The ANOVA decomposition and generalized sparse grid methods for the high-dimensional backward Kolmogorov equation

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Alexander Hullmann

aus

Engelskirchen

Bonn 2014

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

1. Gutachter: Prof. Dr. M. Griebel

2. Gutachter: Prof. Dr. J. Garcke

Tag der Promotion: 25. Februar 2015

Erscheinungsjahr: 2015

Zusammenfassung

Die vorliegende Arbeit behandelt ein Verfahren zur Diskretisierung und Lösung der hochdimensionalen Kolomogorov Rückwärts-Gleichung, welche beispielsweise bei der Bewertung von Optionen auf multivariaten Sprung-Diffusionsprozessen auftritt. Um dieses hochdimensionale Problem effizient behandeln zu können, wenden wir zunächst eine ANOVA-Zerlegung und Trunkation auf die Anfangsbedingung an und erhalten so eine Superposition von nieder- und moderatdimensionalen Problemen. Diese partiellen Integro-Differentialgleichungen diskretisieren wir mit einem θ-Schema in der Zeit und mit einem verallgemeinerten dünnen Gitter im Ort. Zur Lösung der entstehenden linearen Gleichungssysteme verwenden wir neue additive Vorkonditionierungsverfahren sowie eine effiziente Implementierung der Operatoranwendung für nicht-lokale Operatoren. Die Kombination dieser Methoden versetzt uns in die Lage, die Kolmogorov Rückwärts-Gleichung für hochdimensionale Sprung-Diffusionsprozesse effizient zu lösen, was wir mit der numerischen Bewertung einer Europäischen Option demonstrieren, die auf einem zehndimensionalen Kou-Modell basiert.

Der erste Beitrag dieser Arbeit besteht in der Untersuchung, unter welchen Voraussetzungen wir mit einer ANOVA-Zerlegung der Anfangsbedingung oder der Lösung die Kolmogorov Rückwärts-Gleichung in eine Superposition von niederdimensionalen Teilproblemen überführen können. Bisherige Ansätze setzen zumeist eine multivariate Brownsche Bewegung voraus, deren Kovarianzmatrix diagonal ist. Wir fordern schwächere Voraussetzungen und konzentrieren uns von Beginn an auf Sprung-Diffusionsprozesse. Hierbei behandeln wir auch Alternativen zu der in der Literatur gängigen Anker-ANOVA. Schließlich rekombinieren wir die numerischen Teillösungen, um eine Approximation der Gesamtlösung zu erhalten.

Der zweite Beitrag der Arbeit besteht in der additiven Vorkonditionierung des Dünngitter-Erzeugendensystems. Anders als beim vollen Gitter führt ein Multilevelansatz mit Jacobi-Skalierung für H^t-elliptische Probleme nicht zu Konditionszahlen, die sich unabhängig von der Dimension d oder dem Diskretisierungslevel J beschränken lassen. Wir beschreiben ein Lineares Programm, das die Skalierungsfaktoren der Operatormatrix mit dem Ziel einer möglichst kleinen Konditionszahl optimiert. Wir beweisen, dass sich selbst bei der besten positiven Skalierung die Konditionszahl im H^t-Fall wie $\Theta(J^{d-2})$ verhält. Gestatten wir hingegen negative Skalierungsfaktoren oder alternativ eine block-diagonale Vorkonditionierungsmatrix, erzielen wir für eine gewisse Klasse von Problemen, zu denen auch das Poisson-Problem gehört, $\mathcal{O}(1)$ Konditionszahlen unabhängig vom Level J *und* der Dimension d. Für den Laplace-Operator und das reguläre dünne Gitter beobachten wir sogar *fallende* Konditionszahlen bei fixiertem Level J und steigender Dimension d. Unsere Vorkonditionierer können ohne Prewavelets und mit linearem Aufwand in der Anzahl der Unbekannten implementiert werden. Weiterhin setzen wir ein nicht-lineares Verfahren ein, das im Kontext der dünnen Gitter auch als OptiCom bezeichnet wird und die in jedem Iterationsschritt optimalen Skalierungsfaktoren bestimmt. Wenn der Operator eine Darstellung als Summe von Produktoperatoren zulässt, ist die Implementierung der OptiCom dank einer speziellen Matrix-Vektor-Multiplikation in log-linearer Laufzeit bezüglich der Anzahl der Freiheitsgrade möglich, was eine deutliche Verbesserung im Vergleich zum quadratischen Aufwand darstellt, der typischerweise bei der Umsetzung der OptiCom nötig ist. Weiterhin geben wir CG-Varianten für alle beschriebenen Verfahren an.

Der dritte Beitrag dieser Arbeit ist die Einführung eines mehrdimensionalen Kou-Modells und die numerische Umsetzung und Zusammenführung der beschriebenen Verfahren. Hierzu

gehört die Beschreibung des unidirektionalen Prinzips für nicht-lokale Operatoren sowie die Entwicklung einer neuen Rekurrenzformel für die Kou-Operatoranwendung im Galerkin-Fall. Wir können die Trunkationsfehler der Anker- und Gauss-ANOVA-Zerlegung der Anfangsbedingung mit einem Monte Carlo-Verfahren bestimmen, und kombinieren schließlich die Anker-ANOVA mit unserem Dünngitter-Löser für partielle Integro-Differentialgleichungen. Anhand einer kurzen Fehlerdiskussion demonstrieren wir, dass sich ANOVA-Approximationsfehler und Diskretisierungsfehler gegenläufig entwickeln, was sich in unseren Experimenten bestätigt.

Insgesamt können wir nun die hochdimensionale Kolmogorov Rückwärts-Gleichung für bestimmte multivariate Sprung-Diffusionsprozesse numerisch effizient behandeln. Mit unseren Ergebnissen im Bereich der Operatoranwendung und der Vorkonditionierung bleibt der Gesamtaufwand des Verfahrens linear in der Anzahl der Freiheitsgrade, und wir können erstmalig Probleme, die auf einem zehndimensionalen Kou-Modell basieren, mit einer hinreichenden Genauigkeit lösen. Dies wäre mit klassischen Tensorproduktansätzen ohnehin unmöglich, aber auch der Dünngitteransatz profitiert substanziell von der anfänglichen ANOVA-Zerlegung und Trunkation.

Danksagung

Viele Menschen haben ihren Anteil am Entstehen dieser Arbeit. An dieser Stelle möchte ich ihnen meine Dankbarkeit ausdrücken. Zuallererst verdient mein Doktorvater Prof. Dr. M. Griebel großen Dank für die Überlassung des Themas, die Bereitstellung exzellenter Arbeitsbedingungen und dafür, dass er mich seit 2006 durch viele Diskussionen, Anregungen und Herausforderungen ausgebildet und gefördert hat. Außerdem möchte ich mich bei Prof. Dr. J. Garcke für die Übernahme des Zweitgutachtens und viele gute Anregungen bedanken.

Meine kollegiale Arbeitsgruppe hat sehr dazu beigetragen, dass ich immer gerne am Institut für Numerische Simulation (INS) gearbeitet habe. Mein Dank gilt hier insbesondere der Institutssekretärin Babette Weißkopf für ihre große Hilfsbereitschaft in Verwaltungsangelegenheiten und den (Parallelrechner-)Administratoren, ohne die die aufwändigen numerischen Experimente in dieser Arbeit nicht möglich gewesen wären. Meine Kollegen Bastian Bohn, Jens Oettershagen, Christian Kuske und Sebastian Mayer haben mich durch mathematische Diskussionen und das Korrekturlesen der Arbeit unterstützt, aber auch die gemeinsamen Pausen waren eine Bereicherung. Des Weiteren möchte ich mich bei der Universität Bonn, der DFG und dem BMBF bedanken, die phasenweise meine Arbeit am INS finanziert haben. Zu guter Letzt verdient meine Familie größten Dank für ihre Unterstützung.

Contents

1 Introduction **1**

2 ANOVA decomposition of the backward Kolmogorov equation **7**
2.1 Lévy processes and the backward Kolmogorov equation for option pricing . . . 7
2.1.1 Lévy processes and problem setup . . . 8
2.1.2 The corresponding backward Kolmogorov equation . . . 16
2.2 The ANOVA decomposition . . . 21
2.2.1 Definition . . . 21
2.2.2 Extension: Iterated ANOVA . . . 26
2.3 Decomposition of functions based on expected values . . . 27
2.3.1 A direct decomposition of the solution for Lévy processes . . . 28
2.3.2 A special decomposition of the solution for Gaussian processes . . . 30
2.3.3 Decomposition of the final condition . . . 32
2.4 Common approximation schemes . . . 35
2.4.1 Truncation dimension . . . 35
2.4.2 Superposition dimension . . . 36
2.4.3 Combination of truncation and superposition dimension . . . 36

3 Discretization of the moderate-dimensional subproblems **41**
3.1 Function spaces . . . 41
3.2 Variational formulation . . . 42
3.3 Localization . . . 42
3.4 Space and time discretization . . . 44
3.5 Convergence rates . . . 45
3.6 Further simplifications and summary . . . 46

4 Sparse grid spaces and fast operator application **49**
4.1 Definition . . . 50
4.2 Generating system discretization of variational problems . . . 53
4.2.1 Generating system approach . . . 54
4.2.2 Generalized sparse grid . . . 57
4.3 Operator application . . . 58
4.3.1 Single space matrix-vector multiplication algorithm . . . 60
4.3.2 Unidirectional principle for non-local operators . . . 61

5 Preconditioning of high-dimensional elliptic equations **67**
5.1 Norm equivalences based on orthogonal subspaces . . . 68

5.2 The theory of subspace splittings applied to sparse grids . . . 73
5.2.1 Subspace splitting theory . . . 73
5.2.2 Introducing multiple scaling factors . . . 76
5.2.3 Subspace splittings for generalized sparse grids . . . 77
5.2.4 Implementation . . . 78
5.3 Positive scalings for sparse grid subspace correction methods . . . 79
5.3.1 Formulation as Linear Program . . . 79
5.3.2 Bounds for H^t-elliptic problems . . . 82
5.4 Unconstrained scalings for sparse grid subspace correction methods . . . 85
5.5 An orthogonal projection based preconditioner . . . 87
5.5.1 Construction . . . 88
5.5.2 Relation to prewavelets . . . 92
5.5.3 Implementation . . . 94
5.6 OptiCom-approach . . . 96
5.6.1 Definition . . . 96
5.6.2 CG version . . . 97
5.6.3 Application to sparse grids . . . 98
5.6.4 Implementation . . . 100
5.7 Numerical experiments . . . 101
5.7.1 Relation full grid and sparse grid condition numbers . . . 101
5.7.2 Splitting condition numbers . . . 103
5.7.3 Iteration counts . . . 103
5.7.4 Impact of the single space matrix-vector multiplication . . . 106

6 Numerical experiments with the Kou-model 109
6.1 The model and problem setup . . . 110
6.1.1 One-dimensional Kou model . . . 110
6.1.2 Multi-dimensional case and dependence modelling . . . 111
6.1.3 Representation of the multi-dimensional process as Lévy process . . . 112
6.1.4 Payoff function . . . 115
6.2 Galerkin recurrence formula . . . 116
6.3 Experiments with the PIDE solver . . . 120
6.3.1 Toivanen option . . . 120
6.3.2 Two-dimensional put option . . . 120
6.3.3 Preconditioning experiments . . . 124
6.4 The ANOVA approximation for a ten-dimensional Kou model . . . 130
6.4.1 Comparison for models with different decay rates . . . 133
6.4.2 Evaluation at, in and out of the money . . . 135
6.5 The ANOVA-PIDE approach for our ten-dimensional problem . . . 136
6.5.1 Approximation by one one-dimensional subproblem ($d_t = 1, d_s = 0$) . . . 141
6.5.2 Approximation by nine two-dimensional subproblems ($d_t = 1, d_s = 1$) . . . 141
6.5.3 Approximation by eight three-dimensional subproblems ($d_t = 2, d_s = 1$) 141
6.5.4 Approximation by 28 four-dimensional subproblems ($d_t = 2, d_s = 2$) . . . 145
6.5.5 Approximation by seven four-dimensional problems ($d_t = 3, d_s = 1$) . . . 145
6.5.6 Discussion . . . 145

7 Conclusion **149**
7.1 Summary 149
7.2 Outlook 150

Bibliography **151**

1 Introduction

The relevance of mathematical models to many areas like finance, business intelligence and health care is increasing rapidly. The ever-growing means of data collection, storing and analysis lead to more complex and numerically challenging mathematical models. This development can be observed quite well in option pricing: In 1973, Black, Scholes [BS73] and Merton [Mer73] published their seminal work, which allowed to determine the fair price of an option under a set of assumptions, among them that security prices follow a geometric Brownian motion with constant volatility. Of course, the notion of continuous sample paths does not hold in practice simply because of limited trading hours or sudden market events. This motivated the jump-diffusion model [Mer76], where security prices do not only follow the geometric Brownian motion but can also exhibit log-normally distributed jumps. However, it has become clear that the general assumption of log-normality is questionable since the normal distribution underestimates the probability of extreme events. In general, daily returns of six standard deviations can practically be observed in most markets [CT04], although a market move of that magnitude would theoretically occur only about once in a million years. The Kou-model [Kou02] with double exponential jump-diffusion tries to account for the leptokurtic feature of returns. Of course, there is a myriad of other approaches, cf. [CT04], that can also account for heteroscedasticity like stochastic volatility models [Hes93] or time-changed Brownian motions [MCC98]. Analytical option pricing formulae do not exist for all cases, and often, instead of the Black-Scholes partial differential equation (PDE), a partial integro-differential equation (PIDE) has to be solved. The aspect of multi-dimensionality also comes into play when we deal with stochastic volatility components and options on multiple underlyings like interest-rate swaps or indices, ultimately resulting in multi-dimensional PIDEs.

In this thesis, we discuss a numerical solution method for certain instances of the high-dimensional backward Kolmogorov equation (BKE)

$$\frac{\partial}{\partial t}V(t,\mathbf{s}) + \frac{1}{2}\sum_{j,k=1}^{d} q_{jk}s_j s_k \frac{\partial^2}{\partial s_j \partial s_k}V(t,\mathbf{s}) + r\sum_{j=1}^{d} s_j \frac{\partial}{\partial s_j}V(t,\mathbf{s}) - rV(t,\mathbf{s}) \tag{1.1}$$

$$+ \int_{\mathbb{R}^d} V(t,\mathbf{s}e^{\mathbf{y}}) - V(t,\mathbf{s}) - \sum_{j=1}^{d} s_j(e^{y_j}-1)\frac{\partial}{\partial s_j}V(t,\mathbf{s})\nu(\mathrm{d}\mathbf{y}) = 0\,, \tag{1.2}$$

on $(t,\mathbf{s}) \in (0,T)\times\mathbb{R}^d_{>0}$ with $\mathbf{s} = (s_1,\dots,s_d)$, $\mathbf{s}e^{\mathbf{y}} := (s_1e^{y_1},\dots,s_de^{y_d})$ and

$$V(t,\mathbf{s}) \in C^{1,2}\big((0,T)\times\mathbb{R}^d_{>0}\big) \cap C^0\big([0,T]\times\mathbb{R}^d_{\geq 0}\big)\,, \quad V(T,\mathbf{s}) = h(\mathbf{s}), \mathbf{s}\in\mathbb{R}^d_{\geq 0}\,.$$

Note that (1.1) without the integral term in (1.2) resembles the classical Black-Scholes PDE, which is a well-known multi-dimensional convection-diffusion equation [Kwo08, Rei04]. Apart from the pricing of financial derivatives, such PDEs result from diffusion approxima-

tion techniques or the Fokker-Planck approach. Examples are the description of queueing networks [Mit97, SCDD02] and reaction mechanisms in molecular biology [Sjö07, SLE09].

The BKE in (1.1) and (1.2) typically arises in option pricing with

$$V(t,\mathbf{s}) = \mathbb{E}(e^{-r(T-t)}h(\mathbf{S}(T)) \mid \mathbf{S}(t) = \mathbf{s}) \,, \tag{1.3}$$

where h is the final condition or payoff function, $\mathbf{X}(t) = (X_1(t), \dots, X_d(t))$ is a Lévy process with state space $\mathbb{R}^d$ and $\mathbf{S}(t) = (S_1(0)e^{rt+X_1(t)}, \dots, S_d(0)e^{rt+X_d(t)})$ is an exponential Lévy model under the risk-neutral measure with starting point $\mathbf{S}(0) = (S_1(0), \dots, S_d(0))$.

In this thesis, we present theoretical and computational aspects of several new methods specifically designed to deal with the high-dimensionality of the BKE. We demonstrate the efficiency of our methods on the pricing of basket options with a multi-dimensional generalization of the Kou-model [Kou02]. Of course, some of the intermediate results obtained in this course may very well be applied to other PDE or PIDE problems and not just the BKE or the Kou-model. In the next paragraphs, we discuss the aspects of this thesis in more detail.

