC method.



(a) Random walk

(b) Brownian bridge

(c) Principal components

(d) Linear transformation

Figure 6.4. *Convergence behaviour of the different numerical approaches for the CMO pricing problem (6.5) with* $d = 256$.

We next discuss the relation of the convergence behaviour of the numerical methods to the effective dimension of the CMO problem. We already showed that the path construction affects both the performance of the numerical methods (except of MC) and the truncation dimension of the integral. Since the truncation dimension in the classical case differs from the truncation dimension in the anchored case for this problem, it is interesting to see which of these two notions better predicts the convergence behaviour of the numerical methods. Remember that LT does not lead

to an improved convergence of the dimension-adaptive sparse grid methods SGP and SGH compared to RW. This observation can not be explained by the effective dimension in the classical case since LT obtains the optimal result $d_t = 1$ for the CMO problem. The observation is, however, in clear correspondence with the fact that LT provides no reduction of the truncation dimension in the *anchored* case. This indicates that the performance of SGP and SGH depends on the truncation dimension in the anchored case, but not on the truncation dimension in the classical case. Note that the convergence behaviour of the QMC method is rather related to the effective dimension d_t in the classical case than to the anchored one, since this method converges faster and less oscillatory with LT than with RW.

The different effective dimensions in the anchored and in the classical case are related to the fact that in the anchored-ANOVA decomposition the contributions $If_{\mathbf{u}}$ are of varying sign in the CMO problem. Summing the contributions thus leads to cancellation effects which are not seen in the anchored case since the absolute values $|If_{\mathbf{u}}|$ are taken into account there. Nevertheless, also the error indicator[6] of dimension-adaptive sparse grid methods is based on the absolute values $|\Delta_{\mathbf{k}}|$. These values are closely related to $|If_{\mathbf{u}}|$ as we showed in Section 4.3. The methods SGP and SGH can thus also not profit from such cancellation effects and their convergence behaviour therefore rather depends on the effective dimensions in the anchored case than on the effective dimension in the classical case.

Note finally that the truncation dimension d_t in the anchored case explains the impact of the path construction but not the high performance of the SGH method since d_t is high for this problem. The fast convergence is explained by the low superposition dimension $d_s \leq 2$ and by the smoothness of the integrand.

6.2 Path-dependent Options

In this section, we consider the pricing of European-style path-dependent options in the Black-Scholes model.[7] We denote the maturity date of the option by T and assume that the payoff of the option at time T depends on the prices $S(t)$ of an underlying asset at the equidistant points in time $t_k := \Delta t\, k$, $k = 1, \ldots, d$, with $\Delta t := T/d$. We collect these prices in the vector $\mathbf{S} := (S(t_1), \ldots, S(t_d))$.

We assume that the underlying asset follows a geometric Brownian motion

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t). \tag{6.6}$$

Here, μ denotes the drift, σ denotes the volatility and $W(t)$ is the standard Brownian motion. Using the usual Black-Scholes assumptions, see, e.g., [79, 91], the price of the option $V(S, 0)$ at time $t = 0$ is given by the discounted expectation

$$V(S, 0) = e^{-rT}\mathbb{E}[V(\mathbf{S}, T)] \tag{6.7}$$

[6] See Algorithm 4.1.

[7] For more information on the Black-Scholes model we refer to [79].

under the equivalent martingale measure, where $V(\mathbf{S}, T)$ denotes the payoff of the option at time T and r the riskless interest rate.

By Itô's lemma, the analytical solution of the stochastic differential equation (6.6) is given by

$$S(t) = S(0)e^{(r-\sigma^2/2)t+\sigma W(t)}, \tag{6.8}$$

where we replaced the drift μ by the riskless interest rate r to obtain the risk-neutral measure in (6.7). Hence, to simulate the path $\mathbf{S} = (S(t_1), \ldots, S(t_d))$ of the asset is suffices to simulate the discrete path $\mathbf{W} := (W(t_1), \ldots, W(t_d))$ of the Brownian motion.

Using the relation (6.8), we can define

$$g(\mathbf{W}) := V(\mathbf{S}, T).$$

Let $\mathbf{y} = (y_1, \ldots, y_d)$ denote a vector of d independent standard normally distributed random numbers. Then, the vector $\mathbf{W}$ can be generated by

$$\mathbf{W} = \mathbf{A}\mathbf{y}$$

using an arbitrary matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ which satisfies $\mathbf{A}^T\mathbf{A} = \mathbf{C}$, where $\mathbf{C}$ denotes the covariance matrix of the Brownian motion, see Section 5.1. Different choices of the matrix $\mathbf{A}$ thereby correspond to different path constructions such as random walk (RW), Brownian bridge (BB), principal component (PCA) or linear transformation (LT).

This way, we see that the expected value (6.7) depends on d independent standard normally distributed random numbers y_k. It can thus be written as d-dimensional integral

$$V(S, 0) = e^{-rT} \int_{\mathbb{R}^d} g(\mathbf{A}\mathbf{y})\varphi_d(\mathbf{y})d\mathbf{y} \tag{6.9}$$

over $\mathbb{R}^d$ where φ_d is the density of the standard normal distribution. Next we will compute the value of the integral (6.9) and its effective dimension using functions g which correspond to the payoff functions of Asian and barrier options.

6.2.1 Asian options

The pricing of Asian options is a commonly used test problem from finance, see, e.g., [159–161].

Modeling

The payoff of an Asian option with discrete *geometric average* is defined by

$$V(\mathbf{S}, T) = \max\left\{\prod_{k=1}^{d} S(t_k)^{1/d} - K,\, 0\right\}, \tag{6.10}$$

where K denotes the strike price of the option. In case of an discrete *arithmetic average* the payoff is

$$V(\mathbf{S}, T) = \max\left\{\frac{1}{d}\sum_{k=1}^{d} S(t_k) - K,\, 0\right\}. \tag{6.11}$$

While there is no closed form solution for Asian options with arithmetic average, a generalised Black-Scholes formula exists for Asian options with geometric average, which we use to validate our numerical methods and results.

Lemma 6.2 (Closed-form solution). *The price of an Asian option with discrete geometric average (6.10) is given by*

$$V(S,0) = S(0)\,\gamma\,\Phi(\beta + \sigma\sqrt{T_1}) - Ke^{-rT}\Phi(\beta),$$

where

$$\begin{aligned}
\gamma &:= e^{-r(T-T_2)-\sigma^2(T_2-T_1)/2},\\
\beta &:= \frac{\ln(S(0)/K) + (r - \frac{1}{2}\sigma^2)T_2}{\sigma\sqrt{T_1}},\\
T_1 &:= T - \frac{d(d-1)(4d+1)}{6d^2}\Delta t,\\
T_2 &:= T - \frac{(d-1)}{2}\Delta t.
\end{aligned}$$

Proof. The proof uses that the sum of normally distributed random variables is again normally distributed. It can be found in detail in [171]. □

Unless states otherwise, we consider an Asian option with $d = 16$ time steps and the parameters

$$S(0) = 100, \sigma = 0.2, r = 0.1, T = 1 \text{ and } K = 100,$$

which are also used in [159–161]. In the following, we compute the fair price $V(S,0)$ of this option via (6.9) using different quadrature methods and different path generating methods.

To study the impact of the smoothness, we calculate the results additionally for the case $K = 0$. While we obtain a smooth integrand in the case $K = 0$, the integrand has discontinuous first derivatives in the case $K = 100$.

Effective Dimensions

We first determine the effective dimensions of the integral (6.9). The results are shown in Table 6.3 for different proportions $\alpha \in [0.9, 0.9999]$. We first note that the

Asian option pricing problem with $K = 0$ is of very low superposition dimension $d_s \leq 2$ independent of the employed path construction. In the case $K = 100$, d_s decreases if we switch from LT to PCA, to BB and RW.[8] The superposition dimension in the classical case with proportion $\alpha = 99\%$ is one if $K = 0$, see [159], and (about) two if $K = 100$, see [161].

(a) Case $K = 0$

	anchored-ANOVA								classical ANOVA			
	superposition dim.				truncation dimensions							
$1-\alpha$	RW	BB	PCA	LT	RW	BB	PCA	LT	RW	BB	PCA	LT
1e-1	1	1	1	1	9	2	1	1	9	2	1	1
1e-2	1	1	1	1	13	6	2	1	13	6	2	1
1e-3	2	2	1	1	15	14	3	1	15	14	3	1
1e-4	2	2	2	1	16	16	6	1	16	16	6	1

(b) Case[8] $K = 100$

	anchored-ANOVA								classical ANOVA			
	superposition dim.				truncation dimensions							
$1-\alpha$	RW	BB	PCA	LT	RW	BB	PCA	LT	RW	BB	PCA	LT
1e-1	7	2	1	1	10	3	1	1	10	2	1	1
1e-2	8	3	1	1	10	4	2	1	14	7	2	1
1e-3	8	4	2	1	10	7	3	1	15	14	3	1
1e-4	8	4	3	1	13	15	5	1	16	16	6	1

Table 6.3. *Effective dimensions of the Asian option pricing problem (6.9). The nominal dimension is* $d = 16$.

One can moreover see in Table 6.3 that the truncation dimensions in the classical case almost coincide with the truncation dimensions in the anchored case. For instance, for the case $K = 0$ with $\alpha = 0.999$ we obtain $d_t = 15, 14, 3, 1$ using RW, BB, PCA and LT, respectively, for the anchored case as well as for the classical one. The LT-construction achieves the optimal result $d_t = 1$ in both cases. One can show that this holds even for the extreme case $\alpha = 1$, i.e. this problem can be reduced by LT to one with only one nominal dimension.

Quadrature Error and Costs

We now focus on the case $K = 100$ (i.e. the case where the integrand has discontinuous first derivatives) and study the performance of the different numerical

[8] Note that our numerical computations for the anchored case with $K = 100$ and RW or BB might be inaccurate. For these particular problems accurate results are difficult to obtain since the truncation dimension is high (hence many terms have to be integrated) and the integrals are not smooth (hence their computation is expensive).

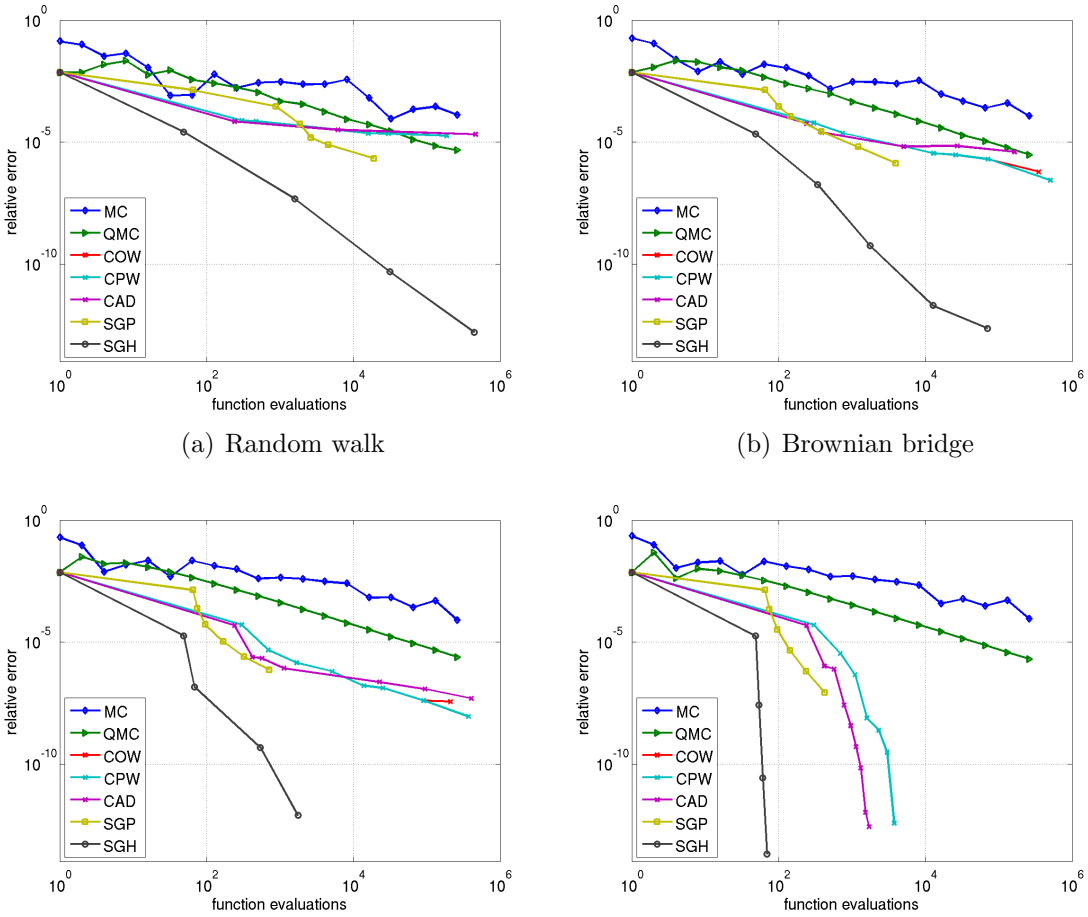
methods with respect of this difficulty. We again use MC, QMC and the sparse grid methods SGP and SPH. In our tests, we will here in addition consider the mixed CUHRE/QMC methods COW, CPW and CAD from Example 3.5 in Section 3.2. Remember that these dimension-wise quadrature methods are not of sparse grid form, but can resolve low regularity in the low-order anchored-ANOVA terms by local adaptivity.

Next we compute the value of the integral (6.9) using the different numerical approaches (MC, QMC, COW, CPW, CAD, SGP and SGH) and different path constructions (RW, BB, PCA, LT) and compare the results to the exact solution from Lemma 6.2. We display the respective convergence behaviour of these methods in Figure 6.5 and Figure 6.6. There, we show the number of function evaluations which is required by each of the different numerical methods to obtain a fixed accuracy.

One can see that the convergence rate of the MC method is again always about 0.5 as expected. QMC is superior to MC and achieves higher convergence rates of almost one, smaller relative errors and a less oscillatory convergence behaviour. This advantage increases if BB, PCA or LT is used. From Figure 6.5 and Figure 6.6 we also observe that the impact of the path construction is considerably bigger in case of the dimension-wise quadrature methods COW, CPW, CAD, SGP and SGH. This is explained by the fact that these methods are tailored to the effective dimension of the problem by the choice of the respective function space weights from Section 3.2.3, or by the dimension-adaptive grid refinement. The convergence of these methods is thus significantly accelerated by path constructions which reduce the effective dimension of the associated integrand. For instance, in the case $K = 0$, compare Figure 6.5, one can see that the performance of the dimension-wise quadrature methods significantly improves if we switch from RW to BB, to PCA and then to LT. While COW, CPW, CAD and SGP provide results which are similar to or even worse than (Q)MC in case of RW, they outperform (Q)MC slightly, clearly and drastically in case of BB, PCA and LT, respectively. To describe the convergence behaviour of the dimension-adaptive methods, again the preasymptotic regime and the asymptotic regime has to be distinguished, compare, e.g., Figure 6.5(d). In the preasymptotic regime, the methods COW, CPW, CAD, SGP and SGH first search for the important dimensions and interactions, whereas, in the asymptotic regime, the important dimensions are identified and the grid is then refined only in these directions. This leads to an extremely rapid convergence in the asymptotic regime, in particular in combination with the LT-construction, see Figure 6.5(d). This is explained by the fact that LT reduces the problem to only one dimension.

A comparison of the convergence rates of the COW, CPW and CAD method shows that the a priori constructions (with order-dependent weights or product weights) and the dimension-adaptive construction of the index set $\mathcal{S}$ lead to very similar results. The results of COW and CPW even coincide in most cases.

For the case $K = 0$, SGH is the by far most efficient method independenty of the



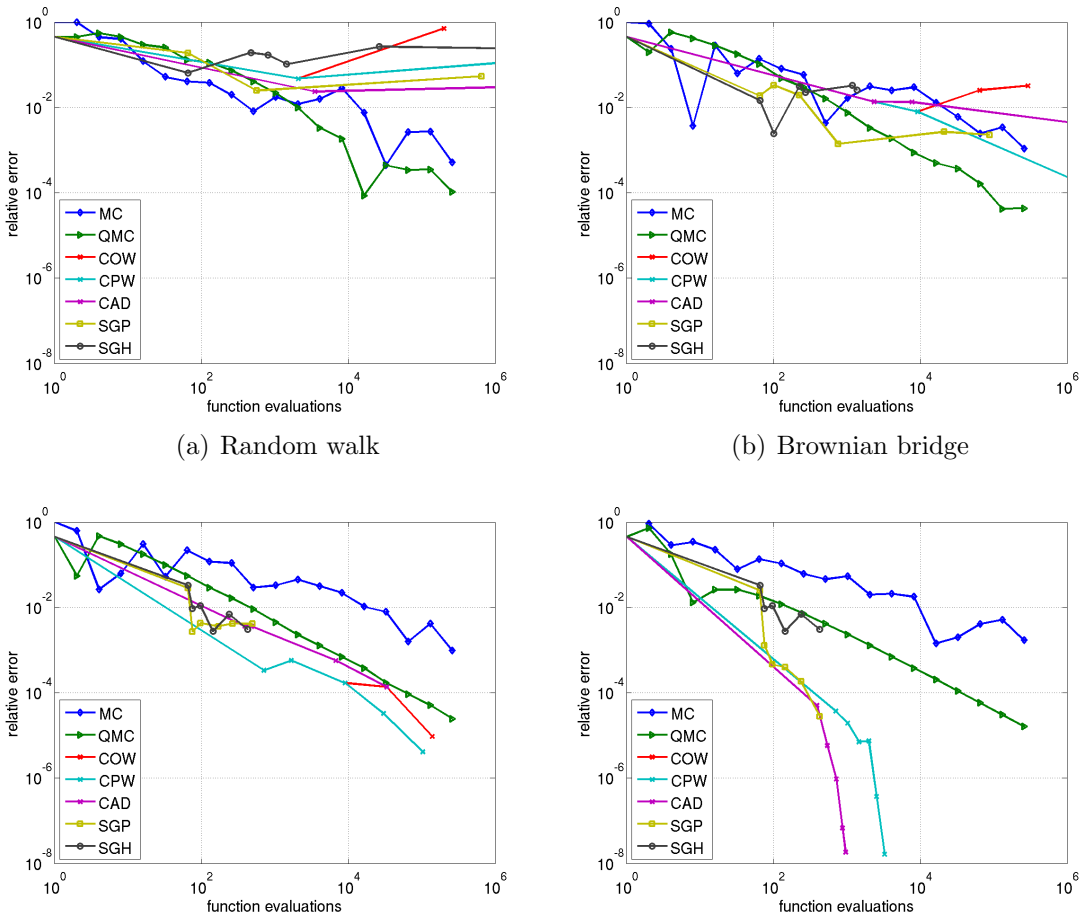
(a) Random walk (b) Brownian bridge

(c) Principal components (d) Linear transformation

Figure 6.5. *Convergence behaviour of the different methods for the Asian option pricing problem (6.9) with* $K = 0$ *and* $d = 16$.

employed path construction. It exploits the low effective dimension by its dimension-adaptive grid refinement and can profit from the smoothness of the integrand much better than all other approaches since it avoids the singular transformation to the unit cube. This way, we obtain relative errors smaller than 10^{12} with only about 10^5, 10^4, 10^3 and 10^2 function evaluations in case of RW, BB, PCA and LT, respectively, which is $7 - 10$ orders of magnitude more precise than the results of QMC.

Comparing the two cases $K = 0$ and $K = 100$, we furthermore see that the convergence rates of the QMC method are only slightly affected by the kink in the integrand, whereas the SG methods clearly suffer from the low degree of regularity. This drawback is to some extent overcome by the COW, the CPW and the CAD method, which are in combination with PCA or LT the most efficient approaches for



(a) Random walk

(b) Brownian bridge

(c) Principal components

(d) Linear transformation

Figure 6.6. *Convergence behaviour of the different methods for the Asian option pricing problem (6.9) with* $K = 100$ *and* $d = 16$.

the case $K = 100$. These methods profit from the low effective dimension and can in addition deal with the low regularity of the integrand by local adaptivity in the low order anchored-ANOVA terms due to the CUHRE approach. With LT, these methods obtain relative errors smaller than 10^{-8} with only about 1,000 function evaluations, see Figure 6.6(d), which is about 100,000 times more accurate than the results of QMC.

6.2.2 Barrier options

This section is concerned with discretely monitored barrier options.[9] Such options can be priced by the integration of high-dimensional functions that have kinks and jumps. To regain smoothness we here use the approach from Section 5.2.3 to avoid the kinks and jumps through conditional sampling, see also [59].

Modeling

Barrier options are path-dependent options which become worthless if the underlying asset reaches a specified barrier. There are various types of barrier options. Here, we consider discretely monitored knock-out barrier options.[10] The payoff of such an option is zero if the underlying asset $S(t)$ crosses a lower barrier $B > 0$ at one of the points in time $t_k = k\,\Delta t$, $k = 1, \dots, d$, with $\Delta t = T/d$. If the barrier is not crossed until the maturity date T, the holder obtains the payoff of a European call option with strike price K. In summary, the payoff is given by

$$V(\mathbf{S}, T) = (S(T) - K)^+ \, \chi(\min_{k=1,\dots,d} S(t_k) > B), \tag{6.12}$$

where we used the notation $(x)^+ := \max\{x, 0\}$ and where $\chi(A)$ denotes the characteristic function which is one if A is true and zero else.

By the martingale approach (6.7) this option can be priced by computing the expected value

$$V(S, 0) = e^{-rT} \mathbb{E}[V(\mathbf{S}, T)]$$

via the integral representation (6.9). The maximum operator and the characteristic function in (6.12) lead to a kink and a jump in the corresponding integrand, however, which makes accurate solutions difficult to achieve by numerical methods.

We here overcome this obstacle by deriving the one-step survival probabilities that the underlying asset does not reach the barrier in the time interval $[t_k, t_{k+1}]$ given its value at time t_k. This way we can exploit special structure of the problem and avoid to integrate the zero area of the integrand, see Section 5.2.3. We obtain the following lemma which is originally due to Jan Baldeaux from the University of New South Wales.

Lemma 6.3 (Smoothing of Barrier Options). *The price of the barrier option (6.12) is given by*

$$V(S, 0) = e^{-rT} \mathbb{E}[\tilde{V}(\mathbf{S}, T)],$$

[9] The results of this section are joint work with Jan Baldeaux from the University of New South Wales, Australia

[10] Knock-in or double barrier options can be treated in a similar way. For continuously monitored barrier options there exist closed form pricing formulas, see [171].

where

$$\tilde{V}(\mathbf{S},T) := \left(S(t_{d-1})e^{r\Delta t}\Phi(d_1+\sigma\sqrt{\Delta t}) + K\Phi(d_1)\right) \tag{6.13}$$

$$\cdot\prod_{k=1}^{d-1}\left(1-\Phi\left(\frac{\log\left(B/S(t_{k-1})\right)-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}\right)\right). \tag{6.14}$$

Here Φ *denotes the cumulative normal distribution function and*

$$d_1 := \frac{\log\left(S(t_{d-1})/M\right)+(r+\sigma^2/2)\Delta t}{\sigma\sqrt{\Delta t}}$$

with $M := \max(K,B)$.

Proof. We proceed incrementally and avoid in each time interval $[t_k, t_{k+1}]$ a crossing of the barrier and hence the jump in the integrand using conditional expectations.

To this end, let $\mathbb{P}[S(t_k) > B|S(t_{k-1})]$ denote the one-step survival probability that the asset is above the barrier B at time t_k conditional on its value at t_{k-1}. Then, by definition of M and by independence of the increments of the underlying Brownian motion, we obtain that

$$\begin{aligned}
\mathbb{E}[V(\mathbf{S},T)] &= \mathbb{E}\left[(S(T)-K)^+\chi(\min_{k=1,\dots,d}S(t_k)>B)\right] \\
&= \mathbb{E}\left[(S(T)-M)^+\chi(\min_{k=1,\dots,d-1}S(t_k)>B)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(S(T)-M)^+|S(t_{d-1})\right]\prod_{k=1}^{d-1}\mathbb{P}[S(t_k)>B|S(t_{k-1})]\right].
\end{aligned} \tag{6.15}$$

By (6.8), the one-step survival probabilities can be derived analytically. A simple computation yields

$$\begin{aligned}
\mathbb{P}[S(t_k)>B|S(t_{k-1})] &= \mathbb{P}\left[S(t_{k-1})\,\exp\{(r-\sigma^2/2)\,\Delta t+\sigma\sqrt{\Delta t}z_k\}>B|S(t_{k-1})\right] \\
&= \mathbb{P}\left[z_k > \frac{\log\left(B/S(t_{k-1})\right)-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}\,|S(t_{k-1})\right] \\
&= 1-\Phi\left(\frac{\log\left(B/S(t_{k-1})\right)-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}\right),
\end{aligned}$$

where z_k denotes a standard normally distributed random number.

The value of the remaining integral in (6.15) is given by

$$\mathbb{E}\left[(S(T)-M)^+|S(t_{d-1})\right] = S(t_{d-1})e^{r\Delta t}\Phi(d_1+\sigma\sqrt{\Delta t})+K\Phi(d_1),$$

Table 6.4. *Contribution (in percentage) of the first $j = 1, 2, 3$ dimensions to the total variance $\sigma^2(f)$ of the Barrier option pricing problem (6.7).*

(a) Standard payoff (6.12)

d	$\sigma^2(f)$	D_1	D_2	D_3	d_t
4	268.8	92	95	99	4
8	268.7	88	92	95	7
16	267.6	85	89	93	14
32	266.4	82	86	91	27

(b) Smoothed payoff (6.13)

d	$\sigma^2(f)$	D_1	D_2	D_3	d_t
4	110.2	71	82	82	4
8	149.7	84	88	88	8
16	182.2	87	91	93	16
32	207.1	88	92	95	32

which directly follows from the Black-Scholes formula for European call options. This way, also the kink in the integrand can be avoided and the dimension is reduced by one. □

We next compute the fair price $V(S, 0)$ of a Barrier option with the parameters

$$S(0) = 100, \sigma = 0.2, r = 0.05, T = 1, K = 95 \text{ and } B = 92$$

and compare the results which we obtain by integration of the payoff (6.12) with the results which we obtain by integration of the smoothed version (6.13).

Note that the simulation of the smoothed version (6.13) with BB, PCA or LT is not straightforward, but possible using the method of Leobacher [104]. Furthermore, it is difficult to address (6.13) directly on $\mathbb{R}^d$, since we used the inverse transformation method to generate conditional samplings. We therefore omit the method SGH in the following numerical experiments.

Effective Dimensions

To study the importance of the dimensions of the pricing problem of barrier options, we consider the classical case and use the values T_j from (2.10) to compute

$$D_j := (1 - T_j)\, 100.$$

The values D_j describe the contribution (in percentage) of the first j dimensions to the total variance of the problem. The results are shown in Table 6.4 for the case that the BB-construction is used. To illustrate the impact of the dimension, we here display the results for $d = 4, 8, 16, 32$.

We observe that in case of the standard payoff (6.12), $91 - 99\%$ of the total variance is concentrated in the first three dimensions. This proportion, however, decreases with increasing dimension. In case of the smoothed payoff (6.13), we see for $d = 4$ and $d = 8$ that the concentration of the variance is partly lost. This indicates that our approach to apply BB via the method [104] does not completely

avoid the incremental character of the conditional sampling approach. Interestingly, the contribution of the first dimensions to the total variance grows with increasing dimension.[11] Nevertheless, the truncation dimension d_t (with $\alpha = 0.99$) remains high, see Table 6.4.[12] It equals or almost equals to the nominal dimension d.

Further numerical computations not displayed here show that RW leads to less concentration of the variance in the first dimensions than BB with and without smoothing. PCA and LT lead to slightly better results than BB in the smoothed case but slightly worse in the standard case. The superposition dimension in the classical case is larger than two for the standard payoff while its two (at least for $d \leq 8$) for the smoothed one. This holds almost independently of the path construction. However, in all cases that we considered here the truncation dimension equals or almost equals to the nominal dimension. We will see below that this effect prevents the efficient pricing of barrier options by SG methods for high (roughly $d > 8$) dimensions and also deteriorates the performance of QMC methods.

We finally consider the impact of conditional sampling on the variance of the integrand. One can see in Table 6.4 that conditional sampling reduces the variance (by a factor of almost two for $d = 8$), but this advantage decreases with increasing dimension.

Quadrature Error and Costs

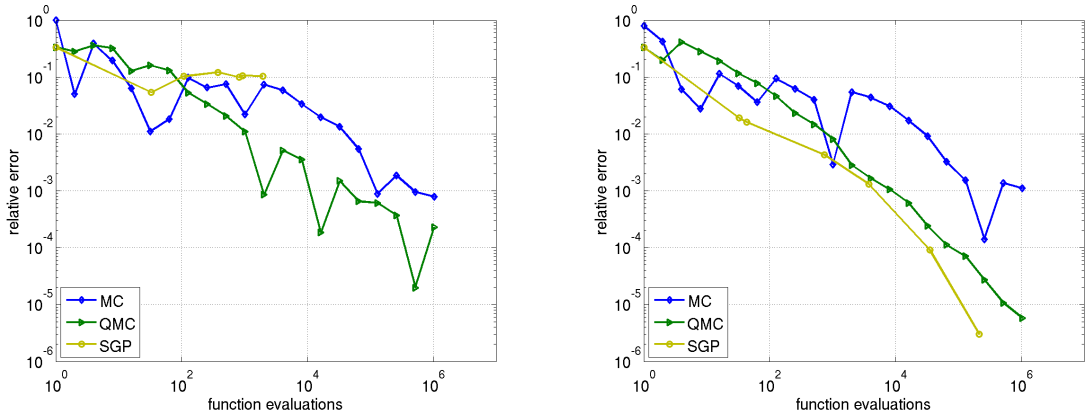
To compare the efficiency of the different numerical methods we computed the relative errors for different sample sizes n using the reference solution $V(S, 0) = 11.4899$. In Figure 6.7 we display the convergence behaviour of the numerical methods MC, QMC and SGP for the case $d = 8$ and the BB-construction.