As a first step, we start with an initial drastic dimensionality reduction of the BKE (1.1) and (1.2) based on the ANOVA decomposition of functions [GH10b, Hoe48]. We now give a simplified account of the method: Essentially we represent the final condition $h(\mathbf{s})$ by a superposition of functions

$$h(\mathbf{s}) \approx \sum_{\mathfrak{m}\in\mathfrak{S}} h_{\mathfrak{m}}(\mathbf{s}_{\mathfrak{m}}) \,, \tag{1.4}$$

where $\mathfrak{S}$ is a subset of the power set of $\mathfrak{D} := \{1, \dots, d\}$, i.e., $\mathfrak{m} \in \mathfrak{S} \Rightarrow \mathfrak{m} \subset \mathfrak{D}$, and the $h_{\mathfrak{m}}$ only depend on the $\#\mathfrak{m}$-dimensional vectors $\mathbf{s}_{\mathfrak{m}} = (s_i)_{i\in\mathfrak{m}}$. It is easier and more insightful to apply the ANOVA decomposition to the representation (1.3) of $V(t,\mathbf{s})$ than to combine it directly with the PIDE (1.1) and (1.2). The representation (1.4) ultimately allows us to approximate $V(t,\mathbf{s})$ by a superposition of expected values

$$V(t,\mathbf{s}) \approx \sum_{\mathfrak{m}\in\mathfrak{S}} \mathbb{E}(e^{-r(T-t)}h_{\mathfrak{m}}(\mathbf{S}_{\mathfrak{m}}(T)) \mid \mathbf{S}_{\mathfrak{m}}(t) = \mathbf{s}_{\mathfrak{m}}) \tag{1.5}$$

based on the marginal Lévy processes $\mathbf{S}_{\mathfrak{m}} = (S_i)_{i\in\mathfrak{m}}$. The resulting moderate-dimensional problems on the right-hand side of (1.5) need to be computed numerically, so we deal with their $\#\mathfrak{m}$-dimensional BKE representation. This way of looking at things is depicted in Figure 1.1. In the end we trade a d-dimensional problem for $\#\mathfrak{S}$ moderate-dimensional problems. This technique has already been applied to the multi-dimensional geometric Brownian motion [RW07, SGW13]. The generalization to other processes and the relation to the anchor ANOVA decomposition is discussed in [RW13]. We evaluate possibilities to apply the ANOVA decomposition directly to the solution and not just the final condition and explore alternatives to the anchor ANOVA.

For the numerical solution of the now moderate-dimensional BKEs, we use a Θ-method for time stepping and a Galerkin discretization in space[1]. However, a classical tensor product approach for the space discretization of V or, to a lesser extent, the moderate-dimensional problems in (1.5) suffers from the so-called curse of dimensionality [Bel61], which means that the cost complexity for the approximation to the solution of a problem grows exponentially

[1] We refer the reader to [CT04, RSW10] for analytical aspects of the BKE and to [SS00, GOV05] for advanced space-time discretization techniques.

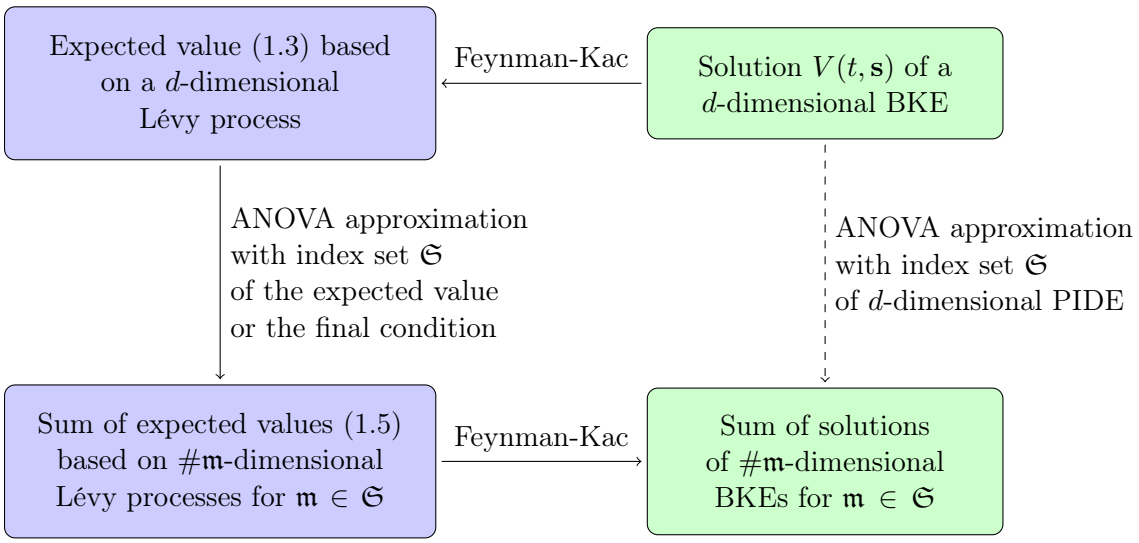

Figure 1.1: This diagram shows how the detour via the expected value formulation (1.3) based on a Lévy process leads to an ANOVA approximation of the d-dimensional BKE

with the dimension d. A d-dimensional mesh with a resolution of h in each direction typically results in a storage and cost complexity of $\Theta(h^{-d})$. Fortunately, it is possible to increase the efficiency by exploring specific a priori assumptions on the solution using sparse grids [BG04]. This discretization technique circumvents the curse of dimensionality to some extent as it results in a complexity of only $\mathcal{O}(h^{-1}\,(\log h^{-1})^{d-1})$, which allows for huge savings for higher values of d, while – depending on the smoothness assumptions on the function – the convergence rate of the error is unchanged or only affected by a logarithmic term. Sparse grid discretization spaces can be described by a non-direct sum of anisotropic full grid spaces, and regular sparse grids have been employed for the efficient solution of PDEs [Zen91, Bun92a, Gri91] for a long time. Generalized sparse grids offer great flexibility in enriching the discretization by different sets of full grid spaces, and have been studied in [BG99, GK09, GH13b] and in [Feu10, GG03, BG12] in connection with a posteriori dimension-adaptivity. In special cases, space-adaptivity [Gri98, Feu10] is beneficial, but generally speaking this method results in additional challenges in terms of algorithms and performance and is not part of this thesis.

After time and space discretization, the resulting systems of linear equations must be solved in a fast way, and usually some iterative method is employed. This results in the need of preconditioning, which is more challenging for generalized sparse grids than for isotropic full grid spaces. For H^t-elliptic problems discretized by a regular full grid, the BPX-preconditioner [BPX90] leads to optimal condition numbers and can be implemented as simple Jacobi scaling of a multilevel system [Gri94b]. For sparse grids, this is not the case and a simple Jacobi scaling leads to condition numbers that cannot be bounded from above independently of the discretization level J with $h = 2^{-J}$. Therefore, we follow three different approaches to find optimal or close-to-optimal scaling parameters for the sparse grid generating system: One is based on a Linear Program that minimizes the splitting condition number with respect to a subspace splitting based on orthogonal complement spaces. We prove that the *best possible* set of positive

scaling parameters results in condition numbers that grow by $\Theta(J^{d-2})$ in J for H^t-elliptic problems. The second method is based on the observation that partially negative scaling parameters can also result in a positive definite operator on the sparse grid space, even though this case is not covered by the classical theory of subspace splittings. With an algebraic transformation that produces partially negative scaling parameters we obtain an optimal iterative scheme with error contraction rates that are bounded independently of the level J and are even independent of the dimension d for some cases. A related result has been observed in [DSS09]. In the case of the d-dimensional Laplacian, we even observe a *decrease* in the condition number for *rising* dimension, all else being equal. We also introduce a preconditioner based on orthogonal projections that is closely related to the method that uses partially negative scaling parameters. In fact, it can be regarded as a convenient form of implementing of prewavelets. The last method we propose is a non-linear variable preconditioner [JN99] that has been coined OptiCom in the context of sparse grids applied to data mining problems [Heg03, Gar06, HGC07] and comes up with the best possible scaling in every iteration step. We show that the cost of an OptiCom iteration step is log-linear with respect to the degrees of freedom if a fast matrix-vector multiplication with the operator matrix is available. This is a significant improvement over the quadratic costs typically associated with the OptiCom. As a fast matrix-vector multiplication the unidirectional principle [BZ96, Bun92b, Zei11] can be employed, but we come up with a specifically designed algorithm that also reduces the constant factor of the cost complexity dramatically. Conjugate gradient versions of all considered iteration schemes are presented, which shows that there is even further cost reduction potential.

Another issue when solving the systems of linear equations that arise from the BKE is the non-locality of the integro-operator (1.2). We obtain linear systems with densely populated matrices, which, treated naively, would lead to computational costs for the matrix-vector multiplication that are quadratic with respect to the number of degrees of freedom N. In [AA00], the convolution integral is evaluated using the fast Fourier transform, which reduces the complexity of the system matrix application to $\mathcal{O}(N \log N)$. There are also other results, which make use of the decay of the integral kernel and the accompanying compressibility of the operator matrix [KK02, Rei10]. For the special case of Kou's jump-diffusion model, an even faster operator application with $\mathcal{O}(N)$ complexity in the finite difference case is available [Toi08]. In this thesis, we introduce a comparable approach for the Galerkin method and exploit it in our numerical solver. We use the unidirectional principle – which was originally developed for partial differential operators – and generalize it to our non-local operator. In combination with the Galerkin recurrence formula for the Kou model and the optimal preconditioning, we obtain a solver which altogether scales only linearly with respect to the number of degrees of freedom of our sparse grid discretization.

Finally, we combine the described methods for the solution of a ten-dimensional BKE based on Kou's jump-diffusion model. It is conceivable that this problem stems from an even higher-dimensional model after the projection onto the principal components of the diffusion covariance matrix and a subsequent truncation at ten space dimensions. We measure the errors of different ANOVA approximations at several points via a Monte Carlo approach and identify index sets $\mathfrak{S}$ that are promising, i.e., that are small but lead to sufficiently accurate solutions. We then approximate the solution of the ten-dimensional BKE by the solutions of lower-dimensional BKEs, which we compute using our sparse grid PIDE solver. By measuring the error at 100

randomly selected points using a Monte Carlo approach, we see that we can obtain for the first time a sufficiently accurate approximation of the solution based on a ten-dimensional Kou model not only at our anchor point but also in its proximity.

To sum this up, the main contributions of this thesis are the following:

- We show that the ANOVA decomposition of the final condition h and in some cases the solution V allows us to deal with moderate-dimensional marginals of our Lévy process, which ultimately lead to moderate-dimensional versions of the BKE. This is a novelty in the sense that in most of the literature only the multivariate Brownian motion is considered.

- We present a Linear Programming approach and an algebraic transformation to come up with quasi-optimal fixed a priori scalings for preconditioning. Moreover, we use the OptiCom for the first time in an iterative fashion with log-linear runtime for generalized sparse grids. We introduce the new single space matrix-vector multiplication that also reduces the constant factor in the runtime complexity of the OptiCom dramatically.

- We present a preconditioner for the generalized sparse grid generating system based on orthogonal projections. We show that for certain elliptic problems, the condition numbers can be bounded independently of the discretization level, the coefficients and the dimension.

- We generalize the unidirectional principle to non-local operators and present a multi-dimensional Kou model with a recurrence formula for the Galerkin discretization.

- We discuss the balance between ANOVA approximation error and PIDE discretization error, and we finally demonstrate that the solution of a ten-dimensional model problem can be efficiently approximated using a combination of the methods developed in this thesis.

Parts of this thesis have already been published in a journal article [GHO15] and proceedings contributions [GH13c, GH14b].

The remainder of this thesis is organized as follows: In Chapter 2 we recall the backward Kolmogorov equation, the underlying multivariate Lévy processes and how the ANOVA decomposition can be used to obtain an approximation by a sum of moderate-dimensional subproblems. In Chapter 3 we briefly describe their variational formulation that eventually leads to a discretization in space and time. Chapter 4 contains the definition of generalized sparse grid spaces and discusses numerical aspects of the implementation and operator application using a generating system. Chapter 5 deals with the optimal or quasi-optimal scaling of the generating system operator matrix. We obtain scaling factors by a Linear Program approach, an algebraic transformation and the non-linear OptiCom. We also present a preconditioner based on orthogonal projections closely related to the algebraic transformation. Furthermore, we discuss the efficient implementation, the dimension-dependence of the constants and CG versions of the iterative methods. Then, in Chapter 6, we present a ten-dimensional Kou-model and combine the ANOVA approximation and our PIDE solver to approximate the solution of a challenging option pricing problem. Final remarks in Chapter 7 conclude this thesis.

2 ANOVA decomposition of the backward Kolmogorov equation

The ANOVA decomposition of functions [GH10b] goes back to [Hoe48] and yields an efficient representation of high-dimensional functions by low-dimensional interactions. Apart from special cases, high-dimensional interactions do exist, but their *contribution* to the *description* of most functions is often negligible. This results in a robust method for the initial approximation of high-dimensional problems by a superposition of moderate-dimensional ones.

The ANOVA decomposition has already been used to approximate the payoff function in the context of basket option pricing with the geometric Brownian motion [RW07, Rei12]. There, the multi-dimensional Black-Scholes PDE is transformed into the heat equation. Then, the ANOVA approximation method rests on the argument that marginalizing out some dimensions of the heat kernel again results in a lower-dimensional heat kernel and thus also a lower-dimensional Black-Scholes PDE [SGW13]. We give a proof that the same principle holds analogously for Lévy processes with a kernel that has no product structure. Using the Feynman-Kac formula for Lévy processes, we see that a low-dimensional final condition or a low-dimensional ANOVA term of the solution of the high-dimensional BKE leads to a low-dimensional BKE, and we can finally approximate the solution of the high-dimensional BKE by a superposition of low-dimensional solutions. Note that a discussion of the anchor ANOVA decomposition for generalizations of the Black-Scholes model is given in [RW13].

In Section 2.1, we recall Lévy processes, discuss some properties of their marginals and apply simplifying transformations to (1.3). We also state the corresponding backward Kolmogorov equation and transform it to a simpler form. In Section 2.2, we give a self-contained view of the ANOVA decomposition of functions. Then, in Section 2.3, we show how the ANOVA approximation of the final condition or the solution of the BKE results in expressions that only include low- and moderate-dimensional marginals of our Lévy process. Finally, in Section 2.4, we discuss some common ANOVA approximation schemes and express them in our setting.

2.1 Lévy processes and the backward Kolmogorov equation for option pricing

We now briefly summarize what kind of problem we focus on in this section: For a certain class of stochastic processes $(\mathbf{S}(t))_{t\in[0,T]}$ with state space $\mathbb{R}^d$, we want to compute the function $V : [0,T] \times \mathbb{R}^d_{\geq 0} \to \mathbb{R}$ with

$$V(t,\mathbf{s}) = e^{-r(T-t)}\mathbb{E}\left[h(\mathbf{S}(T)) \mid \mathbf{S}(t) = \mathbf{s}\right] \tag{2.1}$$

for a given final condition $V(T, \mathbf{s}) = h(\mathbf{s})$, which is a problem that arises in option pricing on multi-dimensional jump-diffusion processes. In Subsection 2.1.1, we go into detail about Lévy processes, their marginals, state the option pricing problem and give some simplifying transformations of (2.1). The function $V(t, \mathbf{s})$ from (2.1) is also the solution of a partial integro-differential equation, which we discuss in Subsection 2.1.2. There is a close relationship between both formulations, which remains intact after all variable transforms.

2.1.1 Lévy processes and problem setup

We start this subsection with a definition of a Lévy process, cf. [CT04].

Definition 2.1 (Lévy process)**.** A càdlàg stochastic process $(\mathbf{X}(t))_{0 \le t < \infty}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $\mathbb{R}^d$ such that $\mathbf{X}(0) = 0$ is called a *Lévy process* if it has the following properties:

1. Independent increments: for every sequence $t_0 < t_1 < \cdots < t_n$, the random variables $\mathbf{X}(t_0), \mathbf{X}(t_1) - \mathbf{X}(t_0), \ldots, \mathbf{X}(t_n) - \mathbf{X}(t_{n-1})$ are independent
2. Stationary increments: $\mathbf{X}(t) - \mathbf{X}(s)$ has the same distribution as $\mathbf{X}(t - s)$, $0 \le s < t < \infty$
3. Stochastic continuity: $\forall\, \epsilon > 0 : \lim_{h \to 0} \mathbb{P}(|\mathbf{X}(t + h) - \mathbf{X}(t)| \ge \epsilon) = 0$

It is noteworthy that the stochastic continuity does not mean the sample paths are continuous. Jumps may occur, but not at fixed times t. This also means that a white noise process, which clearly has independent and stationary increments, is not a Lévy process. The multi-dimensional Brownian motion is the special case of a Lévy process without jumps.