The positive effect of the smoothing is clearly visible. The QMC method converges in the smoothed case faster with a rate of almost one and significantly less oscillatory than in the standard case. While the SG method shows no convergence at all in case of the standard payoff, it attains a high rate of convergence of almost two in case of the smoothed payoff and obtains in this way more precise results than MC and QMC also for small sample sizes n. Note that the smoothing implicates no approximation or model errors, but only serves to avoid the kink, the jump and the zero region of the integrand. We observe similar effects if the RW, PCA or LT-construction methods are used.

For smaller dimensions d, the advantage of the QMC method and in particular of the SG method compared to MC is even larger. The performance of QMC and SG deteriorates moderately and clearly, respectively, with increasing dimension, however, also in case of the smoothed payoff. This is explained by the high truncation

[11] One possible explanation for this effect is the concentration of measure phenomenon [102].

[12] Note that conditional sampling reduces the dimension by one. In combination with the method [102], which we require to apply BB, PCA or LT, this advantage is lost, though.



(a) Standard payoff (6.12) (b) Smoothed payoff (6.13)

Figure 6.7. *Convergence behaviour of the Barrier option pricing problem with* $d = 8$.

dimension of this application, which is neither reduced by BB nor by PCA or LT.

6.3 Performance-dependent Options

In this section, we consider the valuation of performance-dependent options. Performance-dependent option are financial derivatives whose payoff depends on the performance of one asset in comparison to a set of benchmark assets. Their valuation is an application from mathematical finance which requires the computation of high-dimensional integrals. Here, the large number of dimensions arises from a large number of state variables. As additional difficulty is that the integrands are typically discontinuous such that only MC methods can directly be applied without penalty. To overcome these two obstacles (high dimensionality and low regularity) we here use the principal component analysis to reduce the dimension and the domain decomposition from Section 5.2.2 to recover smoothness. For the integration of the smooth terms, we then apply SG methods.

The content of this section is already published in the journal article [56] and in the conference proceedings contributions [54, 57].

6.3.1 Framework and Pricing Formulas

In this section, we first motivate the relevance of performance-dependent options. Then, we formally define performance-dependent options and present closed-form formulas for their fair value in a multivariate Black-Scholes model. For details on the stochastic model and on the derivation of the pricing formulas, we refer to Appendix B.

Motivation

Companies make big efforts to bind their staff to them for longer periods of time in order to prevent a permanent change of executives in important positions. Besides high wages, such efforts are long-term incentive and *bonus schemes*. One widespread form of such schemes consists in giving the participants a conditional award of shares [143]. If the participant stays with the company for at least a prescribed time period he will receive a certain number of company shares at the end of the period. Typically, the exact amount of shares is determined by a *performance criterion* such as the company's gain over the period or its ranking among comparable firms (the peer group). This way, such bonus schemes induce uncertain future costs for the company.

For the corporate management and especially for the shareholders, the actual value of such bonus programs is quite interesting. One way to determine an upper bound of this value is to take the price of vanilla call options on the maximum number of possibly needed shares. This upper bound, however, often significantly overestimates the true value of the bonus program since its specific structure is not respected.

Contingent claim theory states that the accurate value of such bonus programs is given by the fair price of options which include the used performance criteria in their payoff. Such options are called performance-dependent options. Their payoff yields exactly the required shares at the end of the bonus scheme. This way, performance-dependent options minimize the amount of money the company would need to hedge the future payments arising from the bonus scheme, see [96].

Similar performance comparison criteria are currently used in various financial products, for example many hedge funds are employing so-called portable alpha strategies. Recently, also pure performance-based derivatives have entered the market in the form of so-called *alpha certificates*. Here, typically the relative performance of a basket of stocks is compared to the relative performance of a stock index. Such products are either used for risk diversification or for pure performance speculation purposes.

Payoff profile

We next formally define performance-dependent options and their payoff profiles. We assume that there are n companies involved in total. Our company gets assigned label 1 and the $n-1$ benchmark companies are labeled from 2 to n. The stock price of the i-th company varying with time t is denoted by $S_i(t), 1 \leq i \leq n$. The current time is denoted by $t = 0$. All stock prices at the end of the time period $t = T$ are collected in the vector $\mathbf{S} = (S_1(T), \ldots, S_n(T))$.

The character of a performance-dependent option is described by the payoff of the option at time T. To this end, we denote the relative price increase of stock i

over the time interval $[0, T]$ by

$$\Delta S_i := \frac{S_i(T)}{S_i(0)}.$$

We save the performance of the first company in comparison to a given strike price K and in comparison to the benchmark assets at time T in a ranking vector $\mathbf{Rank}(\mathbf{S}) \in \{+, -\}^n$ which is defined by

$$\mathrm{Rank}_1(\mathbf{S}) := \begin{cases} + & \text{if } S_1(T) \geq K, \\ - & \text{else} \end{cases} \quad \text{and} \quad \mathrm{Rank}_i(\mathbf{S}) := \begin{cases} + & \text{if } \Delta S_1 \geq \Delta S_i, \\ - & \text{else} \end{cases}$$

for $i = 2, \ldots, n$. This means, if the first asset outperforms benchmark asset i we denote this by a plus sign in the i-th component of the ranking vector $\mathbf{Rank}(\mathbf{S})$, otherwise, there is a minus sign. For the fair valuation of a bonus scheme, the strike K is typically equal to $S_1(0)$ since this way the payoff represents the risk of the price increase of the company's own stock until time T. In the following, arbitrary strike prices K are allowed, though.

In order to define the payoff of the performance-dependent option, we require bonus factors $a_{\mathbf{R}}$ which define the bonus for each possible ranking $\mathbf{R} \in \{+, -\}^n$. It is important to distinguish here between a possible ranking denoted $\mathbf{R}$ and the realized ranking induced by $\mathbf{S}$ which is denoted by $\mathbf{Rank}(\mathbf{S})$. The payoff of the performance-dependent option at time T is then defined by

$$V(\mathbf{S}, T) := a_{\mathbf{Rank}(\mathbf{S})} \max\{S_1(T) - K, 0\}.$$

We always define $a_{\mathbf{R}} = 0$ if $R_1 = -$, so that the payoff can be written as

$$V(\mathbf{S}, T) = a_{\mathbf{Rank}(\mathbf{S})} (S_1(T) - K). \tag{6.16}$$

In the following, we illustrated some possible choices for the bonus factors $a_{\mathbf{R}}$ which are included in our framework.

Example 6.4 (Performance-independent option).

$$a_{\mathbf{R}} = \begin{cases} 1 & \text{if } R_1 = + \\ 0 & else. \end{cases}$$

In this case, we recover a European call option on the stock S_1.

Example 6.5 (Linear ranking-dependent option).

$$a_{\mathbf{R}} = \begin{cases} m/(n-1) & \text{if } R_1 = + \\ 0 & else. \end{cases}$$

Here, m denotes the number of outperformed benchmark assets. The payoff only depends on the rank of our company in the benchmark. If the company ranks first, there is a full payoff $(S_1(T) - K)^+$. If it ranks last, the payoff is zero. In between, the payoff increases linearly with the number of outperformed benchmark assets.

Example 6.6 (Outperformance option).

$$a_{\mathbf{R}} = \begin{cases} 1 & \text{if } \mathbf{R} = (+, \ldots, +) \\ 0 & else. \end{cases}$$

A payoff only occurs if $S_1(T) \geq K$ and if all benchmark assets are outperformed.

Example 6.7 (Ranking-dependent option with outperformance condition).

$$a_{\mathbf{R}} = \begin{cases} m/(n-1) & \text{if } R_1 = + \text{ and } R_2 = + \\ 0 & else. \end{cases}$$

The bonus depends linearly on the number m of outperformed benchmark companies like in Example 6.5. However, the bonus is only payed if company two is outperformed. Company two could, e.g., be the main competitor of our company.

Pricing Formulas

To define a fair value for performance-dependent options, we use a multidimensional Black-Scholes model (see, e.g., [70, 91]) for the temporal development of all asset prices $S_i(t)$, $i = 1, \ldots, n$, required for the performance ranking.[13] It is based on $d \leq n$ many independent Brownian motions and described in the Appendix B.1. We thereby distinguish two cases: In the first case, which we refer to as *full model*, the number of stochastic processes d equals the number of assets n, whereas $d < n$ in the second case, which we refer to as *reduced model*.

Using the usual Black-Scholes assumptions, see, e.g., [91], the fair price $V(S_1, 0)$ of an option at time $t = 0$ is given by the discounted expectation

$$V(S_1, 0) = e^{-rT}\, \mathbb{E}[V(\mathbf{S}, T)] \tag{6.17}$$

of the payoff under the equivalent martingale measure, where r denotes the riskless interest rate. In the case of a performance-dependent option with payoff (6.16), we see that the expected value (6.17) can be written as a d-dimensional integral, since d independent Brownian motions are involved in the model. The integrand, however,

[13] A valuation approach for American-style performance-dependent options using a fairly general Lévy model for the underlying securities is presented in Egloff *et al.* [37]. There, a least-squares Monte Carlo scheme is used for the numerical solution of the model, but only one benchmark process is considered. Thus, the problem of high-dimensionality does not arise which is one of the main issues in this thesis.

is discontinuous induced by the jumps of the bonus factors $a_{\mathbf{R}}$, see the Examples 6.4-6.7. Therefore, numerical integration methods will perform poorly and only MC integration can be used without penalty. Thus, high accuracy solutions will be hard to obtain.

A closer inspection of the integral (6.17) shows, however, that the discontinuities are located on *hyperplanes*. Hence, we can apply the approach from Section 5.2.2 to decompose the integration domain first into polyhedrons and then each polyhedron into orthants. In this way, we can derive an analytical expression for the expected value (6.17) in terms of smooth functions, in our case multivariate normal distributions. The pricing formulas and their derivation can be found in the appendix. Theorem B.3 states the result for the full model case $d = n$ while Theorem B.4 summarizes the main result in the reduced model case $d < n$.

6.3.2 Numerical Results

In this section, we present numerical examples to illustrate the costs and the benefits which result from the domain decomposition into orthants as presented in Theorem 5.5. In particular, we compare the efficiency of the resulting pricing algorithm to the direct pricing approach

- QMC integration[14] of the expected payoff (6.17) using Sobol point sets (STD)

Observe that our pricing formulas in Theorem B.3 and Theorem B.4, which result from the orthant decomposition in Section 5.2.2, require the evaluation of several multivariate cumulative normal distributions. For their computation, we here systematically compare the use of

- QMC integration based on Sobol point sets (QMC),
- Product integration based on the Clenshaw Curtis rule (P),
- SG integration based on the Clenshaw Curtis rule (SG)

in combination with the Genz-transformation [44].

In the following we investigate four different choices of bonus factors $a_{\mathbf{R}}$ in the payoff function (6.16) according to the Examples 6.4 – 6.7. In all cases, we use the model parameters

$$K = 100, S_1(0) = 100, T = 1 \text{ and } r = 5\%.$$

In the performance-independent case of Example 6.4, an analytical solution is readily obtained by the Black-Scholes formula. In all other cases, we computed reference

[14] Monte Carlo integration led to significantly less accurate results in all our experiments.

Table 6.5. *Option prices $V(S_1, 0)$, discounts compared to the corresponding plain vanilla option and number of computed normal distributions (# Int) for the considered examples using a full Black-Scholes model with $n = 5$.*

Example	$V(S_1, 0)$	Discount	# Int
6.4	9.4499	-	1
6.5	6.2354	34.02%	30
6.6	3.0183	68.06%	2
6.7	4.5612	51.73%	16

values for the option prices on a very fine integration grid to compare the efficiency of the different pricing approaches.

To take into account the cost of the domain decomposition, we measure in this section the total cost of the different pricing algorithms not by the number of required function evaluations but show the overall computing times. To this end, note that all computations were performed on a dual Intel(R) Xeon(TM) CPU 3.06GHz processor.

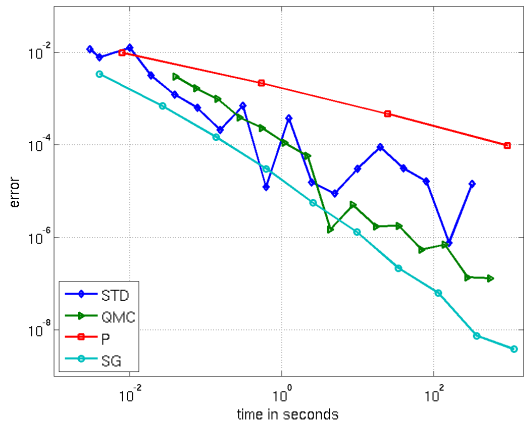
Full Model

We consider the multivariate Black-Scholes model from Section B.1 with $n = 5$ assets, $d = 5$ processes and use the volatility matrix

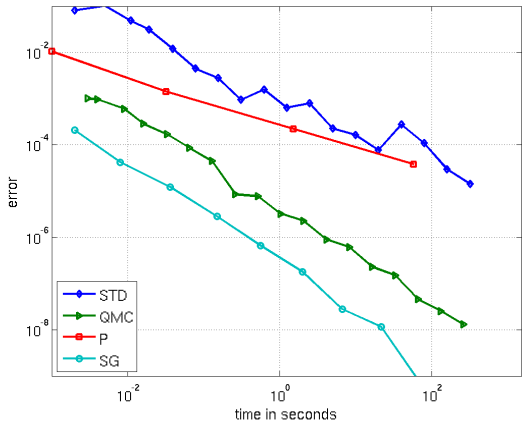
$$\boldsymbol{\sigma} = \begin{pmatrix} 0.1515 & 0.0581 & 0.0373 & 0.0389 & 0.0278 \\ 0.0581 & 0.2079 & 0.0376 & 0.0454 & 0.0393 \\ 0.0373 & 0.0376 & 0.1637 & 0.0597 & 0.0635 \\ 0.0389 & 0.0454 & 0.0597 & 0.1929 & 0.0540 \\ 0.0278 & 0.0393 & 0.0635 & 0.0540 & 0.2007 \end{pmatrix}.$$

The prices of the different performance-dependent options under these model assumptions are displayed in the second column of Table 6.5. In principle, all bonus schemes, which correspond to the Examples 6.5 – 6.7, could be hedged by the plain vanilla option in Example 6.4. The differences of the prices of the performance-dependent options (yielding the accurate value) and the corresponding plain vanilla options are shown in the third column of Table 6.5. We see that the usage of plain vanilla options substantially (up to 68 %) overestimates the fair values of the bonus schemes. The number of d-dimensional normal distributions (# Int) which have to be computed is shown in the last column. It depends on the special choice of the bonus factors as we explain in Appendix B.2.

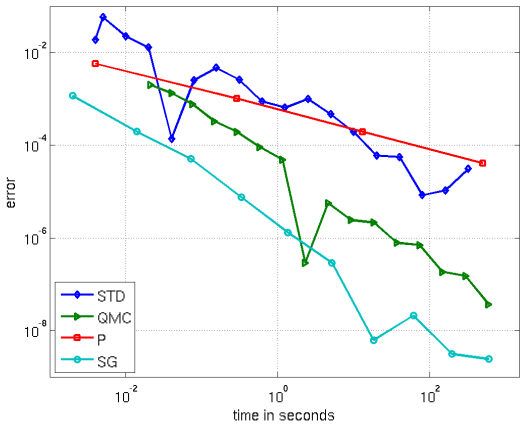
The convergence behaviour of the four different approaches (STD, QMC, P, SG) to price the performance-dependent options from the Examples 6.5 – 6.7 are displayed in Figure 6.8. There, the time is displayed which is needed to obtain a



(a) Example 6.5



(b) Example 6.6



(c) Example 6.7

Figure 6.8. *Errors and timings of the different numerical approaches to price different performance-dependent options using a full Black-Scholes model with* $n = d = 5$.

given accuracy. One can see that the standard approach (STD) and the product integration approach (P) perform worst for all accuracies. The convergence rates are clearly lower than one in all Examples. The integration scheme STD suffers under the irregularity of the integrand which is highly discontinuous and not of bounded variation. The product integration approach suffers under the curse of dimension. The use of the pricing formula from Theorem B.3 combined with QMC or SG integration clearly outperforms the STD approach in terms of efficiency in all considered examples. The QMC scheme exhibits a convergence rate of about one independent of the problem. The combination of SG integration with our pricing

formula (SG) leads to the best overall accuracies and convergence rates in all cases. Even very high accuracy demands can be fulfilled in less than a few seconds.

Reduced Model

In this section, we illustrate the performance of our approach in case of a reduced Black-Scholes market with $n = 30$ assets and $d = 5$ processes. This setting corresponds, e.g., to the case of a performance-dependent option which includes the performance of all companies of the German stock index DAX in its payoff profile. As we can see from the principal component analysis in Figure B.1, this number of processes suffice to explain more than 95 percentage of the total variance. Throughout this section we use $\boldsymbol{\sigma}$ being a 30×5 volatility matrix whose entries are uniformly distributed in $[-1/d, 1/d]$.

The prices of the performance-dependent options from the Examples 6.4 – 6.7 under these model assumptions are displayed in the second column of Table 6.6. In

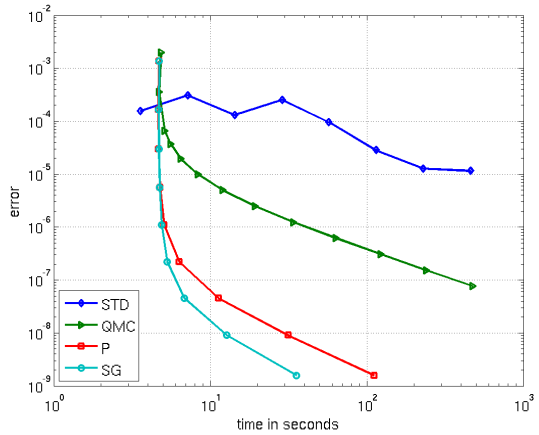
Table 6.6. *Option prices $V(S_1, 0)$, discounts compared to the corresponding plain vanilla option, number of computed normal distributions (# Int), intrinsic dimensions (Dim), and convergence rates of the different numerical approaches for the considered examples using a reduced Black-Scholes model with $n = 30$ and $d = 5$.*

Example	$V(S_1, 0)$	Discount	♯ Int	Dim	STD	QMC	P	SG
6.4	14.4995	-	1	1	1.1	-	-	-
6.5	12.9115	10.95%	41	2	0.58	0.88	1.45	1.55
6.6	1.8774	87.05%	31	5	0.6	1.1	0.27	1.87
6.7	8.6024	40.67%	38	3	0.52	1.3	0.89	1.54

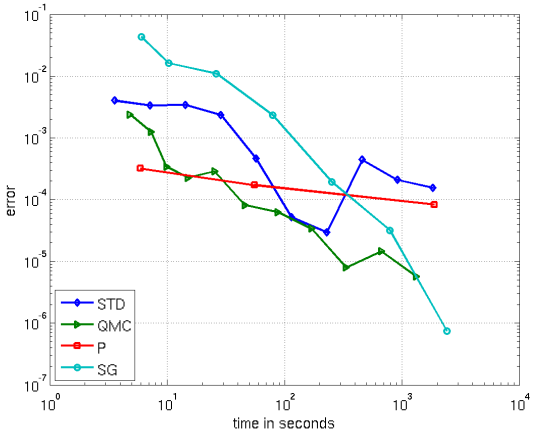
the third column we show the discounts which result from a comparison of the prices of the performance-dependent options (yielding the accurate value of the bonus schemes) and the corresponding plain vanilla options (yielding an upper bound to the value of the bonus schemes). One can see that the usage of plain vanilla options substantially (up to 87 %) overestimates the fair values of the bonus schemes. As we will explain in more detail in Appendix B.2, the complexity and dimensionality of our formula is often substantially reduced depending on the choice of the bonus factors. The number (♯ Int) and the maximum dimension (Dim) of normal distributions which have to be computed in the Examples 6.4 – 6.7 are displayed in the fourth and fifth column of Table 6.6. One can see that the number of required normal distributions is substantially lower than the theoretical bound (5.11) which is $c_{n,d} = 174,437$ for these examples. The maximum dimension varies from one to the nominal dimension five depending on the specific example.

The convergence behaviour of the four different approaches STD, QMC, P and SG to price the performance-dependent options from the Examples 6.5 – 6.7 are

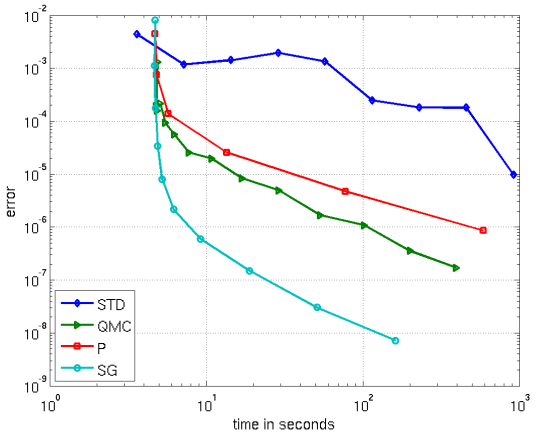
displayed in Figure 6.9. There, the time is displayed which is needed to obtain a



(a) Example 6.5



(b) Example 6.6



(c) Example 6.7

Figure 6.9. *Errors and timings of the different numerical approaches to price different performance-dependent options using a reduced Black-Scholes model with* $n = 30$ *and* $d = 5$.

given accuracy.

In the special case of Example 6.4, the application of Theorem B.4 combined with the transformation of Genz [44] automatically reduces to the analytical solution given by the Black-Scholes formula with the variance $\bar{\sigma}_1$ which is defined in Appendix B.1. The exact solution up to machine precision is obtained in about 4.7 seconds by all integration schemes (QMC, P, SG). This is the time which is needed in the domain decomposition step of our algorithm to compute all vertices $\mathbf{v}$ and all weights $c_{\mathbf{v}}$. In the same time, the STD approach approximates the solution up to an error of $1e{-}03$.

One can see that a simulation of the expected payoff (STD) performs similarly in all examples. Low accuracies are quickly achieved, the convergence rate is slow, though. The rate is about 0.6 in all examples and thus lower than one, as may be expected. The integration scheme suffers under the irregularity of the integrand which is highly discontinuous and not of bounded variation. The QMC scheme clearly outperforms the STD approach in all examples. It exhibits a convergence rate of about one and leads to much smaller errors after the setup time of 4.7 seconds. In contrast to the two previous approaches, the product integration approach (P) exhibits a high dependency on the specific example. While it performs very well in the Examples 6.4 and 6.5 it only converges with a rate of 0.27 in Example 6.6. Here, the curse of dimension, under which the product approach suffers, is clearly visible. While the intrinsic dimensions of Examples 6.4 and 6.5 are only one and two, respectively, the intrinsic dimension of Example 6.6 is five and, thus, equal to the nominal dimension. The combination of SG integration with our pricing formula (SG) leads to the best convergence rates. The curse of dimension can be broken to some extent, while the favorable accuracy of the product approach is maintained. It is the most efficient scheme for the Examples 6.4, 6.5 and 6.7. However, for higher dimensional problems as Example 6.6, this advantage is only visible if very accurate solutions are required. In the preasymptotic regime, the QMC scheme leads to smaller errors.

6.4 Asset-Liability Management in Life Insurance

In this section, we consider the asset-liability management of a portfolio of participating life insurance policies.[15] This is an application from mathematical finance leading to high-dimension integrals where the large number of dimensions mainly arises from small time steps in the time discretizations. We here in particular focus on dimension reduction using the different approaches from Section 5.1 to construct stochastic sample paths.

The remainder of this section is as follows: First, we describe the ALM model. Then, we formulate the problem to compute target figures of the model in terms of high-dimensional integrals. Finally, we present numerical results which illustrate the efficiency of different numerical approaches to compute these integrals using different constructions for the sample paths of the capital market.

The ALM model presented in this chapter is already published in the journal article [52]. The conference proceedings contribution [50] covers the part of this chapter which is concerned with the effective dimension of ALM problems in life insurance. The numerical aspects are partly published in the book contribution [53].

[15]Most of this section is the result of the research project “numerical simulation for asset-liability management in insurance” (03GRNHBN). The project was realized in cooperation with the Zurich Group Germany and was supported by the German Federal Ministry of Education and Research (BMBF) in the framework of its program “Mathematics for innovations in industry and service”.

6.4.1 Model and Integral Representation

In this section, we describe the application problem to compute target figures of stochastic asset-liability management models (ALM) in life insurance. Because of their high complexity, we here only sketch the most important aspects of such models and refer for details to Appendix C.1.

Stochastic ALM models are becoming more and more important for insurance companies due to new accountancy standards, greater globalisation, stronger competition, more volatile capital markets and longer periods of low interest rates. Additional importance arises from the current need of insurance companies to move from an accounting based on book values to a market-based, fair value accountancy standard as required by Solvency II and the International Financial Reporting Standard (IFRS), see, e.g., [88]. This task can be achieved by performing stochastic simulations of ALM models in a risk-neutral environment.

In the following, we start with an abstract representation of ALM models in life insurance in terms of a general state space model. This representation reveals the different building blocks from a computational point of view and is used in the remainder of this article.

Overall Model Structure

Much effort has been spent on the development of stochastic ALM models in the last years, see, e.g., [2,4,5,8,15,24,29,60,66,67,84,93,110,112,113,155,169] and the references therein.

As in these references, we here focus on the situation where a stochastic capital market model is used, while all other model components are assumed to be deterministic. We model all terms in discrete time.[16] The simulation starts at time $t = 0$ and ends at time $t = T$. The time interval $[0, T]$ is decomposed into K periods $[t_{k-1}, t_k]$ with $t_k = k\,\Delta t$ and period length $\Delta t = T/K$. Thereby, a (Markov) multi-period model specifies how the different accounts evolve from one point in time to the next. Then, the overall structure of one time step of such ALM models can often be organized into different modules as illustrated in Figure 6.10. Similar model structures are also used in [33, 52, 84].

The stochastic component, the capital market model (or scenario generator), is usually defined by a system of stochastic differential equations for the involved market components (e.g., stocks and interest rates). It is usually based on an underlying multivariate Brownian motion. The deterministic part of the ALM model includes all model equations specified in the asset model, the management model and the liability model. In the asset-model, the market prices of the different asset classes, the return rate of the company's portfolio and the overall development of

[16] Our starting point is thus either a discrete-time model or the discretization of a continuous-time model.

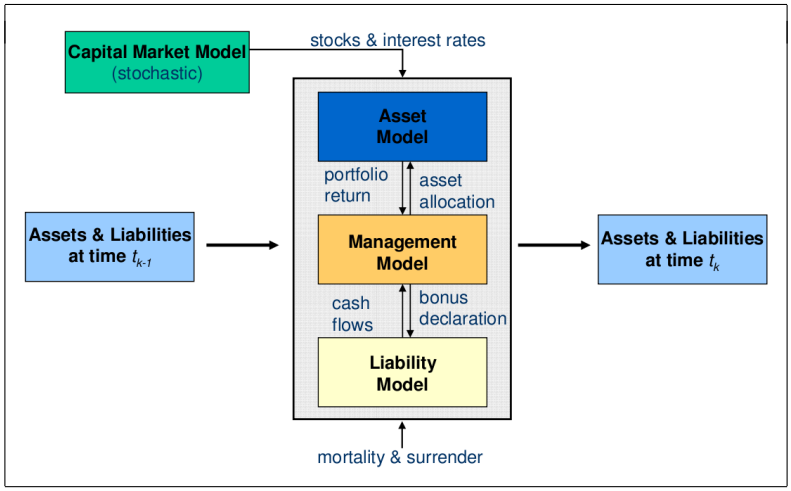


Figure 6.10. *Overall structure of one time step of the ALM model.*

the capital are determined. In the liability model, the premium payments and all survival, death and surrender benefits are collected, which depend on the specific insurance product. In addition, the balance sheet items on the liability side (e.g., the actuarial reserve, the bonus reserve or the equity) are updated. In the management model, the capital allocation, the bonus declaration and the shareholder participation are specified by deterministic management rules which can depend on the capital markets, the cash flows, the reserves and all other accounts.

State Space Representation

To obtain a convenient and compact representation of stochastic ALM models, we assume that the entire state of the insurance company at time t_k can be represented by a *state vector* $\mathbf{X}_k \in \mathbb{R}^M$. This vector contains all information, e.g., balance sheet items and policyholder accounts, which are required to carry the simulation from one point in time to the next, thereby depending on the development of the capital markets. We further assume that the state of the capital markets at time t_k can be described by a vector $\mathbf{S}_k \in \mathbb{R}^D$. It contains, for instance, the current stock prices and interest rates at time t_k.

The initial state of the insurance company at the start of the simulation is given and denoted by $\mathbf{X}_0$. Its temporal development is then specified by the equation

$$\mathbf{X}_k \quad = \quad r(\mathbf{X}_{k-1}, \mathbf{S}_k) \tag{6.18}$$

in a recursive way for $k = 1, \ldots, K$, which mirrors the Markov property of the ALM model. For a given input vector $\mathbf{S}_k$, the state equation $r : \mathbb{R}^{M+D} \to \mathbb{R}^M$ thereby relates the state of the insurance company at time t_k to the state at time t_{k-1}. It includes all model equations specified in the asset model, in the management model

and in the liability model. Most models proposed in the literature or used in practise can be written in the form (6.18) using a sufficiently complex function r.