Lévy-Khintchin representation

First, we fix a *truncation function* $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d_{\ge 0}$ that is bounded and measurable with $\mathcal{T}(\mathbf{z}) = 1 + o(|\mathbf{z}|)$ for $|\mathbf{z}| \to 0$ and $\mathcal{T}(\mathbf{z}) = O(1/|\mathbf{z}|)$ for $|\mathbf{z}| \to \infty$. Then, it is well-known from the theory of Lévy processes that the characteristic exponent $\psi : \mathbb{R}^d \to \mathbb{C}$ of $\mathbf{X}(t)$, which satisfies

$$\mathbb{E}\left(e^{\mathrm{i}\langle \boldsymbol{\xi}, \mathbf{X}(t)\rangle}\right) = e^{t\psi(\boldsymbol{\xi})} \quad \text{for} \quad \boldsymbol{\xi} \in \mathbb{R}^d, t \ge 0\,,$$

allows for the unique Lévy-Khintchin representation

$$\psi(\boldsymbol{\xi}) = -\frac{1}{2}\langle \boldsymbol{\xi}, \mathbf{Q}\boldsymbol{\xi}\rangle + \mathrm{i}\langle \boldsymbol{\theta}, \boldsymbol{\xi}\rangle + \int_{\mathbb{R}^d} \left(e^{\mathrm{i}\langle \boldsymbol{\xi}, \mathbf{z}\rangle} - 1 - \mathrm{i}\langle \boldsymbol{\xi}, \mathbf{z}\rangle \mathcal{T}(\mathbf{z})\right) \nu(\mathrm{d}\mathbf{z}), \tag{2.2}$$

where $\mathbf{Q} = (q_{jk})_{j,k=1}^d \in \mathbb{R}^{d \times d}$ is the covariance matrix of the continuous part of $\mathbf{X}$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d) \in \mathbb{R}^d$ is the drift[1] of $\mathbf{X}$, and ν is the Lévy jump measure that satisfies

$$\int_{\mathbb{R}^d} \min(1, |\mathbf{z}|^2)\nu(\mathrm{d}\mathbf{z}) < \infty\,. \tag{2.3}$$

[1] The term *drift* has to be used with caution, as $\boldsymbol{\theta}$ is not the actual expected value of $\mathbf{X}(1)$ but depends implicitly on the truncation function $\mathcal{T}$.

Given a truncation function $\mathcal{T}$, we can uniquely describe any Lévy process by the triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$.

The condition (2.3) means that there is only a finite amount of large jumps and a possibly infinite amount of small jumps. The purpose of the truncation function $\mathcal{T}$ is to make the distinction between small and large jumps. According to [CT04] P. Lévy used $\mathcal{T}(\mathbf{z}) = \frac{1}{1+|\mathbf{z}|^2}$, while most recent textbooks use $\mathcal{T}(\mathbf{z}) = \mathbb{1}_{\{|\mathbf{z}|<1\}}$. Note that in special cases trivial truncation functions are possible: For processes with a finite number of jumps $\mathcal{T}(\mathbf{x}) = 0$ is acceptable, while processes with a finite expected value admit $\mathcal{T}(\mathbf{x}) = 1$. Later on, we will make such an assumption and set $\mathcal{T}(\mathbf{z}) = 1$, but for the time being we try to be as general as possible.

Not necessarily all components $X_j(t), j = 1, \ldots, d$, of a Lévy processes with

$$\mathbf{X}(t) = (X_1(t), \ldots, X_d(t))$$

have a finite expected value or variance. For those $X_j(t)$ that do, we now devise a simple way to compute these quantities given the characteristic exponent (2.2). Let us shortly assume that we are given a random variable X with characteristic function ϕ. It holds that

$$\phi'(\xi) = \mathbb{E}[\mathrm{i}Xe^{\mathrm{i}\xi X}] \quad \Rightarrow \quad -\mathrm{i}\phi'(0) = \mathbb{E}[X]$$

and

$$\phi''(\xi) = \mathbb{E}[\mathrm{i}^2X^2e^{\mathrm{i}\xi X_j(t)}] \quad \Rightarrow \quad -\phi''(0) = \mathbb{E}[X^2]\,.$$

We can now express the variance of X as

$$\mathrm{var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = -\phi''(0) - \big(\mathrm{i}\phi'(0)\big)^2 = \big(\phi'(0)\big)^2 - \phi''(0)\,.$$

Thus, in order to compute the expected value and variance of $X_j(t), j = 1, \ldots, d$, we have to compute the first and second derivatives of the characteristic functions

$$\begin{aligned}\phi_{X_j(t)}(\xi) = \mathbb{E}[e^{\mathrm{i}\xi X_j(t)}] = \mathbb{E}[e^{\mathrm{i}\langle\xi\mathbf{e}_j,\mathbf{X}(t)\rangle}] &= e^{t\psi(\xi\mathbf{e}_j)} \\ &= \exp\left(t\left(-\frac{\xi^2 q_{jj}}{2} + \mathrm{i}\xi\theta_j + \int_{\mathbb{R}^d} e^{\mathrm{i}\xi z_j} - 1 - \mathrm{i}\xi z_j\mathcal{T}(\mathbf{z})\nu(\mathrm{d}\mathbf{z})\right)\right)\,.\end{aligned}$$

We get

$$\begin{aligned}E[X_j(t)] = -\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}\xi}e^{t\psi(\xi\mathbf{e}_j)}\bigg|_{\xi=0} &= -\mathrm{i}\left(t\frac{\mathrm{d}}{\mathrm{d}\xi}\psi(\xi\mathbf{e}_j)\right)\underbrace{e^{t\psi(\xi\mathbf{e}_j)}}_{=1}\bigg|_{\xi=0} \\ &= -\mathrm{i}t\left(-\xi q_{jj} + \mathrm{i}\theta_j + \int_{\mathbb{R}^d} \mathrm{i}z_j e^{\mathrm{i}\xi z_j} - \mathrm{i}z_j\mathcal{T}(\mathbf{z})\nu(\mathrm{d}\mathbf{z})\right)\bigg|_{\xi=0} \\ &= t\left(\theta_j + \int_{\mathbb{R}^d} z_j(1-\mathcal{T}(\mathbf{z}))\nu(\mathrm{d}\mathbf{z})\right)\,. \qquad (2.4)\end{aligned}$$

The expression (2.4) shows that the expected value of $X_j(t)$ grows linearly in time and consists of the drift plus the jump part adjusted for the truncation function. This tells us why for $\mathcal{T}(\mathbf{z}) = 1$ the parameter θ_j can be referred to as the expected value of $X_j(1)$ and for $\mathcal{T}(\mathbf{z}) = 0$ as the drift of the continuous part. However, it always depends on the jump measure ν whether

this kind of representation is possible. It holds that

$$\begin{aligned}\operatorname{var}(X_j(t)) &= \left(\frac{\mathrm{d}}{\mathrm{d}\xi} e^{t\psi(\xi\mathbf{e}_j)}\Big|_{\xi=0}\right)^2 - \left(\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^2 e^{t\psi(\xi\mathbf{e}_j)}\Big|_{\xi=0} \\ &= \left(\left(\frac{\mathrm{d}}{\mathrm{d}\xi} t\psi(\xi\mathbf{e}_j)\right) \underbrace{e^{t\psi(\xi\mathbf{e}_j)}}_{=1}\Big|_{\xi=0}\right)^2 - \frac{\mathrm{d}}{\mathrm{d}\xi}\left(\left(\frac{\mathrm{d}}{\mathrm{d}\xi} t\psi(\xi\mathbf{e}_j)\right) e^{t\psi(\xi\mathbf{e}_j)}\right)\Big|_{\xi=0} \\ &= \left(\frac{\mathrm{d}}{\mathrm{d}\xi} t\psi(\xi\mathbf{e}_j)\Big|_{\xi=0}\right)^2 - \left(\frac{\mathrm{d}}{\mathrm{d}\xi} t\psi(\xi\mathbf{e}_j)\right)^2\Big|_{\xi=0} - \left(\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^2 t\psi(\xi\mathbf{e}_j)\Big|_{\xi=0}, \end{aligned} \tag{2.5}$$

where the first two terms in (2.5) cancel out, and we only need to form the second derivative of $\psi(\xi\mathbf{e}_j)$ with respect to ξ and get

$$\operatorname{var}(X_j(t)) = -t\left(-q_{jj} + \int_{\mathbb{R}^d} (-1) z_j^2 e^{\mathrm{i}\xi z_j} \nu(\mathrm{d}\mathbf{z})\right)\Big|_{\xi=0} = t\left(q_{jj} + \int_{\mathbb{R}^d} z_j^2 \nu(\mathrm{d}\mathbf{z})\right) . \tag{2.6}$$

As was said before, it is not clear that (2.6) exists and is finite. For the model for ν that we pick in Chapter 6, however, this is the case.

Forming marginals of a Lévy process

We now want to study the effects of forming marginals or dropping components of $\mathbf{X}(t) \in \mathbb{R}^d$ for all $t \geq 0$. To this end, we first need to introduce some notation: Let $\mathfrak{D} := \{1, 2, \ldots, d\}$ be the set of all available dimensions. Given a subset $\mathfrak{m} \subset \mathfrak{D}$ that contains the dimensions or components that we want to consider, we can form an $\mathfrak{m}$-marginal $\mathbf{X}_\mathfrak{m}(t)$ with

$$(\mathbf{X}_\mathfrak{m}(t))_i = (\mathbf{X}(t))_i \quad \text{for} \quad i \in \mathfrak{m} .$$

Obviously, this marginal is an $\#\mathfrak{m}$-dimensional vector, but we do not refer to it as an element of $\mathbb{R}^{\#\mathfrak{m}}$ but choose

$$\mathbb{R}^\mathfrak{m} := \{\#\mathfrak{m}\text{-dim vectors or mappings } v \text{ with } v : \mathfrak{m} \to \mathbb{R}\} \simeq \mathbb{R}^{\#\mathfrak{m}}$$

instead. Similarly, we sometimes choose to write $\mathbb{R}^\mathfrak{D}$ for sequences with $\#\mathfrak{D} = d$ elements as an alternative to $\mathbb{R}^d$ even though they are isomorphic.

The process $\mathbf{X}_\mathfrak{m}$ is again Lévy with state space $\mathbb{R}^\mathfrak{m}$, but there is more to say about it. First, we choose the slightly modified truncation function

$$\mathcal{T}_\delta(\mathbf{z}) = \mathbb{1}_{\{|\mathbf{z}|_\infty \leq \delta\}}(\mathbf{z}) \quad \text{with} \quad \delta > 0 , \tag{2.7}$$

which has the product structure

$$\mathcal{T}_\delta(\mathbf{z}) = \mathbb{1}_{\{|\mathbf{z}|_\infty \leq \delta\}} = \mathbb{1}_{\{|\mathbf{z}_\mathfrak{m}|_\infty \leq \delta\}} \cdot \mathbb{1}_{\{|\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}|_\infty \leq \delta\}} \tag{2.8}$$

for $\mathfrak{m} \subset \mathfrak{D}$. There exist other truncation functions that allow for a product representation similar to (2.8), but (2.7) is relatively easy to handle and serves our purpose well.

The following theorem shows how to compute the triplet of the marginal Lévy process $\mathbf{X}_\mathfrak{m}$ subject to the truncation function $\mathcal{T}_\delta(\mathbf{x}) = \mathbb{1}_{\{|\mathbf{x}|_\infty \leq \delta\}}(\mathbf{x})$.

Theorem 2.2. *Let $\mathbf{X}$ be a Lévy process with triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$ and let $\mathbf{X}_\mathfrak{m}$ be the $\mathfrak{m}$-marginal of $\mathbf{X}$. Then, $\mathbf{X}_\mathfrak{m}$ is also a Lévy process with triplet $(\mathbf{Q}_\mathfrak{m}, \tilde{\boldsymbol{\theta}}_\mathfrak{m}, \nu_\mathfrak{m})$, where*

$$\mathbf{Q}_\mathfrak{m} \in \mathbb{R}^{\mathfrak{m}\times\mathfrak{m}} \quad \text{with} \quad (\mathbf{Q}_\mathfrak{m})_{ij} = q_{ij} \quad \text{for} \quad i,j \in \mathfrak{m}\,,$$

$$\tilde{\boldsymbol{\theta}}_\mathfrak{m} \in \mathbb{R}^{\mathfrak{m}} \quad \text{with} \quad (\tilde{\boldsymbol{\theta}}_\mathfrak{m})_i = \theta_i + \int_{[-\delta,\delta]^\mathfrak{m}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})} z_i \nu(\mathrm{d}\mathbf{z}) \quad \text{for} \quad i \in \mathfrak{m}\,,$$

and

$$\nu_\mathfrak{m}(A) = \nu(A\times\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}) \quad \text{for} \quad A \in \mathcal{B}(\mathbb{R}^\mathfrak{m})\,,$$

which means $\nu_\mathfrak{m}$ is the $\mathfrak{m}$-marginal of ν.

Proof. If $\mathbf{X}$ is a Lévy process, its marginal $\mathbf{X}_\mathfrak{m}$ is obviously again Lévy, see Definition 2.1, and we need to find the characteristic triplet of $\mathbf{X}_\mathfrak{m}$. Computing the characteristic function of the $\mathfrak{m}$-marginal of a random vector amounts to setting the dimensions that are marginalized out, i.e., $\mathcal{D}\setminus\mathfrak{m}$, to zero in the characteristic function. This carries over to the characteristic exponent $\psi_\mathfrak{m}$ of the process $\mathbf{X}_\mathfrak{m}$ with respect to the characteristic exponent ψ of $\mathbf{X}$. Independently of $t \geq 0$ and for all $\boldsymbol{\xi}_\mathfrak{m} \in \mathbb{R}^\mathfrak{m}$, it holds that

$$e^{t\psi_\mathfrak{m}(\boldsymbol{\xi}_\mathfrak{m})} = \mathbb{E}\left(e^{\mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{X}_\mathfrak{m}(t)\rangle}\right) = \mathbb{E}\left(e^{\mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m}\diamond\mathbf{0}_{\mathfrak{D}\setminus\mathfrak{m}},\mathbf{X}(t)\rangle}\right) = e^{t\psi(\boldsymbol{\xi}_\mathfrak{m}\diamond\mathbf{0}_{\mathfrak{D}\setminus\mathfrak{m}})}\,,$$

where

$$(\boldsymbol{\xi}_\mathfrak{m}\diamond\mathbf{0}_{\mathfrak{D}\setminus\mathfrak{m}})_i = \begin{cases}(\boldsymbol{\xi}_\mathfrak{m})_i & \text{for } i\in\mathfrak{m}\,,\\ 0 & \text{else}\,.\end{cases}$$

We conclude

$$\psi_\mathfrak{m}(\boldsymbol{\xi}_\mathfrak{m}) = \psi(\boldsymbol{\xi}_\mathfrak{m}\diamond\mathbf{0}_{\mathfrak{D}\setminus\mathfrak{m}})$$

and set all entries $\boldsymbol{\xi}_{\mathfrak{D}\setminus\mathfrak{m}}$ to 0 in (2.2), which yields

$$\begin{aligned}\psi(\boldsymbol{\xi}_\mathfrak{m}\diamond\mathbf{0}_{\mathfrak{D}\setminus\mathfrak{m}}) = &-\frac{1}{2}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{Q}_\mathfrak{m}\boldsymbol{\xi}_\mathfrak{m}\rangle + \mathrm{i}\langle\boldsymbol{\theta}_\mathfrak{m},\boldsymbol{\xi}_\mathfrak{m}\rangle\\ &+\int_{\mathbb{R}^d}\left(e^{\mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle} - 1 - \mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle\mathbb{1}_{\{|\mathbf{z}_\mathfrak{m}|_\infty\leq\delta\}}\cdot\mathbb{1}_{\left\{|\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}|_\infty\leq\delta\right\}}\right)\nu(\mathrm{d}\mathbf{z})\,. \qquad (2.9)\end{aligned}$$

The covariance matrix $\mathbf{Q}_\mathfrak{m}$ of the triplet of our $\mathfrak{m}$-marginal is simply a submatrix of $\mathbf{Q}$. We now focus on the integral term (2.9) and split the domain of integration

$$\begin{aligned}&\int_{\mathbb{R}^d}\left(e^{\mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle} - 1 - \mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle\mathbb{1}_{\{|\mathbf{z}_\mathfrak{m}|_\infty\leq\delta\}}\cdot\mathbb{1}_{\left\{|\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}|_\infty\leq\delta\right\}}\right)\nu(\mathrm{d}\mathbf{z})\\ =&\int_{\mathbb{R}^\mathfrak{m}\times[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}}}\left(e^{\mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle} - 1 - \mathrm{i}\langle\boldsymbol{\xi}_\mathfrak{m},\mathbf{z}_\mathfrak{m}\rangle\mathbb{1}_{\{|\mathbf{z}_\mathfrak{m}|_\infty\leq\delta\}}\cdot\underbrace{\mathbb{1}_{\left\{|\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}|_\infty\leq\delta\right\}}}_{=1}\right)\nu(\mathrm{d}\mathbf{z}) \qquad \text{p.t.o.}\end{aligned}$$

$$+\int_{\mathbb{R}^{\mathfrak{m}}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\Big(e^{\mathrm{i}\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\cdot\underbrace{\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}|_\infty\le\delta\}}}_{=0}\Big)\nu(\mathrm{d}\mathbf{z})$$

$$=\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{\mathrm{i}\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\Big)\int_{[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}}}\nu(\mathrm{d}\mathbf{z}) \tag{2.10}$$

$$+\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{\mathrm{i}\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1\Big)\int_{(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\nu(\mathrm{d}\mathbf{z})\,. \tag{2.11}$$

Now we want to recombine (2.10) and (2.11). To this end, we consider the term

$$\mathrm{i}\left\langle\boldsymbol{\xi}_{\mathfrak{m}},\int_{\mathbb{R}^{\mathfrak{m}}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\mathbf{z}_{\mathfrak{m}}\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\nu(\mathrm{d}\mathbf{z})\right\rangle \tag{2.12}$$

$$=\int_{\mathbb{R}^{\mathfrak{m}}}\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\int_{(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\nu(\mathrm{d}\mathbf{z}) \tag{2.13}$$

Note that (2.12) exists since $\big|\mathbf{z}_{\mathfrak{m}}\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\big|\le\sqrt{d}\delta<\infty$ and

$$\nu\big(\mathbb{R}^{\mathfrak{m}}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})\big)\le\nu(\mathbb{R}^{\mathfrak{D}}\setminus[-\delta,\delta]^{\mathfrak{D}})<\infty\,.$$