The computation of one scenario of the model (6.18) then involves the computation of the vector

$$\mathbf{X} \;=\; (\mathbf{X}_1, \ldots, \mathbf{X}_K)$$

for the states of the insurance company at the different points in time. The states thereby depend on the vector

$$\mathbf{S} \;=\; (\mathbf{S}_1, \ldots, \mathbf{S}_K)$$

which describes the underlying capital market scenario.

Benchmark Model

As an example for the abstract model (6.18), we here consider the general ALM model framework from [52]. It describes the temporal development of a portfolio of participating life insurance policies. The items of the balance sheet at time t_k which are simulated in this model are shown in Table 6.7. The asset side consists of the market value C_k of the company's assets at time t_k. On the liability side, the first item is the book value of the actuarial reserve D_k.[17] The second item is the allocated bonus B_k which constitutes the part of the surpluses that have been credited to the policyholders via the profit participation. The free reserve F_k is a buffer account for future bonus payments. It consists of surpluses which have not yet been credited to the individual policyholder accounts, and is used to smooth capital market oscillations in order to achieve a stable and low-volatile return participation of the policyholders. The last item, the equity Q_k, consists of the part of the surpluses which is kept by the shareholders of the company. The model parameters and model equations which are used to describe the temporal development of the different balance sheet items are summarized in Figure 6.11 and are briefly explained in the following. For more details we refer to Appendix C.1.

Table 6.7. *Simplified balance sheet at time t_k.*

Assets		Liabilities	
Capital	C_k	Actuarial reserve	D_k
		Allocated bonus	B_k
		Free reserve	F_k
		Equity	Q_k

[17] i.e., the guaranteed savings part of the policyholders after deduction of risk premiums and administrative costs.

The capital market model contains the price of a stock and the short interest rate. The temporal dynamics of the stock price is modelled by a geometric Brownian motion while the short interest rates are obtained from the Cox-Ingersoll-Ross (CIR) model which is coupled to the stock price model via a constant correlation factor ρ. This system, which is based on a two-dimensional Brownian motion, is then discretized according to the period length Δt with an explicit Euler-Maruyama discretization yielding discrete stock prices s_k and short interest rates r_k for each period k, see the equations (C1) and (C2) in Figure 6.11.

In the asset model, the market prices $b_k(\tau)$ of the bonds of the company at time t_k are determined, see equation (A1), which in turn depend on their duration τ and on the short interest rates r_k. This way, the portfolio return rate p_k is specified (A2) which contributes to the development of the capital C_k of the company in each period (A3).

In the management model, the capital allocation, the bonus declaration and the shareholder participation are specified. The asset allocation in stocks and bonds is dynamic. Thereby, the proportion of funds invested in stocks is linked to the current amount of reserves of the company, see (M2). This implements a CPPI (constant proportion portfolio insurance) capital allocation strategy. The remaining part of the capital is invested in zero coupon bonds using a buy-and-hold trading strategy, see (M1) and (M3). For the declaration of the policyholder interest z_k, the mechanism from [66] is used (M4), which is based on the reserve situation of the company. To distribute the total surplus G_k in each period k among policyholders and shareholders, a fixed percentage α of the surplus is saved in the free reserve F_k while the remaining part is added to the equity Q_k, see (M5).

In the liability model, the actuarial reserve D_k^i and the allocated bonuses B_k^i for each policyholder i, $i = 1, \ldots, m$, are updated, see (L2) and (L3). They depend on the biometry assumptions and on the specific insurance products under consideration. Mortality and surrender are thereby assumed to be deterministic. The probabilities q_k^i and u_k^i that the policyholder i dies or surrenders, respectively, in period k are taken from experience-based tables and determine the number δ_k^i of contracts in the portfolio (L1). The surplus G_k in period k, see (L4), consists of the interest surplus, which results from the spread $p_k - z_k$ of portfolio and policyholder interest, and of the surrender surplus, which depends on the surrender factor ϑ. Finally, the equity Q_k is obtained (L5) so that the sum of the assets equals the sum of the liabilities.

Next, we formulate this particular model in terms of the state space representation (6.18). The state space $\mathbf{X}_k$ at time t_k of this model consists of all accounts of the insurance company and of the policyholders. We use

$$\mathbf{X}_k = \left(B_k^1, \ldots, B_k^m, D_k^1, \ldots, D_k^m, \delta_k^1, \ldots, \delta_k^m, n_k, \ldots, n_{k-\tau+1}, C_k, F_k\right)$$

and thus have $M = 3m + \tau + 2$, where m is the number of policyholders and τ the

Capital market model:

Input parameters: κ, θ, σ_r (short interest rates), μ, σ_s (stock prices)

(C1) Short interest rates $r_k = r_{k-1} + \kappa(\theta - r_{k-1})\Delta t + \sigma_r \sqrt{|r_{k-1}|}\, \Delta W_{k,1}$

(C2) Stock prices $s_k = s_{k-1} \exp\{(\mu - \sigma_s^2/2)\Delta t + \sigma_s\, \Delta W_{k,2}\}$

Management model:

Input parameters: β (asset allocation), ω, γ (bonus declaration), α (shareholders)

(M1) New investment $N_k = C_{k-1} + P_k - \sum_{j=1}^{\tau-1} n_{k-j}\, b_{k-1}(\tau - j)$

(M2) Investment in stocks $A_k = \max\{\min\{N_k,\, \beta F_{k-1}\},\, 0\}$

(M3) Number new bonds $n_k = (N_k - A_k)/b_{k-1}(\tau)$

(M4) Policyholder interest $z_k = \max\{z,\, \omega(F_{k-1}/(D_{k-1} + B_{k-1}) - \gamma)\}$

(M5) Free reserve $F_k = \max\{F_{k-1} + \min\{G_k, \alpha\, G_k\},\, 0\}$

Asset model:

Input parameters: τ (bond duration)

(A1) Bond prices $b_k(\tau) = A(\tau)\, \exp\{-B(\tau)\, r_k\}$

(A2) Portfolio return rate $p_k = (\Delta A_k + \sum_{j=0}^{\tau-1} n_{k-j}\, \Delta b_{k,j})/(C_{k-1} + P_k)$

(A3) Capital $C_k = (1 + p_k)(C_{k-1} + P_k) - E_k - T_k - S_k$

Liability model:

Input parameters: q_k^i, u_k^i (mortality, surrender), $z, \vartheta, P_k^i, E_k^{i,G}, T_k^{i,G}, S_k^{i,G}$ (product)

(L1) Number of contracts $\delta_k^i = \left(1 - q_k^i - u_k^i\right) \delta_{k-1}^i$

(L2) Actuarial reserve $D_k^i = ((1 + z)(D_{k-1}^i + P_k^i) - q_k^i T_k^{i,G})/(1 - q_k^i) - E_k^{i,G}$

(L3) Allocated bonus $B_k^i = (1 + z_k)\, B_{k-1}^i + (z_k - z)\, (D_{k-1}^i + P_k^i)$

(L4) Surplus $G_k = p_k F_{k-1} + (p_k - z_k)\, (D_{k-1} + B_{k-1} + P_k) + (1/\vartheta - 1) S_k$

(L5) Equity $Q_k = C_k - D_k - B_k - F_k$

Figure 6.11. Summary of the most important model parameters and equations. Lower indices k refer to the point in time t_k whereas upper indices i refer to the i-th policyholder. The values $\Delta W_{k,\ell} = W_\ell(t_k) - W_\ell(t_{k-1})$ for $\ell \in \{1, 2\}$, $\Delta A_k = A_k(s_k/s_{k-1} - 1)$ and $\Delta b_{k,j} = b_k(\tau - j - 1) - b_{k-1}(\tau - j)$ denote the increments of the underlying Brownian motion, the changes of the stock investments and the changes of the bond prices from the beginning to the end of period k, respectively. The terms $A(\tau)$ and $B(\tau)$ are constants which are defined in the CIR model. The values P_k, E_k, T_k and S_k denote the total premium income and the total survival, death, and surrender payments in period k. Like D_k and B_k, these values result by summation of the individual policyholder accounts.

duration of the bonds. From the state space $\mathbf{X}$ all remaining variables in the model can be derived. Since any term of the model is recursively defined, see Figure 6.11, a state equation of the form $\mathbf{X}_k = r(\mathbf{X}_{k-1}, \mathbf{S}_k)$ can be formulated. It includes all model equations from Figure 6.11 except Equations C1 and C2. The state space of the capital market model is two-dimensional and given by

$$\mathbf{S}_k = (s_k, r_k).$$

Integral Representation

Due to the large variety of path-dependencies, guarantees and option-like features of insurance products and management rules in ALM models, closed-form representations of statistical target figures, such as expected values, are in general not available. Therefore, numerical methods have to be used for their simulation. In this section, we rewrite the problem to simulate ALM models of type (6.18) as an high-dimensional integration problem and apply deterministic quadrature methods for its numerical computation.

A single simulation run $\mathbf{X}$ of the ALM model (6.18) corresponds to a particular capital market scenario $\mathbf{S}$. It can be analysed by looking, e.g., at the balance sheet positions or at cross sections of the portfolio at certain times. Here, we focus on stochastic simulations of the ALM model (6.18). To this end, a large number of scenarios is generated and statistical performance figures such as expected values are considered and evaluated. These measures are based on the most important state variables, e.g. the equity or the investment return, and can be written in the form

$$P = \mathbb{E}[f_P(\mathbf{X})] \tag{6.19}$$

for some evaluation function $f_P : \mathbb{R}^{M \cdot K} \to \mathbb{R}$. A simple example for such a function f_P evaluates the equity from the state vector $\mathbf{X}_K$ at the time t_K.

We next discuss the numerical computation of the performance figure P. Thereby, we assume that the capital market scenarios $\mathbf{S}$ result from the discretization and simulation of a system of stochastic Itô differential equations which is based on a D-dimensional Brownian motion. Since K time steps are used in the discretization this results in an $(D \cdot K)$-dimensional problem represented by

$$\begin{aligned} \mathbf{S} &= f_C(\mathbf{W}), \\ \mathbf{W} &= f_B(\mathbf{Y}), \end{aligned} \tag{6.20}$$

where $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_K)$ denotes the discrete path which contains the values of the Brownian motion at times $t_1, \ldots, t_K$ and where $\mathbf{Y}$ denotes a vector which contains $D \cdot K$ independent standard normally distributed random numbers. The functions $f_C : \mathbb{R}^{D \cdot K} \to \mathbb{R}^{D \cdot K}$ and $f_B : \mathbb{R}^{D \cdot K} \to \mathbb{R}^{D \cdot K}$ represent the generation of the capital market scenarios $\mathbf{S}$ and the path construction of the underlying Brownian

Algorithm 6.1: Computation of the expected value (6.19) by (Q)MC methods.

for $i = 1, 2, \ldots, n$ **do**

1)	Generate normally distributed random numbers	$\mathbf{Y}^i \in \mathbb{R}^{D \cdot K}$
2)	Construct path of Brownian motion	$\mathbf{W}^i = f_B(\mathbf{Y}^i)$
3)	Generate capital market scenario	$\mathbf{S}^i = f_C(\mathbf{W}^i)$
4)	Evaluate ALM model equations	$\mathbf{X}^i = f_I(\mathbf{S}^i)$
5)	Compute performance figure	$P^i = f_P(\mathbf{X}^i)$

Compute the average $P \approx \frac{1}{n}\left(P^1 + \ldots + P^n\right)$.

motion $\mathbf{W}$, respectively. For the specific ALM model from Section 6.4.1 the equation $\mathbf{S} = f_C(\mathbf{W})$ is explicitly given by Equations C1 and C2 in Figure 6.11. Possible choices for the function f_B are

$$f_B(\mathbf{Y}) = \mathbf{A} \cdot \mathbf{Y}.$$

The matrix $\mathbf{A}$ then describes the path construction. It can be arbitrary as long as $\mathbf{A}^T\mathbf{A} = \mathbf{C}$, where $\mathbf{C}$ denotes the covariance matrix of the D-dimensional Brownian motion. Specific choices are discussed in Section 5.1.

This way the performance figure P can be computed by numerical quadrature methods. The main steps of a standard (Q)MC algorithm (see Section 3.1) using n scenarios for the approximation of the expected value (6.19) are summarised in Algorithm 6.1. Thereby, the function $f_I : \mathbb{R}^{D \cdot K} \to \mathbb{R}^{M \cdot K}$ denotes the explicit (i.e. non-recursive) representation of the recursion (6.18) and thus contains all equations of the ALM model.

Since the distribution of the vector $\mathbf{X} \in \mathbb{R}^{M \cdot K}$, which contains the states of the insurance company, depends on the normally distributed vector $\mathbf{Y} \in \mathbb{R}^d$ with $d = D \cdot K$, see (6.18) and (6.20), the performance figure (6.19) can be represented as a d-dimensional integral

$$P = \int_{\mathbb{R}^d} h(\mathbf{y})\, \varphi_d(\mathbf{y})\; d\mathbf{y} = \int_{[0,1]^d} f(\mathbf{x})\; d\mathbf{x}. \tag{6.21}$$

Here, the function $h : \mathbb{R}^d \to \mathbb{R}$ is explicitly given by $h = f_P \circ f_I \circ f_C \circ f_B$, see also Algorithm 6.1, and $\varphi_d(\mathbf{y}) = (2\pi)^{-d/2} e^{-\mathbf{y}^T\mathbf{y}/2}$ denotes the Gaussian density function. The usual transformation with the cumulative normal distribution function Φ yields the integrand $f(\mathbf{x}) = h(\Phi^{-1}(x_1), \ldots, \Phi^{-1}(x_d))$.

We see that, for instance, the computation of a performance figure in an ALM model with a two-factor capital market and a monthly discretization for a time horizon of $T = 10$ years corresponds to $D = 2$ and $K = 120$ and thus results in a 240-dimensional integral.

6.4.2 Numerical Results

We now apply MC simulation, QMC integration based on Sobol point sets and dimension-adaptive SG integration based on the Gauss-Hermite rule (SGH) to compute of the integral (6.21). The dimension-adaptive SG method SGP turned out to be less efficient than SGH in most of our experiments. We here thus only show the results for SGH and refer to this method by SG.

We in particular investigate the impact of the different path generating methods on the dimension reduction of the integral and on the performance of the numerical methods. Further numerical experiments using different parameter setups can be found in the Appendix C.2. The impact of the parameters on the outcome of the model is studied in detail in [52].

Setting

In our numerical experiments we consider the following sample portfolio.

Example 6.8 (Sample Portfolio). We consider a model portfolio with $50,000$ contracts which is represented by $m = 500$ equal-sized model points. The data of each model point $i = 1, \dots, m$, is generated according to the distribution assumptions that entry and exit age are normally distributed with mean 36 and 62 and variance 10 and 4, respectively. In addition, the side conditions are respected that entry and exit age are in intervals $[15, 55]$ and $[55, 70]$, respectively. The current age at the start of the simulation is uniformly distributed between entry and exit age. The probability that the contracts of a model point belong to female policyholders is assumed to be 55%. The policyholders of model point i pay a constant monthly premium P^i which we generated uniformly distributed in the interval $[50, 500]$. For simplicity, we assume that the policies have not received any bonus payments before the start of the simulation, i.e., $B_0^i = 0$ for all $i = 1, \dots, m$. Furthermore, we assume that the policyholders die according to the probabilities q_k^i which we take from the DAV 2004R mortality table. Surrender occurs according to the exponentially distributed probabilities $u_k^i = 1 - e^{-0.03\Delta t}$ which are taken from [80].

Furthermore, we assume that all policyholders hold the following sample insurance product.

Example 6.9 (Sample Product). We consider an endowment insurance with death benefit and constant premium payments. All policyholders receive at maturity their guaranteed benefit D_k^i and the value of their bonus account B_k^i provided they are still alive at this point in time. For the computation of D_k^i, the guaranteed interest $z = 3\%$ is used. In case of death prior to maturity, the sum of all premium payments and the value of the bonus account is returned. In case of surrender, the

policyholder receives $\vartheta(D_k^i + B_k^i)$ with $\vartheta = 0.9$.

We use the following parameters to specify the management model. The participation ratio for the bonus declaration is chosen as $\omega = 25\%$ corresponding to the neutral scenario in [66]. The target reserve rate is assumed to be $\gamma = 15\%$. The capital allocation follows the CPPI strategy with $\beta = 1$. The bonds are traded with duration $\tau = 1/12$ years.

The parameters for the short rate and the stock prices are $\kappa = 0.1, \theta = 4\%, \sigma_r = 5\%$ and $\mu = 5\%, \sigma_s = 10\%$. They are taken from [38] and [93], respectively, where they have been estimated based on historical data for the German market. The correlation is set to $\rho = -0.1$.

We assume that the two accounts F_k and Q_k are merged into one single account, also denoted by Q_k, which is appropriate if the policyholders are also the owners of the company, see [66], and which corresponds to the case $\alpha = 1$. It is finally assumed that the total initial reserves of the company are given by $Q_0 = \gamma_0 D_0$ with $\gamma_0 = 0.1$. We always choose a period length of $\Delta t = 1/12$ years, but consider different numbers of periods in the simulation.

Effective Dimensions

We now demonstrate that the path generating method f_B in (6.20) has a significant influence on the effective dimension and on the performance of the deterministic numerical methods.

We compare the component-wise application of the random walk (RW), the Brownian bridge (BB), and the principal component construction (PCA) and, in addition, the eigenvalue decomposition (EVD), see Section 5.1. The truncation dimensions d_t which result from these four different constructions of the short interest rates and stock prices are listed in Table 6.8 for different nominal dimensions d. Note that we here show the classical effective dimensions (2.8) based on the (classical) ANOVA decomposition.

Table 6.8. *Truncation dimensions d_t of the ALM problem and the convergence rates of the QMC and SG method using different path constructions.*

	Truncation dimension d_t				QMC convergence rates				SG convergence rates			
d	RW	BB	PCA	EVD	RW	BB	PCA	EVD	RW	BB	PCA	EVD
16	16	3	10	9	0.6	0.8	0.8	0.8	2.1	2.0	1.6	1.4
32	32	7	12	12	0.6	0.8	0.8	0.9	1.5	1.6	1.3	1.8
64	64	8	16	15	0.5	0.8	0.9	0.8	0.1	0.5	0.6	0.3
128	126	11	22	20	0.5	0.8	0.9	0.9	0.1	0.1	0.2	0.2

One can see that the BB, PCA and EVD path constructions all lead to effective

dimensions d_t which are much smaller than the nominal dimensions d and are only slightly increasing with increasing d. If instead the RW discretization is used then the effective dimension is nearly equal to the nominal dimension. PCA and EVD lead to almost the same results which is explained by the rather small correlation $\rho = -0.1$ between the two underlying Brownian motions. The lowest effective dimensions are achieved by the BB construction.

Further numerical computations show that the our ALM model problem is also of very low effective dimension d_s in the superposition sense independently of the path construction. We show in Appendix C.2 that this property stays maintained if we vary input parameters and model components. We conjecture that this property might also hold for more complex ALM models in life insurance as the ones considered here.

A partial explanation for the low effective dimensions and a common feature of ALM problems is that the high nominal dimension mainly arises from the discretization of an underlying continuous time process. The corresponding integrals can thus be written as an approximation to some infinite-dimensional integrals with respect to the Wiener measure.[18] In these cases, the integrands are contained in some weighted function spaces whose weights are related to the eigenvalues of the covariance operator of the Wiener measure. The eigenvalues, sorted by their magnitude, are decaying proportional to j^{-2} (where j is the number of the eigenvalue) which induces strong tractability as shown in [148], see also Appendix A, and may explain the low effective truncation dimension.

We further show in Table 6.8 the impact of the path construction on the convergence rates of the QMC and SG method. The convergence behaviour of the MC method is not affected by the path construction since the total variance of the problems remains unchanged and is thus not displayed. One can see that the QMC method achieves higher convergence rates than MC if the paths are generated with BB, PCA or EVD. In these cases the rates are almost identical ranging from 0.75 to 0.92 and show almost no dependence on the nominal dimension d. If the random walk construction is used instead, then the convergence rates of the QMC method deteriorate with increasing d and no longer outperform the MC rate of 0.5 for $d \geq 64$. This effect illustrates the importance of the path generating method if QMC methods are applied to ALM simulations. It even more significantly affects other QMC points sets, like, e.g., the Halton sequence, whose points are in higher dimensions not as uniformly distributed as the Sobol points. One can finally see that the behaviour of the SG method is less clearly related to the truncation dimension d_t and to the path construction[19] but is rather affected by the nominal dimension d.

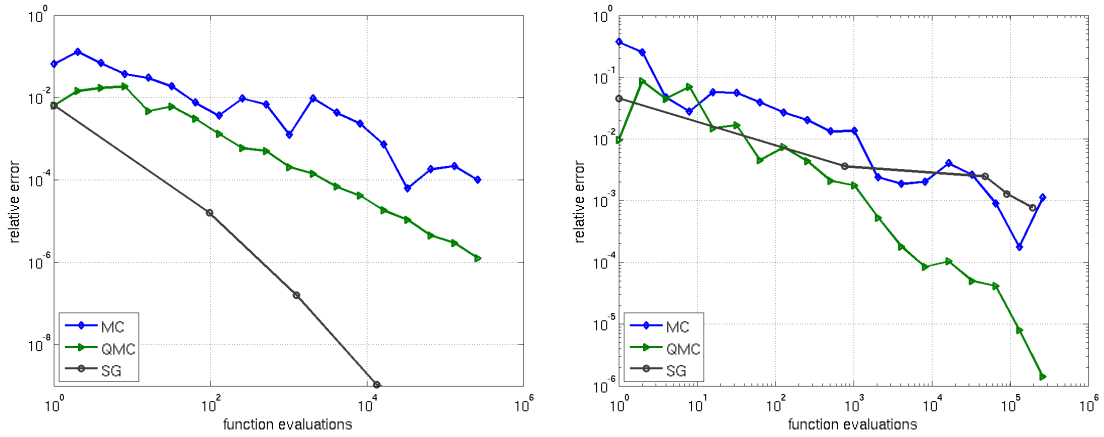
[18] See, e.g., [105] and the references listed there.

[19] The reason are two different interacting effects. The BB, PCA and EVD constructions lead to integrands of low truncation dimension but with kinks which are not parallel to the coordinate axes. In the RW construction the integrands are of high truncation dimension but some of the

While the SG method attains high convergence rates larger than one in the moderately high dimensional case $d \leq 32$, the rates deteriorate with increasing d and only a very slow or even no convergence is observed in the higher dimensional cases.

Quadrature Error and Costs

We finally display in Figure 6.12 the convergence behaviour of the MC, QMC and SG method for our benchmark model. There, the number of function evaluations is shown which is required by each of the three numerical methods to obtain a fixed accuracy. We here use the BB-path construction and consider the two cases $K = 16$ and $K = 128$ which correspond to integration problems with nominal dimensions $d = 32$ and $d = 256$ and effective dimensions $d_t = 7$ and $d_t = 15$, respectively.



(a) $d = 32$ (b) $d = 256$

Figure 6.12. *Convergence behaviour of the different numerical approaches to compute the expected value (6.19) for the benchmark ALM model introduced in Section 6.4.1.*

One can see that the QMC method clearly outperforms MC simulation with a convergence rate close to one and independent of the dimension. The convergence of the QMC method is also less erratic than the convergence of the MC method. If $1,000$ function evaluations are used, the QMC method is about ten times as accurate as the MC method with the same number of scenarios. For higher accuracy requirements, the advantage of the QMC method is even more pronounced. These results can not be explained by the classical QMC convergence theory but by the low effective dimension of the ALM problems, which we reported in Section C.2.

kinks are axis-parallel. SG profit from low truncation dimension but suffer from kinks which are not axis-parallel.

The SG method is the by far most efficient method for the moderate high-dimensional setup ($d = 32$). With $1,000$ function evaluations already an relative accuracy of 10^{-7} is achieved, which is thousand times as accurate than the result of the QMC method. The performance of the SG method deteriorates with rising dimensions, however, showing that the curse of dimension can not be completely avoided in this application. For very high dimensions ($d = 256$), the SG method is no longer competitive to QMC. This is explained by the fact that the arising integrands have discontinuous first derivatives. The points of low regularity result from the minimum operators, maximum operators or absolute values in the model equation (C1), (M2) and (M4), see Figure 6.11. To ensure a satisfactory performance of the SG method despite the low regularity, the smoothness must first be recovered by suitable smoothing transformations (e.g., by a smoothing of maximum and minimum operators or by a decomposition of the integration domain into domains where the integrand is smooth). These approaches are discussed in Section 5.2. Their application to ALM problems is interesting but not straightforward, and requires further investigations, which we leave for future work. Several additional numerical experiments which illustrate the impact of the regularity and of the dimension on the performance of the numerical methods can be found in the Appendix C.2.

6.5 Summary and Discussion

We conclude this chapter with a short summary and with recommendations. In this chapter, we applied different sparse grid methods to various application problems from finance and insurance and compared their performance to the performance of Monte Carlo and quasi-Monte Carlo integration.

Moreover, to gain insight into the importance of dimensions and to explain the convergence behaviour of the numerical methods, we computed the effective dimensions of our model problems in the classical case and in the anchored case. We saw that most of our problems are of very low effective dimension, or can be transformed to be of low effective dimension by coordinate transformations. Using the example of the CMO problem, we moreover observed that the behaviour of the dimension-adaptive methods, such as SGP and SGH, can not be explained by the effective dimension in the classical case, but by the effective dimension in the anchored one.

In summary, we showed in correspondence with the theoretical results from Chapter 3 and Chapter 4 that the efficient applicability of sparse grid methods strongly depends on the two key properties

- high degree of smoothness
- low effective dimension

of the integrand. Remember that the sparse grid approach is very general. It leaves the underlying univariate quadrature rule as well as the underlying index set

open. Note that different choices of these components may considerably affect the convergence behaviour of the sparse grid method. We give some comments and guidelines based on the numerical experiments in this chapter.

- To profit from smoothness it is important to chose an underlying univariate quadrature rule with a high polynomial degree of exactness such as a Gauss rule. Moreover, the singular transformation to the unit cube should be avoided, which is possible with help of the Gauss-Hermite or the Genz-Keister rule.

- For problems with higher nominal dimensions (roughly $d > 10$) it is essential to lift the dependence of the method on the dimension through a careful choice of the underlying index set. The lower the effective dimension, the smaller choices of the index set are possible without deteriorating the accuracy of the method. A general, automatic and efficient way to construct a suitable index set is given by the dimension-adaptive approach from Gerstner and Griebel [49].

- In higher dimensions it is important to start with a one-point quadrature rule on the lowest level (i.e. with $m_1 = 1$, see Section 4.1.1) of the sparse grid construction to reduce the increase of the number of grid points. It is of advantage if the univariate sequence of quadrature rules is nested but this property is not essential in our experience. The Gauss-Hermite sequence, which provided the best results in most of our numerical experiments, is not nested, for instance.

In summary, we recommend the use of the dimension-adaptive sparse grid method based on the slowly increasing Gauss-Hermite sequence (SGH) for integrals with Gaussian weight and integrands which are smooth and of low effective dimension. Our numerical results in Section 6.1 with interest derivatives clearly showed the superiority of this method to Monte Carlo, quasi-Monte Carlo and existing sparse grid methods even in hundreds of dimensions.

Most application problems do unfortunately not satisfy the two conditions low effective dimension and high degree of smoothness. While this situation has only a moderate and a small impact on QMC and MC, respectively, it strongly deteriorates the performance of sparse grid methods as we illustrated, e.g., in Section 6.2. Sparse grid integration is therefore not suited as a black box method for problems from finance and insurance.

Nevertheless, the results of this chapter demonstrate that this difficulty can be overcome in several cases by coordinate transformations and domain decompositions. In particular the LT-construction from Imai and Tan [82] turned out to be a very efficient coordinate transformation in combination with dimension-adaptive methods. We showed that domain decompositions are beneficial for the problems to price barrier options and performance-dependent options, compare Section 6.2 and

Section 6.3. This way, we could exploit the underlying structure of the problems and we believe that sparse grid methods can with the help of similar approaches also be successfully applied to many further problems. However, none of the approaches is applicable in a general and generic way to a wider class of problems. Low regularity therefore remains the main challenge for the efficient applicability of sparse grids methods in finance and insurance.

Here, the flexibility in the construction of the dimension-wise quadrature approach may help to design methods which can deal also with low regularity in an automatic way. As a first step into this direction we defined mixed CUHRE/QMC methods. We applied these methods in Section 6.2 to the problem to price an Asian option. In combination with PCA or LT, the methods performed most efficient in our tests and outperformed quasi-Monte Carlo by several orders of magnitude despite the low regularity of the integrand. We believe that further improvements of this approach are possible if modeling and discretization errors are balanced in a more sophisticated way and if instead of the CUHRE method a different locally adaptive method is employed which treats the integrals directly on $\mathbb{R}^d$ such that the singular transformation to the unit cube is avoided.

Chapter 7
Conclusions

In this thesis we dealt with the efficient numerical computation of high-dimensional integrals as they commonly arise in applications from finance and insurance.

Starting with the anchored-ANOVA decomposition, we introduced the new notion of effective dimension in the anchored case and showed that integration errors can be bounded by these dimensions. Moreover, we provided numerical experiments with applications from finance which indicate that the performance of sparse grid methods can be better explained with the help of our new notion of effective dimension than with the classical one.