Now, we consider (2.10) and (2.11) again and subtract (2.13) and subsequently add (2.12). Then, we can recombine both integrals

$$\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{i\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\Big)\int_{[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}}}\nu(\mathrm{d}\mathbf{z})$$

$$+\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{i\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\Big)\int_{(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\nu(\mathrm{d}\mathbf{z})$$

$$+\mathrm{i}\left\langle\boldsymbol{\xi}_{\mathfrak{m}},\int_{\mathbb{R}^{\mathfrak{m}}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\mathbf{z}_{\mathfrak{m}}\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\nu(\mathrm{d}\mathbf{z})\right\rangle$$

$$=\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{i\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\Big)\int_{\substack{[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}}\cup\\(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}}\nu(\mathrm{d}\mathbf{z}) \tag{2.14}$$

$$+\mathrm{i}\left\langle\boldsymbol{\xi}_{\mathfrak{m}},\int_{\mathbb{R}^{\mathfrak{m}}\times(\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}\setminus[-\delta,\delta]^{\mathfrak{D}\setminus\mathfrak{m}})}\mathbf{z}_{\mathfrak{m}}\nu(\mathrm{d}\mathbf{z})\right\rangle\,. \tag{2.15}$$

Substituting (2.14) and (2.15) for (2.9) and adding (2.15) to $\boldsymbol{\theta}_{\mathfrak{m}}$, we finally get

$$\psi_{\mathfrak{m}}(\boldsymbol{\xi}_{\mathfrak{m}})=-\frac{1}{2}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{Q}_{\mathfrak{m}}\boldsymbol{\xi}_{\mathfrak{m}}\rangle+\mathrm{i}\left\langle\tilde{\boldsymbol{\theta}}_{\mathfrak{m}},\boldsymbol{\xi}_{\mathfrak{m}}\right\rangle$$
$$+\int_{\mathbb{R}^{\mathfrak{m}}}\Big(e^{i\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle}-1-\mathrm{i}\,\langle\boldsymbol{\xi}_{\mathfrak{m}},\mathbf{z}_{\mathfrak{m}}\rangle\,\mathbb{1}_{\{|\mathbf{z}_{\mathfrak{m}}|_\infty\le\delta\}}\Big)\nu_{\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{m}})\,,$$

which proves our theorem. □

As mentioned earlier, a non-trivial truncation function $\mathcal{T}$ is only necessary if there are many

large jumps such that $\mathbb{E}[|\mathbf{X}(t)|]$ does not exist and there are infinitely many small jumps such that $\nu([-1,1]^{\mathfrak{D}}) = \infty$. From now on, we assume that $\mathbb{E}[|\mathbf{X}(t)|] < \infty$ is well-defined and choose $\mathcal{T}(\mathbf{z}) = 1$. This choice has the benefit that $\boldsymbol{\theta} = \mathbb{E}[\mathbf{X}(1)]$, so the triplet directly tells us the expected value of the process per unit time. Note that this is our previous truncation function $\mathcal{T}_\delta(\mathbf{z})$ for $\delta \to \infty$. Then Theorem 2.2 yields that $\tilde{\boldsymbol{\theta}}_{\mathfrak{m}} \to \boldsymbol{\theta}_{\mathfrak{m}}$. This means that we no longer need to adjust the drift $\boldsymbol{\theta}_{\mathfrak{m}}$ when forming a marginal but can simply choose a subvector of $\boldsymbol{\theta}$. This is very intuitive and we get

Proposition 2.3. *Let $\mathcal{T}(\mathbf{z}) = 1$ be the truncation function and let $\mathbf{X}$ be a Lévy process with finite expected value and triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$. Then, the $\mathfrak{m}$-marginal $\mathbf{X}_{\mathfrak{m}}$ of $\mathbf{X}$ is again a Lévy process with triplet $(\mathbf{Q}_{\mathfrak{m}}, \boldsymbol{\theta}_{\mathfrak{m}}, \nu_{\mathfrak{m}})$, where*

$$\mathbf{Q}_{\mathfrak{m}} \in \mathbb{R}^{\mathfrak{m}\times\mathfrak{m}} \quad \textit{with} \quad (\mathbf{Q}_{\mathfrak{m}})_{ij} = (\mathbf{Q})_{ij} \quad \textit{for} \quad i,j \in \mathfrak{m}\,,$$

$$\boldsymbol{\theta}_{\mathfrak{m}} \in \mathbb{R}^{\mathfrak{m}} \quad \textit{with} \quad (\boldsymbol{\theta}_{\mathfrak{m}})_i = (\boldsymbol{\theta})_i \quad \textit{for} \quad i \in \mathfrak{m}\,,$$

and

$$\nu_{\mathfrak{m}}(A) = \nu(A \times \mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}) \quad \textit{for} \quad A \in \mathcal{B}(\mathbb{R}^{\mathfrak{m}})\,.$$

Option pricing formulation and transformations

A European option is the right but not the obligation to buy (call option) or sell (put option) a specified quantity of an underlying at time T in the future for a fixed price K. The Black-Scholes model [BS73] assumes that security prices follow a geometric Brownian motion

$$S(t) = S_0 \exp\left((\mu - \tfrac{\sigma^2}{2})t + \sigma W(t)\right)\,. \tag{2.16}$$

Then, arbitrage considerations show that the fair price of a European option $V(t,\mathbf{s})$ is given by the discounted expected value (2.1) of the payoff function h under the risk-neutral measure. For the geometric Brownian motion (2.16), the risk-neutral measure simply means that the true drift μ has to be replaced by the risk-free interest rate r.

In this thesis, we assume an exponentiated Lévy model for $\mathbf{S}(t)$ in (2.1), which reads

$$\mathbf{S}(t) = \mathbf{S}(0)\exp(rt + \mathbf{X}(t)) := (S_1(0)\exp(rt + X_1(t)), \ldots, S_d(0)\exp(rt + X_d(t))\,, \tag{2.17}$$

where $\mathbf{X}(t) = (X_1(t), \ldots, X_d(t))$ is a Lévy process with triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$. Note that the summation of rt and the exp-function are applied component-wise. Compared with the geometric Brownian motion, (2.17) is able to model jumps and thus can account for the leptokurtic feature of asset returns, so that theoretical prices match the market prices more accurately [CT04]. Note that under the risk-neutral measure, the processes

$$e^{-tr}S_i(t) = e^{X_i(t)} \quad \text{for} \quad i = 1, \ldots, d \tag{2.18}$$

need to be martingales with respect to the canonical filtration of the multivariate process $\mathbf{X}(t)$, see [RSW10]. A quick glance at the Lévy-Khintchin representation (2.2) for $\mathcal{T}(\mathbf{z}) = 1$ reveals

that this translates into

$$\begin{aligned} & \mathbb{E}[e^{X_j(t)}] = e^{t\psi(-\mathbf{i}\mathbf{e}_j)} = \exp\left(t\left(\frac{q_{jj}}{2} + \theta_j + \int_{\mathbb{R}} e^{z_j} - 1 - z_j\nu_j(\mathrm{d}z_j)\right)\right) \stackrel{!}{=} 1 \\ \Leftrightarrow \quad & \theta_j = -\frac{q_{jj}}{2} - \int_{\mathbb{R}} e^{z_j} - 1 - z_j\nu_j(\mathrm{d}z_j) \end{aligned} \tag{2.19}$$

for $i = 1, \dots, d$, where $\mathbf{e}_j$ is the j-th unit vector and ν_j is the $\{j\}$-marginal of ν. Jump-diffusion models lead to incomplete markets and a risk-neutral measure needs to be selected by, e.g., the rational expectations equilibrium, see [Kou07]. However, this is not the focus of this thesis.

We now want to compute (2.1) for a given payoff function h. In order to simplify matters, we first do a variable transform. Note that in the following the exp- and log-functions need to be understood component-wise.

Theorem 2.4. *We set*

$$u(\tau, \mathbf{x}) = e^{(T-t)r}V(t, \mathbf{s}) \;, \quad \tau = T - t \;, \quad \mathbf{x} = \log \mathbf{s} + r\tau \quad \textit{and} \quad g(\log \mathbf{s}) = h(\mathbf{s}) \tag{2.20}$$

for $t \geq 0$ and $\mathbf{x} \in \mathbb{R}^d, \mathbf{s} \in \mathbb{R}^d_{\geq 0}$. Then, we can express u in terms of the process $\mathbf{X}$ by

$$u(\tau, \mathbf{x}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})] \;. \tag{2.21}$$

Proof. The proof simply uses the equalities in (2.20), that is

$$\begin{aligned} u(\tau, \mathbf{x}) = e^{(T-t)r}V(t, \mathbf{s}) &= \mathbb{E}[h(\mathbf{S}(T)) \mid \mathbf{S}(t) = \mathbf{s}] \\ &= \mathbb{E}[h(\exp(rT + \mathbf{X}(T)) \mid \exp(rt + \mathbf{X}(t)) = \mathbf{s})] \\ &= \mathbb{E}[g(rT + \mathbf{X}(T)) \mid \mathbf{X}(t) = \log \mathbf{s} + rT - rt - rT] \\ &= \mathbb{E}[g(rT + \underbrace{\mathbf{X}(T) - \mathbf{X}(t)}_{\sim \mathbf{X}(T-t)} + \underbrace{\mathbf{X}(t)}_{=\mathbf{x}-rT}) \mid \mathbf{X}(t) = \mathbf{x} - rT] \\ &= \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})] \;, \end{aligned}$$

which is exactly our claim. □

We consider (2.21) to be more convenient than (2.1). In order to make approximation methods for u effective, a further reparameterization may be necessary. We now try to express (2.21) in terms of a process $\mathbf{Y}(\tau) = \mathbf{B}^T\mathbf{X}(\tau), \tau \geq 0$, where $\mathbf{B} \in \mathbb{R}^{d\times d}$ is an orthonormal matrix. Orthonormal transforms are frequently used for related problems [Oet11, IT09], as $|\det \mathbf{B}| = 1$ makes integral transformations particularly easy.

Theorem 2.5. *Let $\mathbf{B} \in \mathbb{R}^{d\times d}$ be orthonormal. We set*

$$v(\tau, \mathbf{y}) = u(\tau, \mathbf{x}) \;, \qquad \mathbf{y} = \mathbf{B}^T\mathbf{x} \quad \textit{and} \quad g_{\mathbf{B}}(\mathbf{y}) = g(\mathbf{x}) \;. \tag{2.22}$$

Then, we can express (2.21) in terms of $\mathbf{Y}$ and v by

$$v(\tau, \mathbf{y}) = \mathbb{E}[g_{\mathbf{B}}(\mathbf{Y}(\tau) + \mathbf{y})] \;. \tag{2.23}$$

Proof. The proof simply uses the equalities in (2.22), that is

$$v(\tau, \mathbf{y}) = u(\tau, \mathbf{B}\mathbf{y}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{B}\mathbf{y})] = \mathbb{E}[g(\mathbf{B}(\mathbf{B}^T\mathbf{X}(\tau) + \mathbf{y}))] = \mathbb{E}[g_{\mathbf{B}}(\mathbf{Y}(\tau) + \mathbf{y})] \,,$$

which proves our claim. □

The question is what matrix $\mathbf{B}$ results in a representation (2.23) that is more convenient than (2.21). Before we answer this, we need to have a look at the properties of $\mathbf{Y}$.

Theorem 2.6. *Let $\mathbf{B} \in \mathbb{R}^{d\times d}$ be an orthogonal matrix. If $\mathbf{X}$ is a Lévy process with triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$, the process $\mathbf{Y}(\tau) = \mathbf{B}^T\mathbf{X}(\tau), \tau \geq 0$, is also a Lévy process with characteristic triplet $(\mathbf{B}^T\mathbf{Q}\mathbf{B}, \mathbf{B}^T\boldsymbol{\theta}, \nu_{\mathbf{B}})$ and*

$$\nu_{\mathbf{B}}(A) = \nu(\mathbf{B}A) \quad \textit{for} \quad A \in \mathcal{B}(\mathbb{R}^d) \,,$$

where $\mathbf{B}A$ denotes the set A after a linear transformation by the matrix $\mathbf{B}$.

Proof. Clearly, the properties of Lévy processes from Definition 2.1 are satisfied for $\mathbf{Y}$ as well. We now obtain the characteristic exponent $\psi_{\mathbf{Y}}$ of $\mathbf{Y}$ from the characteristic exponent $\psi_{\mathbf{X}}$ of $\mathbf{X}$:

$$\begin{aligned} e^{t\psi_{\mathbf{Y}}(\boldsymbol{\xi})} &= \mathbb{E}\left(e^{i\langle\boldsymbol{\xi},\mathbf{Y}(t)\rangle}\right) = \mathbb{E}\left(e^{i\langle\boldsymbol{\xi},\mathbf{B}^T\mathbf{X}(t)\rangle}\right) = \mathbb{E}\left(e^{i\langle\mathbf{B}\boldsymbol{\xi},\mathbf{X}(t)\rangle}\right) = e^{t\psi_{\mathbf{X}}(\mathbf{B}\boldsymbol{\xi})} \\ &= \exp\left(t\left(-\frac{1}{2}\left\langle\boldsymbol{\xi}, \mathbf{B}^T\mathbf{Q}\mathbf{B}\boldsymbol{\xi}\right\rangle + i\left\langle\mathbf{B}^T\boldsymbol{\theta}, \boldsymbol{\xi}\right\rangle + \int_{\mathbb{R}^d}\left(e^{i\langle\boldsymbol{\xi},\mathbf{B}^T\mathbf{z}\rangle} - 1 - i\left\langle\boldsymbol{\xi}, \mathbf{B}^T\mathbf{z}\right\rangle\right)\nu(\mathrm{d}\mathbf{z})\right)\right) \,. \end{aligned} \tag{2.24}$$

Now, we change the integration measure from ν to $\nu_{\mathbf{B}}$ in (2.24) and thus substitute $\mathbf{B}^T\mathbf{z}$ by $\mathbf{z}$ in the integral. This transformation is simple, since $\mathbf{B}$ is orthonormal and hence $|\det\mathbf{B}| = 1$. Obviously, the result is again a Lévy-Khintchin representation with triplet $(\mathbf{B}^T\mathbf{Q}\mathbf{B}, \mathbf{B}^T\boldsymbol{\theta}, \nu_{\mathbf{B}})$. □

Remark 2.7. If $\mathbf{X}$ is a martingale, this also applies to $\mathbf{Y} = \mathbf{B}^T\mathbf{X}$. However, if $e^{\mathbf{X}}$ is a martingale, this does not necessarily hold for $e^{\mathbf{B}^T\mathbf{X}}$. We give a simple counterexample with

$$\mathbf{Q} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \,, \quad \boldsymbol{\theta} = \begin{pmatrix} -1 \\ -\frac{1}{2} \end{pmatrix} \,, \quad \mathbf{B} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \,, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \,.$$

For Lévy process $\mathbf{X}(\tau)$ with triplet $(\mathbf{Q}, \boldsymbol{\theta}, 0)$, we know that

$$\mathbb{E}[e^{\mathbf{X}(\tau)}] = \begin{pmatrix} e^{\tau\psi(-\mathrm{i}\mathbf{e}_1)} \\ e^{\tau\psi(-\mathrm{i}\mathbf{e}_2)} \end{pmatrix} = \begin{pmatrix} e^{\tau(\frac{1}{2}2-1)} \\ e^{\tau(\frac{1}{2}1-\frac{1}{2})} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \,.$$

However, the process $\mathbf{Y} = \mathbf{B}^T\mathbf{X}$ has the triplet $(\mathbf{B}^T\mathbf{Q}\mathbf{B}, \mathbf{B}^T\boldsymbol{\theta}, 0)$ with

$$\mathbf{B}^T\mathbf{Q}\mathbf{B} = \boldsymbol{\Sigma} \quad \text{and} \quad \mathbf{B}^T\boldsymbol{\theta} = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$$

and thus

$$\mathbb{E}[e^{\mathbf{Y}(\tau)}] = \begin{pmatrix} e^{\tau\psi_{\mathbf{Y}}(-\mathrm{i}\mathbf{e}_1)} \\ e^{\tau\psi_{\mathbf{Y}}(-\mathrm{i}\mathbf{e}_2)} \end{pmatrix} = \begin{pmatrix} e^{\tau(\frac{1}{2}1-\frac{1}{2})} \\ e^{\tau(\frac{1}{2}2+1)} \end{pmatrix} = \begin{pmatrix} 1 \\ e^{2\tau} \end{pmatrix} \,.$$

Obviously, the process $e^{\mathbf{B}^T\mathbf{X}}$ is no martingale.

The question is now what $\mathbf{B} \in \mathbb{R}^{d\times d}$ results in a favorable process $\mathbf{Y}$. One obvious choice is the matrix which diagonalizes $\mathbf{Q}$ by

$$\mathbf{Q} = \mathbf{B}\mathbf{\Sigma}\mathbf{B}^T \Leftrightarrow \mathbf{B}^T\mathbf{Q}\mathbf{B} = \mathbf{\Sigma}\,, \tag{2.25}$$

where $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1^2, \dots, \sigma_d^2)$ and $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_d^2 > 0$. In this case, the columns of $\mathbf{B}$ are the associated eigenvectors to the eigenvalues $(\sigma_i^2)_{i=1}^d$. Now, any decay in the spectrum of $\mathbf{Q}$ is directly visible in the diffusion part of $\mathbf{Y}$, which has triplet $(\mathbf{\Sigma}, \mathbf{B}^T\boldsymbol{\theta}, \nu_{\mathbf{B}})$. As we will see in Chapter 6, the rate of decay in the covariance spectrum directly relates to the accuracy of ANOVA approximations.