Based on the anchored-ANOVA decomposition, we developed a novel general class of methods for the computation of high-dimensional integrals, which we referred to as dimension-wise quadrature methods. To construct our new methods, we first truncated the anchored-ANOVA decomposition and then integrated the remaining terms using appropriate low-dimensional quadrature rules. We discussed a priori and a posteriori approaches for the truncation and derived bounds for the resulting modeling and discretization errors. We showed that the presented error bounds also apply to sparse grid methods as they can be regarded as special cases of our general approach. Furthermore, we explained that sparse grid methods intertwine the truncation of the anchored-ANOVA series and the subsequent discretization. This way, modeling and discretization error can be balanced in an optimal way by the choice of the underlying index set. We determined the optimal index sets for integrands from weighted tensor products of Sobolev spaces and derived new cost and error bounds for the respective sparse grid methods.

We considered sparse grid methods in more detail and presented two new variants based on the delayed Genz-Keister and on the slowly increasing Gauss-Hermite sequence, respectively. Both methods can treat integrals with Gaussian weight directly on $\mathbb{R}^d$. This way, singular transformations to the unit cube can be avoided and the smoothness of the integrand is exploited in an optimal way. To further improve the performance of dimension-adaptive sparse grid methods, we combined these methods for the first time with the linear transformation method from Imai and Tan. We provided several numerical experiments with problems from finance which lead to smooth integrands. Our results show that the dimension-adaptive sparse grid method based on the slowly increasing Gauss-Hermite sequence is most efficient for these problems. It outperforms Monte Carlo, quasi-Monte Carlo and existing sparse grid methods by several orders of magnitude even in hundreds of dimensions.

We also discussed several approaches to overcome the difficulty that many integrands in finance are not smooth, which deteriorates the performance of sparse grid methods enormously. To this end, we presented dimension-wise quadrature methods which are not of sparse grid form, but use the CUHRE method for the integration of the low-order anchored-ANOVA terms and quasi-Monte Carlo methods for the higher-order ones. This way, we obtained mixed CUHRE/QMC methods which are to our knowledge the first numerical quadrature methods which can profit from low effective dimension by dimension-adaptivity and can at the same time deal with low regularity by local adaptivity. A correct balancing of modeling and discretization errors is then more difficult than with sparse grid methods. Numerical experiments with the Asian option as a test function from finance with discontinuous first derivatives demonstrate, however, that this disadvantage is more than compensated by the benefits of the local adaptivity. We moreover studied the approach to decompose the integration domain into subdomains where the integrand is smooth. This way, we successfully regained the fast performance of sparse grid methods for some special classes of problems from finance. We demonstrated the superiority of these approaches to standard methods for the pricing problems of barrier options and performance-dependent options.

Finally, we considered the simulation of stochastic asset-liability management (ALM) models in life insurance, which is one of the most complex and sophisticated problems in insurance. We here successfully applied, to our knowledge for the first time, quasi-Monte Carlo and sparse grid methods to ALM simulation problems. To this end, we first rewrote the ALM simulation problem as a multivariate integration problem, which we then solved with the help of our deterministic quadrature methods in combination with adaptivity and dimension reduction techniques. We provided numerical experiments based on a general ALM model framework, which includes the most important features of life insurance product management and most of the models previously proposed in the literature. Our results demonstrate that the deterministic methods often converge faster and less erratic than Monte Carlo integration even for complex models with many variables and model equations. Furthermore, we showed by an analysis of variance that ALM problems are often of very low effective dimension, or can be transformed to be of low effective dimension by coordinate transformations. This way, we also provided a theoretical explanation for the success of the deterministic quadrature methods.

Of course, there are issues which could not be dealt with in this thesis and which remain open. It would be interesting, for instance, to identify the function classes for which the effective dimension in the anchored case coincides with the effective dimension in the classical case. We indicated that the pricing problems of Asian options or zero coupon bonds belong to such a function class, but not the CMO problem. Note that we always used the center of the integration domain as anchor

point. Of course also other choices are possible and it would be of interest to analyse the impact of the anchor point in more detail.

Other possible areas for future research include improvements of our dimension-wise quadrature methods, e.g., by a more sophisticated balancing of modeling and discretization errors. For applications from finance, our mixed CUHRE/QMC methods can be further improved if instead of the CUHRE method a different locally adaptive method is employed which treats the integrals directly on $\mathbb{R}^d$ such that the singular transformation to the unit cube can be avoided. The local error estimator of such a method could, e.g., be based on Genz-Keister points [45]. Note further that our dimension-wise approach can not only be used for integration but also for the representation and approximation of high-dimensional functions in the sense of [140], which we also leave for future research.

With respect to sparse grid methods it is of particular importance that more general and generic approaches are developed to overcome the difficulty that many integrands in finance are not smooth. This issue remains the main challenge for the efficient applicability of sparse grids methods in finance and insurance. Moreover, it would be interesting to study the efficient computation of further quantities, which can not be formulated as expected values or integrals, but which are also important for financial institutions, such as quantiles. Quasi-Monte Carlo methods can compute also quantiles very efficiently as shown by Papageorgiou and Paskov [131] using the example of Value at Risk calculations. To our knowledge, it is not yet known if similar or even better results can be obtained with sparse grid methods.

We finally remark that most of our methods and results are not restricted to applications from finance and insurance, but can also be used in other application areas such as chemistry or physics.

Appendix A
Tractability of Integration

Tractability is a very active research area in information-based complexity which investigates the dependence of an algorithm on the dimension of a problem. In this chapter, we summarize known results on tractability of uniform integration in the worst case setting.[1] The results are mostly taken from the recent survey articles [125,126]. We start in Section A.1 by introducing reproducing kernel Hilbert spaces since for such spaces particularly many results can be derived. Then, the worst case error and the notion of tractability is defined. In Section A.2, we relate the worst case error in different function spaces to different discrepancy measures. In Section A.3, we report bounds on the discrepancy. The classical bounds depend on the number n of sample points and hold for a fixed dimension d, whereas the more recent estimates, which have been obtained to study tractability, hold for variable n and variable d. The discrepancy of a point set can be defined in many different ways. Similarly, there are many different notions of tractability which can be studied in many different function spaces. In the following, we mainly consider L_2-discrepancies. They are related to integration in the Sobolev spaces of functions which have bounded mixed derivatives in the L_2-norm.

A.1 Reproducing Kernel Hilbert Spaces

In this section, we introduce the basic definitions and tools which we are required in the subsequent sections to summarize results on tractability of multivariate integration.

Reproducing kernel Hilbert spaces

The theory of reproducing kernel Hilbert spaces (RKHS) was introduced in [3] and further investigated in [158]. It is widely used for the numerical analysis of algorithms since it allows to define function spaces in a particularly concise and elegant way by means of so-called reproducing kernel functions.

Let $K : [0,1]^d \times [0,1]^d \to \mathbb{R}$ be non-negative definite. Then, there exists a unique Hilbert space $H = H(K)$ with scalar product $(\cdot,\cdot)_H$ such that

- $K(\cdot, \mathbf{t}) \in H(K)$ for all $\mathbf{t} \in [0,1]^d$

[1] Worst case setting means that the performance of the algorithm is measured by the worst case error in a some function spaces. It is closely related to the average case setting which is the main focus in [142]. Results on weighted (e.g. Gaussian) integration can be found in, e.g., [27, 74, 124].

- $(f, K(\cdot, \mathbf{t}))_H = f(\mathbf{t})$ for all $\mathbf{t} \in [0,1]^d$ and all $f \in H(K)$.

The Hilbert space $H(K)$ is called the reproducing kernel Hilbert space to the kernel K. By Riesz's representation theorem also the opposite direction is true: Let H be a Hilbert space in which the point evaluation is a bounded linear functional, i.e., $f(\mathbf{t}) \leq C\|f\|_H$ for all $f \in H$ and a constant $C > 0$. Then, there exists a unique kernel K, such that H is its reproducing kernel Hilbert space. A Hilbert space is thus a reproducing kernel Hilbert space if and only if the point evaluation is a bounded linear functional in this space. Several properties of reproducing kernel Hilbert spaces are known, see, e.g., [158]. For instance, the tensor products of reproducing kernel Hilbert spaces are the reproducing kernel Hilbert spaces to the tensor product kernels, i.e,

$$H(K_1 \otimes K_2) = H(K_1) \otimes H(K_2)$$

for two dimensions. Many examples of reproducing kernel Hilbert spaces can be found in [3, 142, 144, 158]. Note that $L_2([0,1]^d)$ is not a reproducing kernel Hilbert space, since the point evaluation is not well-defined in this space.

Worst Case Errors

Let Q_n denote a quadrature rule in d dimensions of the form (3.3) which uses n points $\mathbf{x}^i$ and weights w_i. The worst case error of Q_n in a Hilbert space H with norm $\|\cdot\|_H$ is defined by its worst case performance over the unit ball in H, i.e.,

$$e(Q_n) := \sup_{f \in H,\, \|f\|_H \leq 1} |If - Q_n f|.$$

We will see that it is important to compare this quantity to the initial error

$$e(Q_0) = \sup_{f \in H,\, \|f\|_H \leq 1} |If|$$

where we formally set $Q_0 = 0$.

If we assume that $H = H(K)$ is a reproducing kernel Hilbert space with kernel K, then the worst case error can be represented as

$$e^2(Q_n) = \int_{[0,1]^d \times [0,1]^d} K(\mathbf{s}, \mathbf{t})\, d\mathbf{s} d\mathbf{t} - 2 \sum_{i=1}^{n} w_i \int_{[0,1]^d} K(\mathbf{s}, \mathbf{x}^i) d\mathbf{s} + \sum_{i,j=1}^{n} w_i w_j K(\mathbf{x}^i, \mathbf{x}^j) \quad \text{(A.1)}$$

see, e.g., [65]. The initial error is given by

$$e^2(Q_0) = \int_{[0,1]^d \times [0,1]^d} K(\mathbf{s}, \mathbf{t})\, d\mathbf{s} d\mathbf{t}.$$

It is known that worst case settings in reproducing kernel Hilbert spaces correspond to particular average case settings. Indeed, $e(Q_n)$ equals to the average case error with respect to the Gaussian measure μ which has mean zero and the covariance matrix K, i.e.,

$$e^2(Q_n) = \int_{C([0,1]^d)} (If - Q_n f)^2 \mu(df), \tag{A.2}$$

see [142] for details.

Moreover, by the Cauchy Schwarz inequality the general Koksma-Hlawka inequality

$$|If - Q_n f| \leq e(Q_n) \|f\|_H \tag{A.3}$$

holds for all $f \in H(K)$, see, e.g., [40]. Thereby, $e(Q_n)$ describes the quality of the quadrature rule and $\|f\|_H$ the regularity of the integrand. The inequality is sharp in the sense that for each Q_n there exist a function $f \in H(K)$, $f \neq 0$, such that equality holds.

Of particular interest is the problem to find points $\mathbf{x}^i$ and weights w_i such that the quadrature rule Q_n minimizes the worst case error $e(Q_n)$. The minimal worst case error is defined as

$$e(n,d) := \inf \{e(Q_n) : \ Q_n \text{ as in (3.3)}\}.$$

It is often compared to the initial error $e(0,d)$. To this end, let

$$n(\varepsilon, d) := \min \{n : \ e(n,d) \leq \varepsilon \, e(0,d)\}$$

denote the minimal number of points which are required by Q_n to reduce the initial error by a factor of ε. Bounds on $e(n,d)$ in terms of n and d and on $n(\varepsilon,d)$ in terms of ε and d are known for many reproducing kernel Hilbert space $H(K)$. We report known estimates for these quantities for particular spaces $H(K)$ in Section A.3.

It is known from complexity theory that the worst case error $e(Q_n)$ is minimized by linear algorithms which use non-adaptive integration points. Nevertheless, for the integration of a particular function f, a quadrature rule Q_n may perform much better as its worst case performance and adaption may lead to significantly better results than non-adaption.

Tractability

Based on the behaviour of $n(\varepsilon,d)$ the notion of tractability for the normalised error criterion $e(n,d)/e(0,d)$ is defined. It describes the dependence of an algorithm on the dimension of a problem. Integration for a Hilbert space H is called tractable in the class of quadrature rules (3.3) if there exist constants $C, p, q \geq 0$, such that

$$n(\varepsilon, d) \leq C \, d^q \varepsilon^{-p} \tag{A.4}$$

for all $d \in \mathbb{N}$ and $\varepsilon \in (0,1)$, i.e., if $n(\varepsilon, d)$ is polynomial in d and ε^{-1}. Integration is called strongly tractable if (A.4) holds with $q = 0$, i.e., if $n(\varepsilon, d)$ is independent of d and polynomial in ε^{-1}.[2] The smallest q and p for which (A.4) holds are called the d-exponent and the ε-exponent of tractability. To illustrate the notion of tractability, we consider some examples:

- The expected L_2-error of the MC method (3.4) with n samples is

  $$\varepsilon(n, d) = \sigma^2(f)\, n^{-1/2}$$

  where $\sigma^2(f)$ is the variance of f. This shows that integration is MC-tractable in the space H if and only if the variance $\sigma^2(f)$ is uniformly bounded for all d and all $f \in H$. Then, the ε-exponent of tractability is $p^\star = 2$.

- Let $H = C^r([0,1]^d)$ be the space of r-times continuously differentiable functions. Then, there exist constants $c_{r,d}$ and $C_{r,d} > 0$ such that

  $$c_{r,d}\, \varepsilon^{-d/r} \leq n(\varepsilon, d) \leq C_{r,d}\, \varepsilon^{-d/r},$$

  which shows that integration is intractable in this space. If we assume that $r = d$, then we get for fixed d that $n(\varepsilon, d) = O(\varepsilon^{-1})$, i.e., $n(\varepsilon, d)$ is polynomial in ε^{-1}. It is not known, though, if integration is tractable or intractable in this space since $C_{r,d}$ is exponentially large in d and $c_{r,d}$ is exponentially small in d.

- Let $H^{r,\dots,r}([0,1]^d)$ denote the Sobolev space of functions with r bounded mixed derivatives. It is known that integration is intractable in this space for all $r \geq 1$. In weighted Sobolev spaces $H^{r,\dots,r}_{\boldsymbol{\gamma}}([0,1]^d)$ integration is tractable if the function space weights $\boldsymbol{\gamma}$ decay sufficiently fast. We discuss this example in more detail in Section A.3.

A.2 Notions of Discrepancy

In this section, we relate the worst case error to the discrepancy of a point set. The notion of discrepancy is much older than the notion of tractability and goes back to the work of Weyl in 1916. It has been investigated extensively for the analysis of QMC methods and pseudo-random number generators. It appears in the classical Koksma–Hlawka inequality (see, e.g., [119]), which bounds the integration error of a

[2] Note that we defined tractability and strong tractability with respect to the normalized error $e(n,d)/e(0,d)$. Instead, one can use $n(\varepsilon, d) := \min\{n : e(n,d) \leq \varepsilon\}$. This leads to the notion of tractability with respect to the absolute error. We only defined polynomially tractability. There are further notions of tractability such as weak tractability or T-tractability. Polynomially tractability or T-tractability implies weak tractability. The lack of weak tractability is called intractability, see [126].

QMC method by the product of the discrepancy of the QMC sample points and of the variation of the integrand. The discrepancy describes the quality of the QMC points in the sense that it measures the irregularity of the distribution of the point set in the unit cube. Many different discrepancy measures (and corresponding notions of variation) are possible and have been studied in the literature, see, e.g., [76,125,126]. Some are based on geometric properties of the point set whereas others are based on the worst case errors for integration in certain function spaces.

Here, we start with a general notion of discrepancy which is taken from [40] and which includes many discrepancy measures proposed in the literature as special cases. The starting point of this definition is a function $B(\mathbf{x},\mathbf{t}) : [0,1]^d \times D \to \mathbb{R}$ where $D \subseteq [0,1]^{\bar{d}}$ and $\bar{d}$ is a positive integer.[3] We assume that $B(\mathbf{x}, \cdot\,) \in L_2(D)$ for all $\mathbf{x} \in [0,1]^d$ and that $\mathbf{x} \mapsto B(\mathbf{x}, \cdot\,)$ is a continuous mapping. The *local discrepancy* of the quadrature rule Q_n with respect to the function B is defined as

$$g(\mathbf{t}) = \sum_{i=1}^{n} w_i B(\mathbf{x}^i,\mathbf{t}) - \int_{[0,1]^d} B(\mathbf{x},\mathbf{t})d\mathbf{x}. \tag{A.5}$$

Using the local discrepancy, the *L_2-discrepancy* of the quadrature rule Q_n with respect to the function B is defined as

$$D(Q_n) = \left(\int_D |g(\mathbf{t})|^2 d\mathbf{t}\right)^{1/2}. \tag{A.6}$$

The function B induces a symmetric and non-negative definite kernel

$$K(\mathbf{x},\mathbf{y}) = \int_D B(\mathbf{x},\mathbf{t})B(\mathbf{y},\mathbf{t})d\mathbf{t} \tag{A.7}$$

and thus also a reproducing kernel Hilbert space $H(K)$, see Section A.1. If we represent the square of the discrepancy $D^2(Q_n)$ in terms of the kernel function, one obtains that it equals to the right hand side of (A.1) and therefore

$$D(Q_n) = e(Q_n). \tag{A.8}$$

The L_2-discrepancy of Q_n with function B can thus be interpreted as the worst case error of Q_n in the reproducing kernel Hilbert space with associated kernel (A.7) or, see (A.2), as the average case error of Q_n with respect to the Gaussian measure which has mean zero and the covariance matrix (A.7).

Next, we define the *weighted L_2-discrepancy* with respect to the function B. It incorporates ANOVA[4]-like projections of the quadrature points onto lower-dimensional

[3] In [40] the case $D = [0,1]^d$ is considered. We use the more general case $D \subseteq [0,1]^{\bar{d}}$ in order to include also unanchored discrepancies in the definition (A.6).

[4] The analysis of variance (ANOVA) and related dimension-wise decompositions of a function $f : [0,1]^d \to \mathbb{R}$ are the topic of Chapter 2.

faces of the unit cube. Weighted notions of discrepancy are used to analyse algorithms in reproducing kernel Hilbert spaces where dimensions and/or interaction are not of equal importance.

For a set $\mathbf{u} \subseteq \{1, \dots, d\}$ and a vector $\mathbf{t} \in [0,1]^d$ let $\mathbf{t}_\mathbf{u}$ denote the vector from $[0,1]^{|\mathbf{u}|}$ containing the components of $\mathbf{t}$ whose indices are in $\mathbf{u}$. Furthermore, let $d\mathbf{t}_\mathbf{u} = \prod_{j \in \mathbf{u}} dt_j$ and $(\mathbf{t}_\mathbf{u}, \mathbf{1})$ the vector from $[0,1]^d$ whose j-th component is t_j if $j \in \mathbf{u}$ and one otherwise. Let

$$\boldsymbol{\gamma} = \{\gamma_\mathbf{u} : \mathbf{u} \subseteq \{1, \dots, d\}\} \tag{A.9}$$

denote a given sequence of 2^d-many weights $\gamma_\mathbf{u} \in [0,1]$. This definition of general weights includes the following important special cases:

- *Equal weights:* $\gamma_\mathbf{u} = 1$ for all $\mathbf{u} \subseteq \{1, \dots, d\}$.
- *Product weights:* For a given sequence of weights $\gamma_j \geq 0$, $j = 1, \dots, d$, product weights are defined by
$$\gamma_\mathbf{u} = \prod_{j \in \mathbf{u}} \gamma_j.$$
- *Finite-order weights:* For some integer d_s independent of d, finite-order weights satisfy $\gamma_\mathbf{u} = 0$ if $|\mathbf{u}| > d_s$.

Based on the weights $\gamma_\mathbf{u}$ the weighted L_2-discrepancy of the quadrature rule Q_n with respect to the function B is defined as

$$D_{\boldsymbol{\gamma}}(Q_n) := \left(\sum_{\mathbf{u} \subseteq \{1,\dots,d\}} \gamma_\mathbf{u} \int_{[0,1]^{|\mathbf{u}|}} |g(\mathbf{t}_\mathbf{u}, \mathbf{1})|^2 d\mathbf{t}_\mathbf{u} \right)^{1/2} \tag{A.10}$$

with the local discrepancy g as in (A.5). For the specific choice of weights $\gamma_\mathbf{u} = 0$ if $|\mathbf{u}| < d$ and $\gamma_{\{1,\dots,d\}} = 1$, the weighted L_2-discrepancy (A.10) reduces to the L_2-discrepancy (A.6).

Next, we discuss specific choices of the function B, for which we obtain from (A.6) the *r-smooth L_2-discrepancy* [133] and the *B-discrepancy* [126]. Both discrepancy measures include the *L_2-star discrepancy* [119] as special case. For the same choices of B, we get from (A.10) the *generalised L_2-star discrepancy* [76] and the *weighted L_2-star discrepancy* [148]. Unless states otherwise, we set $D = [0,1]^d$ in the following.

L_2-star discrepancy

The L_2-star discrepancy (see, e.g., [119]) is a very popular discrepancy measure which has been investigated extensively for the analysis of QMC methods and

pseudo-random number generators. It is obtained from (A.6) if we choose the function B as

$$B(\mathbf{x}, \mathbf{t}) = \chi_{[\mathbf{0},\mathbf{t})}(\mathbf{x}), \tag{A.11}$$

i.e., as the characteristic function which is one if $\mathbf{x} \in [\mathbf{0}, \mathbf{t})$ and zero otherwise. Here, $[\mathbf{0}, \mathbf{t})$ denotes the box $\prod_{j=1}^{d}[0, t_j)$. From (A.5), we then get

$$g(\mathbf{t}) = \sum_{i=1}^{n} w_i \, \chi_{[\mathbf{0},\mathbf{t})}(\mathbf{x}^i) - t_1 \cdot \ldots \cdot t_d.$$

For a quadrature rule Q_n with equal weights $w_i = 1/n$, the sum on the right hand side equals to the number of points of Q_n which lie in the box $[\mathbf{0}, \mathbf{t})$ which shows that g is the classical local discrepancy of a point set. The L_2-star discrepancy then results from averaging over $g(\mathbf{t})$ according to (A.6). It is a special case of the B-discrepancy, of the r-smooth L_2-discrepancy and of the weighted L_2-star discrepancy which we discuss below.

From (A.7) and (A.11) we obtain that

$$K(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{d}(1 - \max\{x_j, y_j\}).$$

This kernel corresponds to the reproducing kernel Hilbert space

$$H(K) = H_0^{1,\ldots,1}([0,1]^d) = \bigotimes_{j=1}^{d} H_0^1([0,1])$$

where $H_0^1([0,1]) = \{f \in H^1([0,1]) : f(1) = 0\}$ is the Sobolev space of absolutely continuous functions whose first derivatives belong to $L_2([0,1])$ anchored at one. Because of (A.8), the L_2-star discrepancy can be interpreted as the worst case error $e(Q_n)$ in this particular tensor product space. From (A.1), one obtains by direct integration the explicit formula

$$\begin{aligned} D^2(Q_n) \;=\; & \frac{1}{3^d} - \frac{1}{2^{d-1}} \sum_{i=1}^{n} w_i \prod_{j=1}^{d}(1 - (x_j^i)^2) \\ & + \sum_{i,k=1}^{n} w_i w_k \prod_{j=1}^{d}(1 - \max\{x_j^i, x_j^k\}) \end{aligned} \tag{A.12}$$

for the L_2-star discrepancy, which can be evaluated in $O(dn^2)$ operations. For moderate d, faster algorithms are found in [72], which require $O(n(\log n)^d)$ operations. The initial L_2-star discrepancy is given by $D(Q_0) = 3^{-d/2}$ and is thus exponentially small in d. It corresponds to the initial error $e(Q_0)$ in the space $H(K)$.

If the L_2-norm in the definition (A.6) is replaced by the L_p-norm with $1 \leq p < \infty$, then (A.6) with B as in (A.11) corresponds to the L_p-star discrepancy. The L_p-star discrepancies are directly related to the worst case error of multivariate integration in the spaces of absolutely continuous functions whose first derivatives belong to $L_q[(0,1)]$ with $1/p + 1/q = 1$, see [119]. If the L_∞-norm is used in (A.6) then one obtains the L_∞-star discrepancy (also denoted by $\star$-discrepancy) which corresponds to the variation in the sense of Hardy and Krause and which has the initial discrepancy one.

B-discrepancy

The B-discrepancy as introduced in [126] is obtained from (A.6) if we choose

$$B(\mathbf{x}, \mathbf{t}) = \chi_{A(\mathbf{t})}(\mathbf{x}), \tag{A.13}$$

where $A(\mathbf{t}) \subseteq [0,1]^{\bar{d}}$ denotes a measurable set.[5] The L_2-star discrepancy is a special case of (A.13) which is obtained if we choose

$$A(\mathbf{t}) = \prod_{j=1}^{d} [0, t_j).$$

Note that the boxes $[\mathbf{0}, \mathbf{t})$, which are used in this definition, are anchored at the origin $(0, \ldots, 0)$. For

$$A(\mathbf{t}) = \prod_{j=1}^{d} [\min\{a_j, t_j\}, \max\{a_j, t_j\})$$

we get a generalized notion of discrepancy; the discrepancy anchored at the point $\mathbf{a} \in [0,1]^d$. If $A(\mathbf{t})$ denotes the box which has one corner at $\mathbf{t}$ and the opposite corner at the vertex of the unit cube which is closed to $\mathbf{t}$, then the centered discrepancy is obtained which was studied in [76]. Finally, one can averages over all boxes in the unit cube. For $\bar{d} = 2d$, $D = \{(\mathbf{x}, \mathbf{y}) \in [0,1]^{2d} : \mathbf{x} \leq \mathbf{y}\}$, $\mathbf{t} = (\mathbf{x}, \mathbf{y})$ and

$$A(\mathbf{t}) = \prod_{j=1}^{d} [x_j, y_j)$$

one gets the L_2-*unanchored discrepancy* (also called extreme discrepancy) which was introduced in [116]. Further choices of the sets $A(\mathbf{t})$ and D are possible and lead to

[5] In [126], the more general case $A(\mathbf{t}) \subseteq \mathbb{R}^{\bar{d}}$ is considered. Then also the *ball discrepancy* is included in the definition.

further discrepancy measures such as the *quadrant discrepancies*[6] anchored at the point $\mathbf{a} \in [0,1]^d$, see [126]. The B-discrepancies can of course also be studied in the L_p-norm with $1 \leq p \leq \infty$.

The kernel function and reproducing kernel Hilbert space which relate to the discrepancy anchored at the point $\mathbf{a}$ are given by

$$K(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{d} \frac{1}{2}(|x_j - a_j| + |y_j - a_j| - |x_j - y_j|)$$

and

$$H(K) = \bigotimes_{j=1}^{d} H^1_{a_j}([0,1]),$$

where $H^1_{a_j}([0,1]) = \{f \in H^1([0,1]) : f(1-a_j) = 0\}$ denotes the Sobolev space anchored at $1 - a_i$.

The kernel function and the reproducing kernel Hilbert space which correspond to the unanchored discrepancy are given by

$$K(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{d} (\min\{x_j, y_j\} - x_j y_j)$$

and

$$H(K) = \bigotimes_{j=1}^{d} F^1([0,1]),$$

where $F^1([0,1]) = \{f \in H^1([0,1]) : f(0) = f(1) = 0\}$ denotes the Sobolev space of periodic functions with boundary condition.

r-smooth L_2-discrepancy

The B-discrepancies are related to integration problems with classes of functions which have bounded mixed first derivatives. Such function spaces are widely used to compare the quality of QMC methods. For the analysis of SG methods, spaces of functions with *higher regularity* are more appropriate. To this end, we choose the function

$$B(\mathbf{x}, \mathbf{t}) = \frac{1}{((r-1)!)^d}(\mathbf{t} - \mathbf{x})_+^{r-1}, \tag{A.14}$$

with the parameter $r \geq 1$ and $(\mathbf{t}-\mathbf{x})_+^{r-1} = \prod_{i=1}^{d}(\max\{t_i - x_i,\, 0\})^r$. This choice of B together with (A.6) corresponds to the *r-smooth L_2-discrepancy* introduced in [133]. For $r = 1$, (A.14) simplifies to (A.11) and the L_2-star discrepancy is obtained.

[6] The centered discrepancy is the quadrant discrepancy anchored at the point $(1/2, \ldots, 1/2)$.

The kernel $K(\mathbf{x},\mathbf{y}) = \int_{[0,1]^d} B(\mathbf{x},\mathbf{t})B(\mathbf{y},\mathbf{t})d\mathbf{t}$ which corresponds to the r-smooth L_2-discrepancy is the reproducing kernel of the Sobolev space

$$H(K) = H_0^{r,\ldots,r}([0,1]^d) = \bigotimes_{i=1}^{d} H_0^r([0,1])$$

where $H_0^r([0,1])$ is the space of functions whose $(r-1)$-st derivatives are absolutely continuous, which satisfy the boundary conditions $f^{(j)}(0) = 0$ for $j = 0,\ldots,r-1$ and whose r-th derivative belongs to $L_2[(0,1)]$. By a modification of the kernel function it is possible to remove the boundary conditions and to consider the Sobolev space $H^{r,\ldots,r}([0,1]^d)$ with the norm $\|f\|_H^2 = \sum_{j=1}^{r-1}(f^{(j)}(0))^2 + \|f^{(r)}\|_2^2$, see [125]. The r-smooth L_2-discrepancy of a SG algorithms in the space $H_0^{r,\ldots,r}([0,1]^d)$ can be derived in only $O(n(\log n)^{2-d} + d(\log(n))^4)$ operations, if the tensor product structure of sparse grids is exploited, see [40].

Weighted L_2-star discrepancy

If we consider the weighted L_2-discrepancy (A.10) with B as in (A.11) and with equal weights $\gamma_{\mathbf{u}} = 1$, then the *generalized L_2-star discrepancy* is obtained, which was introduced in [76]. Although the square of the generalized L_2-star discrepancy is defined as the sum of 2^d terms, see (A.10), it can be computed in only $O(dn^2)$ operations using the explicit formula

$$\begin{aligned} D_{\mathbf{1}}^2(Q_n) &= \left(\frac{4}{3}\right)^d - 2\sum_{i=1}^{n} w_i \prod_{j=1}^{d} \frac{3-(x_j^i)^2}{2} \\ &\quad + \sum_{i,k=1}^{n} w_i w_k \prod_{j=1}^{d} \left(2 - \max\{x_j^i, y_j^k\}\right) \end{aligned} \tag{A.15}$$

which is proved in [76]. The initial generalized L_2-star discrepancy is given by $D_{\mathbf{1}}(Q_0) = (4/3)^{d/2}$. It is thus exponentially large in d.

Next, we consider *product weights* $\gamma_{\mathbf{u}} = \prod_{j\in\mathbf{u}} \gamma_j$. Then, one recovers from (A.10) with B as in (A.11) the *weighted L_2-star discrepancy* which was introduced in [148] to analyse QMC methods in weighted Sobolev spaces. One obtains from (A.7) that

$$K(\mathbf{x},\mathbf{y}) = \prod_{j=1}^{d} \left(1 + \gamma_j(1 - \max\{x_j^i, y_j^k\})\right).$$

This kernel corresponds to the space

$$H(K) = H_{\boldsymbol{\gamma}}^{r,\ldots,r}([0,1]^d) = \{f \in H^{r,\ldots,r}([0,1]^d) : \|f\|_{\boldsymbol{\gamma},H} < \infty\}$$

with the norm

$$\|f\|^2_{\boldsymbol{\gamma},H} = \sum_{\mathbf{u}\subseteq\{1,\ldots,d\}} \gamma_{\mathbf{u}}^{-1} \int_{[0,1]^{|\mathbf{u}|}} \left| \frac{\partial^{|\mathbf{u}|}}{\partial \mathbf{x}_{\mathbf{u}}} f(\mathbf{x}_{\mathbf{u}}, \mathbf{1}) \right|^2 d\mathbf{x}_{\mathbf{u}}. \tag{A.16}$$

For the weighted L_2-star discrepancy, the explicit formula

$$\begin{aligned} D^2_{\boldsymbol{\gamma}}(Q_n) \;=\; & \prod_{j=1}^{d} \left(1 + \frac{\gamma_j}{3}\right) - 2 \sum_{i=1}^{n} w_i \prod_{j=1}^{d} \left(1 + \frac{\gamma_j}{2}(1 - (x_j^i)^2)\right) \\ & + \sum_{i,k=1}^{n} w_i w_k \prod_{j=1}^{d} \left(1 + \gamma_j (1 - \max\{x_j^i, y_j^k\})\right) \end{aligned} \tag{A.17}$$

holds, whose computation requires $O(dn^2)$ operations. It is derived in [86] for quadrature rules with equal weights.

If we use *general weights* $\gamma_{\mathbf{u}}$ as in (A.9), then the associated kernel function is

$$K(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{u}\subseteq\{1,\ldots,d\}} \gamma_{\mathbf{u}} K_{\mathbf{u}}(\mathbf{x}, \mathbf{y})$$

with

$$K_{\mathbf{u}}(\mathbf{x}, \mathbf{y}) = \prod_{j\in\mathbf{u}} \left(1 + \frac{1}{2}(|x_j - a_j| + |y_j - a_j| - |x_j - y_j|)\right).$$

The corresponding reproducing kernel Hilbert space $H(K)$ is the sum of the spaces $H(K_{\mathbf{u}})$ for all $\mathbf{u}$ for which $\gamma_{\mathbf{u}} > 0$ holds. The norm in this space is given by (A.16). In general, the kernel K and the space $H(K)$ are thus defined as the sum of 2^d-many terms. If some of the weights are zero (e.g. if *finite-order weights* are used), then $H(K)$ is not a tensor product space. The notion of weighted L_2-discrepancies can be further generalized by using the L_p-norm with $1 \le p \le \infty$ in (A.10), see [148].

A.3 Tractability Results

In this section, we report bounds on the minimal discrepancies which can be achieved by quadrature rules Q_n of the form (3.3) with n points $\mathbf{x}^i$ and weights w_i. We start with classical bounds which depend on the number n of sample points and hold for a fixed dimension d. Then, we proceed with more recent estimates which hold for variable n and variable d. Finally, we use the latter estimates to study tractability in different function spaces. Thereby, we restrict ourselves to the L_2-star discrepancy (A.12) and to the weighted L_2-star discrepancy (A.17). They are related to integration in the Sobolev spaces of functions which have bounded mixed derivatives in the L_2-norm. For many other discrepancy measures similar estimates are valid, see [125].

Minimal L_2-discrepancies

The minimal discrepancy is defined as

$$D(n,d) := \inf \{D(Q_n) : \, Q_n \text{ as in (3.3)}\}.$$

Because of (A.8), it equals to the minimal worst case error $e(n,d)$ from Section A.1. Next, we summarize known bounds on the minimal discrepancy $D(n,d)$ and compare it to the discrepancies $D(Q_n)$ which can be achieved by QMC and SG algorithms.

We start with the L_2-star discrepancy and the case that d is fixed. Then, it is known that there exist constants $c_d > 0$ and $C_d > 0$ such that

$$c_d \, \frac{(\log n)^{(d-1)/2}}{n} \leq D(n,d) \leq C_d \, \frac{(\log n)^{(d-1)/2}}{n}$$

for all $n \geq 2$.

QMC methods are based on low-discrepancy sequences. Such sequences satisfy the bound

$$D(Q_n) \leq C_d \frac{(\log n)^d}{n}$$

for all $n \geq 2$. QMC methods are thus almost (up to a different power of the logarithmic factor) optimal with respect to the asymptotic behaviour of their L_2-star discrepancy. Explicit bounds on the L_2-star discrepancy of SG quadrature rules are given in [163] which imply

$$D(Q_n) \leq C_d \frac{(\log n)^{3d/2}}{n}$$

for all $n \geq 2$. Hence, SG methods show a similar almost optimal asymptotic behaviour for fixed d. Recently, also methods Q_n have been found based on randomized digital net constructions, which are optimal with respect to the asymptotic behaviour of their L_2-star discrepancy, see [126] and the references therein. Note that the constants in the above estimates depend on d and may also be exponentially large in d. In this case it is not clear if a good asymptotic behaviour of $D(Q_n)$ can be utilized in practical computations.

Tractability of the absolute L_2-star discrepancy

To investigate the impact of the dimension on the size of the constants, we next consider the exponent $p^\star$ of the L_2-star discrepancy which is defined as the smallest number p for which there exists a constant $C > 0$ such that

$$D(n,d) \leq C n^{-1/p}$$

for all $n, d \in \mathbb{N}$.[7] Note that here neither the constant C nor the exponent p depends on d.

Next, we report lower and upper bound on $p^\star$. An inspection of the case $d = 1$, shows that $p^\star \geq 1$. For quadrature rules with equal weights it is proved in [108] that $p^\star \geq 1.0699$. For sparse grid methods the lower bound $p^\star \geq 2.1933$ is found in [137]. In a non-constructive way it is shown in [164] that $p^\star \leq 1.4778$. The best known construction obtains $p = 2.454$, see [163], where sparse grid methods are used. In conclusion, the results show that quadrature rules can be constructed whose L_2-star discrepancy depends at most polynomially on the dimension d. In view of (A.8) this in particular implies that integration in the Sobolev space $H_0^{1,\ldots,1}([0,1]^d)$ is strongly tractable with respect to the absolute worst case error.

Intractability of the relative L_2-star discrepancy

It is questionable, though, if the question for the exponent $p^\star$ of the absolute L_2-star discrepancy is properly scaled. Since the initial L_2-star discrepancy $D(Q_0) = 3^{-d/2}$ is exponentially small in d, it indicates that integration is getting easier with respect to the dimension which is not in agreement with the behaviour of the numerical algorithms. This problem is avoided if the relative (or normalised) L_2-star discrepancy $D(Q_n)/D(Q_0)$ is considered. Because of (A.8), this quantity is directly related to the notion of tractability for the normed worst case error, see Section A.1.

To describe the behaviour of the relative L_2-star discrepancy, let

$$n(\varepsilon, d) := \min \{n : \, D(n,d)/D(0,d) \leq \varepsilon\}$$

denote the minimal number of points which are necessary to reduce the initial minimal discrepancy by a factor of ε. For quadrature methods with equal weights $w_i = 1/n$, it is known that

$$\left(\frac{9}{8}\right)^d (1-\varepsilon^2) \leq n(\varepsilon, d) \leq \left(\frac{3}{2}\right)^d \varepsilon^{-2}.$$

For quadrature methods with arbitrary weights $w_i \in \mathbb{R}$, the asymptotic estimates

$$1.0628^d(1+o(1)) \leq n(\varepsilon, d) \leq \left(\frac{3}{2}\right)^d \varepsilon^{-2}$$

can be proved, where the lower bound holds for all $\varepsilon < 1$, see [124]. Hence, in both cases $n(\varepsilon, d)$ increases exponentially fast in d. The normalised L_2-star discrepancy

[7] Note that the exponent $p^\star$ of the absolute L_2-star discrepancy equals to the ε-exponent of strong tractability for the absolute error criterion in the Sobolev space $H^{1,\ldots,1}([0,1]^d)$ anchored at one.

is thus intractable. Moreover, integration in the Sobolev space $H_0^{1,\dots,1}([0,1]^d)$ is intractable with respect to the normed worst case error. Similar results hold for the L_p-star discrepancies with $1 \le p < \infty$. The L_∞-star discrepancy satisfies $n(\varepsilon, d) \le Cd\varepsilon^{-2}$, though. It is thus tractable and depends only linearly on the dimension.

Tractability of the weighted L_2-star discrepancy

Next, we discuss tractability for the weighted L_2-star discrepancy (A.17). It relates to the tractability of integration in the Sobolev space $H^{1,\dots,1}([0,1]^d)$ without a boundary condition. Its minimal discrepancy, its minimal number of points, which are required to reduce the initial discrepancy by a factor of ε, and its exponent are denoted by $D_{\boldsymbol{\gamma}}(n,d)$, $n_{\boldsymbol{\gamma}}(\varepsilon,d)$ and $p^\star_{\boldsymbol{\gamma}}$, respectively.

For *equal weights* $\gamma_{\mathbf{u}} = 1$ and for quadrature methods with equal weights $w_i = 1/n$ it holds that

$$1.0563^d(1-\varepsilon^2) \le n_{\mathbf{1}}(\varepsilon,d) \le \left(\frac{9}{8}\right)^d \varepsilon^{-2}.$$

For arbitrary weights $w_i \in \mathbb{R}$ it is known that

$$1.0463^d(1+o(1)) \le n_{\mathbf{1}}(\varepsilon,d) \le \left(\frac{9}{8}\right)^d \varepsilon^{-2}$$

for all $\varepsilon < 1$, see [124]. The lower bounds show that the generalized L_2-star discrepancy (A.15) is exponentially large in d. Therefore, integration in the Sobolev space $H^{1,\dots,1}([0,1]^d)$ is intractable.

Next, we consider *finite-order weights* $\gamma_{\mathbf{u}}$. They imply strong tractability of multivariate integration for the relative error and tractability for the absolute error criterion, see [126]. Furthermore, it is found in [147] that the worst case errors of Halton, Sobol and Niederreiter QMC points depend at most polynomial on the dimension if finite-order weights are used. From these estimates one obtains that

$$n_{\boldsymbol{\gamma}}(\varepsilon,d) \le (Cd\log d)^{d_s}(\log\varepsilon^{-1} + \log(Cd\log d))^{d_s}\,\varepsilon^{-1},$$

where the exponent d_s denotes the order of the finite-order weights. Similar results are known for shifted lattice rules whose generating vector is tailored to the finite-order weights using the component-by-component construction, see [147].

For *product weights* $\gamma_{\mathbf{u}} = \prod_{j\in\mathbf{u}} \gamma_j$ and for quadrature rules with equal weights $w_i = 1/n$ it is shown in [148] that

$$1.055^{s_d}\,(1-\varepsilon^2) \le n_{\boldsymbol{\gamma}}(\varepsilon,d) \le (1.1836^{s_d}-1)\,\varepsilon^{-2}$$

for all $\varepsilon < 1$, where

$$s_d = \sum_{j=1}^{d} \gamma_j$$

is the sum of all weights. These estimates imply that $n_{\boldsymbol{\gamma}}(\varepsilon, d)$ is uniformly bounded in d and polynomially bounded in ε^{-1} if and only if

$$\sum_{j=1}^{\infty} \gamma_j < \infty. \tag{A.18}$$

In this case, we obtain that integration in the space $H_{\boldsymbol{\gamma}}^{1,\ldots,1}([0,1]^d)$ is strongly tractable. Furthermore, it holds that $n_{\boldsymbol{\gamma}}(\varepsilon, d)$ is polynomially bounded in d and ε^{-1} if and only if

$$\limsup_{d\to\infty} \left(\sum_{j=1}^{d} \frac{\gamma_j}{\log d} \right) < \infty,$$

which is equivalent to the statement that integration in the space $H_{\gamma}^{1,\ldots,1}([0,1]^d)$ is tractable.

The case of *higher smoothness* $r > 1$ is studied in [124] and leads to similar conclusions: For equal weights one obtains that integration in the Sobolev space $H^{r,\ldots,r}([0,1]^d)$ is intractable for all $r \geq 1$. For product weights γ_j it is proved that if

$$\sum_{j=1}^{\infty} \gamma_j = \infty$$

then integration in the weighted Sobolev space $H_{\boldsymbol{\gamma}}^{r,\ldots,r}([0,1]^d)$ is not strongly tractable. If

$$\limsup_{d\to\infty} \left(\sum_{j=1}^{d} \frac{\gamma_j}{\log d} \right) = \infty$$

then integration in the weighted Sobolev space $H_{\boldsymbol{\gamma}}^{r,\ldots,r}([0,1]^d)$ is intractable. It is not known if these conditions are also necessary for no strong tractability and intractability if $r > 1$.

Finally, we consider the exponent $p_{\boldsymbol{\gamma}}^{\star}$ of the weighted L_2-star discrepancy with product weights. It follows from the results in [148] for quadrature rules with equal weights that the exponent $p_{\boldsymbol{\gamma}}^{\star}$ of QMC quadrature exists if and only if (A.18) holds. Then, $p_{\boldsymbol{\gamma}}^{\star} \in [1,2]$. If $\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty$, then the optimal exponent $p_{\boldsymbol{\gamma}}^{\star} = 1$ is achieved with QMC methods. If the weights satisfy

$$\gamma_j = O(j^{-\alpha})$$

with $\alpha > 1$ then the exponent of SG quadrature is

$$p_{\boldsymbol{\gamma}}^{\star} = \max\left\{1, \frac{2}{\alpha - 1}\right\},$$

see [138]. Sparse grids are thus optimal for $\alpha \geq 3$ with $p^\star_\gamma = 1$. They far from optimal for $\alpha \in (1, 2)$.

In summary, the results show that deterministic methods of the form (3.3) can never completely avoid the curse of dimension in classical Sobolev spaces with bounded mixed derivatives. For weighted Sobolev spaces, however, integration is tractable if the function space weights decay sufficiently fast. In this case, tractability holds in the class of QMC methods as well as in the class of SG quadrature rules. Furthermore, the use of quadrature rules with arbitrary weights instead of only positive weights do not help to break intractability with respect to the normalized weighted L_2-star discrepancy. It is known that this is not true in general. There are spaces for with integration is tractable for the class of linear quadrature formulas whereas integration is intractable for the class of positive quadrature formulas, see [125].

Appendix B
Performance-dependent Options

In this chapter, we provide more details on the valuation of performance-dependent options as discussed in Section 6.3. To this end, we first recall the multivariate Black-Scholes model which we use to describe the temporal development of the involved asset prices. Then, we derive closed-form representations of the prices of performance-dependent options for full and for reduced models. In the reduced model case the formula is based on the orthant decomposition from Theorem 5.5.

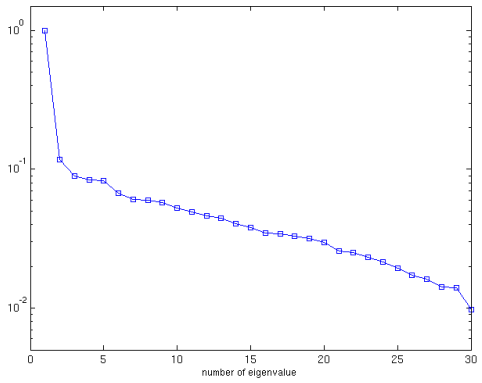
B.1 Multidimensional Black-Scholes Model

For the valuation of derivatives in markets with several interacting assets, the multidimensional Black-Scholes model [70, 91] has been used with great success. There, it is assumed that the stock prices are driven by $d \leq n$ stochastic processes modeled by the Black-Scholes-type system of stochastic partial differential equations (SDEs)

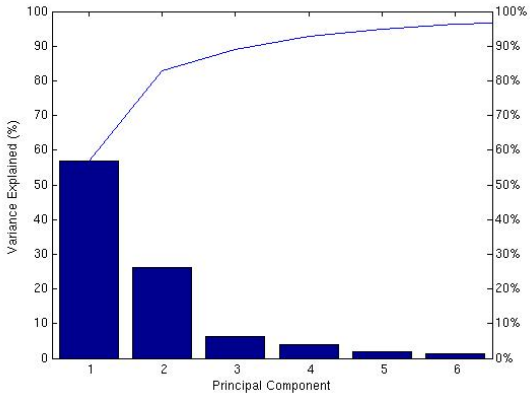
$$dS_i(t) = S_i(t) \left(\mu_i dt + \sum_{j=1}^{d} \sigma_{ij} dW_j(t) \right) \tag{B.1}$$

for $i = 1, \ldots, n$. Here, μ_i denotes the drift of the i-th stock, $\boldsymbol{\sigma}$ the $n \times d$ volatility matrix of the stock price movements and $W_j(t)$ the corresponding Wiener processes. The matrix $\boldsymbol{\sigma}\boldsymbol{\sigma}^T$ is assumed to be positive definite. If $d = n$, we call the corresponding model *full*. If $d < n$, the model is called *reduced*.

Let us remark here that for small benchmarks usually the full model with a square volatility matrix $\boldsymbol{\sigma}$ is used. The entries of the volatility matrix are typically estimated from historical market data. However, for larger benchmarks, the parameter estimation problem becomes more and more ill-conditioned resulting in eigenvalues of $\boldsymbol{\sigma}\boldsymbol{\sigma}^T$ which are close to zero. Then, reduced models with $d < n$ are often employed. If the benchmark consists of all assets in a stock index, this reduction can be achieved, for instance, by grouping assets in the same area of business. The matrix entry σ_{ij} then reflects the correlation of stock i with business area j. Such a grouping can often be obtained without much loss of information e.g. using Principal Component Analysis (PCA), as was confirmed empirically by research from Meade and Salkin [109] and Laloux *et al.* [99], see also [141]. For illustration, we estimated the covariance matrix of the 30 stock prices in the German stock index DAX using the daily data from 1. January 2004 – 31. December 2004. The results of the PCA are shown in Figure B.1.



(a) The eigenvalues plotted in a logarithmic scale in decaying order with the largest eigenvalue scaled to one.



(b) Percentage of the total variance which is explained by the first six eigenvectors.

Figure B.1. *The eigenvalues of the covariance matrix of the German stock index DAX.*

One can see that the eigenvalues decay exponentially and that already five components suffice to explain more than 95 percentage of the total variance.

By Itô's formula, the explicit solution of the SDE (B.1) is given by

$$S_i(T) = S_i(\mathbf{X}) = S_i(0) \exp\left(\mu_i T - \bar{\sigma}_i + \sqrt{T} \sum_{j=1}^{d} \sigma_{ij} X_j\right) \tag{B.2}$$

for $i = 1, \ldots, n$ with

$$\bar{\sigma}_i := \frac{1}{2} \sum_{j=1}^{d} \sigma_{ij}^2 \, T$$

and $\mathbf{X} = (X_1, \ldots, X_d)$ being a $N(\mathbf{0}, \mathbf{I})$-normally distributed random vector.

B.2 Pricing Formulas

We next derive closed-form formulas for the prices of performance-dependent options for full and for reduced models.

Martingale approach

The multivariate Black-Scholes model induces a complete market which gives the existence of a unique equivalent martingale measure. Using the usual Black-Scholes assumptions, see, e.g., [91], the option price $V(S_1, 0)$ is given by the discounted

expectation

$$V(S_1, 0) = e^{-rT}\,\mathbb{E}[V(\mathbf{S}, T)] \tag{B.3}$$

of the payoff under the equivalent martingale measure. To this end, the drift μ_i is replaced by the riskless interest rate r for each stock i. In the case of a performance-dependent option with payoff (6.16) we get

$$V(S_1, 0) = e^{-rT}\,\mathbb{E}\left[\sum_{\mathbf{R}\in\{+,-\}^n} a_{\mathbf{R}}(S_1(T) - K)\,\chi_{\mathbf{R}}(\mathbf{S})\right].$$

Thereby, the expectation runs over all possible rankings $\mathbf{R}$ and the characteristic function $\chi_{\mathbf{R}}(\mathbf{S})$ is defined by

$$\chi_{\mathbf{R}}(\mathbf{S}) = \begin{cases} 1 & \text{if } \mathbf{Rank}(\mathbf{S}) = \mathbf{R}, \\ 0 & \text{else.} \end{cases}$$

Plugging in the density function $\varphi(\mathbf{x}) := \varphi_{\mathbf{0},\mathbf{I}}(\mathbf{x})$ of the $N(\mathbf{0}, \mathbf{I})$-distributed random vector $\mathbf{X}$ (note that $\mathbf{S} = \mathbf{S}(\mathbf{X})$), we get

$$V(S_1, 0) = e^{-rT} \int_{R^d} \sum_{\mathbf{R}\in\{+,-\}^n} a_{\mathbf{R}}(S_1(T) - K)\,\chi_{\mathbf{R}}(\mathbf{S})\varphi(\mathbf{x})\; d\mathbf{x} \tag{B.4}$$

which will be the starting point of the following analysis.

Full model

In this section, we assume that the number of stochastic processes d equals the number of assets n. We derive the price of a performance-dependent option as a multivariate integral and show how this integral can be evaluated in terms of multivariate normal distributions. In the following, we nevertheless distinguish between d and n in order to be able to reuse some of the results also for the reduced model case.

To prove our main theorem we need the following two lemmas which relate the payoff conditions to multivariate normal distributions Φ. To this end, recall that the multivariate normal probability with limits $\mathbf{b} = (b_1, \ldots, b_d)$, mean zero and covariance matrix $\mathbf{A}$ is defined by

$$\Phi(\mathbf{A}, \mathbf{b}) := \int_{-\infty}^{b_1} \ldots \int_{-\infty}^{b_d} \varphi_{\mathbf{0},\mathbf{A}}(\mathbf{x})\; dx_d \ldots dx_1. \tag{B.5}$$

Here, the multivariate Gaussian density with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ is defined by

$$\varphi_{\boldsymbol{\mu},\mathbf{A}}(\mathbf{x}) := \frac{1}{(2\pi)^{d/2}(\det \mathbf{A})^{1/2}}\; e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{A}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \tag{B.6}$$

Lemma B.1. *Let* $\mathbf{b}, \mathbf{q} \in \mathbb{R}^d$ *and* $\mathbf{A} \in \mathbb{R}^{d\times d}$ *with full rank, then*

$$\int_{\mathbf{Ax}\geq\mathbf{b}} e^{\mathbf{q}^T\mathbf{x}}\varphi(\mathbf{x})d\mathbf{x} = e^{\frac{1}{2}\mathbf{q}^T\mathbf{q}}\Phi(\mathbf{AA}^T, \mathbf{Aq}-\mathbf{b}).$$

We use $\int_{\mathbf{Ax}\geq\mathbf{b}}$ *as abbreviation for the integration over the set* $\{\mathbf{x}\in\mathbb{R}^d : \mathbf{Ax}\geq\mathbf{b}\}$.

Proof. A straightforward computation shows

$$e^{\mathbf{q}^T\mathbf{x}}\varphi(\mathbf{x}) = e^{\frac{1}{2}\mathbf{q}^T\mathbf{q}}\varphi_{\mathbf{q},\mathbf{I}}(\mathbf{x})$$

for all $\mathbf{x} \in \mathbb{R}^d$. Using the substitution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y} + \mathbf{q}$ we obtain

$$\begin{aligned}\int_{\mathbf{Ax}\geq\mathbf{b}} e^{\mathbf{q}^T\mathbf{x}}\varphi(\mathbf{x})d\mathbf{x} &= e^{\frac{1}{2}\mathbf{q}^T\mathbf{q}}\int_{\mathbf{Ax}\geq\mathbf{b}}\varphi_{\mathbf{q},\mathbf{I}}(\mathbf{x})d\mathbf{x}\\ &= e^{\frac{1}{2}\mathbf{q}^T\mathbf{q}}\int_{\mathbf{y}\geq\mathbf{b}-\mathbf{Aq}}\varphi_{0,\mathbf{AA^T}}(\mathbf{y})\ d\mathbf{y}\end{aligned}$$

and thus the assertion. □

For the second lemma, we first need to define a comparison relation $\geq_{\mathbf{R}}$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with respect to the ranking $\mathbf{R}$:

$$\mathbf{x} \geq_{\mathbf{R}} \mathbf{y} \ :\Leftrightarrow\ R_i(x_i - y_i) \geq 0 \ \text{ for } 1 \leq i \leq n.$$

Thus, the comparison relation $\geq_{\mathbf{R}}$ is the usual component-wise comparison where the direction depends on the sign of the corresponding entry of the ranking vector $\mathbf{R}$.

Lemma B.2. *We have* $\mathbf{Rank}(\mathbf{S}) = \mathbf{R}$ *exactly if* $\mathbf{AX} \geq_{\mathbf{R}} \mathbf{b}$ *with*

$$\mathbf{A} := \sqrt{T}\begin{pmatrix}\sigma_{11} & \dots & \sigma_{1d}\\ \sigma_{11}-\sigma_{21} & \dots & \sigma_{1d}-\sigma_{2d}\\ \vdots & & \vdots\\ \sigma_{11}-\sigma_{n1} & \dots & \sigma_{1d}-\sigma_{nd}\end{pmatrix} \text{ and } \mathbf{b} := \begin{pmatrix}\ln\frac{K}{S_1(0)} - rT + \bar\sigma_1\\ \bar\sigma_1 - \bar\sigma_2\\ \vdots\\ \bar\sigma_1 - \bar\sigma_n\end{pmatrix}$$

where $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{X} \in \mathbb{R}^d$ *and* $\mathbf{b} \in \mathbb{R}^n$.

Proof. Using (B.2) we see that $\mathrm{Rank}_1(\mathbf{S}) = +$ is equivalent to

$$S_1(T) \geq K \quad\Longleftrightarrow\quad \sqrt{T}\sum_{j=1}^{d}\sigma_{1j}X_j \geq \ln\frac{K}{S_1(0)} - rT + \bar\sigma_1$$

which yields the first row of the system $\mathbf{AX} \geq_{\mathbf{R}} \mathbf{b}$. Moreover, for $i = 2, \ldots, n$ the outperformance criterion $\mathrm{Rank}_i(\mathbf{S}) = +$ can be written as

$$\frac{S_1(T)}{S_1(0)} \geq \frac{S_i(T)}{S_i(0)} \quad \Longleftrightarrow \quad \sqrt{T}\sum_{j=1}^{d}(\sigma_{1j} - \sigma_{ij})X_j \geq \bar{\sigma}_1 - \bar{\sigma}_i$$

which yields rows 2 to n of the system. □

Now we can state the following pricing formula which, in a slightly more special setting, is originally due to Korn [96].

Theorem B.3. *The price of a performance-dependent option with payoff (6.16) is for the model (B.1) in the case $d = n$ given by*

$$V(S_1, 0) = \sum_{\mathbf{R}\in\{+,-\}^n} a_{\mathbf{R}} \left(S_1(0)\, \Phi(\mathbf{A_R A_R^T}, -\mathbf{d_R}) - e^{-rT}\, K\Phi(\mathbf{A_R A_R^T}, -\mathbf{b_R}) \right)$$

where the vectors $\mathbf{b_R}$, $\mathbf{d_R}$ and the matrix $\mathbf{A_R}$ are defined by $(\mathbf{b_R})_i := R_i \mathbf{b}_i$, $(\mathbf{d_R})_i := R_i \mathbf{d}_i$ and $(\mathbf{A_R})_{ij} := \mathbf{R}_i \mathbf{A}_{ij}$. Thereby, $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are defined as in Lemma B.2 and the vector $\mathbf{d} \in \mathbb{R}^n$ is defined by $\mathbf{d} := \mathbf{b} - \sqrt{T}\mathbf{A}\sigma_1$ with σ_1^T being the first row of the volatility matrix $\boldsymbol{\sigma}$.