2.1.2 The corresponding backward Kolmogorov equation

Having recalled Lévy processes and the option pricing problem, we now come to the backward Kolmogorov equation. We reproduce the transformations we made to get from $V(t, \mathbf{s})$ to $v(\tau, \mathbf{y})$ in the context of PIDEs.

The backward Kolmogorov equation for pricing options

The function V is known to satisfy the PIDE (2.27) and (2.28) in the following theorem, see [CT04, RSW10] for proofs and further information.

Theorem 2.8 (BKE for European options)**.** *Let* $\mathbf{S}$ *be an exponential Lévy model* (2.17) *with Lévy triplet* $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$*, which has a non-vanishing diffusion matrix* $\mathbf{Q}$*, and let* ν *satisfy*

$$\int_{|\mathbf{z}|\geq 1} e^{z_i}\nu(\mathrm{d}\mathbf{z}) < \infty \tag{2.26}$$

for $i = 1, \dots, d$*. Then, the function*

$$V \in C^{1,2}\left((0,T)\times\mathbb{R}^d_{>0}\right) \cap C^0\left([0,T]\times\mathbb{R}^d_{\geq 0}\right)$$

given by (2.1) *is a solution of the backward PIDE for European options*

$$\frac{\partial V}{\partial t}(t,\mathbf{s}) + \frac{1}{2}\sum_{i,j=1}^d s_i s_j q_{ij}\frac{\partial^2 V}{\partial s_i \partial s_j}(t,\mathbf{s}) + r\sum_{i=1}^d s_i \frac{\partial V}{\partial s_i}(t,\mathbf{s}) - rV(t,\mathbf{s}) \tag{2.27}$$

$$+ \int_{\mathbb{R}^d}\left(V(t,\mathbf{s}e^{\mathbf{z}}) - V(t,\mathbf{s}) - \sum_{i=1}^d s_i\left(e^{z_i}-1\right)\frac{\partial V}{\partial s_i}(t,\mathbf{s})\right)\nu(\mathrm{d}\mathbf{z}) = 0 \tag{2.28}$$

on $(0,T)\times\mathbb{R}^d_{>0}$*, where* $V(t,\mathbf{s}e^{\mathbf{z}}) := V(t, s_1e^{z_1}, \dots, s_de^{z_d})$*,* $\mathbf{s} = (s_1, \dots, s_d)$ *and* $\mathbf{z} = (z_1, \dots, z_d)$*, and where the final condition is given by*

$$V(T,\mathbf{s}) = h(\mathbf{s}) \quad \textit{for} \quad \mathbf{s} \in \mathbb{R}^d_{\geq 0}\,. \tag{2.29}$$

Transformation to the infinitesimal generator

Let us define the transition operator

$$[P_t f](\mathbf{x}) = \mathbb{E}[f(\mathbf{x} + \mathbf{X}(t))] \, .$$

The infinitesimal generator $L^{\mathbf{X}}$ of the process $\mathbf{X}$ is then defined as

$$L^{\mathbf{X}} f = \lim_{t \downarrow 0} \frac{P_t f - f}{t} \, .$$

Obviously, (2.21) means that

$$u(\tau, \mathbf{x}) = [P_\tau g](\mathbf{x})$$

and thus

$$\frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) = [L^{\mathbf{X}} u](\tau, \mathbf{x}) \, .$$

In the next theorem, we recover the infinitesimal generator of $\mathbf{X}$ by expressing our BKE in terms of $u(\tau, \mathbf{x})$. This is a useful transformation as we also eliminate all non-constant coefficients.

Theorem 2.9. *Using the transformations $u(\tau, \mathbf{x}) = e^{r(T-t)} V(t, \mathbf{s})$ with $\tau = T - t$ and $\mathbf{x} = \log \mathbf{s} + r\tau$, we can rewrite the BKE (2.27) and (2.28) as*

$$\frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) = [L^{\mathbf{X}} u](\tau, \mathbf{x}) \, , \tag{2.30}$$

where

$$[L^{\mathbf{X}} u](\tau, \mathbf{x}) = \frac{1}{2} \sum_{i,j=1}^{d} q_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) + \sum_{i=1}^{d} \theta_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \tag{2.31}$$

$$+ \int_{\mathbb{R}^d} \left(u(\tau, \mathbf{x} + \mathbf{z}) - u(\tau, \mathbf{x}) - \sum_{i=1}^{d} z_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \right) \nu(\mathrm{d}\mathbf{z}) \tag{2.32}$$

is the infinitesimal generator of $\mathbf{X}$. The final condition $V(T, \mathbf{s}) = h(\mathbf{s})$ becomes an initial condition $u(0, \mathbf{x}) = g(\mathbf{x})$ with $g(\log \mathbf{s}) = h(\mathbf{s})$.

Proof. We do a step by step transformation of the PIDE. Because of $u(\tau, \mathbf{x}) = e^{r(T-t)} V(t, \mathbf{s})$,

$$\frac{\mathrm{d}}{\mathrm{d}t} u(\tau, \mathbf{x}) = -\frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) - r \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \, ,$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t} e^{r(T-t)} V(t, \mathbf{s}) = -r e^{r(T-t)} V(t, \mathbf{s}) + e^{r(T-t)} \frac{\partial V}{\partial t}(t, \mathbf{s})$$

we get

$$\frac{\partial V}{\partial t}(t, \mathbf{s}) = -e^{-r(T-t)} \frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) - r e^{-r(T-t)} \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) + r e^{-r(T-t)} u(\tau, \mathbf{x}) \, . \tag{2.33}$$

By replacing $\frac{\partial V}{\partial t}(t, \mathbf{s})$ with the right-hand side of (2.33), the reactive term

$$-rV(t, \mathbf{s}) = -re^{-r(T-t)}u(\tau, \mathbf{x})$$

of (2.27) cancels out. Now, we come to the convection and diffusion part. Because of

$$\frac{\mathrm{d}}{\mathrm{d}s_i}e^{-r(T-t)}u(\tau, \mathbf{x}) = e^{-r(T-t)}\frac{1}{s_i}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \tag{2.34}$$

we get

$$s_i\frac{\partial V}{\partial s_i}(t, \mathbf{s}) = e^{-r(T-t)}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \ . \tag{2.35}$$

This means that the terms

$$r\sum_{i=1}^{d} s_i\frac{\partial V}{\partial s_i}(t, \mathbf{s}) = re^{-r(T-t)}\sum_{i=1}^{d}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x})$$

in (2.27) also cancel out with the respective terms on the right hand side of (2.33). Differentiating (2.34) with respect to $\frac{\mathrm{d}}{\mathrm{d}s_i}$ again yields

$$\frac{\mathrm{d}}{\mathrm{d}s_i}\frac{\mathrm{d}}{\mathrm{d}s_i}e^{-r(T-t)}u(\tau, \mathbf{x}) = e^{-r(T-t)}\frac{1}{s_i^2}\frac{\partial^2 u}{\partial x_i^2}(\tau, \mathbf{x}) - e^{-r(T-t)}\frac{1}{s_i^2}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \ ,$$

which results in

$$s_i^2\frac{\partial^2 V}{\partial s_i^2}(t, \mathbf{s}) = e^{-r(T-t)}\frac{\partial^2 u}{\partial x_i^2}(\tau, \mathbf{x}) - e^{-r(T-t)}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \ , \tag{2.36}$$

and differentiating with respect to $\frac{\mathrm{d}}{\mathrm{d}s_j}$ for $j \neq i$ results in

$$s_i s_j\frac{\partial^2 V}{\partial s_i \partial s_j}(t, \mathbf{s}) = e^{-r(T-t)}\frac{\partial^2 u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) \ . \tag{2.37}$$

Using (2.33), (2.35), (2.36) and (2.37), the line (2.27) transforms to

$$e^{-r(T-t)}\left(-\frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^{d} q_{ij}\frac{\partial^2 u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) - \sum_{i=1}^{d}\frac{q_{ii}}{2}\frac{\partial u}{\partial x_i}(\tau, \mathbf{x})\right) \ , \tag{2.38}$$

and we are left to deal with the integral term in (2.28). It is easy to see that

$$V(t, \mathbf{s}e^{\mathbf{z}}) = e^{-r(T-t)}u(\tau, \log \mathbf{s} + \mathbf{z} + r\tau) = e^{-r(T-t)}u(\tau, \mathbf{x} + \mathbf{z})$$

and in combination with (2.35) we can express (2.28) as

$$e^{-r(T-t)}\int_{\mathbb{R}^d}\left(u(\tau, \mathbf{x} + \mathbf{z}) - u(\tau, \mathbf{x}) - \sum_{i=1}^{d}(e^{z_i} - 1)\frac{\partial u}{\partial x_i}(\tau, \mathbf{x})\right)\nu(\mathrm{d}\mathbf{z}) \qquad \text{p.t.o.}$$

$$=e^{-r(T-t)} \int_{\mathbb{R}^d} \left(u(\tau, \mathbf{x}+\mathbf{z}) - u(\tau, \mathbf{x}) - \sum_{i=1}^{d} z_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \right) \nu(\mathrm{d}\mathbf{z}) \tag{2.39}$$

$$- e^{-r(T-t)} \sum_{i=1}^{d} \int_{\mathbb{R}} (e^{z_i} - 1 - z_i)\, \nu_i(\mathrm{d}z_i) \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \,. \tag{2.40}$$

Now, we combine (2.38), (2.39) and (2.40) and multiply the equation by $e^{r(T-t)}$. After a bit of reordering this yields

$$\begin{aligned} &- \frac{\partial u}{\partial \tau}(\tau, \mathbf{x}) + \frac{1}{2} \sum_{i,j=1}^{d} q_{ij} \frac{\partial u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) \\ &+ \sum_{i=1}^{d} \left(-\frac{q_{ii}}{2} - \int_{\mathbb{R}} (e^{z_i} - 1 - z_i)\, \nu_i(\mathrm{d}z_i) \right) \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \\ &+ \int_{\mathbb{R}^d} \left(u(\tau, \mathbf{x}+\mathbf{z}) - u(\tau, \mathbf{x}) - \sum_{i=1}^{d} z_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \right) \nu(\mathrm{d}\mathbf{z}) = 0 \end{aligned} \tag{2.41}$$

We then add $+\frac{\partial u}{\partial \tau}(\tau, \mathbf{x})$ to both sides of the equation and in (2.41) we substitute (2.19) by θ_j for $j = 1, \ldots, d$. What is left is to deal with the final condition at $t = T$ or the initial condition at $\tau = 0$, respectively. Obviously,

$$u(0, \mathbf{x}) = e^{r \cdot 0} V(T, \exp(\mathbf{x} - r \cdot 0)) = h(\exp(\mathbf{x})) = g(\mathbf{x}) \,,$$

which proves the theorem. □

Diagonalization of the data covariance matrix

We now diagonalize the covariance matrix $\mathbf{Q}$ by a further coordinate transform $\mathbf{x} = \mathbf{B}\mathbf{y}$ of (2.30), where $\mathbf{B}$ is an orthonormal matrix and

$$\mathbf{Q} = \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^T \Leftrightarrow \mathbf{B}^T\mathbf{Q}\mathbf{B} = \boldsymbol{\Sigma} \,.$$

This transformation and the subsequent truncation of small eigenvalues have already been discussed in [NHW10].

Theorem 2.10. *The PIDE* (2.30) *can be rewritten as*

$$\frac{\partial v}{\partial \tau} v(\tau, \mathbf{y}) = [L^{\mathbf{Y}} v](\tau, \mathbf{y}) \,, \tag{2.42}$$

where $L^{\mathbf{Y}}$ *is the infinitesimal generator* (2.31) *and* (2.32) *of the process* $\mathbf{Y}(\tau) = \mathbf{B}^T\mathbf{X}(\tau)$ *with triplet* $(\boldsymbol{\Sigma}, \mathbf{B}^T\boldsymbol{\theta}, \nu_{\mathbf{B}})$. *Here,* $\nu_{\mathbf{B}}$ *is the measure with*

$$\nu_{\mathbf{B}}(A) = \nu(\mathbf{B}A) \quad \textit{for} \quad A \in \mathcal{B}(\mathbb{R}^d) \,,$$

where $\mathbf{B}A$ *is the set* A *under the linear transformation* $\mathbf{B}$. *The initial condition is given by*

$v(0, \mathbf{y}) = g_{\mathbf{B}}(\mathbf{y}) := g(\mathbf{B}\mathbf{y})$.

Proof. The variable transform does not affect the time derivative. The first-order derivatives change to

$$\frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) = \sum_{k=1}^{d} b_{ik} \frac{\partial v}{\partial y_k}(\tau, \mathbf{y})$$

for $i = 1, \ldots, d$ and thus

$$\sum_{i=1}^{d} \theta_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) = \sum_{i=1}^{d} \theta_i \sum_{k=1}^{d} b_{ik} \frac{\partial v}{\partial y_k}(\tau, \mathbf{y}) = \sum_{k=1}^{d} \left(\sum_{i=1}^{d} \theta_i b_{ik} \right) \frac{\partial v}{\partial y_k}(\tau, \mathbf{y}) \, .$$

Obviously, the drift vector $\boldsymbol{\theta}$ of $\mathbf{X}(t)$ results in a drift $\mathbf{B}^T\boldsymbol{\theta}$ of $\mathbf{Y}(t)$. The second-order derivatives become

$$\frac{\partial^2 u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) = \sum_{k,l=1}^{d} b_{ik} b_{jl} \frac{\partial^2 v}{\partial y_k \partial y_l}(\tau, \mathbf{y}) \, .$$

for $i, j = 1, \ldots, d$. Now we add up all second order terms that appear in (2.31)

$$\begin{aligned}
\sum_{i,j=1}^{q} q_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j}(\tau, \mathbf{x}) &= \sum_{i,j,k,l=1}^{d} q_{ij} b_{ik} b_{jl} \frac{\partial^2 v}{\partial y_k \partial y_l}(\tau, \mathbf{y}) \\
&= \sum_{i,k,l=1}^{d} b_{ik} (\mathbf{Q}\mathbf{B})_{il} \frac{\partial^2 v}{\partial y_k \partial y_l}(\tau, \mathbf{y}) \\
&= \sum_{k,l=1}^{d} (\mathbf{B}^T\mathbf{Q}\mathbf{B})_{lk} \frac{\partial^2 v}{\partial y_k \partial y_l}(\tau, \mathbf{y}) \\
&= \sum_{k=1}^{d} \sigma_k^2 \frac{\partial^2 v}{\partial y_k^2}(\tau, \mathbf{y}) \, .
\end{aligned}$$

We see that the covariance matrix of the diffusion part of $\mathbf{Y}(t)$ is $\mathbf{B}^T\mathbf{Q}\mathbf{B} = \boldsymbol{\Sigma}$. Now, we turn to the jump part (2.32)

$$\begin{aligned}
&\int_{\mathbb{R}^d} \left(u(\tau, \mathbf{x} + \mathbf{z}) - u(\tau, \mathbf{x}) - \sum_{i=1}^{d} z_i \frac{\partial u}{\partial x_i}(\tau, \mathbf{x}) \right) \nu(\mathrm{d}\mathbf{z}) \\
&= \int_{\mathbb{R}^d} \left(v(\tau, \mathbf{B}^T\mathbf{x} + \mathbf{B}^T\mathbf{z}) - v(\tau, \mathbf{y}) - \sum_{i=1}^{d} z_i \sum_{k=1}^{d} b_{ik} \frac{\partial v}{\partial y_k}(\tau, \mathbf{y}) \right) \nu(\mathrm{d}\mathbf{z}) \\
&= \int_{\mathbb{R}^d} \left(v(\tau, \mathbf{y} + \mathbf{B}^T\mathbf{z}) - v(\tau, \mathbf{y}) - \sum_{k=1}^{d} (\mathbf{B}^T\mathbf{z})_k \frac{\partial v}{\partial y_k}(\tau, \mathbf{y}) \right) \nu(\mathrm{d}\mathbf{z}) \qquad (2.43)
\end{aligned}$$

Now, we change the integration measure from ν to $\nu_{\mathbf{B}}$ in (2.43) and thus substitute $\mathbf{B}^T\mathbf{z} \to \mathbf{z}$

in the integral. Finally, the initial condition changes to

$$v(0, \mathbf{y}) = u(0, \mathbf{B}\mathbf{y}) = g(\mathbf{B}\mathbf{y}) = g_{\mathbf{B}}(\mathbf{y}) \ .$$

This proves the theorem. □

Remark 2.11. For non-jump processes, choosing an orthonormal matrix $\mathbf{B}$ that diagonalizes $\mathbf{Q}$ reveals the decay in the spectrum of $\mathbf{Q}$, but depending on the jump measure ν there might be better choices available. However, they would most likely not eliminate all mixed partial derivatives in (2.31), so we do not follow this route. See [Oet11] for a related optimization problem.

2.2 The ANOVA decomposition

The d-dimensional ANOVA decomposition for functions has successfully been employed in a variety of fields, e.g., finance [GH10b] and molecular dynamics [Ham10]. In the following subsection, we give a self-contained rigorous description of the method that accounts for the generalizations used in both references. In Subsection 2.2.2, we give a short outlook on a non-linear extension of the ANOVA decomposition.