Proof. The characteristic function $\chi_{\mathbf{R}}(\mathbf{S})$ in the integral (B.4) can be eliminated using Lemma B.2 and we get

$$V(S_1, 0) = e^{-rT} \sum_{\mathbf{R}\in\{+,-\}^n} a_{\mathbf{R}} \int_{\mathbf{Ax}\geq_{\mathbf{R}}\mathbf{b}} (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}. \tag{B.7}$$

By (B.2), the integral term can be written as

$$S_1(0)e^{rT-\bar{\sigma}_1} \int_{\mathbf{Ax}\geq_{\mathbf{R}}\mathbf{b}} e^{\sqrt{T}\sigma_1^T\mathbf{x}}\, \varphi(\mathbf{x})d\mathbf{x} - K \int_{\mathbf{Ax}\geq_{\mathbf{R}}\mathbf{b}} \varphi(\mathbf{x})d\mathbf{x}.$$

Application of Lemma B.1 with $\mathbf{q} = \sqrt{T}\sigma_{\mathbf{1}}$ shows that the first integral equals

$$e^{\frac{1}{2}\mathbf{q}^T\mathbf{q}} \int_{\mathbf{y}\geq_{\mathbf{R}}\mathbf{b}-\mathbf{Aq}} \varphi_{0,\mathbf{AA^T}}(\mathbf{y})\; d\mathbf{y} = e^{\bar{\sigma}_1} \int_{\mathbf{y}\geq\mathbf{d_R}} \varphi_{0,\mathbf{A_R A_R^T}}(\mathbf{y})\; d\mathbf{y} = e^{\bar{\sigma}_1}\Phi(\mathbf{A_R A_R^T}, -\mathbf{d_R}).$$

By a further application of Lemma B.1 with $\mathbf{q} = \mathbf{0}$ we obtain that the second integral equals $K\Phi(\mathbf{A_R A_R^T}, -\mathbf{b_R})$ and thus the assertion holds. □

Note that this decomposition not only provides the option price as a sum of normal distributions but can also be used to show which rankings appear with which

probabilities under the model assumptions. Note further that the pricing formula in Theorem B.3 allows a stable and efficient valuation of performance-dependent options in the case of moderate-sized benchmarks. For a large number n of benchmark assets, one is, however, confronted with the following problems:

- In total, 2^n rankings have to be considered and thus an with n exponentially growing number of cumulative normal distributions have to be computed.
- For each normal distribution, an n-dimensional integration problem has to be solved which gets increasingly more difficult with rising n.
- In larger benchmarks, stock prices are typically highly correlated. As a consequence, some of the eigenvalues of the covariance matrix $\boldsymbol{\sigma}$ will be very small which makes the integration problems ill-conditioned.
- There is a large number $(n(n+1)/2)$ of free model parameters in the volatility matrix which are difficult to estimate robustly for large n.

In conclusion, the pricing formula in Theorem B.3 can only be applied to small benchmarks, although it is very useful in this case. In the next section, we aim to derive a similar pricing formula for reduced models which incorporate less processes than companies ($d < n$). This way, substantially fewer rankings have to be considered and much lower-dimensional integrals have to be computed which allows the pricing of performance-dependent options even for large benchmarks.

Reduced model

In this section, we assume that the number of stochastic processes d is less than the number of assets n. Lemma B.2 and thus representation (B.7) of the option price remains also valid in the reduced model. Note, however, that $\mathbf{A}$ is now an $(n \times d)$-matrix which prevents the direct application of Lemma B.1. At this point, a geometrical point of view is advantageous to illustrate the effect of performance comparisons in the reduced model.

The matrix $\mathbf{A}$ and the vector $\mathbf{b}$ define a set of n hyperplanes in the space $\mathbb{R}^d$. The set $\mathbf{Ax} \geq_{\mathbf{R}} \mathbf{b}$ thus describes a polyhedron P, which is bounded by up to n hyperplanes in the d-dimensional space. Each polyhedron thereby corresponds to a unique ranking $\mathbf{R}$ and hence to a particular bonus factor $a_{\mathbf{R}}$. The integrand (B.4) therefore exhibits kinks or jumps along the boundary of each polyhedron, i.e., in the set of points $\mathbf{x}$ which satisfy the linear system $\mathbf{Ax} = \mathbf{b}$.

By identifying all cells P in the hyperplane arrangement, as explained in Section 5.2.2, the representation (B.7) of the option price can be rewritten as

$$V(S_1, 0) = e^{-rT} \sum_{P \in \mathcal{A}} a_{\mathbf{R}} \int_P (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}. \tag{B.8}$$

By integrating the payoff function over each cell of the hyperplane arrangement separately, the option value can be determined as a sum over all integral values weighted with the corresponding bonus factors. Note that only smooth integrands appear in this approach. To circumvent the problem that the integration region is now a general polyhedron we use the orthant decomposition of each polyhedral cell, as explained in Section 5.2.2.

Now, we are able to give a pricing formula for performance-dependent options also in the reduced model case.

Theorem B.4. *The price of a performance-dependent option with payoff (6.16) is for the model (B.1) in the case $d \leq n$ given by*

$$V(S_1,0) = \sum_{\mathbf{v}\in\mathcal{V}} c_{\mathbf{v}} \left(S_1(0)\Phi(\mathbf{A}_{\mathbf{v}}\mathbf{A}_{\mathbf{v}}^T, -\mathbf{d}_{\mathbf{v}}) - e^{-rT}K\Phi(\mathbf{A}_{\mathbf{v}}\mathbf{A}_{\mathbf{v}}^T, -\mathbf{b}_{\mathbf{v}}) \right)$$

with $\mathbf{A}_{\mathbf{v}}, \mathbf{b}_{\mathbf{v}}$ as in (5.13) and $\mathbf{d}_{\mathbf{v}}$ being the corresponding subvector of $\mathbf{d}$. The weights $c_{\mathbf{v}}$ are given by

$$c_{\mathbf{v}} := \sum_{\mathbf{w}\in\mathcal{V}:\ \mathbf{v}\in P_{\mathbf{w}}} s_{\mathbf{v},\mathbf{w}} a_{\mathbf{w}}. \tag{B.9}$$

Proof. By Lemma 5.3 we see that the integral representation (B.8) is equivalent to a summation over all vertices $\mathbf{v} \in \mathcal{V}$, i.e.

$$V(S_1,0) = e^{-rT} \sum_{\mathbf{v}\in\mathcal{V}} a_{\mathbf{v}} \int_{P_{\mathbf{v}}} (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}.$$

By Lemma 5.4 we can decompose the polyhedron $P_{\mathbf{v}}$ into a signed sum of orthants and obtain

$$V(S_1,0) = e^{-rT} \sum_{\mathbf{v}\in\mathcal{V}} a_{\mathbf{v}} \sum_{\mathbf{w}\in\mathcal{V}_{\mathbf{v}}} s_{\mathbf{v},\mathbf{w}} \int_{O_{\mathbf{w}}} (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}.$$

By the second part of Lemma 5.4 we know that only $c_{n,d}$ different integrals appear in the above sum. Rearranging the terms leads to

$$V(S_1,0) = e^{-rT} \sum_{\mathbf{v}\in\mathcal{V}} c_{\mathbf{v}} \int_{O_{\mathbf{v}}} (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}.$$

Since now the integration domains $O_{\mathbf{v}}$ are orthants, Lemma B.1 can be applied exactly as in the proof of Theorem B.3 which finally implies the theorem. □

To compute the weights $c_{\mathbf{v}}$, all cells $P_{\mathbf{w}}$ incident in $\mathbf{v}$ have to be traversed and their ranking vectors have to be be determined. This can be done symbolically by

Algorithm B.1: Valuation algorithm for performance-dependent options in the reduced model case.

1) To compute the set of all intersection points $\mathcal{V}$:
- a) compute the set of vertices of the hyperplane arrangement
- b) compute the bounding box of these vertices
- c) compute the set of boundary intersection points

2) For each intersection point $\mathbf{v} \in \mathcal{V}$:
- a) determine the submatrix $\mathbf{A}_\mathbf{v}$ and the subvectors $\mathbf{d}_\mathbf{v}$ and $\mathbf{b}_\mathbf{v}$
- b) evaluate the cumulative normal distributions $\Phi(\mathbf{A}_\mathbf{v}\mathbf{A}_\mathbf{v}^T, -\mathbf{d}_\mathbf{v})$ and $\Phi(\mathbf{A}_\mathbf{v}\mathbf{A}_\mathbf{v}^T, -\mathbf{b}_\mathbf{v})$
- c) for all vertices $\mathbf{w} \in \mathcal{V}$ whose polyhedra $P_\mathbf{w}$ contain $\mathbf{v}$: determine the reflection signs $s_{\mathbf{v},\mathbf{w}}$ and bonus factors $a_\mathbf{w}$
- d) compute the weight $c_\mathbf{v}$ using formula (B.9)

3) Compute the price of the option as the weighted sum over all normal distributions according to Theorem B.4

flipping the signs in the ranking vector of $P_\mathbf{v}$ which correspond to the hyperplanes intersecting in $\mathbf{v}$. By the non-degeneracy condition there are at most 2^d cells adjacent to each vertex which bounds the number of terms in the definition of $c_\mathbf{v}$. Moreover, the number of vertices in $\mathcal{V}$ equals $c_{n,d}$ which yields the number of integrals which have to be computed in the worst case. The structure of the valuation algorithm is summarized in Algorithm B.1.

Example B.5. Consider the bonus scheme from Example 6.5 with $n = 3$, $d = 2$ and the hyperplane arrangement from Figure 5.6. Then, the bonus factors $a_j := a_{\mathbf{v}_j}$ are given by

$$a_1 = 0,\ a_2 = 0,\ a_3 = 0,\ a_4 = \frac{1}{2},\ a_5 = 1,\ a_6 = 0,\ a_7 = \frac{1}{2}.$$

Following the steps in the proof of Theorem B.4 and employing the decomposition from Example 5.6 we see that the price of this option satisfies

$$\begin{aligned} V(S_1, 0) &= e^{-rT}\left(\frac{1}{2}\,I(P_4) + I(P_5) + \frac{1}{2}\,I(P_7)\right) \\ &= e^{-rT}\left(-\frac{1}{2}\,I(O_1) - \frac{1}{2}\,I(O_2) + \frac{1}{2}\,I(O_5) - \frac{1}{2}\,I(O_6) + \frac{1}{2}\,I(O_7)\right) \end{aligned}$$

where we define

$$I(B) := \int_B (S_1(T) - K)\varphi(\mathbf{x})d\mathbf{x}.$$

Let us remark that, if the payoff function has a special structure, many weights $c_{\mathbf{v}}$ are zero in the formula from Theorem B.4. This way, the corresponding normal distributions do not have to be computed. This is, for example, true for the outperformance option of Example 6.6.

In addition, if the vertex $\mathbf{v}$ is located on the artificial boundary, see for example vertex $\mathbf{v}_3$ in Figure 5.6, the corresponding orthant is defined by $k < d$ intersecting hyperplanes. As a consequence, only a k-dimensional normal distribution instead of a d-dimensional one has to be computed. Consider, for example, a bonus scheme which is defined by the bonus factors

$$a_{\mathbf{R}} = \begin{cases} \displaystyle\sum_{\{i:R_i=+\}} \bar{a}_i & \text{if } \mathbf{R}_1 = + \\ 0 & \text{else} \end{cases} \tag{B.10}$$

for some given $\bar{a}_i \in \mathbb{R}$, where the sum goes over all $i \in \{2, \ldots, n\}$ where $R_i = +$. Example 6.5 is a special case of such a scheme with $\bar{a}_i \equiv 1/(n-1)$. The pricing formula for such a scheme only contains vertices which are located on at least $d-2$ boundary hyperplanes. Thus, independently of d and n, at most two-dimensional normal distributions have to be evaluated. Moreover, the number of two-dimensional normal distributions is bounded by $n-1$. This behaviour is most easily understood if the payoff function of the bonus scheme (B.10) is rewritten in the equivalent form

$$V(\mathbf{S}, T) = \sum_{i=2}^{n} \bar{a}_i \left(S_1(T) - K\right)^+ \chi_{\Delta S_1(T) \geq \Delta S_i(T)}$$

which shows that only the two-dimensional joint distributions of the random variables $S_1(T)$ and $S_i(T)$ are required for $i = 2, \ldots, n$. Note that these special cases are automatically recognized by our algorithm and only the minimum number of integrals with the corresponding minimal dimensions are computed.

An additional advantage of the formulas from Theorem B.3 and B.4 is given by the fact that option price sensitivities (the Greek letters) can be obtained by analytical differentiation. Their computation can thus be integrated in the valuation algorithm without much additional effort.

Let us finally remark that we restricted ourselves to payoff profiles which depend on relative performance comparisons to a specific benchmark. Payoff profiles which include absolute performance criteria, e.g., performance comparisons with different strike prices, can also be included. The corresponding valuation formulas then include weighted sums of gap option prices.

Appendix C
Asset-Liability Management in Life Insurance

In this chapter, we provide more details on asset-liability management models in life insurance. This application was considered in Section 6.4.

First, in Section C.1, we describe our general ALM model framework for the simulation of the future development of a life insurance company in more detail. Then, we study in Section C.2 to which extent mathematical properties of the model, such as its variances, its smoothness and its effective dimensions, relate to the convergence behaviour of MC, QMC and SG methods.

C.1 Modeling

The scope of asset-liability management (ALM) is the responsible administration of the assets and liabilities of insurance contracts. To this end, the insurance company has to attain two goals simultaneously. On the one hand, the available capital has to be invested profitably, usually in bonds but also, up to a certain percentage, in stocks (asset management). On the other hand, the obligations against policyholders, which depend on the specific insurance policies, have to be met (liability management).

Here, we focus on portfolios of participating (with-profit) policies which make up a significant part of the life insurance market. The holder of such a policy gets a fixed guaranteed interest and, in addition, a variable reversionary bonus which is annually added to the policyholder's account and allows the policyholder to participate in the investment returns of the company. Thereby, the insurance company has to declare in each year which part of the investment returns is given to the policyholders as reversionary bonus, which part is saved in a reserve account for future bonus payments and which part is kept by the shareholders of the company. These management decisions depend on the financial situation of the company as well as on strategic considerations and legal requirements. A maximisation of the shareholders' benefits has to be balanced with a competitive bonus declaration for the policyholders. Moreover, the exposure of the company to financial, mortality and surrender risks has to be taken into account. These problems, which easily become quite complex due to the wide range of guarantees and option-like features of insurance products and management rules, are investigated with the help of ALM analyses. In this context, it is necessary to estimate the medium- and long-term

development of all assets and liabilities of the company as well as the interactions between them and to determine their sensitivity to the different types of risks. This can either be achieved by the computation of particular scenarios (stress tests) which are based on historical data, subjective expectations, and guidelines of regulatory authorities or by a stochastic modeling and simulation of the market development, the policyholder behaviour and all involved accounts, see, e.g., [84].

In recent years, the latter approach has attracted more and more attention as it takes financial uncertainties more realistically into account than an analysis of a small number of deterministically given scenarios. Additional importance arises from the current need of insurance companies to move from an accounting based on book values to a market-based, fair value accountancy standard as required by Solvency II and the International Financial Reporting Standard (IFRS), see, e.g., [88]. This task can be achieved by performing stochastic simulations of ALM models in a risk-neutral environment. Much effort has been spent on the development of such models in the last years, see, e.g., [2, 4, 5, 8, 15, 24, 29, 60, 66, 67, 84, 93, 110, 112, 113, 155, 169] and the references therein. Here, most authors focus on the fair valuation and contract design of unit-linked and participating life insurance policies. Exceptions are [60, 93] where the financial risks and returns of participating policies are analysed under the real world probability measure. Most of the articles in the existing literature (exceptions are [4, 5, 29, 113]) restrict themselves to single-premium contracts and neglect mortality to simplify the presentation or to obtain analytical solutions. However, in the presence of surrender, generalisations which include periodic premiums and mortality risk are not always straightforward, see, e.g., [6].

Here, we develop a general model framework for the ALM of life insurance products. The complexity of the model is chosen such that, on the one hand, most of the models previously proposed in the literature and the most important features of life insurance product management are included. As a consequence, closed-form solutions will only be available in special cases. On the other hand, the model is supposed to remain transparent and modular, and it should be possible to simulate the model efficiently. Therefore, we use a discrete time framework in which all terms can be derived easily and can be computed recursively. We use a stochastic two-factor model to simulate the behaviour of the capital markets, while the development of the biometric parameters is assumed to be deterministic. The asset allocation is dynamic with the goal of keeping a certain percentage of stocks. The bonus declaration mechanism is based on the reserve situation of the company as proposed in [66]. Surrender is modelled and analysed using experience-based surrender tables. Different life insurance product specifics are incorporated via premium, benefit and surrender characteristics in a fairly general framework. In contrast to most of the existing literature, where only the valuation or the development of a single policy is considered, we model the development of a portfolio of policies using model points.

Each model point corresponds to an individual policyholder account or to a pool of similar policyholder accounts which can be used to reduce the computational complexity, in particular in the case of very large insurance portfolios. Thus we can also investigate effects which arise from the pooling of non-homogeneous contracts, as in [69], where the pooling of two contracts is considered.

Overall Model Structure

We model all terms in discrete time. Here, we denote the start of the simulation by time $t = 0$ and the end of the simulation by $t = T$ (in years). The time interval $[0, T]$ is decomposed into K periods $[t_{k-1}, t_k]$ with $t_k = k\,\Delta t$, $k = 1, \ldots, K$ and a period length $\Delta t = T/K$ of one month.[1]

Balance Sheet Model The main focus of our model is on simulating the temporal development of the most important balance sheet items for a portfolio of insurance policies. In this section, we indicate the overall structure of the balance sheet. The determination of the single balance sheet items and the modeling of their future development is the subject of the following sections. The balance sheet items at time t_k, which are used in our model, are shown in Table 6.7.

The asset side consists of the market value C_k of the company's assets at time t_k. On the liability side, the first item is the book value of the actuarial reserve D_k, i.e., the guaranteed savings part of the policyholders after deduction of risk premiums and administrative costs. The second item is the book value of the allocated bonuses B_k which constitute the part of the surpluses that have been credited to the policyholders via the profit participation. The free reserve F_k is a buffer account for future bonus payments. It consists of surpluses which have not yet been credited to the individual policyholder accounts, and is used to smooth capital market oscillations and to achieve a stable and low-volatile return participation of the policyholders. The last item, the equity or company account Q_k, consists of the part of the surpluses which is kept by the shareholders of the company and is defined by

$$Q_k = C_k - D_k - B_k - F_k$$

such that the sum of the assets equals the sum of the liabilities. Similar to the bonus reserve in [66], Q_k is a hybrid determined as the difference between a market value C_k and the three book values D_k, B_k and F_k. It may be interpreted as hidden reserve of the company as discussed in [93].

Similar balance sheets have already been considered in the literature. The sum $M_k = D_k + B_k$ corresponds to the policy reserve in [66], the policyholders' account in [93] or to the customer account in [112]. We prefer to separate the two accounts

[1] Shorter or longer period lengths can be realised in a straightforward way.

in order to thoroughly distinguish the effects of the bonus distribution from the guaranteed benefits. The free reserve F_k and the company account Q_k in our model correspond to the bonus account (also termed undistributed reserve) and to the insurer's account in [26]. These two accounts are sometimes merged into one single account. This, however, is only appropriate if the policyholders are also the owners of the company, see [66].

Performance Figures Typically, stochastic simulations of the balance sheet items are performed for which many different scenarios are generated. To analyse the results of such a simulation then statistical measures are considered which result from an averaging over all scenarios. As a risk measure we consider the path-dependent cumulative probability of default

$$\mathrm{PD}_k = \mathbb{P}\left(\min_{j=1,\dots,k} Q_j < 0\right).$$

In the next sections, we also take a look at the expected values of the balance sheet items $\mathbb{E}[C_k]$, $\mathbb{E}[B_k]$, $\mathbb{E}[F_k]$ and $\mathbb{E}[Q_k]$ for $k = 1, \dots, K$. These profit and risk figures can easily be computed during the simulation. Similarly, it is straightforward to include the computation of further performance measures like the variance, the value-at-risk, the expected shortfall or the return on risk capital.

To determine the sensitivity of a given performance figure f to one of the model parameters v, we compute the partial derivative $f'(v) = \partial f(v)/\partial v$ by a finite difference approximation.[2] For a better comparison, often also the relative change in f to a small change in v is considered, which is given by $f'(v)/f(v)$. For $v = r_0$, this measure of sensitivities is also called effective duration or interest-rate elasticity. For a discussion and further collection of useful risk measures we refer to [15, 29, 84].

To compute these performance figures and to model the future development of the items of the balance sheet, the capital markets, the behaviour of the policyholders and the decisions of the company's management have to be taken into account. Here, we use a stochastic *capital market model*, while all other model components (*asset model*, *management model* and *liability model*) are assumed to be deterministic. We already described the overall structure of the model in Section 6.4. Next, we consider the different modules, see also Figure 6.10, in more detail.

Capital Market Model

We assume that the insurance company invests its capital either in fixed interest assets, i.e., bonds, or in a variable return asset, i.e., a stock or a basket of stocks.

[2] There exist more recent approaches to calculate sensitivities using the Monte Carlo method, like e.g. smoking adjoints [58], likelihood ratio methods [59] or Malliavin calculus [39], see also [41].

The future development of the capital market is specified by a coupled system of two stochastic differential equations, one for the short interest rate and one for the stock price. This system is then discretized with a period length of Δt. The simulation of the model can either be performed under the objective probability measure which is used for risk analyses, see, e.g., [60, 93], and which is the main focus of this work, or under the risk-neutral probability measure, which is appropriate for the fair valuation of embedded options or the identification of fair contract designs, see, e.g., [4, 66, 67, 112].

Continuous Capital Market Model For the modeling of the interest rate environment we use the Cox-Ingersoll-Ross (CIR) model [25]. The CIR model is an one-factor mean-reversion model which specifies the dynamics of the short interest rate $r(t)$ at time t by the stochastic differential equation

$$dr(t) = \kappa(\theta - r(t))dt + \sqrt{r(t)}\sigma_r dW_r(t), \tag{C.1}$$

where $W_r(t)$ is a standard Brownian motion, $\theta > 0$ denotes the mean reversion level, $\kappa > 0$ denotes the reversion rate and $\sigma_r \geq 0$ denotes the volatility of the short rate dynamic. The CIR model has the following appealing properties: First, it produces short interest rates which are always positive if the parameters fulfil the condition $\kappa\theta > \sigma_r^2/2$. In addition, assuming the absence of arbitrage and a market price $\lambda(t, r)$ of interest rate risk of the special form $\lambda(t, r) = \lambda_0\sqrt{r(t)}$ with a parameter $\lambda_0 \in \mathbb{R}$, the short interest rate under the risk-neutral measure follows the same square-root process as in (C.1) but with the parameters $\hat{\kappa} = \kappa + \lambda_0\sigma_r$ and $\hat{\theta} = \kappa\theta/\hat{\kappa}$. Moreover, in the CIR model the prices of arbitrage-free bonds can be derived in closed form, see (C.9).

To model the stock price uncertainty, we assume that the stock price $s(t)$ at time t evolve according to a geometric Brownian motion

$$ds(t) = \mu s(t)dt + \sigma_s s(t)dW_s(t), \tag{C.2}$$

where $\mu \in \mathbb{R}$ denotes the drift rate and $\sigma_s \geq 0$ denotes the volatility of the stock return.[3] By Itô's lemma, the explicit solution of this stochastic differential equation is given by

$$s(t) = s(0)\, e^{(\mu - \sigma_s^2/2)t + \sigma_s W_s(t)}. \tag{C.3}$$

Usually, stock and bond returns are correlated. We thus assume that the two Brownian motions satisfy $dW_s(t)dW_r(t) = \rho dt$ with a constant correlation coefficient $\rho \in [-1, 1]$.

The same two-factor model is used in [29]. In [15, 87], the Vasicek model is employed instead of the CIR model. In [93], stocks and bonds are modelled via a

[3] For a simulation under the risk-neutral measure Q, the drift μ is replaced by $r(t)$.

coupled system of two geometric Brownian motions with different drift and volatility parameters. In [7, 145], more complex jump-diffusion processes and Markov-modulated geometric Brownian motions are employed. The simulation of the latter models is more involved, though. For more detail about these and other models which can be used to simulate the bond and stock prices, we refer to [14, 79, 92].

The capital market parameters $\kappa, \theta, \sigma_r, \mu, \sigma_s$ and ρ can be estimated on the basis of historical data. This way, the objective dynamics of $r(t)$ and $s(t)$ are characterised. The market price of risk parameter λ_0 can then be identified by calibrating the theoretical bond prices (C.9) to observed market prices, see, e.g., [38]. Alternatively, the parameters $\hat{\kappa}, \hat{\theta}, \sigma_r$ and σ_s, which identify the risk neutral measure, can be obtained by calibrating the models (C.1) and (C.2) to observed market prices, see, e.g., [14, 29]. To derive the remaining parameters, estimates of μ, θ and ρ are required. In a more complex model, the constant coefficients in (C.1) could be replaced by time-dependent parameter functions. Then, the model can be fitted to the currently observed term structure of interest rates. However, the bond prices can not be derived analytically anymore, but have to be computed by numerical integration, see [14].

Discrete Capital Market Model In the discrete time case, the short interest rate and the stock prices are defined by $r_k = r(t_k)$ and $s_k = s(t_k)$. For the solution of equation (C.1), we use an Euler-Maruyama discretization[4] with step size Δt, which yields

$$r_k = r_{k-1} + \kappa(\theta - r_{k-1})\Delta t + \sigma_r \sqrt{|r_{k-1}|}\, \Delta W_{r,k}, \tag{C.4}$$

where $\Delta W_{r,k} = W_r(t_k) - W_r(t_{k-1})$ denotes the increment of the Brownian motion in the k-th period. For the stock prices, we use (C.3) and obtain

$$s_k = s_{k-1} \exp\left\{(\mu - \sigma_s^2/2)\Delta t + \sigma_s \Delta W_{s,k}\right\}, \tag{C.5}$$

with $\Delta W_{s,k} = W_s(t_k) - W_s(t_{k-1})$. The increments $\Delta W_{r,k}$ and $\Delta W_{s,k}$ can be generated by $\Delta\, W_{r,k} = \sqrt{\Delta t}\xi_{r,k}$ and $\Delta W_{s,k} = \sqrt{\Delta t}\,(\rho\xi_{r,k} + \sqrt{1-\rho^2}\xi_{s,k})$ with two independent $N(0,1)$-distributed random variables $\xi_{r,k}$ and $\xi_{s,k}$. Since

$$\mathrm{Cov}(\rho\xi_{r,k} + \sqrt{1-\rho^2}\xi_{s,k}, \xi_{r,k}) = \rho,$$

the correlation between the two Wiener processes $W_s(t)$ and $W_r(t)$ is respected. Note that several alternative path generation methods exist that can be used instead of this random walk construction and that these constructions often have better properties with respect to the numerical simulation of the model as we have discussed in Section 5.1. We will compare different approaches in Section C.2.

[4] An alternative to the Euler-Maruyama scheme, which is more time consuming but avoids time discretization errors, is to sample from a noncentral chi-squared distribution, see [59]. The time discretization error is not the focus of this article, though, and we refer to [23]. More information on the numerical solution of stochastic differential equations can be found, e.g., in [59, 94].

Management Model

In this section, we first discuss the allocation of the available capital between stocks and bonds, which determines the portfolio return rate p_k in each period k. Then, the bonus declaration mechanism is illustrated, which defines the interest rate z_k which is given to the policyholders. Finally, the shareholder participation is discussed.

Capital allocation We assume that the company rebalances its assets at the beginning of each period. Thereby, the company aims to have a fixed portion $\beta \in [0, 1]$ of its assets invested in stocks, while the remaining capital is invested in zero coupon bonds with a fixed duration of τ periods.[5] The price of such a bond at time t_k is denoted by $b_k(\tau)$. We assume that no bonds are sold before their maturity.

Let P_k be the premium income of the company at the beginning of period k and let C_{k-1} be the total capital at the end of the previous period. The part N_k of $C_{k-1} + P_k$ which is available for a new investment at the beginning of period k is then given by

$$N_k = C_{k-1} + P_k - \sum_{i=1}^{\tau-1} n_{k-i}\, b_{k-1}(\tau - i),$$

where n_j denotes the number of zero coupon bonds which were bought at the beginning of period j. The capital A_k which is invested in stocks at the beginning of period k is then determined by

$$A_k = \max\{\min\{N_k,\, \beta(C_{k-1} + P_k)\},\, 0\} \tag{C.6}$$

so that the side conditions $0 \leq A_k \leq \beta(C_{k-1} + P_k)$ are satisfied. The remaining money $N_k - A_k$ is used to buy $n_k = (N_k - A_k)/b_{k-1}(\tau)$ zero coupon bonds with duration $\tau\Delta t$. Note that due to long-term investments in bonds it may happen that $N_k < 0$. This case of insufficient liquidity leads to $n_k < 0$ and thus to a short selling of bonds.

Note that we considered an alternative capital allocation strategy in Section 6.4. There, we used

$$A_k = \max\{\min\{N_k,\, \beta F_{k-1}\},\, 0\} \tag{C.7}$$

with $\beta \in \mathbb{R}^+$, so that the proportion of funds invested in stocks is linked to the current amount of reserves. This implements a CPPI capital allocation strategy, see [113]. A comparison of these two strategies can be found in [53].

Bonus declaration Due to regulatory requirements, companies can only guarantee a relatively low interest rate to their policyholders. As a compensation, policyholders are usually entitled to additional variable bonus payments, which are periodically

[5] A slightly more general trading strategy is discussed in [29].

credited to the policyholders' accounts and allow the policyholders to participate in the investment returns of the company (contribution principle). The exact specification of this reversionary bonus varies from one insurance product to another and often depends on the financial situation of the company as well as on strategical considerations and legal requirements. The bonus is declared annually by the company, with the goal to provide a low-volatile, stable and competitive return participation (average interest principle). The company builds up reserves in years of good returns, which are used to keep the bonus stable in years of low returns. Thereby, a high and thus competitive declaration has to be balanced with a solid financial strength of the company. Various mathematical models for the declaration mechanism are discussed in the literature, see, e.g., [4, 8, 26, 60, 66, 93, 112]. In [60, 66, 93], the bonus interest rate for the next year is already declared at the beginning of this year (principle of advance declaration) as it is required in some countries, e.g., Germany, by regulation. The declaration is based on the current reserve rate γ_{k-1} of the company, which is defined in our framework by the ratio of the free reserve to the allocated liabilities, i.e.,

$$\gamma_{k-1} = \frac{F_{k-1}}{D_{k-1} + B_{k-1}}.$$

In this work, we follow the approach proposed in [66] where the annual interest rate is defined by

$$\hat{z}_k = \max\{\hat{z}, \omega(\gamma_{k-1} - \gamma)\}.$$

Here, $\hat{z}$ denotes the annual guaranteed interest rate, $\gamma \geq 0$ the target reserve rate of the company and $\omega \in [0, 1]$ the distribution ratio or participation coefficient which determines how fast excessive reserves are reduced. Typical values are $\omega \in [0.2, 0.3]$ and $\gamma \in [0.1, 0.4]$. This way, a fixed fraction of the excessive reserve is, in accordance with the average interest rate principle, distributed to the policyholders in case of a satisfactorily large reserve. If the reserve rate γ_{k-1} is below the target reserve rate γ, only the guaranteed interest is paid, see [66] for details.

In our model, the bonus is declared annually, always at the beginning of the first period of each year. In case of a monthly discretization, this bonus has to be converted into a monthly interest

$$z_k = \begin{cases} (1 + \hat{z}_k)^{1/12} - 1 & \text{if } k \bmod 12 = 1 \\ z_{k-1} & \text{otherwise} \end{cases}$$

which is given to the policyholders in each period k of this year.

Shareholder participation Excess returns $p_k - z_k$, conservative biometry and cost assumptions as well as surrender fees lead to a surplus G_k in each period k which has to be divided among the two accounts free reserve F_k and equity Q_k. In case

of a positive surplus, we assume that a fixed percentage $\alpha \in [0,1]$ is saved in the free reserve while the remaining part is added to the equity account. Here, a typical assumption is a distribution according to the 90/10-rule which corresponds to the case $\alpha = 0.9$. If the surplus is negative, we assume that the required capital is taken from the free reserve. If the free reserves do not suffice, the company account has to cover the remaining deficit. The free reserve is then defined by

$$F_k = \max\{F_{k-1} + \min\{G_k, \alpha\, G_k\},\, 0\}. \tag{C.8}$$

The exact specification of the surplus G_k and the development of the equity Q_k is derived in the liability model.

Asset Model

In this section, market prices of the different asset classes, the portfolio return rate p_k and the development of the capital C_k in the k-th time step of the model are derived.

Bond prices In the CIR model, the price $b(t,\tau)$ at time t of a zero coupon bond with maturity at time $t + \tau\Delta t$ can be derived in closed form by

$$b(t,\tau) = A(\tau)\, e^{-B(\tau)\, r(t)} \tag{C.9}$$

as an exponential affine function of the prevailing short interest rate $r(t)$ with

$$A(\tau) = \left(\frac{2h e^{(\hat\kappa + h)\tau\Delta t/2}}{2h + (\hat\kappa + h)(e^{h\tau\Delta t} - 1)} \right)^{2\kappa\theta/\sigma_r^2}, \quad B(\tau) = \frac{2(e^{h\tau\Delta t} - 1)}{2h + (\hat\kappa + h)(e^{h\tau\Delta t} - 1)},$$

and $h = \sqrt{\hat\kappa^2 + 2\sigma_r^2}$. In the discrete time case, we then define $b_k(\tau) = b(t_k, \tau)$.

Portfolio return rate As a result of the capital allocation, at the beginning of period k, the capital A_k from (C.6) is invested into stocks with value s_{k-1}. In addition, the company owns n_{k-i} bonds with maturity $\tau - i$ and value $b_{k-1}(\tau - i)$ for each $i = 1, \ldots, \tau - 1$. The portfolio return rate p_k in period k resulting from the above allocation procedure is then determined by

$$p_k = \left(\Delta A_k + \sum_{i=0}^{\tau-1} n_{k-i}\, \Delta b_{k,i} \right) / (C_{k-1} + P_k), \tag{C.10}$$

where $\Delta A_k = A_k(s_k/s_{k-1} - 1)$ and $\Delta b_{k,i} = b_k(\tau - i - 1) - b_{k-1}(\tau - i)$ denote the changes of the market values of the stock and of the bond investments from the beginning to the end of period k, respectively.

Projection of the assets Let P_k denote the premium, which is obtained by the company at the beginning of period k, and let E_k, T_k and S_k denote the survival, the death, and the surrender payments to policyholders, which take place at the end of period k. These cashflows are defined in the liability model. The capital C_k is then recursively given by

$$C_k = (C_{k-1} + P_k)(1 + p_k) - E_k - T_k - S_k. \tag{C.11}$$

Liability Model

In this section, we first discuss the modeling of the decrement of policies due to mortality and surrender. Then, we derive the development of the policyholder's accounts and, finally, the development of all items on the liability side of the balance sheet.

Decrement model For efficiency, the portfolio of all insurance contracts is often represented by a reduced number m of model points. Each model point then represents a group of policies which are similar with respect to cash flows and technical reserves, see, e.g., [84]. A model point then contains averaged values for all criteria like the entry and maturity time, the age and the gender of the policyholders.

Let q_k^i denote the probability that a policyholder of model point i dies in the k-th period. In the following, we assume that this probability is given deterministically. This is motivated by the fact that the systematic development of mortality can be predicted much more accurately than, e.g., the development of the capital markets. Moreover, unsystematic mortality risk can be controlled by means of portfolio diversification. The probabilities q_k^i typically depend on the age, the year of birth and the gender of the policyholder. They are collected and regularly updated by the insurance companies. In the following, we always assume that mortality occurs deterministically in accordance with the actuarial assumptions.

Most insurance policies include surrender options for the policyholder. Usually, two different approaches for the valuation of these rights are distinguished, see [6]. Either surrender is considered as an exogenously determined event and modelled like death using experience-based decrement tables, or it is assumed that surrender options are rationally exercised by policyholders. While the second approach is extensively investigated in the literature, see, e.g., [2, 5, 66], only very few publications (see [80] and the references therein) investigate the effects of exogenously given surrender decisions. In this work we assume that the probabilities u_k^i that a policyholder of model point i surrenders in the k-th period are exogenously given. This is the appropriate approach if surrender decisions are mainly driven by the personal consumption plans of the policyholders, see, e.g., [29]. Here, a typical assumption is that the probabilities u_k^i only depend on the elapsed contract time. As in [80],

we assume that the time t until surrender follows an exponential distribution with exponent $\lambda = 0.03$. The probabilities of surrender are then given by

$$u_k^i = 1 - e^{-\lambda \Delta t}. \tag{C.12}$$

Since the probabilities of surrender are, in contrast to the probabilities of death, not included into the actuarial premium calculation, the effects of surrender on the success of the company significantly differ from the effects due to mortality.

Let δ_k^i denote the expected number of contracts in model point i at the end of period k. This number evolves over time according to

$$\delta_k^i = \left(1 - q_k^i - u_k^i\right) \delta_{k-1}^i. \tag{C.13}$$

By pooling, all contracts of a model point expire at the same time. In the simulation the model point is then simply dissolved. In this work, we do not consider the evolution of new contracts during the simulation.

Insurance products Next, the guaranteed part and the bonus part of the insurance products are specified. We always assume that premiums are paid at the beginning of a period while benefits are paid at the end of the period. Furthermore, we assume that all administrative costs are already included in the premium.

For each model point $i = 1, \ldots, m$, the guaranteed part of the insurance product is defined by the specification of the following four characteristics:

- *premium characteristic:* $(P_1^i, \ldots, P_K^i)$ where P_k^i denotes the premium of an insurance holder in model point i at the beginning of period k if he is still alive at that time.
- *survival benefit characteristic:* $(E_1^{i,G}, \ldots, E_K^{i,G})$ where $E_k^{i,G}$ denotes the guaranteed payments to an insurance holder in model point i at the end of period k if he survives period k.
- *death benefit characteristic:* $(T_1^{i,G}, \ldots, T_K^{i,G})$ where $T_k^{i,G}$ denotes the guaranteed payment to an insurance holder in model point i at the end of period k if he dies in period k.
- *surrender characteristic:* $(S_1^{i,G}, \ldots, S_K^{i,G})$ where $S_k^{i,G}$ denotes the guaranteed payment to an insurance holder in model point i at the end of period k if he surrenders in period k.

Here, K denotes the last period of the simulation. The characteristics, which can be different for each model point, are usually derived from an insurance tariff which contains the functional interrelations between premium, benefit, death and surrender characteristics.

Given the benefit characteristics of a product, the premiums are determined by the equivalence principle which states that the present value of the death and survival benefits must equal the present value of the premium at the start of the insurance, see, e.g., [4, 168]. Here, the present values are computed according to a given technical interest rate z using the traditional actuarial approach. Typically, z is fixed at the start of the contract and constitutes an interest rate guarantee. We assume here that z is equal for all contracts.

The actuarial reserve D_k^i of an insurance contract at each point in time is defined as the difference of the present value of the expected future death and survival benefits and the present value of the expected future premiums which are calculated according to the actuarial assumptions. An efficient computation of the actuarial reserve D_k^i of model point i at the end of period k is possible by using the recursion

$$D_k^i = \frac{1+z}{1-q_k^i}(D_{k-1}^i + P_k^i) - E_k^{i,G} - \frac{q_k^i}{1-q_k^i}T_k^{i,G}. \tag{C.14}$$

Multiplication by $(1-q_k^i)$ shows that this equation results from the fact that the expected actuarial reserve $(1-q_k^i)D_k^i$ at the end of period k is given by the sum of the actuarial reserve of the previous period and the premium after guaranteed interest $(1+z)(D_{k-1}^i+P_k^i)$ minus the expected survival and death benefits $(1-q_k^i)E_k^{i,G}+q_k^i T_k^{i,G}$, see, e.g., [4, 168].

In addition to the guaranteed benefits which depend on the technical interest rate z, policyholders are also entitled to a bonus interest $z_k - z$ defined in Section C.1, which depends on the development of the financial markets. Depending on the specific insurance product, the allocated bonuses are distributed to each policyholder either at maturity of his contract or earlier in case of death or surrender. Let $E_k^{i,B}$, $T_k^{i,B}$ and $S_k^{i,B}$ denote the bonus payments to an insurance holder in model point i at the end of period k in case of survival, death and surrender, respectively. The sum of all bonuses allocated to a policyholder of model point i at the end of period k is collected in an individual bonus account B_k^i. Its value is recursively defined by

$$B_k^i = \frac{1+z_k}{1-q_k^i}B_{k-1}^i + \frac{z_k - z}{1-q_k^i}(D_{k-1}^i + P_k^i) - E_k^{i,B} - \frac{q_k^i}{1-q_k^i}T_k^{i,B}. \tag{C.15}$$

Similar to above, this equation results from the fact that the expected value[6] $(1-q_k^i)B_k^i$ of the bonus account at the end of period k is given by the sum of allocated bonuses in the past after interest $(1+z_k)B_{k-1}^i$ and the bonus payment $(z_k-z)(D_{k-1}^i+P_k^i)$ of period k minus the expected bonus payments $(1-q_k^i)E_k^{i,G}+q_k^i T_k^{i,G}$ in case of survival and death, respectively.

[6] Note that B_k^i still depends on financial uncertainty. The expected value is only taken with respect to the death probabilities.

The total payments E_k^i, T_k^i and S_k^i to a policyholder of model point i at the end of period k in case of survival, death and surrender are then given by

$$E_k^i = E_k^{i,G} + E_k^{i,B}, \quad T_k^i = T_k^{i,G} + T_k^{i,B} \text{ and } S_k^i = S_k^{i,G} + S_k^{i,B}. \tag{C.16}$$

By adding (C.14) and (C.15), we see that the sum $M_k^i = D_k^i + B_k^i$ of the two policyholder accounts satisfies

$$M_k^i = \frac{1+z_k}{1-q_k^i}(M_{k-1}^i + P_k^i) - E_k^i - \frac{q_k^i}{1-q_k^i} T_k^i \tag{C.17}$$

which has a similar structure as the recursion for the actuarial reserve (C.14).

Example C.1 (Sample Characteristics). For illustration, we consider the endowment insurance with death benefit and constant premium payments from Example 6.9. The guaranteed components of the four characteristics are then defined by

$$P_k^i = P^i, \quad E_k^{i,G} = \chi_k(d^i)\, E^{i,G}, \quad T_k^{i,G} = k\, P^i \text{ and } S_k^{i,G} = \vartheta D_k^i,$$

where $\chi_k(d^i)$ denotes the indicator function which is one if $k = d^i$ and zero otherwise. The bonus payments at the end of period k are given by

$$E_k^{i,B} = \chi_k(d^i) B_k^i, \quad T_k^{i,B} = B_k^i \text{ and } S_k^{i,B} = \vartheta\, B_k^i.$$

We used this sample characteristics in our numerical experiments in Section 6.4.2 and Section C.2.

Projection of the liabilities We next determine the cash flows which are occurring to and from the policyholders in our model framework. The premium P_k, which is obtained by the company at the beginning of period k, the survival payments E_k, the death payments T_k, and the surrender payments S_k to policyholders, which take place at the end of period k, are obtained by summation of the individual cash flows (C.16), i.e.,

$$P_k = \sum_{i=1}^m \delta_{k-1}^i P_k^i, \; E_k = \sum_{i=1}^m \delta_k^i E_k^i, \; T_k = \sum_{i=1}^m q_k^i \delta_{k-1}^i T_k^i, \; S_k = \sum_{i=1}^m u_k^i \delta_{k-1}^i S_k^i, \tag{C.18}$$

where the numbers δ_k^i are given by (C.13) and where m denotes the number of model points. The actuarial reserve D_k and the allocated bonus B_k are derived similarly by summation of the individual policyholder accounts (C.14) and (C.15), i.e.,

$$D_k = \sum_{i=1}^m \delta_k^i D_k^i \text{ and } B_k = \sum_{i=1}^m \delta_k^i B_k^i.$$

From the equations (C.13), (C.17) and (C.18), we derive that the sum $M_k = \sum_i \delta_k^i M_k^i = D_k + B_k$ is recursively given by

$$M_k = (1 + z_k)(M_{k-1} + P_k) - E_k - T_k - S_k/\vartheta, \tag{C.19}$$

where we assumed $S_k^{i,G} = \vartheta D_k^i$ as in Example 6.9 and $\vartheta > 0$.

To define the free reserve F_k, we next determine the gross surplus G_k in period k. By the so-called contribution formula, the surplus is usually divided into the components interest surplus, risk surplus, cost surplus and surrender surplus. In our model only the interest surplus and the surrender surplus show up. The interest surplus is given by the difference between the total capital market return $p_k\,(F_{k-1} + M_{k-1} + P_k)$ on policyholder capital and the interest payments $z_k\,(M_{k-1} + P_k)$ to policyholders. The surrender surplus is given by $S_k/\vartheta - S_k$. The gross surplus in period k is thus obtained by

$$G_k = p_k\,F_{k-1} + (p_k - z_k)\,(M_{k-1} + P_k) + (1/\vartheta - 1)S_k.$$

The free reserve F_k is then derived using equation (C.8). Altogether, the company account Q_k is determined by

$$Q_k = C_k - D_k - B_k - F_k. \tag{C.20}$$

For convenience, the most important model parameters and equations are summarised in Figure 6.11. Note that the cash flows and all balance sheet items are expected values with respect to our deterministic mortality and surrender assumptions, but random numbers with respect to our stochastic capital market model.

Note that our model framework includes many of the models previously proposed in the literature as special cases. If we consider only one policy with a single initial payment P^i at time $t = 0$, a term of K years, a yearly discretization and neglect mortality and surrender, then the non-zero entries in the characteristics of this (pure savings) product are defined by $P_1^i = P^i$ and $E_K^{i,G} = (1+z)^K P^i$. Moreover, by (C.19) it holds that

$$M_k^i = (1 + z_k)M_{k-1}^i, \quad M_0^i = P^i \ \text{ for } k = 1, \ldots, K.$$

If we further replace (C.8) by $F_k = F_{k-1} + G_k$, we exactly recover the model for European-type participation contracts proposed in [66]. If we include mortality, define the guaranteed death benefit by $T_k^{i,G} = (1+z)^k P^i$ and declare the policyholder interest according to $z_k = \max\{\omega\, p_k, z\}$, we further recover the model for single premium contracts proposed in [4]. The definition $P_k^i = P^i$ for all $k = 1, \ldots, K$ leads to the constant premium case in [5] but without the linear approximation for the benefit adjustment.

In [52, 53], we performed a sensitivity analysis of the presented ALM model to show that the model captures the most important behaviour patterns of the balance sheet development of participating life insurance products, that it incorporates effects such as solvency risks, liquidity risks and diversification effects and that it can be used to analyse the impact of mortality and surrender and the influence of different capital allocation strategies and of different bonus declaration mechanisms.

C.2 Numerical Analysis

In this section, we determine the variances, the smoothness and the effective dimensions of different parameter setups and show how these properties relate to the convergence behaviour of MC, QMC and SG methods.

Parameter Setups

As remarked in Section 6.4.1, the efficiency of numerical integration methods depends on several mathematical properties of the integrand, such as its variance, its effective dimension and its smoothness, all of which are affected by the choice of the input parameters of the ALM model. To investigate the numerical issues which arise from different choices of parameters and the influence of different model components, we consider the following simple basic setup 1 and several extensions 2 – 10. Thereby, in each of the extensions, either one additional feature is added to the basic setup or one particular component of the basic setup is replaced by a different one. In setup 11, we then consider the combination of several extensions. All setups are special cases of the ALM model in Figure 6.11 and result from different choices of the input parameters.

1) Basic model We start with a basic model which corresponds to the model from [66] for European-type participation contracts with conservative bonus declaration. Thereby, we consider a homogeneous portfolio of $50,000$ pure savings policies, which are exactly represented by one model point. The policyholders of this model point are assumed to be male and of age 42. Their contracts started at the age 36 and mature at age 62. The policyholders pay a constant monthly premium $P_k^i = 50$ and receive the guaranteed interest $z = 3\%$. The assets follow Equation C2 (Figure 6.11) with parameters $\mu = 5\%$ and $\sigma_s = 10\%$. This case is represented in the model framework by setting $m = 1$, $\beta = 1$, $\omega = 0$ and $q_k^i \equiv u_k^i \equiv 0$. We assume that the two accounts F_k and Q_k are merged into one single account, also denoted by Q_k, which is appropriate if the policyholders are also the owners of the company, see [66], and which corresponds to the case $\alpha = 1$. We further assume that the policies have not received any bonus payments before the start of the simulation, i.e., $B_0^i = 0$. It is finally assumed that the total initial reserves of the company are

given by $Q_0 = \gamma_0 D_0$ with $\gamma_0 = 0.1$. We always choose a period length of $\Delta t = 1/12$ years, but consider different numbers of periods in the simulation.

2) Mortality and surrender As a representative for a more complex insurance product, we consider the endowment insurance with death benefit and surrender option from Example 6.9. We take the probabilities q_k^i of death from the DAV 2004R mortality table and choose exponentially distributed surrender probabilities $u_k^i = 1 - e^{-0.03\Delta t}$.

3) Non-homogeneous portfolio We here consider a more complex representative model portfolio with 500 different model points as described in Example 6.8.

4) High volatility To illustrate the effect of more volatile capital markets, we here choose a volatility of $\sigma_s = 30\%$ instead of 10%.

5) and 6 Moderate and aggressive bonus payments To illustrate the effect of the bonus declaration mechanism, we choose $\omega = 25\%$ and $\omega = 100\%$, which correspond to the neutral and aggressive scenarios in [66], respectively. The target reserve rate is assumed to be $\gamma = 15\%$.

7) Shareholder participation In this setup, we choose $\alpha = 0.9$ which corresponds to a distribution of the surplus between free reserve F_k and equity Q_k according to the 90/10-rule. We assume $Q_0 = 0$ which means that the shareholders will not make additional payments to the company to avoid a ruin. This way, $\mathbb{E}[Q_K]$ serves as a direct measure for the investment returns of the shareholders in the time interval $[0, t_K]$. The initial reserves of the company are collected in the free reserve, i.e., $F_0 = 0.1 D_0$.

8) CIR model In this setup, we assume that the capital is only invested into bonds (i.e. $\beta = 0$) with a duration of $\tau = 1/12$ years. The short interest rates follow Equation C1 (Figure 6.11), where we use the parameters $\kappa = 0.1$, $\theta = 4\%$ and $\sigma_r = 5\%$. The bond prices then result from Equation A1. The terms $A(\tau)$ and $B(\tau)$ in this equation involve the market price of interest rate risk which we assume to be $\lambda_0 = -5\%$. At time t_0, we assume a uniform bond allocation, i.e., $n_j = (1-\beta)C_0 / \sum_{i=0}^{\tau-1} b_0(i)$ for $j = 1-\tau, \ldots, 0$.

9) CIR + GBM As a representative for a more complex capital market model, we consider a correlated system of geometric Brownian motion and square-root diffusion as in Equation C1 and C2 (Figure 6.11) with a correlation of $\rho = -0.1$. The parameters of the geometric Brownian motion and of the CIR model are as

above, but with $\sigma_s = 5\%$. The capital allocation is performed with the target stock ratio $\beta = 10\%$.

10) CPPI strategy In this setup, we again use the correlated system of geometric Brownian motion and square-root diffusion as in the previous setup. In contrast to setup 9, we here replace $\beta(C_{k-1} + P_k)$ in Equation M2 by βF_{k-1} with $\beta = 1$ such that the proportion of funds invested in stocks is linked to the current amount of reserves. This implements a CPPI (constant proportion portfolio insurance) capital allocation strategy.

11) Compound model We finally consider the simulation of a more complex ALM model which is obtained by a combination of the setups 1 – 3, 5 and 10. It models the development of a portfolio of endowment insurances with death benefit and incorporates the surrender of contracts, a reserve-dependent bonus declaration, a dynamic asset allocation and a two-factor stochastic capital market. We used this model also for our numerical experiments in Section 6.4.2.

Properties of the Model Components

Here, we merely consider results which have an impact on the performance of the numerical methods. An detailed assessment of the influence of the different model parameters on the expected value of the equity account and on other performance figures such as, e.g., the default probabilities, can be found in [52].

We in particular focus on the distribution of the equity account Q_K at time t_K in the setups 1 – 11. We compute the expected value of Q_K and the variance of Q_K for all setups by numerical integration on a very fine simulation grid with 2^{20} QMC sample points.[7] The results for $K = 16$ and $K = 128$ time steps are shown in Table C.1(a) and Table C.1(b), respectively.

One can see that the small volatility and the mean-reverting property of the short interest rates in the CIR model result in rather small variances in the setups 8 – 11. The by far highest variance arises in the setup 4. A comparison of the cases $K = 16$ and $K = 128$ shows that longer time horizons with more periods lead to higher variances in all considered setups as expected. Setup 11 with $K = 128$ is of striking small variance as it combines several components which reduce the variance of the basic setup such as the decrement of contracts (setup 2), the bonus declaration (setup 5) and the CIR model (setup 10).

[7] For the setups 1 – 4 these values can also be derived analytically. For setup 1 and setup 4 one obtains $\mathbb{E}[Q_K] = \sum_{k=1}^{K} P_k^i \left(\exp\{\mu(K-k+1)\Delta t\} - P_k^i(1+z)^{K-k+1}\right) + \exp\{\mu K \Delta t\} C_0 - (1+z)^K D_0$, which we used as a first test to validate the correctness of the implementation and the accuracy of the numerical methods. The closed-form solution of a similar model but with bonus payments can be found in [4].

Note further that the integrands which correspond to the different setups not only differ in their variance but are also contained in different smoothness classes. This is indicated in Table C.1(a) and Table C.1(b) by the smoothness parameter r, which denotes the maximum number of continuous derivatives of the equity Q_K as function of the vector $\mathbf{Y} \in \mathbb{R}^d$. While Q_K is a C^∞-smooth function in the setups 1–4, it is only a C^0-smooth function in the other cases. This loss of regularity results from the maximum and minimum operators in the management rules M2, M4 and M5 for the capital allocation, the bonus declaration and the shareholder participation and from the absolute value in the model equation C1.

In addition to the expected values and variances, we also computed the (classical) truncation dimension (2.8) of the function Q_K for all setups. We thereby used the Brownian bridge construction for the stock prices and short interest rates. The path generation does not affect the distribution or smoothness of Q_K, but has a significant influence on the truncation dimension as we already showed in Section 6.4.2. One can see in Table C.1(a) and in Table C.1(b) that the truncation dimensions d_t are in all cases significantly smaller than the nominal dimensions d. In the setups 9 – 11 with $K = 128$, the nominal dimension is $d = 256$, while the truncation dimension is only $d_t = 15$. The highest truncation dimension appears in setup 6 where we have $d = 128$ and $d_t = 23$. In the setups 1 – 4 and 8, we observe that the truncation dimension is almost independent of the nominal dimension. It is in all cases smaller than eight and even only one in the setups 1 – 4. In setup 11 one can finally see that the combination of several extensions does not necessarily increase the effective dimension. It may even be smaller than the maximum effective dimension of the individual components. This indicates that also significantly more complex ALM models might be of low effective dimension.

Further numerical computations show that the considered ALM model problems are also of very low effective dimension d_s in the superposition sense. For $d = 32$ we obtain for almost all setups that the integral (6.19) is 'nearly' additive, i.e. $d_s = 1$, independent of d and independent of the chosen construction of the capital market paths.[8] Only for setup 7 we get the superposition $d_s = 2$.

In summary, one can see that the use of the different model components leads to comparably small changes in the expected value but it significantly affects the variance, the smoothness and the effective dimension of the function Q_K. Interestingly, all of our model problems led to integrands which are of low effective dimension in the superposition sense. They are also of rather low effective dimension in the truncation sense if the BB path construction is used. We conjecture that these properties might also hold for more complex ALM models in life insurance as the ones considered here.

[8] It is more difficult and expensive to compute the superposition than the truncation dimension so that we here have to restrict the maximum dimension to $d = 32$.

Impact of the Model Components

As in Section 6.4.2, we now apply MC simulation, QMC integration based on Sobol point sets and dimension-adaptive SG integration based on the Gauss-Hermite rule to compute the integral (6.19) for all eleven setups.

To demonstrate the impact of the nominal dimension d on the performance of the numerical methods, we again distinguish the two cases $K = 16$ and $K = 128$. They correspond to integrals, where the nominal dimension ranges from 16 to 256, see Table C.1(a) and Table C.1(b). Here, we again use the Brownian bridge path construction for the stock prices and short interest rates to obtain low effective dimensions avoiding the additional computational costs of the PCA and EVD constructions. To measure the accuracy of the three different numerical approaches we proceeded as follows: We approximate the integral (6.19) with $n = 1, 2, 4, \ldots, 2^{18}$ MC and QMC samples. As the considered SG method determines the required number of nodes automatically, we here successively refine the approximation until the grid size exceeded 2^{18} nodes. By a comparison of these results with reference solutions, the convergence rates and the error constants of the numerical methods are then computed using a least square fit. To eliminate the influence of the initial seed in the MC method, we show the average convergence rates and constants which are obtained after twenty independent runs of the MC method with different seeds.

For the eleven setups and the three numerical methods, the convergence rates and the constants are displayed in Table C.1(a) and Table C.1(b) for the cases $K = 16$ and $K = 128$, respectively.

One can see that the MC method on average converges with the rate $1/2$ independently of the selection of the model parameters and of the number of time steps while the average constant of the approximation significantly varies from setup to setup. This is explained by the different variances σ of the considered setups and the fact that the expected MC error is proportional to the ratio $\sigma/\sqrt{n}$. For instance, the constants in the setups 8 – 11, which are of comparably small variance, are considerably smaller than, e.g., the constant in setup 4 which is of particular high variance. This means that also the number of scenarios, which have to be simulated to obtain a prescribed precision, varies from setup to setup according to its variance. Computational results not displayed in Table C.1(a) and Table C.1(b) show that with 500 scenarios on average only one digit accuracy, i.e., a relative error of about 10%, can be expected.[9] For two digits precision, about $20,000$ sample points are required on average. Three digits accuracy are only attained in very few cases with the considered maximum number of sample points of 2^{18}.

One can further see that the QMC method outperforms MC simulation in all setups. It converges faster with a convergence rate of 0.7 – 0.8 and has also smaller

[9] Note that we show the average results of 20 independent MC runs. Single runs of the MC method may be much better as well as much worse than the reported results.

Table C.1. Expected values $\mathbb{E}$, variances σ, smoothness parameters r, nominal dimensions d and effective dimensions d_t of the equity Q_K for the eleven setups. Moreover, convergence rates (rate) and constants (const) of the MC, QMC and SG method.