2.2.1 Definition

Let $\mu_i, i = 1, \dots, d$, be measures on $\mathbb{R}$. We define the product measure

$$\mu_{\mathfrak{m}} := \bigotimes_{i \in \mathfrak{m}} \mu_i \tag{2.44}$$

for all subsets

$$\mathfrak{m} \subset \mathfrak{D} = \{1, \dots, d\} \ .$$

We focus on d-variate functions in the Hilbert space $L_2(\mu_{\mathfrak{D}})$ with inner product

$$(u, v)_{\mathfrak{D}} := \int_{\mathbb{R}^{\mathfrak{D}}} u(\mathbf{x}) v(\mathbf{x}) \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) \ .$$

Analogously, the inner products on the spaces $L_2(\mu_{\mathfrak{m}})$ are then

$$(u, v)_{\mathfrak{m}} := \int_{\mathbb{R}^{\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}}) v(\mathbf{x}_{\mathfrak{m}}) \mu_{\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}})$$

for $\mu_{\mathfrak{m}}$-measurable functions u, v and $\mathbf{x}_{\mathfrak{m}} = (x_i)_{i \in \mathfrak{m}}$. Functions in $L_2(\mu_{\mathfrak{m}})$ are only $\#\mathfrak{m}$-variate, and we would like to embed them in $L_2(\mu_{\mathfrak{D}})$. To this end, we need to introduce some notation. We can split $\mathbf{x} = (x_1, \dots, x_d)$ into $\mathbf{x}_{\mathfrak{m}} = (x_i)_{i \in \mathfrak{m}}$ and $\mathbf{x}_{\mathfrak{D} \setminus \mathfrak{m}} = (x_i)_{i \in \mathfrak{D} \setminus \mathfrak{m}}$, and later on we also need the concatenation $\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{x}_{\mathfrak{D} \setminus \mathfrak{m}} = \mathbf{x} \in \mathbb{R}^{\mathfrak{D}}$.

Now, we choose *unit functions* $\gamma_i \in L_2(\mu_i)$ for all $i = 1, \dots, d$ and define

$$V_{\mathfrak{m}} := \{v \in L_2(\mu_{\mathfrak{D}}) : \exists v_{\mathfrak{m}} \in L_2(\mu_{\mathfrak{m}}), v(\mathbf{x}) = v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D} \setminus \mathfrak{m}}(\mathbf{x}_{\mathfrak{D} \setminus \mathfrak{m}})\} \ , \tag{2.45}$$

where $\gamma_{\mathfrak{n}} : \mathbb{R}^{\mathfrak{n}} \to \mathbb{R}$ is given by the product

$$\gamma_{\mathfrak{n}}(\mathbf{x}_{\mathfrak{n}}) := \prod_{i \in \mathfrak{n}} \gamma_i(x_i) \ . \tag{2.46}$$

Obviously, $V_{\mathfrak{D}} = L_2(\mu_{\mathfrak{D}})$. Note, that the $V_{\mathfrak{m}} \subset V_{\mathfrak{D}}, \mathfrak{m} \subset \mathfrak{D}$, are linear subspaces, since for the functions $v = v_{\mathfrak{m}} \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}} \in V_{\mathfrak{m}}$, $u = u_{\mathfrak{m}} \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}} \in V_{\mathfrak{m}}$, we directly see that

$$\begin{aligned} \alpha u(\mathbf{x}) + v(\mathbf{x}) &= \alpha u_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) + v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \\ &= (\alpha u_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) + v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}})) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \in V_{\mathfrak{m}} \ . \end{aligned}$$

We are now interested in the projections $Q_{\mathfrak{m}}^V : V_{\mathfrak{D}} \to V_{\mathfrak{m}}$, which, for $u \in V_{\mathfrak{D}}$, satisfy

$$(Q_{\mathfrak{m}}^V u, v) = (u, v) \qquad \forall\, v \in V_{\mathfrak{m}} \ . \tag{2.47}$$

Theorem 2.12. *Given the Hilbert space $V_{\mathfrak{D}}$, and the constants*

$$c_{\mathfrak{n}} := \int_{\mathbb{R}^{\mathfrak{n}}} \gamma_{\mathfrak{n}}^2(\mathbf{x}_{\mathfrak{n}}) \mu_{\mathfrak{n}}(\mathrm{d}\mathbf{x}_{\mathfrak{n}}) = \prod_{i \in \mathfrak{n}} \int_{\mathbb{R}} \gamma_i^2(x_i) \mu_i(\mathrm{d}x_i) \ , \tag{2.48}$$

the functional

$$[Q_{\mathfrak{m}}^V u](\mathbf{x}) := c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \tag{2.49}$$

is the projection $Q_{\mathfrak{m}}^V : V_{\mathfrak{D}} \to V_{\mathfrak{m}}, \mathfrak{m} \subset \mathfrak{D}$, that satisfies (2.47).

Proof. First, we have to show that $Q_{\mathfrak{m}}^V u \in V_{\mathfrak{m}}$, which basically means that we have to check that the integral in (2.49) is in $L_2(\mu_{\mathfrak{m}})$. With Cauchy-Schwarz we see that

$$\begin{aligned} & \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \\ \leq\ & c_{\mathfrak{D}\setminus\mathfrak{m}}^{1/2} \Big(\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u^2(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \Big)^{1/2} \end{aligned}$$

and due to the monotonicity of integration, we get

$$\begin{aligned} & \int_{R^{\mathfrak{m}}} \Big(\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \Big)^2 \mu_{\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}}) \\ \leq\ & c_{\mathfrak{D}\setminus\mathfrak{m}} \int_{R^{\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u^2(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}}) \\ =\ & c_{\mathfrak{D}\setminus\mathfrak{m}} \cdot \|u\|_{\mathfrak{D}}^2 < \infty \ , \end{aligned} \tag{2.50}$$

and we can conclude that $Q_{\mathfrak{m}}^V u \in V_{\mathfrak{m}}$.

Now we have to show that $Q_{\mathfrak{m}}^V u$ also satisfies (2.47). For $v(\mathbf{x}) = v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \in V_{\mathfrak{m}}$,

we have

$$\begin{aligned}(Q_{\mathfrak{m}}^V u, v) &= \int_{\mathbb{R}^{\mathfrak{D}}} c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \cdot \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \\ &\quad \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) \\ &= c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \cdot \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \gamma_{\mathfrak{D}\setminus\mathfrak{m}}^2(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \\ &\quad \cdot \int_{\mathbb{R}^{\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \mu_{\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}}) \ .\end{aligned}$$

Further simplification gives

$$\begin{aligned}(Q_{\mathfrak{m}}^V u, v_{\mathfrak{m}}) &= \int_{\mathbb{R}^{\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}}) \\ &= \int_{\mathbb{R}^{\mathfrak{D}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot v_{\mathfrak{m}}(\mathbf{x}_{\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}}(\mathrm{d}(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}})) \\ &= (u, v_{\mathfrak{m}}) \ ,\end{aligned}$$

which proves the Theorem. □

We want to approximate $\#\mathfrak{D}$-variate functions $u \in V_{\mathfrak{D}}$ by functions from $V_{\mathfrak{m}}$ with $\#\mathfrak{m} \ll \#\mathfrak{D}$ only. Since we are in a Hilbert-space setting, the projection $Q_{\mathfrak{m}}^V$ from Theorem 2.12 is also the best-approximation in the space $V_{\mathfrak{m}}$. The representation of high-dimensional functions by a superposition of functions from $V_{\mathfrak{m}}$ and $V_{\mathfrak{n}}$, the best-approximation in, e.g., $V_{\mathfrak{m}} \cup V_{\mathfrak{n}}$ with $\mathfrak{m} \not\subseteq \mathfrak{n}$ and $\mathfrak{n} \not\subseteq \mathfrak{m}$, requires a little extra effort and leads to so-called orthogonal complement spaces. Note that in the following passage, we implicitly make use of the isomorphism

$$L_2(\mu_{\mathfrak{D}}) \simeq \bigotimes_{i \in \mathfrak{D}} L_2(\mu_i) \tag{2.51}$$

to switch to tensor product notation when more convenient.

We define the orthogonal complements

$$W_{\mathfrak{m}} := \bigotimes_{i \in \mathfrak{m}} (L_2(\mu_i) \ominus \operatorname{span}\{\gamma_i\}) \otimes \bigotimes_{i \in \mathfrak{D}\setminus\mathfrak{m}} \operatorname{span}\{\gamma_i\} \ ,$$

where $L_2(\mu_i) \ominus \operatorname{span}\{\gamma_i\}$ is simply the $(\cdot,\cdot)_{\{i\}}$-orthogonal complement of $\operatorname{span}\{\gamma_i\}$ in $L_2(\mu_i)$. Thus,

$$W_{\mathfrak{m}} \perp W_{\mathfrak{n}} \quad \text{for} \quad \mathfrak{m} \neq \mathfrak{n} \tag{2.52}$$

and furthermore

$$\begin{aligned}V_{\mathfrak{m}} &= \bigotimes_{i \in \mathfrak{m}} (L_2(\mu_i) \ominus \operatorname{span}\{\gamma_i\} \oplus \operatorname{span}\{\gamma_i\}) \otimes \bigotimes_{i \in \mathfrak{D}\setminus\mathfrak{m}} \operatorname{span}\{\gamma_i\} \\ &= \bigoplus_{\mathfrak{n} \subset \mathfrak{m}} \bigotimes_{i \in \mathfrak{n}} (L_2(\mu_i) \ominus \operatorname{span}\{\gamma_i\}) \otimes \bigotimes_{i \in \mathfrak{D}\setminus\mathfrak{n}} \operatorname{span}\{\gamma_i\} = \bigoplus_{\mathfrak{n} \subset \mathfrak{m}} W_{\mathfrak{n}} \ .\end{aligned}$$

For $\mathfrak{m} = \mathfrak{D}$, we see that we can uniquely decompose any function

$$u \in V_{\mathfrak{D}} = \bigoplus_{\mathfrak{m}\subset\mathfrak{D}} W_{\mathfrak{m}} \quad \text{into} \quad u = \sum_{\mathfrak{m}\subset\mathfrak{D}} u_{\mathfrak{m}} \quad \text{with} \quad u_{\mathfrak{m}} \in W_{\mathfrak{m}}, \mathfrak{m} \subset \mathfrak{D} \, . \tag{2.53}$$

Moreover, we can conclude that

$$W_{\mathfrak{p}} \perp V_{\mathfrak{m}} = \sum_{\mathfrak{n}\subset\mathfrak{m}} W_{\mathfrak{n}} \quad \text{for} \quad \mathfrak{p} \not\subset \mathfrak{m} \, . \tag{2.54}$$

The next Lemma shows how we obtain the best-approximation in the $W_{\mathfrak{m}}$ spaces.

Lemma 2.13. *The orthogonal projection $Q_{\mathfrak{m}}^W : V \to W_{\mathfrak{m}}$ is given by the functional*

$$Q_{\mathfrak{m}}^W := \sum_{\mathfrak{n}\subset\mathfrak{m}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V \, . \tag{2.55}$$

Proof. It is easy to see that $Q_{\mathfrak{m}}^W u \in V_{\mathfrak{m}}$. By establishing the orthogonality to all $W_{\mathfrak{p}}, \mathfrak{p} \subsetneq \mathfrak{m}$, we know that $Q_{\mathfrak{m}}^W u \in W_{\mathfrak{m}}$. So, for any $\mathfrak{p} \subsetneq \mathfrak{m}$, we pick an index $i \in \mathfrak{m}$ with $i \notin \mathfrak{p}$. Then, for all $w_{\mathfrak{p}} \in W_{\mathfrak{p}}$ it holds that

$$\Big(\sum_{\mathfrak{n}\subset\mathfrak{m}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u, w_{\mathfrak{p}} \Big) = \Big(\sum_{\substack{\mathfrak{n} \text{ with} \\ \mathfrak{p}\subset\mathfrak{n}\subset\mathfrak{m}}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u, w_{\mathfrak{p}} \Big) \tag{2.56}$$

$$= \Big(\sum_{\substack{\mathfrak{n} \text{ with} \\ \mathfrak{p}\subset\mathfrak{n}\subset\mathfrak{m}\setminus\{i\}}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u + \sum_{\substack{\mathfrak{n} \text{ with} \\ \mathfrak{p}\cup\{i\}\subset\mathfrak{n}\subset\mathfrak{m}}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u, w_{\mathfrak{p}} \Big) \tag{2.57}$$

$$= \Big(\sum_{\substack{\mathfrak{n} \text{ with} \\ \mathfrak{p}\subset\mathfrak{n}\subset\mathfrak{m}\setminus\{i\}}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u - \sum_{\substack{\mathfrak{n}' \text{ with} \\ \mathfrak{p}\subset\mathfrak{n}'\subset\mathfrak{m}\setminus\{i\}}} (-1)^{\#\mathfrak{m}-|\mathfrak{n}'|} Q_{\mathfrak{n}'\cup\{i\}}^V u, w_{\mathfrak{p}} \Big) = 0 \, , \tag{2.58}$$

where we used the orthogonality relation (2.54) in (2.56), split the summation in two parts in (2.57), set $\mathfrak{n}' := \mathfrak{n} \setminus \{i\}$ and used the projection property

$$(Q_{\mathfrak{n}\cup\{i\}}^V u, w_{\mathfrak{p}}) = (u, w_{\mathfrak{p}}) = (Q_{\mathfrak{n}}^V u, w_{\mathfrak{p}})$$

in (2.58). So we know that $Q_{\mathfrak{m}}^W u \in W_{\mathfrak{m}}$. We still need to show that this is a projection onto $W_{\mathfrak{m}}$ by

$$(Q_{\mathfrak{m}}^W u, w_{\mathfrak{m}}) = \Big(\sum_{\mathfrak{n}\subset\mathfrak{m}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u, w_{\mathfrak{p}} \Big) = (Q_{\mathfrak{n}}^V u, w_{\mathfrak{m}}) = (u, w_{\mathfrak{m}})$$

for all $w_{\mathfrak{m}} \in W_{\mathfrak{m}}$, where we exploited the orthogonality of $w_{\mathfrak{m}}$ to any function from a space $V_{\mathfrak{n}}, \mathfrak{n} \subsetneq \mathfrak{m}$. □

For a given product measure $\mu_{\mathfrak{D}}$ and functions $(\gamma_i)_{i=1}^D$, we are now able to restate the decomposition (2.53) of function $u \in L_2(\mu_{\mathfrak{D}})$ as

$$u = \sum_{\mathfrak{m}\subset\mathfrak{D}} Q_{\mathfrak{m}}^W u \, .$$

Let us now assume we have a subset $\mathfrak{S} \subset \mathcal{P}(\mathfrak{D})$ that satisfies the admissibility condition

$$\mathfrak{m} \in \mathfrak{S}, \mathfrak{n} \subset \mathfrak{m} \Rightarrow \mathfrak{n} \in \mathfrak{S} \; . \tag{2.59}$$

Then

$$u_{\mathfrak{S}} := \sum_{\mathfrak{m} \in \mathfrak{S}} Q_{\mathfrak{m}}^W u \tag{2.60}$$

is the $L_2(\mu_{\mathfrak{D}})$ best approximation of u in the space $\oplus_{\mathfrak{m}\in\mathfrak{S}} W_{\mathfrak{m}}$. Due to the orthogonality of the decomposition of u, the squared error of this kind of approximation can easily be stated in the $\|\cdot\|_{\mathfrak{D}}$-norm by

$$\|u - u_{\mathfrak{S}}\|_{\mathfrak{D}}^2 = \sum_{\mathfrak{m} \in \mathcal{P}(\mathfrak{D})\setminus\mathfrak{S}} \|u_{\mathfrak{m}}\|_{\mathfrak{D}}^2 \; .$$

In Section 2.4 we discuss common choices for the set $\mathfrak{S}$. Note that

$$\begin{aligned} u_{\mathfrak{S}} &= \sum_{\mathfrak{m}\in\mathfrak{S}} Q_{\mathfrak{m}}^W u = \sum_{\mathfrak{m}\in\mathfrak{S}} \sum_{\mathfrak{n}\subset\mathfrak{m}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} Q_{\mathfrak{n}}^V u & (2.61)\\ &= \sum_{\mathfrak{n}\in\mathfrak{S}} \Big(\sum_{\mathfrak{m}\in\mathfrak{S},\mathfrak{m}\supset\mathfrak{n}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} \Big) Q_{\mathfrak{n}}^V u \; , & (2.62) \end{aligned}$$

which means that in order to compute $u_{\mathfrak{S}}$, we have to compute every projection $Q_{\mathfrak{n}}^V u, \mathfrak{n} \in \mathfrak{S}$, at most once. For further results related to the ANOVA decomposition we recommend [KSWW10].

Example 2.14 (Classical ANOVA on the unit hypercube)**.** The ANOVA decomposition on $[0,1]^d$ is a special case with unit functions $\gamma_i = 1, i = 1, \ldots, d$, and the measures

$$\mu_i(A) = \lambda(A \cap [0,1]) \qquad \forall \; A \in \mathcal{B}(\mathbb{R}) \; ,$$

where λ is the Lebesgue-measure and $\mathcal{B}(\mathbb{R})$ the Borel-set of $\mathbb{R}$. Obviously, the constants from (2.48) are given by $c_{\mathfrak{m}} = 1$ for all $\mathfrak{m} \subset \mathfrak{D}$.