(a) $K = 16$ periods

no.	1	2	3	4	5	6	7	8	9	10	11
sample setup	basic setup	mort. surr.	non-homog.	high vola.	moder. bonus	aggr. bonus	share-holder	CIR model	GBM+ CIR	CPPI strat.	comp. model
$\mathbb{E}$	5.9	6.1	6.6	5.9	5.8	5.6	5.8	4.6	4.9	4.7	5.7
σ	38	37	46	359	36	33	22	0.1	0.4	0.2	0.5
r	∞	∞	∞	∞	0	0	0	0	0	0	0
d	16	16	16	16	16	16	16	16	32	32	32
d_t	1	1	1	1	1	1	4	7	7	9	7
MC method											
rate	0.52	0.49	0.45	0.5	0.41	0.49	0.53	0.49	0.52	0.51	0.49
const	0.63	0.73	0.52	1.88	0.32	0.64	0.59	0.05	0.1	0.08	0.1
QMC method											
rate	0.77	0.8	0.75	0.8	0.79	0.7	0.77	0.71	0.8	0.8	0.79
const	0.5	0.53	0.45	2.26	0.5	0.25	0.64	0.02	0.05	0.05	0.04
SG method											
rate	3.01	2.33	2.37	2.26	0.35	-0.2	0.09	1.22	1.31	1.63	1.62
const	0.52	0.07	0.1	3.4	0.01	2e-3	0.11	2e-4	4e-3	0.01	0.01

(b) $K = 128$ periods

no.	1	2	3	4	5	6	7	8	9	10	11
sample setup	basic setup	mort. surr.	non-homog.	high vola.	moder. bonus	aggr. bonus	share-holder	CIR model	GBM+ CIR	CPPI strat.	comp. model
$\mathbb{E}$	26.5	26	22.8	26.5	14.2	6.6	24.2	10.9	15.3	12.7	10.2
σ	1233	937	791	16627	596	437	490	155	138	160	17
r	∞	∞	∞	∞	0	0	0	0	0	0	0
d	128	128	128	128	128	128	128	128	256	256	256
d_t	1	1	1	1	8	23	16	8	15	15	15
MC method											
rate	0.5	0.51	0.53	0.48	0.51	0.51	0.5	0.48	0.48	0.52	0.49
const	0.79	0.84	1.01	2.45	1.42	2	0.57	0.64	0.42	0.7	0.22
QMC method											
rate	0.81	0.8	0.71	0.69	0.81	0.71	0.72	0.73	0.75	0.75	0.78
const	0.93	0.82	0.6	3.08	0.6	0.93	0.54	0.5	0.44	0.44	0.14
SG method											
rate	1.65	1.64	1.56	1.5	0.14	0.21	0.1	0.63	0.29	0.33	0.15
const	2.5	1.58	0.99	70.7	0.28	3.31	1.36	0.07	0.37	0.04	0.01

or comparably large error constants. It is therefore also more accurate than MC even for small sample sizes. On average it suffices to generate about 100 scenarios for one digit, $1,000$ scenarios for two digits and $10,000$ scenarios for three digits precision, i.e., low accuracy requirements of two digits precision are obtained by the QMC method about twenty times faster than by MC simulation. For higher accuracy requirements, the advantage of the QMC method is even more pronounced. Furthermore, an inspection of the high-dimensional case $K = 128$ shows that the QMC convergence rate as well as the constant of the approximation is almost independent of the dimension. These results can not be explained by the classical QMC convergence theory but by the low effective dimension of the ALM problems. We further see that the fast convergence behaviour of the QMC method is not affected by the smoothness parameter r. This shows on the other hand that the QMC method can hardly profit from setups with a high degree of smoothness $r = \infty$.

One can finally see that the performance of the SG method varies significantly from setup to setup. We observe that it is the by far most efficient method with a very high convergence rate of up to three for all setups which lead to smooth integrands with $r = \infty$. In the moderately high dimensional case $K = 16$ also the constants (except in setup 4) are clearly lower than the constants of the MC and QMC approximations. In these cases three digits precision (and more) are already attained with only about 50 points if $r = \infty$. The convergence rates and the constants slightly deteriorate with rising dimensions, though, showing that the curse of dimension can not be completely avoided by the SG method. Higher variances seem not to affect the convergence rates of the SG method but lead to increasing constants of the approximation, see, e.g., setup 4. With respect to the smoothness of the integrands, we see that the low degree of regularity has a much more pronounced impact on the SG convergence in the setups $5 - 7$ than in the setups $8 - 11$. This is explained by the fact that in the setups $8 - 11$ the arising maximum and minimum operators in the model equation M2 and the absolute value in the model equation C1 only apply in very rare cases (e.g. if the discrete version of the CIR model produces negative interest rates). In the setups $5 - 7$, the non-smooth model equations are of higher importance such that we consequently observe only a very slow or even no convergence. To ensure a satisfactory performance of the SG method in these cases, the smoothness must first be recovered by suitable smoothing transformations (e.g., by a smoothing of maximum and minimum operators or by a decomposition of the integration domain into domains where the integrand is smooth).

Bibliography

[1] P. Acworth, M. Broadie, and P. Glasserman, *A comparison of some Monte Carlo and quasi-Monte Carlo methods for option pricing*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, P. Hellekalek and H. Niederreiter, eds., Springer, 1998, pp. 1–18.

[2] M. Albizzati and H. Geman, *Interest rate risk management and valuation of the surrender option in life insurance policies*, J. Risk and Insurance, 61 (1994), pp. 616–637.

[3] N. Aronzaijn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68 (1950), pp. 337–404.

[4] A. Bacinello, *Fair pricing of life insurance participating contracts with a minimum interest rate guaranteed*, Astin Bulletin, 31 (2001), pp. 275–297.

[5] ——, *Pricing guaranteed life insurance participating policies with annual premiums and surrender option*, Astin Bulletin, 7 (2003), pp. 1–17.

[6] ——, *Modelling the surrender conditions in equity-linked life insurance*, Insurance: Mathematics and Economics, 37 (2005), pp. 270–296.

[7] L. Ballotta, *A Lévy process-based framework for the fair valuation of participating life insurance contracts*, Insurance: Mathematics and Economics, 37 (2005), pp. 173–196.

[8] L. Ballotta, S. Haberman, and N. Wang, *Guarantees in with-profit and unitized with-profit life insurance contracts: Fair valuation problem in presence of the default option*, Insurance: Mathematics and Economics, 73 (2006), pp. 97–121.

[9] R. Bellmann, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[10] J. Bernsten, T. Espelid, and A. Genz, *Algorithm 698: DCUHRE – an adaptive multidimensional integration routine for a vector of integrals*, ACM Transactions on Mathematical Software, 17 (1991), pp. 452–456.

[11] T. Bonk, *A new algorithm for multi-dimensional adaptive numerical quadrature*, in Adaptive Methods – Algorithms, Theory, and Applications, NNFM 46, W. Hackbusch and G. Wittum, eds., Vieweg, 1994, pp. 54–68.

[12] P. Boyle, *Options: A Monte Carlo approach*, J. Financial Economics, 4 (1977), pp. 323–338.

[13] P. Boyle, M. Broadie, and P. Glasserman, *Monte Carlo methods for security pricing*, J. Economic Dynamics and Control, 21 (1997), pp. 1267–1321.

[14] D. Brigo and F. Mercurio, *Interest Rate Models - Theory and Practice*, Springer, 2001.

[15] E. Briys and F. Varenne, *On the risk of life insurance liabilities: Debunking some common pitfalls*, J. Risk and Insurance, 64 (1997), pp. 37–57.

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

[17] H.-J. Bungartz and M. Griebel, *A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives*, J. Complexity, 15 (1999), pp. 167–199.

[18] H.-J. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica, 13 (2004), pp. 1–123.

[19] R. Caflisch, W. Morokoff, and A. Owen, *Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension*, J. Comp. Finance, 1 (1997), pp. 27–46.

[20] R. Cools, *Monomial cubature rules since "Stroud": A compilation–part 2*, J. Comput. Appl. Math., 112 (1999), pp. 1–2.

[21] ——, *An encyclopedia of cubature formulas*, J. Complexity, 19 (2003), pp. 445–453.

[22] R. Cools and P. Rabinowitz, *Monomial cubature rules since "Stroud": A compilation*, J. Comput. Appl. Math., 48 (1993), pp. 309–326.

[23] S. Corsaro, P. Angelis, Z. Marino, and F. Perla, *On high-performance software development for the numerical simulation of life insurance policies*, in Numerical Methods for Finance, J. Appleby, D. Edelman, and J. Miller, eds., vol. 8 of Financial Mathematics Series, Chapman & Hall/CRC, 2007, pp. 87–111.

[24] C. Cottin and A. Kurz, *Asset–Liability–Management in der Lebensversicherung*, Preprint 16, Fachhochschule Bielefeld, 2003.

[25] J. Cox, J. Ingersoll, and S. Ross, *A theory of the term structure of interest rates*, Econometrica, 53 (1985), pp. 385–407.

[26] J. Cummins, K. Miltersen, and S. Persson, *International comparison of interest rate guarantees in life insurance.* Astin Colloquium, Bergen, 2004.

[27] F. Curbera, *Delayed curse of dimension for Gaussian integration*, J. Complexity, 16 (2000), pp. 474–506.

[28] P. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, 1984.

[29] M. De Felice and F. Moriconi, *Market based tools for managing the life insurance company*, Astin Bulletin, 1 (2005), pp. 79–111.

[30] R. DeVore, *Nonlinear approximation*, Acta Numerica, 7 (1998), pp. 51–150.

[31] J. Dick, *Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order*, SIAM J. Numer. Anal., 46 (2008), pp. 1519–1553.

[32] J. Dick, I. Sloan, X. Wang, and H. Woźniakowski, *Liberating the weights*, J. Complexity, 20 (2004), pp. 593–623.

[33] D. Diers, *Interne Unternehmensmodelle in der Schaden- und Unfallversicherung*, ifa-Schriftenreihe, Ulm, 2007.

[34] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, 1987.

[35] H. Edelsbrunner, *Algebraic decomposition of non-convex polyhedra*, in Proc. 36th Ann. IEEE Sympos. Found. Comput. Sci., IEEE Computer Society, 1995, pp. 248–257.

[36] B. Efron and C. Stein, *The jackknife estimate of variance*, Annals of Statistics, 9 (1981), pp. 586–596.

[37] D. Egloff, W. Farkas, and M. Leippold, *American options with stopping time constraints*, Appl. Math. Finance, (2008). forthcoming.

[38] T. Fischer, A. May, and B. Walther, *Anpassung eines CIR–1–Modells zur Simulation der Zinsstrukturkurve*, Blätter der DGVFM, 2 (2003), pp. 193–206.

[39] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi, *An application of Malliavin calculus to Monte Carlo methods in finance*, Finance & Stochastics, 4 (1999), pp. 391–412.

[40] K. Frank and S. Heinrich, *Computing discrepancies of Smolyak quadrature rules*, J. Complexity, 12 (1996), pp. 287–314.

[41] C. Fries, *Mathematical Finance: Theory, Modeling, Implementation*, John Wiley & Sons, 2007.

[42] A. Genz, *Fully symmetric interpolatory rules for multiple integrals*, SIAM J. Numer. Anal., 23 (1986), pp. 1273–1283.

[43] ——, *A package for testing multiple integration subroutines*, in Numerical Integration: Recent Developments, Software and Applications, P. Keast and G. Fairweather, eds., Kluwer, 1987, pp. 337–340.

[44] A. Genz, *Numerical computation of multivariate normal probabilities*, J. Comput. Graph. Statist., 1 (1992), pp. 141–150.

[45] A. Genz and B. Keister, *Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight*, J. Comp. Appl. Math., 71 (1996), pp. 299–309.

[46] A. Genz and A. Malik, *An adaptive algorithm for numerical integration over an n-dimensional rectangular region*, J. Comp. Appl. Math., 6 (1980), pp. 295–302.

[47] T. Gerstner, *Sparse grid quadrature methods for computational finance*. Habilitation, University of Bonn, 2007.

[48] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, Numerical Algorithms, 18 (1998), pp. 209–232.

[49] T. Gerstner and M. Griebel, *Dimension–adaptive tensor–product quadrature*, Computing, 71 (2003), pp. 65–87.

[50] T. Gerstner, M. Griebel, and M. Holtz, *The Effective Dimension of Asset-Liability Management Problems in Life Insurance*, in Proc. Third Brazilian Conference on Statistical Modelling in Insurance and Finance, C. Fernandes, H. Schmidli, and N. Kolev, eds., 2007, pp. 148–153.

[51] ——, *Efficient deterministic numerical simulation of stochastic asset-liability management models in life insurance*, Insurance: Mathematics and Economics, (2008). In revision.

[52] T. Gerstner, M. Griebel, M. Holtz, R. Goschnick, and M. Haep, *A general asset-liability management model for the efficient simulation of portfolios of life insurance policies*, Insurance: Mathematics and Economics, 42 (2008), pp. 704–716.

[53] ——, *Numerical simulation for asset-liability management in life insurance*, in Mathematics – Key Technology for the Future, Part 6, H.-J. Krebs and W. Jäger, eds., Springer, 2008, pp. 319–341.

[54] T. Gerstner and M. Holtz, *Geometric tools for the valuation of performance-dependent options*, in Computational Finance and its Applications II, M. Costantino and C. Brebbia, eds., London, 2006, WIT Press, pp. 161–170.

[55] T. Gerstner and M. Holtz, *The orthant decomposition of hyperplane arrangements*, working paper, University Bonn, 2006.

[56] T. Gerstner and M. Holtz, *Valuation of performance-dependent options*, Appl. Math. Finance, 15 (2008), pp. 1–20.

[57] T. Gerstner, M. Holtz, and R. Korn, *Valuation of performance-dependent options in a Black-Scholes framework*, in Numerical Methods for Finance, J. Appleby, D. Edelman, and J. Miller, eds., vol. 8 of Financial Mathematics Series, Chapman & Hall/CRC, 2007, pp. 203–214.

[58] M. Giles and P. Glasserman, *Smoking adjoints: Fast Monte Carlo Greeks*, Risk, 19 (2006), pp. 88–92.

[59] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, 2003.

[60] O. Goecke, *Über die Fähigkeit eines Lebensversicherers Kapitalmarktrisiken zu transformieren*, Blätter der DGVFM, 2 (2003), pp. 207–227.

[61] M. Griebel, *Sparse grids and related approximation schemes for higher dimensional problems*, in Foundations of Computational Mathematics (FoCM05), Santander, L. Pardo, A. Pinkus, E. Suli, and M. Todd, eds., Cambridge University Press, 2006, pp. 106–161.

[62] M. Griebel and M. Holtz, *Dimension-wise integration of high-dimensional functions with applications to finance*, INS Preprint 0809, Institute for Numerical Simulation, University of Bonn, 2008.

[63] M. Griebel and S. Knapek, *Optimized tensor-product approximation spaces*, Constructive Approximation, 16 (2000), pp. 525–540.

[64] M. Griebel, F. Kuo, and I. Sloan, *The smoothing effect of the ANOVA decomposition*. Technical Report, University of New South Wales, 2008.

[65] M. Griebel and H. Woźniakowski, *On the optimal convergence rate of universal and non-universal algorithms for multivariate integration and approximation*, Mathematics of Computation, 75 (2006), pp. 1259–1286.

[66] A. Grosen and P. Jorgensen, *Fair valuation of life insurance liabilities: The impact of interest rate guarantees, surrender options and bonus policies*, Insurance: Mathematics and Economics, 26 (2000), pp. 37–57.

[67] ——, *Life insurance liabilities at market value: An analysis of insolvency risk, bonus policy, and regulatory intervention rules in a barrier option framework*, J. Risk and Insurance, 69 (2002), pp. 63–91.

[68] T. Hahn, *Cuba - a library for multidimensional numerical integration*, Computer Physics Communications, 168 (2005), pp. 78–95.

[69] M. Hansen and K. Miltersen, *Minimum rate of returns guarantees: The Danish case*, Scand. Actuarial J., 4 (2002), pp. 280–318.

[70] J. Harrison and S. Pliska, *Martingales and stochastic integrals in the theory of continuous trading*, Stochastic Process. Appl., 11 (1981), pp. 215–260.

[71] M. Hegland, *Adaptive sparse grids*, ANZIAM J., 44 (2003), pp. C335–C353.

[72] S. Heinrich, *Efficient algorithms for computing the L2-discrepancy*, Math. Comp., 65 (1995), pp. 1621–1633.

[73] F. Heiss and V. Winschel, *Likelihood approximation by numerical integration on sparse grids*, J. Econometrics, 144 (2008), pp. 62–80.

[74] F. Hickernell, I. Sloan, and G. Wasilkowski, *On tractability of weighted integration over bounded and unbounded regions in* $\mathbb{R}^s$, Math. Comp., 73 (2004), pp. 1885–1901.

[75] F. Hickernell and H. Woźniakowski, *Integration and approximation in arbitrary dimensions*, Adv. Comput. Math., 12 (2000), pp. 25–58.

[76] F. J. Hickernell, *A generalized discrepancy and quadrature error bound*, Mathematics of Computation, 67 (1998), pp. 299–322.

[77] A. Hinrichs and E. Novak, *Cubature formulas for symmetric measures in higher dimensions with few points*, Math. of Comp., 76 (2007), pp. 1357–1372.

[78] W. Hoeffding, *A class of statistics with asympotically normal distributions*, Annals of Math. Statist., 19 (1948), pp. 293–325.

[79] J. Hull, *Options, Futures and other Derivative Securities*, Prentice Hall, Upper Saddle River, 2000.

[80] W. Hürlimann, *Fair pricing using deflators and decrement copulas: The unit linked endowment approach*, Blätter der DGVFM, 3 (2004), pp. 421–437.

[81] J. Imai and K. Tan, *Minimizing effective dimension using linear transformation*, in Monte Carlo and Quasi-Monte Carlo Methods 2002, H. Niederreiter, ed., Springer, 2004, pp. 275–292.

[82] ——, *A general dimension reduction technique for derivative pricing*, J. Comp. Finance, 10 (2006), pp. 129–155.

[83] P. Jäckel, *Monte-Carlo Methods in Finance*, John Wiley & Sons, 2002.

[84] R. Jaquemod, *Abschlussbericht der DAV-Arbeitsgruppe Stochastisches Unternehmensmodell für deutsche Lebensversicherungen*, Schriftenreihe angewandte Versicherungsmathematik, Versicherungswirtschaft, 2005.

[85] T. Jinag and A. Owen, *Quasi-regression with shrinkage*, Math. Comput. Simul., 62 (2003), pp. 231–241.

[86] S. Joe, *Formulas for the computation of the weighted L_2 discrepancy*, Research Report 55, University of Waikato, 1997.

[87] P. Jorgensen, *Life insurance contracts with embedded options, valuation, risk management, and regulation*, J. Risk Finance, (2001), pp. 19–30.

[88] ——, *On accounting standards and fair valuation of life insurance and pension liabilities*, Scand. Actuarial J., 5 (2004), pp. 372–394.

[89] K. Judd, *Numerical Methods in Economics*, MIT Press, Cambridge, 1998.

[90] M. Kalos and P. Whitlock, *Monte Carlo Methods*, John Wiley & Sons, 1986.

[91] I. Karatzas, *Lectures on the Mathematics of Finance*, vol. 8 of CRM Monograph Series, American Mathematical Society, 1997.

[92] I. Karatzas and S. E. Shreve, *Methods of Mathematical Finance*, Springer, New York, 1998.

[93] A. Kling, A. Richter, and J. Russ, *The interaction of guarantees, surplus distribution, and asset allocation in with-profit life insurance policies*, Insurance: Mathematics and Economics, 40 (2007), pp. 164–178.

[94] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, 1992.

[95] A. Kolmogorov, *On the representation of continuous functions of several variables by superpositions of continuous functions of several variables and addition*, Dokl. Akad. Nauk SSSR, 114 (1957), pp. 953–956. English translation: American Math. Soc. Transl. (2), 28, 55-59, 1963.

[96] R. Korn, *A valuation approach for tailored options.* Technical Report, University Kaiserslautern, 1996.

[97] F. Kuo, I. Sloan, G. Wasilkowski, and H. Woźniakowski, *On decompositions of multivariate functions.* Technical Report, University of New South Wales, 2008.

[98] F. Y. Kuo and I. Sloan, *Lifting the curse of dimensionality*, Notices of the American Mathematical Society, 52 (2005), pp. 1320–1328.

[99] L. Laloux, P. Cizeau, J. Bouchaud, and M. Potters, *Noise dressing of financial correlation matrices*, Phys. Rev. Lett., 83 (1999), pp. 1467–1470.

[100] J. Lawrence, *Polytope volume computation*, Math. Comp., 57 (1991), pp. 259–271.

[101] P. L'Ecuyer and C. Lemieux, *Variance reduction via lattice rules*, Management Science, 46 (2000), pp. 1214–1235.

[102] M. Ledoux, *The Concentration of Measure Phenomenon*, Mathematical Surveys and Monographs, American Mathematical Society, 2001.

[103] C. Lemieux and A. Owen, *Quasi-regression and the relative importance of the ANOVA components of a function*, in Monte Carlo and Quasi-Monte Carlo Methods 2000, K. Fang, F. Hickernell, and H. Niederreiter, eds., Springer, 2002, pp. 331–344.

[104] G. Leobacher, *Stratified sampling and quasi-Monte Carlo simulation of Lévy processes*, Monte Carlo Methods and Applications, 12 (2006), pp. 231–238.

[105] V. Linetsky, *The path integral approach to financial modeling and options pricing*, Computational Economics, 11 (1998), pp. 129–163.

[106] R. Liu and A. Owen, *Estimating mean dimensionality of analysis of variance decompositions*, J. the American Statistical Association, 101 (2006), pp. 712–721.

[107] J. Lu and D. Darmofal, *Higher-dimensional integration with Gaussian weight for applications in probabilistic design*, SIAM J. Sci. Comput., 26 (2004), pp. 613–624.

[108] J. Matousek, *The exponent of discrepancy is at least 1.0669*, J. Complexity, 14 (1998), pp. 448–453.

[109] N. Meade and G. Salkin, *Index funds – construction and performance measurement*, J. Operational Research Society, 40 (1989), pp. 871–879.

[110] M. Mertens, *Faire Mindestverzinsung in überschussberechtigten Lebensversicherungsverträgen*, PhD thesis, University of Mainz, 2003.

[111] H. Meuer, E. Strohmaier, J. Dongarra, and H. Simon, `http://top500.org`, 2008.

[112] K. Miltersen and S. Persson, *Guaranteed investment contracts: Distributed and undistributed excess returns*, Scand. Actuarial J., 23 (2003), pp. 257–279.

[113] T. Moller and M. Steffensen, *Market-Valuation Methods in Life and Pension Insurance*, International Series on Actual Science, Cambridge University Press, 2007.

[114] B. Moro, *The full Monte*, RISK, 8 (1995).

[115] W. Morokoff, *Generating quasi-random paths for stochastic processes*, SIAM Review, 40 (1998), pp. 765–788.

[116] W. Morokoff and R. Caflisch, *Quasi-random sequences and their discrepancies*, SIAM J. Sci. Comput., 15 (1994), pp. 1251–1279.

[117] B. Moskowitz and R. Caflisch, *Smoothness and dimension reduction in quasi-Monte Carlo methods*, J. Math. Comp. Modeling, 23 (1996), pp. 37–54.

[118] T. Nahm, *Error estimation and index refinement for dimension-adaptive sparse grid quadrature with applications to the computation of path integrals*, Master's thesis, University Bonn, 2005.

[119] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.

[120] S. Ninomiya and S. Tezuka, *Toward real-time pricing of complex financial derivatives*, Appl. Math. Finance, 3 (1996), pp. 1–20.

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

[122] ——, *Simple cubature formulas with high polynomial exactness*, Constructive Approximation, 15 (1999), pp. 499–522.

[123] E. Novak, K. Ritter, and A. Steinbauer, *A multiscale method for the evaluation of Wiener integrals*, in Approximation Theory IX, Volume 2: Computational Aspects, C. Chui and L. Schumaker, eds., Vanderbilt Univ. Press, 1998, pp. 251–258.

[124] E. Novak and H. Woźniakowski, *Intractability results for integration and discrepancy.*, J. Complexity, 17 (2001), pp. 388–441.

[125] ——, *When are integration and discrepancy tractable?*, in Foundations of Computational Mathematics (FoCM99), Oxford, R. A. DeVore, A. Iserles, and E. Süli, eds., Cambridge University Press, 2001, pp. 211–266.

[126] ——, *L_2 discrepancy and multivariate integration*, in Analytic Number Theory: Essays in Honour of Klaus Roth, W. Chen, W. Gowers, H. Halberstam, W. Schmidt, and R. Vaughan, eds., Cambridge University Press, 2009, pp. 359–388.

[127] D. Nuyens and R. Cools, *Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel hilbert spaces*, Math. Comp., 75 (2006), pp. 903–920.

[128] P. Orlik and H. Terao, *Arrangements of Hyperplanes*, Springer, 1992.

[129] A. Owen, *Dimension distribution and quadrature test functions*, Statistica Sinica, 13 (2003), pp. 1–17.

[130] A. Papageorgiou, *The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration*, J. Complexity, 18 (2002), pp. 171–186.

[131] A. Papageorgiou and S. Paskov, *Deterministic simulation for risk management*, J. Portfolio Management, 25 (1995), pp. 122–127.

[132] A. Papageorgiou and J. Traub, *Faster evaluation of multidimensional integrals*, Computers in Phyics, 11 (1997), pp. 574–578.

[133] S. Paskov, *Average case complexity of multivariate integration for smooth functions*, J. Complexity, 9 (1993), pp. 291–312.

[134] S. Paskov and J. Traub, *Faster valuation of financial derivatives*, J. Portfolio Management, 22 (1995), pp. 113–120.

[135] K. Petras, *Smolyak cubature of given polynomial degree with few nodes for increasing dimension*, Numer. Math., 93 (2003), pp. 729–753.

[136] A. Pinkus, *Approximating by ridge functions*, in Surface Fitting and Multiresolution Methods, A. Le Mehaute, C. Rabut, and L. Schumaker, eds., Vanderbilt Univ. Press, 1997, pp. 279–292.

[137] L. Plaskota, *The exponent of discrepancy of sparse grids is at least 2.1933*, Advances in Computational Mathematics, 12 (2000), pp. 3–24.

[138] L. Plaskota and G. W. Wasilkowski, *The exact exponent of sparse grid quadratures in the weighted case*, J. Complexity, 17 (2001), pp. 840–848.

[139] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[140] H. Rabitz and O. Alis, *General foundations of high-dimensional model representations*, J. Mathematical Chemistry, 25 (1999), pp. 197–233.

[141] C. Reisinger, *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben*, PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2004.

[142] K. Ritter, *Average-case Analysis of Numerical Problems*, vol. 1733 of Lecture Notes in Mathematics, Springer, 2000.

[143] S. Rodrick, *Incentive Compensation and Employee Ownership*, National Center for Employee Ownership, 2004.

[144] S. Saitoh, *Integral Transforms, Reproducing Kernels and Their Applications*, Research Notes in Mathematics, Chapman & Hall/CRC, 1997.

[145] T. Siu, *Fair valuation of participating policies with surrender options and regime switching*, Insurance: Mathematics and Economics, 37 (2005), pp. 533–552.

[146] I. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Oxford University Press, New York, 1994.

[147] I. Sloan, X. Wang, and H. Woźniakowski, *Finite-order weights imply tractability of multivariate integration*, J. Complexity, 20 (2004), pp. 46–74.

[148] I. Sloan and H. Woźniakowski, *When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals?*, J. Complexity, 14 (1998), pp. 1–33.

[149] I. H. Sloan, F. Y. Kuo, and S. Joe, *Constructing randomly shifted lattice rules in weighted Sobolev spaces*, SIAM J. Numer. Anal., 40 (2002), pp. 1650 – 1665.

[150] S. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR, 4 (1963), pp. 240–243.

[151] I. Sobol, *Sensitivity estimates for nonlinear mathematical models*, Mathematical Modeling and Computational Experiment, 1 (1993), pp. 407–414.

[152] ——, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Math. Comput. Simulation, 55 (2001), pp. 271–280.

[153] A. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, 1971.

[154] M. Sullivan, *Discrete-time continuous-state interest rate models*, J. Economic Dynamics and Control, 25 (2001), pp. 1001–1017.

[155] A. Tanskanen and J. Lukkarinen, *Fair valuation of path-dependent participating life insurance contracts*, Insurance: Mathematics and Economics, 33 (2003), pp. 595–609.

[156] S. Tezuka, *On the necessity of low-effective dimension*, J. Complexity, 21 (2005), pp. 710–721.

[157] J. Traub, H. Woźniakowski, and G. Wasilkowski, *Information-Based Complexity*, Academic Press, New York, 1988.

[158] G. Wahba, *Spline Models for Observational Data*, vol. 59, CBMS-NSF Regional Conf. Ser. Appl. Math., SIAM, Philadelphia, 1990.

[159] X. Wang, *On the effects of dimension reduction techniques on some high-dimensional problems in finance*, Operations Research, 54 (2006), pp. 1063–1078.

[160] X. Wang and K.-T. Fang, *The effective dimension and quasi-Monte Carlo integration*, J. Complexity, 19 (2003), pp. 101 – 124.

[161] X. Wang and I. Sloan, *Why are high-dimensional finance problems often of low effective dimension?*, SIAM J. Sci. Comput., 27 (2005), pp. 159 – 183.

[162] G. Wasilkowski, F. Kuo, and B. Waterhouse, *Randomly shifted lattice rules for unbounded integrands*, J. Complexity, 22 (2006), pp. 630–651.

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

[164] ——, *The exponent of discrepancy is at most 1.4778*, Math. Comp., 66 (1997), pp. 1125–1132.

[165] ——, *Weighted tensor product algorithms for linear multivariate problems*, J. Complexity, 15 (1999), pp. 402–447.

[166] G. Wasilkowski and H. Woźniakowski, *Finite-order weights imply tractability of linear multivariate problems*, J. Approximation Theory, 130 (2004), pp. 57–77.

[167] B. Waterhouse, *New developments in the construction of lattice rules*, PhD thesis, University of New South Wales, 2007.

[168] K. Wolfsdorf, *Versicherungsmathematik, Teil 1: Personenversicherung*, Teubner, 1986.

[169] M. Wüthrich, H. Bühlmann, and H. Furrer, *Market-Consistent Actuarial Valuation*, EAA Lecture Notes, Springer, 2008.

[170] C. Zenger, *Sparse grids*, in Parallel Algorithms for Partial Differential Equations, W. Hackbusch, ed., vol. 31 of Notes on Numerical Fluid Mechanics, Vieweg, 1991.

[171] P. Zhang, *Exotic Options*, World Scientific Publishing, 1998.