Example 2.15 (Anchor ANOVA on $\mathbb{R}^d$)**.** Typically, it is very expensive to compute the integrals in the projection (2.49). The so-called anchor ANOVA [GH10b] is much cheaper, as it needs only one point evaluation. Formally, this corresponds to $\mu_i = \delta_{a_i}$, where δ_{a_i} is Dirac's delta at point $a_i \in \mathbb{R}$. Obviously, $\mu_{\mathfrak{D}} = \delta_{\mathbf{a}}$ with $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{R}^d$. The constants $c_{\mathfrak{n}}$ from (2.48) are given by

$$c_{\mathfrak{n}} := \int_{\mathbb{R}^{\mathfrak{n}}} \gamma_{\mathfrak{n}}^2(\mathbf{x}_{\mathfrak{n}}) \mu_{\mathfrak{n}}(\mathrm{d}\mathbf{x}_{\mathfrak{n}}) = \gamma_{\mathfrak{n}}^2(\mathbf{a}_{\mathfrak{n}}).$$

Then, the projections $Q_{\mathfrak{m}}^V, \mathfrak{m} \subset \mathfrak{D}$, from (2.49) are given by

$$\begin{aligned} [Q_{\mathfrak{m}}^V u](\mathbf{x}) &= \gamma_{\mathfrak{D}\setminus\mathfrak{m}}^{-2}(\mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) \\ &= \frac{\gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}})}{\gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}})} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) \; . \end{aligned} \tag{2.63}$$

If the unit functions are chosen as $\gamma_i = 1, i = 1, \ldots, d$, the projection (2.63) simplifies to

$$[Q_{\mathfrak{m}}^V u](\mathbf{x}) = u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}})$$

and in the case of standard normal densities

$$\gamma_i(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}} \quad \text{for} \quad i = 1, \dots, d$$

the expression (2.63) results in

$$[Q_{\mathfrak{m}}^V u](\mathbf{x}) = \frac{e^{-\frac{\|\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}\|^2}{2}}}{e^{-\frac{\|\mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}\|^2}{2}}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) = e^{-\frac{\|\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}\|^2 - \|\mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}\|^2}{2}} u(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{a}_{\mathfrak{D}\setminus\mathfrak{m}}) \, .$$

Even though the anchor ANOVA works well in practice, it needs to be said that formally the function space $L_2(\mu_{\mathfrak{D}})$ is only one-dimensional, since $\mu_{\mathfrak{D}}(\mathbb{R}^d \setminus \{\mathbf{a}\}) = 0$ and thus all functions with the same value at point $\mathbf{a}$ are considered to be identical as they can only differ on a null set.

Example 2.16 (Weighted ANOVA on $\mathbb{R}^d$)**.** Alternatively, $\mu_{\mathfrak{D}}$ can realize a density on $\mathbb{R}^d$, and typically a Gaussian density is used. A remark on this can be found in [GH10b]. In that case, the choice of $\gamma_i = 1$ for $i = 1, \dots, d$ results in $c_{\mathfrak{m}} = 1, \mathfrak{m} \subset \mathfrak{D}$.

Example 2.17 (CI in molecular dynamics)**.** In [GH10a] an approach complementary to Example 2.16 is proposed: The one-dimensional measures μ_i are Lebesgue measures, and the one-dimensional unit functions γ_i (or particle functions g_i in the molecular dynamics context) satisfy $\|\gamma_i\|_i = 1, i = 1, \dots, d$. Note that we only treated the real valued case, but the extension to complex numbers is straightforward.

Remark 2.18 (Limitations). We note that there is no sensible ANOVA decomposition on $\mathbb{R}^d$ with $\mu_{\mathfrak{D}} = \lambda^d$, where λ^d is the d-dimensional Lebesgue measure and $\gamma_i = 1, i = 1, \dots, d$. Obviously, $c_{\mathfrak{m}} = \int_{\mathbb{R}^{\mathfrak{m}}} 1 \lambda^{\#\mathfrak{m}}(\mathrm{d}\mathbf{x}_{\mathfrak{m}}) = \infty$ for all $\mathfrak{m} \subset \mathfrak{D}$, so we are no longer dealing with an orthogonal subspace projection when applying $Q_{\mathfrak{m}}^V$ as defined in (2.49).

It is also tempting to go to a higher level of abstraction and to describe everything with general scalar products instead of integrations. It turns out that this is not straightforwardly possible, as we use Fubini's theorem extensively and also the monotonicity of integration in (2.50) does not hold for arbitrary norms.

2.2.2 Extension: Iterated ANOVA

One obvious approach to improve the accuracy of an ANOVA approximation is to enlarge the index set $\mathfrak{S}$ in (2.60). Alternatively, we can repeat the procedure on the residual $r = u - u_{\mathfrak{S}}$ with a different set of functions $\gamma_i, i = 1, \dots, d$, in an iterative fashion. To that end, we need to determine functions $\gamma_i, i = 1, \dots, d$, such that $\otimes_{i=1}^d \gamma_i$ is a good approximation to u. From the proper generalized decomposition method, we know that we can choose the first term of a Schmidt decomposition for $d = 2$, and for $d \geq 3$ an alternating directions approach is possible [FHMM13].

Then the iterated ANOVA decomposition works like this: We fix a product measure $\mu_{\mathfrak{D}}$, the initial function $u^{(1)} := u$, the index set $\emptyset \neq \mathfrak{S} \subset \mathfrak{D}$ and then run the following algorithm iteratively for $m = 1, 2, \dots$

1. Determine a set of one-dimensional functions $\gamma_i^{(m)} \in L_2(\mu_i), i = 1, \dots, d$, and fix the corresponding projections $(Q_{\mathfrak{u}}^V)^{(m)}$

2. Compute $u_{\mathfrak{S}}^{(m)} := \sum_{\mathfrak{u} \subset \mathfrak{S}} \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{u}} (-1)^{\#\mathfrak{m} - \#\mathfrak{u}} \Big) (Q_{\mathfrak{u}}^V)^{(m)} u^{(m)}$ as an approximation to $u^{(m)}$

3. Compute the residual $u^{(m+1)} := u^{(m)} - u_{\mathfrak{S}}^{(m)}$

4. If $\left\| u^{(m+1)} \right\|_{\mathfrak{D}}^2 \geq \epsilon$, set $m \leftarrow m + 1$ and start from 1.

Then, after M iteration steps, the approximation of u by the iterated ANOVA is given by $\sum_{m=1}^{M} u_{\mathfrak{S}}^{(m)}$ and has the error

$$\left\| u - \sum_{m=1}^{M} u_{\mathfrak{S}}^{(m)} \right\|_{\mathfrak{D}}^2 = \left\| u^{(M+1)} \right\|_{\mathfrak{D}}^2 < \epsilon \, .$$

It is noteworthy that for $\mathfrak{S} = \{\emptyset\}$, this method results in a typical low-rank approximation [ACF10, GKT13], since in that case all $u_{\mathfrak{S}}^{(m)}, m = 1, \dots, M$ are products of one-dimensional functions $\gamma_i^{(m)}, i = 1, \dots, d$.

The iterated ANOVA presented here is a non-linear method and the convergence properties are not clear. We consider the relation to low-rank models interesting but we now turn back to linear ANOVA decompositions.

2.3 Decomposition of functions based on expected values

In Section 2.2, we gave a self-contained view on the ANOVA decomposition of functions. Even though the low-dimensional interaction terms are cheap to handle, their computation can be expensive, which is why we often resort to the anchor ANOVA from Example 2.15. However, it is also advisable to exploit the structure of the functions we want to decompose.

This section deals with the ANOVA decomposition of functions that can be expressed by (2.21) or (2.23), respectively. To be more specific, we would like to compute $u : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ with

$$u(\tau, \mathbf{x}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})] \, , \tag{2.64}$$

where $\mathbf{X}$ is a Lévy process. Assuming that $\mathbf{X}(\tau)$ is distributed as $\eta_{\mathfrak{D}}^{\tau}$, we can rewrite (2.64) as

$$u(\tau, \mathbf{x}) = \int_{\mathbb{R}^d} g(\mathbf{x} + \mathbf{y}) \eta_{\mathfrak{D}}^{\tau}(\mathrm{d}\mathbf{y}) \, . \tag{2.65}$$

When d is relatively high, the discretization of u as well as the integration task (2.65) become challenging, and this motivates the ANOVA approximation of u by a superposition of low-

dimensional projections into the subspaces $V_\mathfrak{m}, \mathfrak{m} \in \mathfrak{S}$. Recall from (2.61) and (2.62) that

$$\begin{aligned} u_\mathfrak{S}(\tau, \mathbf{x}) &= \sum_{\mathfrak{m}\in\mathfrak{S}} [Q_\mathfrak{m}^W u](\tau, \mathbf{x}) = \sum_{\mathfrak{m}\in\mathfrak{S}} \sum_{\mathfrak{n}\subset\mathfrak{m}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} [Q_\mathfrak{n}^V u](\tau, \mathbf{x}) \\ &= \sum_{\mathfrak{n}\in\mathfrak{S}} \Big(\sum_{\mathfrak{m}\in\mathfrak{S},\mathfrak{m}\supset\mathfrak{n}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} \Big) [Q_\mathfrak{n}^V u](\tau, \mathbf{x}) \,, \end{aligned} \tag{2.66}$$

where the projections Q^W and Q^V are only applied to the space variable $\mathbf{x}$. Depending on $\mathfrak{S}$, $u_\mathfrak{S}$ is a good representation of u, but the main question is whether we can compute the projections $Q_\mathfrak{n}^V u$ in (2.66) cheaply, i.e., without having to compute u from (2.64) first. There are two approaches:

- Computing the *exact* low-dimensional terms $Q_\mathfrak{n}^V u$.
  - In Subsection 2.3.1, we choose unit functions $\gamma_i, i = 1, \dots, d$ that are reciprocal to the measures μ_i, i.e., $\mu_i(\mathrm{d}x) = \gamma_i(x)^{-1}\mathrm{d}x$. That way we obtain the exact $[Q_\mathfrak{n}^V u](\tau, \mathbf{x})$ for all $\tau \geq 0$ at the same time, but the approach is somewhat impractical from a numerical perspective.
  - In Subsection 2.3.2, we consider processes $\mathbf{X}(\tau)$ with components that are independent Brownian motions and space weights μ_i that are also Gaussians. We show how we can compute $[Q_\mathfrak{n}^V u](\tau, \mathbf{x})$ for an a priori chosen $\tau = T$. This is helpful but does not allow to deal with general Lévy processes.

- In Subsection 2.3.3 we do not decompose and approximate the solution u but instead the initial condition g. To this end, we replace g in (2.64) by $g_\mathfrak{S}$ and regard the terms

  $$[\tilde{Q}_\mathfrak{m}^V u](\tau, \mathbf{x}) := \int_{\mathbb{R}^\mathfrak{m}} [Q_\mathfrak{m}^V g](\mathbf{x} + \mathbf{y}_\mathfrak{m}) \eta_\mathfrak{m}^\tau(\mathrm{d}\mathbf{y}_\mathfrak{m})$$

  as approximations to $Q_\mathfrak{m}^V u$. This turns out to be the practically most relevant approach, and we will use it in our numerical experiments in Chapter 6.

2.3.1 A direct decomposition of the solution for Lévy processes

In the ideal case, the computation of (2.64) and the projections $Q_\mathfrak{n}^V$ commute. This is in fact the case for a special choice of measures μ_i and unit functions γ_i for $i = 1, \dots, d$. Then, we can express (2.66) as

$$\begin{aligned} u_\mathfrak{S}(\tau, \mathbf{x}) &= \sum_{\mathfrak{n}\in\mathfrak{S}} \Big(\sum_{\mathfrak{m}\in\mathfrak{S},\mathfrak{m}\supset\mathfrak{n}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} \Big) Q_\mathfrak{n}^V \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})] \\ &= \sum_{\mathfrak{n}\in\mathfrak{S}} \Big(\sum_{\mathfrak{m}\in\mathfrak{S},\mathfrak{m}\supset\mathfrak{n}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} \Big) \mathbb{E}\left[\left[Q_\mathfrak{n}^V g \right] (\mathbf{X}_\mathfrak{n}(\tau) + \mathbf{x}) \right] \,. \end{aligned} \tag{2.67}$$

Note that the terms in (2.67) only depend on the marginal processes $\mathbf{X}_\mathfrak{n}$, so that the underlying PIDEs (2.30) are only $\#\mathfrak{n}$-dimensional.

Setting

We assume functions $\gamma_i \in L_1(\mathbb{R}), i = 1, \dots, d$, that are strictly positive and satisfy

$$\mu_i(\mathrm{d}x) = \gamma_i(x)^{-1}\mathrm{d}x \ . \tag{2.68}$$

This implies that the measures $\mu_i, i = 1, \dots, d$, are absolutely continuous with respect to the Lebesgue measure and the constants (2.48) are given by

$$c_{\{i\}} = \int_{\mathbb{R}^{\{i\}}} \gamma_i(x_i)^2 \mu_i(\mathrm{d}x_i) = \int_{\mathbb{R}^{\{i\}}} \gamma_i(x_i)\mathrm{d}x_i < \infty \ .$$

The following Theorem shows that under these special circumstances, the $\mathfrak{m}$-marginal $\mathbf{X}_\mathfrak{m}$ of the Lévy process $\mathbf{X}$ is sufficient for computing $Q_\mathfrak{m}^V u$.

Theorem 2.19. *Given an initial condition $g \in L_2(\mu_\mathfrak{D})$, the ANOVA projections of the function u from* (2.65) *are given by*

$$[Q_\mathfrak{m}^V u](\tau, \mathbf{x}) = \mathbb{E}[[Q_\mathfrak{m}^V g](\mathbf{X}_\mathfrak{m}(\tau) + \mathbf{x})] \ .$$

Proof. We apply the projection as defined in (2.49) to u from (2.65) and obtain

$$\begin{aligned}
[Q_\mathfrak{m} u](\tau, \mathbf{x}) &= c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \int_{\mathbb{R}^\mathfrak{D}} g(\mathbf{x}_\mathfrak{m} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}) \eta_\mathfrak{D}^\tau(\mathrm{d}\mathbf{y}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \\
&= c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \int_{\mathbb{R}^\mathfrak{D}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_\mathfrak{m} + \mathbf{y}_\mathfrak{m}) \diamond (\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}})) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \eta_\mathfrak{D}^\tau(\mathrm{d}\mathbf{y}) \ , \qquad (2.69)
\end{aligned}$$

where we used $\gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}})\mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) = \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}$, see (2.68), and changed the order of integration in (2.69). Now, we use the simple integral translation $\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}} \to \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}$ to eliminate the dependence on $\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}$. This leads to

$$\begin{aligned}
[Q_\mathfrak{m} u](\tau, \mathbf{x}) &= c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \int_{\mathbb{R}^\mathfrak{D}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_\mathfrak{m} + \mathbf{y}_\mathfrak{m}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \eta_\mathfrak{D}^\tau(\mathrm{d}\mathbf{y}) \\
&= \int_{\mathbb{R}^\mathfrak{D}} c_{\mathfrak{D}\setminus\mathfrak{m}}^{-1} \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_\mathfrak{m} + \mathbf{y}_\mathfrak{m}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \cdot \gamma_{\mathfrak{D}\setminus\mathfrak{m}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \eta_\mathfrak{D}^\tau(\mathrm{d}\mathbf{y}) \\
&= \int_{\mathbb{R}^\mathfrak{D}} [Q_\mathfrak{m}^V g](\mathbf{x} + \mathbf{y}_\mathfrak{m}) \eta_\mathfrak{m}^\tau(\mathrm{d}\mathbf{y}_\mathfrak{m}) = \mathbb{E}[[Q_\mathfrak{m}^V g](\mathbf{X}_\mathfrak{m}(\tau) + \mathbf{x})] \ . \qquad (2.70)
\end{aligned}$$

Note that (2.70) no longer depends on $\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}$, and we can safely replace the integration measure $\eta_\mathfrak{D}^\tau(\mathrm{d}\mathbf{y})$ by $\eta_\mathfrak{m}^\tau(\mathrm{d}\mathbf{y}_\mathfrak{m})$, which happens to be the distribution of the marginal Lévy process $\mathbf{X}_\mathfrak{m}(\tau)$ at time τ. □

The result of Theorem 2.19 shows that the expected value and the ANOVA projections commute for all $\tau \in [0, T]$. This, however, is only the case if the measures μ_i and the unit functions γ_i are related like in (2.68).

Let us briefly assume that the $Q_\mathfrak{m}^V, \mathfrak{m} \subset \mathfrak{D}$, are given by

$$[Q_\mathfrak{m}^V g](\mathbf{x}) = \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g(\mathbf{x}_\mathfrak{m} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \ , \tag{2.71}$$

even though these $Q_{\mathfrak{m}}^V$ do not not constitute well-defined ANOVA projections, see Remark 2.18. In this case, the $Q_{\mathfrak{m}}^V$ directly commute with the computation of the expected value, since

$$\begin{aligned}[Q_{\mathfrak{m}}^V u](\tau, \mathbf{x}) &= \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}}} g(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}) \eta_{\mathfrak{D}}^{\tau}(\mathrm{d}\mathbf{y}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \\ &= \int_{\mathbb{R}^{\mathfrak{D}}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \eta_{\mathfrak{D}}^{\tau}(\mathrm{d}\mathbf{y}) \\ &= \int_{\mathbb{R}^{\mathfrak{D}}} [Q_{\mathfrak{m}}^V g](\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \eta_{\mathfrak{m}}^{\tau}(\mathrm{d}\mathbf{y}_{\mathfrak{m}}) \, . \end{aligned} \tag{2.72}$$

Note that the equality (2.72) is possible due to a variable substitution $\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}} \to \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}$, which has no further effects when integrating with the Lebesgue measure over $\mathbb{R}^{\mathfrak{D}}$, just as in (2.69). However, as we said already, (2.71) is not a valid projection, and this was just for discussion.

Let us conclude this subsection with a short error discussion. Note that for bounded functions f with $f \lesssim \gamma_{\mathfrak{D}}$ asymptotically for $\|\mathbf{x}\| \to \infty$, we have

$$\|f\|_{\mathfrak{D}}^2 = \int_{\mathbb{R}^d} f(\mathbf{x})^2 \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) = \int_{\mathbb{R}^d} f(\mathbf{x})^2 \gamma_{\mathfrak{D}}(\mathbf{x})^{-1} \mathrm{d}\mathbf{x} \lesssim \int_{\mathbb{R}^d} \gamma_{\mathfrak{D}}(\mathbf{x}) \mathrm{d}\mathbf{x} < \infty \, ,$$

and thus $f \in L_2(\mu_{\mathfrak{D}})$. Now, we consider the error we make by using the ANOVA approximation: As we compute the exact and $(\cdot,\cdot)_{\mathfrak{D}}$-orthogonal ANOVA components of the function f, we obtain

$$\|f - f_{\mathfrak{S}}\|_{L_2(\mu_{\mathfrak{D}})}^2 = \sum_{\mathfrak{m} \in \mathfrak{D}\setminus\mathfrak{S}} \|Q_{\mathfrak{m}}^W f\|_{L_2(\mu_{\mathfrak{D}})}^2 \, ,$$

so $\mu_{\mathfrak{D}}$ is directly relevant for measuring our error. For Gaussian densities

$$\gamma_i(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad \text{for} \quad i = 1, \dots, d \, ,$$

the corresponding μ_i would no longer be finite measures but put an exponentially increasing weight on the tails. So, measuring our error with respect to $\mu_{\mathfrak{D}}$ is somewhat of a relative error criterion: In the tails of $\gamma_{\mathfrak{D}}$, where the function f with $f \lesssim \gamma_{\mathfrak{D}}$ is small, the error of our approximative method needs to be small as well since it is magnified by $\gamma_{\mathfrak{D}}^{-1}$. However, as our ultimate goal is to compute numerical solutions which need some domain truncation sooner or later, this consideration is only philosophical in nature.

In summary, the balancing requirement (2.68) allows us to directly compute the ANOVA components of the function $u(\tau, \cdot)$ by solving only lower-dimensional subproblems (2.67). In the next subsection, we present another special case for which this is possible.

2.3.2 A special decomposition of the solution for Gaussian processes

In this subsection, we assume

$$\gamma_i = 1 \quad \text{and} \quad \mu_i(\mathrm{d}x) = \rho_{\lambda_i}(x) \mathrm{d}x \quad \text{for} \quad i = 1, \dots, d \, ,$$

where $\lambda_i > 0$, and where

$$\rho_{\lambda_i}(x) = \frac{1}{\sqrt{2\pi\lambda_i}} e^{-\frac{x^2}{2\lambda_i}}$$

denotes the density of a $\mathcal{N}(0, \lambda_i)$-distributed Gaussian variable. The multivariate case is then given by

$$\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) = \rho_{\boldsymbol{\lambda}}(\mathbf{x})\mathrm{d}\mathbf{x}$$

with $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)$ and

$$\rho_{\boldsymbol{\lambda}}(\mathbf{x}) = \prod_{i=1}^{d} \rho_{\lambda_i}(x_i) \ . \tag{2.73}$$

The marginal densities of (2.73) are given by

$$\rho_{\boldsymbol{\lambda}_{\mathfrak{m}}}(\mathbf{x}_{\mathfrak{m}}) = \prod_{i \in \mathfrak{m}} \rho_{\lambda_i}(x_i) \quad \text{for} \quad \mathfrak{m} \subset \mathfrak{D}$$

and because of the product structure it holds that

$$\rho_{\boldsymbol{\lambda}}(\mathbf{x}) = \rho_{\boldsymbol{\lambda}_{\mathfrak{m}}}(\mathbf{x}_{\mathfrak{m}}) \cdot \rho_{\boldsymbol{\lambda}_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{x}_{\mathfrak{D}\setminus\mathfrak{m}}) \ .$$

Furthermore, we assume that the Lévy process in (2.64) has triplet $(\boldsymbol{\Sigma}, \mathbf{0}, 0)$ with $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \dots, \sigma_d^2)$ and $\sigma_1^2 \geq \cdots \geq \sigma_d^2 \geq 0$. This means $\mathbf{X}$ has no drift, no jump term and only a diagonal covariance matrix. At first sight, this case looks artificial, but any problem based on a geometric Brownian motion for $\mathbf{S}(t)$, i.e., an exponentiated Lévy process without jumps in (2.17), can be expressed in this setting using the transformations in Subsection 2.1.1, see also [Rei04, RW07]. Here, the $\sigma_1^2, \dots, \sigma_d^2$ can be thought of as the eigenvalues of the original covariance matrix, see Theorem 2.6 for the necessary orthogonal transformation. This results in a process $\mathbf{X}$ that starts in 0, and $\mathbf{X}(T)$ for $T > 0$ is distributed with density

$$\eta_{\mathfrak{D}}^{T}(\mathrm{d}\mathbf{x}) = \rho_{T\boldsymbol{\sigma}^2}(\mathbf{x})\mathrm{d}\mathbf{x} \ ,$$

where $T\boldsymbol{\sigma}^2 = (T\sigma_1^2, \dots, T\sigma_d^2)$.

In Theorem 2.19 we showed that under special circumstances the ANOVA projections $Q_{\mathfrak{m}}^{V}$ commute with the computation of the expected value in (2.21). A similar result holds for Gaussian space weights and the multi-dimensional Brownian motion as well, as the following theorem shows.

Theorem 2.20. *Let $\mathbf{X}$ be a Lévy process with triplet $(\operatorname{diag}\boldsymbol{\sigma}^2, \mathbf{0}, 0)$, where $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_d^2)$. Furthermore, let $Q_{\mathfrak{m}}^{V}, \mathfrak{m} \subset \mathfrak{D}$, be the ANOVA projections based on the space measure $\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) = \rho_{\boldsymbol{\lambda}}(\mathbf{x})\mathrm{d}\mathbf{x}$, where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)$ is the vector of variances used in (2.73). We denote the ANOVA projections based on the space measure $\tilde{\mu}_{\mathfrak{D}}^{T}(\mathrm{d}\mathbf{x}) = \rho_{\boldsymbol{\lambda}+T\boldsymbol{\sigma}^2}(\mathbf{x})\mathrm{d}\mathbf{x}$ by $\tilde{Q}_{\mathfrak{m}}^{V,T}$. Then, for a function u given by (2.64), it holds that*

$$\begin{aligned}
[Q_{\mathfrak{m}}^{V}u](T, \mathbf{x}) =& \int_{\mathbb{R}^{\mathfrak{m}}} \left(\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \rho_{(\boldsymbol{\lambda}+T\boldsymbol{\sigma}^2)_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \right) \eta_{\mathfrak{m}}^{T}(\mathrm{d}\mathbf{y}_{\mathfrak{m}}) \\
=& \mathbb{E}[[\tilde{Q}_{\mathfrak{m}}^{V,T} g](\mathbf{X}_{\mathfrak{m}}(T) + \mathbf{x}_{\mathfrak{m}})] \ .
\end{aligned}$$

Proof. Let us briefly recall (2.65) for $\tau = T$

$$u(T, \mathbf{x}) = \int_{\mathbb{R}^d} g(\mathbf{x} + \mathbf{y}) \eta_{\mathfrak{D}}^T(\mathrm{d}\mathbf{y})$$

and that $g(\mathbf{x}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}) = g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond (\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}))$. Then, the projection $Q_{\mathfrak{m}}^V u$ from (2.49) yields

$$\begin{aligned}
[Q_{\mathfrak{m}}^V u](T, \mathbf{x}) &= \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond (\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}})) \eta_{\mathfrak{D}}^T(\mathrm{d}\mathbf{y}) \mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \\
&= \int_{\mathbb{R}^{\mathfrak{m}}} \Big(\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond (\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}})) \\
&\quad \cdot \rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}) \rho_{\boldsymbol{\lambda}_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \mathrm{d}\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}} \Big) \rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{m}}}(y_{\mathfrak{m}}) \mathrm{d}\mathbf{y}_{\mathfrak{m}} \,.
\end{aligned} \tag{2.74}$$

A simple variable substitution $\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} + \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}} \to \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}$ in (2.74) yields

$$\begin{aligned}
[Q_{\mathfrak{m}}^V u](T, \mathbf{x}) = \int_{\mathbb{R}^{\mathfrak{m}}} \Big(& \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \\
& \cdot \int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} \rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}) \rho_{\boldsymbol{\lambda}_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} - \mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{y}_{\mathfrak{D}\setminus\mathfrak{m}} \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \Big) \\
& \cdot \rho_{T\boldsymbol{\sigma}_{\mathfrak{m}}^2}(y_{\mathfrak{m}}) \mathrm{d}\mathbf{y}_{\mathfrak{m}} \,.
\end{aligned} \tag{2.75}$$

In (2.75), we convolute a $\mathfrak{D}\setminus\mathfrak{m}$-dimensional Gaussian density $\rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{D}\setminus\mathfrak{m}}}$ with a Gaussian density $\rho_{\boldsymbol{\lambda}_{\mathfrak{D}\setminus\mathfrak{m}}}$, which results in a Gaussian with density $\rho_{(\boldsymbol{\lambda}+T\boldsymbol{\sigma}^2)_{\mathfrak{D}\setminus\mathfrak{m}}}$, so that we can deduce

$$[Q_{\mathfrak{m}}^V u](T, \mathbf{x}) = \int_{\mathbb{R}^{\mathfrak{m}}} \Big(\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}}) \diamond \mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \rho_{(T\boldsymbol{\sigma}^2 + \boldsymbol{\lambda})_{\mathfrak{D}\setminus\mathfrak{m}}}(\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}) \mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}} \Big) \rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{m}}}(y_{\mathfrak{m}}) \mathrm{d}\mathbf{y}_{\mathfrak{m}} \,.$$

Obviously,

$$\rho_{(T\boldsymbol{\sigma}^2)_{\mathfrak{m}}}(y_{\mathfrak{m}}) \mathrm{d}\mathbf{y}_{\mathfrak{m}} = \eta_{\mathfrak{m}}^T(\mathrm{d}\mathbf{y}) \,,$$

which concludes the proof. □

Theorem 2.20 says that we can compute the ANOVA component $Q_{\mathfrak{m}}^V u$ of the solution by solving an $\#\mathfrak{m}$-dimensional problem with the projection $\tilde{Q}_{\mathfrak{m}}^{V,T}$ of the initial condition g and the marginal Lévy processes $\mathbf{X}_{\mathfrak{m}}(T)$. This is different from the result of Theorem 2.19, since there we got the correct decomposition for all $\tau \geq 0$ and here $\tilde{Q}_{\mathfrak{m}}^{V,T}$ is T-dependent, which means that we get the correct ANOVA decomposition of u for one point in time only, namely $\tau = T$.

2.3.3 Decomposition of the final condition

So far, we computed the correct ANOVA approximation of the solution u given by (2.64) and learned that this problem reduces to computing ANOVA projections of the initial condition and computing expected values based on marginals of the Lévy process $\mathbf{X}$. However, we needed special assumptions like (2.68) or Gaussian space weights and Brownian motions as in Subsection 2.3.2.

Now, we assume

$$\mu_i(\mathbb{R}) = 1, \gamma_i = 1 \quad \text{for} \quad i = 1, \dots, d \tag{2.76}$$

and compute an ANOVA approximation of the initial condition

$$g_{\mathfrak{S}} = \sum_{\mathfrak{n} \in \mathfrak{S}} \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{n}} (-1)^{\#\mathfrak{m} - \#\mathfrak{n}} \Big) Q_{\mathfrak{n}}^V g$$

based on the sets in $\mathfrak{m} \in \mathfrak{S}$. Note that the assumptions (2.76) allow for most of the common choices for μ_i: Lebesgue measure on the unit interval, the Dirac measure in an anchor point and the Gaussian measure on $\mathbb{R}$. Assuming that $g_{\mathfrak{S}} \approx g$, we get an approximation

$$\tilde{u}_{\mathfrak{S}}(\tau, \mathbf{x}) := \mathbb{E}[g_{\mathfrak{S}}(\mathbf{X}(\tau) + \mathbf{x})] \approx \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})] = u(\tau, \mathbf{x}) \ ,$$

to u from (2.64). Luckily, we can compute $\tilde{u}_{\mathfrak{S}}$ quite efficiently, since

$$\begin{aligned} \tilde{u}_{\mathfrak{S}}(\tau, \mathbf{x}) &= \mathbb{E}\Big[\sum_{\mathfrak{n} \in \mathfrak{S}} \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{n}} (-1)^{\#\mathfrak{m} - \#\mathfrak{n}} \Big) [Q_{\mathfrak{n}}^V g](\mathbf{X}(\tau) + \mathbf{x}) \Big] \\ &= \sum_{\mathfrak{n} \in \mathfrak{S}} \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{n}} (-1)^{\#\mathfrak{m} - \#\mathfrak{n}} \Big) \mathbb{E}[[Q_{\mathfrak{n}}^V g](\mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}})] \ . \end{aligned} \tag{2.77}$$

As $\gamma_i = 1$ for $i = 1, \dots, d$, the projection $Q_{\mathfrak{n}}^V g$ only depends on variables indexed by $\mathfrak{n}$. That means we only need to consider the $\mathfrak{n}$-marginal of $\mathbf{X}(\tau)$ and $\mathbf{x}$ in (2.77). According to Proposition 2.3, we know that $\mathbf{X}_{\mathfrak{n}}$ is an $\#\mathfrak{n}$-dimensional process with Lévy triplet $(\mathbf{Q}_{\mathfrak{n}}, \boldsymbol{\theta}_{\mathfrak{n}}, \nu_{\mathfrak{n}})$. We can now use any simulation-based method or PIDE solver on the problems (2.77) we would have used on (2.64), but with a significantly reduced computational complexity as we are only dealing with at most $\#\mathfrak{S}$ problems with dimensionalities $\#\mathfrak{n}, \mathfrak{n} \in \mathfrak{S}$, instead of one d-dimensional one.

Error estimates

The question is how much our solution $u(\tau, \cdot)$ is perturbed by the approximation $g_{\mathfrak{S}}$ of our initial condition g. Let us assume that the measure $\eta_{\mathfrak{D}}^\tau$ is absolutely continuous with respect to the Lebesgue measure and that we can express the distribution of $\mathbf{X}(\tau)$ as

$$\eta_{\mathfrak{D}}^\tau(\mathrm{d}\mathbf{y}) = k_{\mathfrak{D}}^\tau(\mathbf{y}) \mathrm{d}\mathbf{y} \ ,$$

where $k_{\mathfrak{D}}^\tau(\mathbf{y})$ is a density function. Then we have

$$\begin{aligned} \|u(\tau, \cdot) - \tilde{u}_{\mathfrak{S}}(\tau, \cdot)\|_{L_2(\mu_{\mathfrak{D}})}^2 &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} (g(\mathbf{x} + \mathbf{y}) - g_{\mathfrak{S}}(\mathbf{x} + \mathbf{y})) k_{\mathfrak{D}}^\tau(\mathbf{y}) \mathrm{d}\mathbf{y} \right)^2 \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) \\ &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} (g(\mathbf{y}) - g_{\mathfrak{S}}(\mathbf{y})) k_{\mathfrak{D}}^\tau(\mathbf{y} - \mathbf{x}) \mathrm{d}\mathbf{y} \right)^2 \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) \\ &\leq \int_{\mathbb{R}^d} \|g - g_{\mathfrak{S}}\|_\infty^2 \underbrace{\|k_{\mathfrak{D}}^\tau(\cdot - \mathbf{x})\|_1^2}_{=1} \mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) = \|g - g_{\mathfrak{S}}\|_\infty^2 \ , \end{aligned} \tag{2.78}$$

where we have used Hölder's inequality in (2.78). So we can bound the $L_2(\mu_{\mathfrak{D}})$ error of the solution by the L_∞ error of the ANOVA approximation of the initial condition g. Instead of (2.78), it is also possible to obtain an estimate

$$\|u(\tau,\cdot)-\tilde{u}_{\mathfrak{S}}(\tau,\cdot)\|^2_{L_2(\mu_{\mathfrak{D}})} \le \int_{\mathbb{R}^d} \|g-g_{\mathfrak{S}}\|^2_{L_2}\|k^\tau_{\mathfrak{D}}(\cdot-\mathbf{x})\|^2_{L_2}\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x}) = \|g-g_{\mathfrak{S}}\|^2_{L_2}\|k^\tau_{\mathfrak{D}}\|^2_{L_2}\ ,$$

but two obvious problems arise here: $g-g_{\mathfrak{S}}$ is only in $L_2(\mu_{\mathfrak{D}})$ but not necessarily in the unweighted L_2-space, and $k^\tau_{\mathfrak{D}}$ is integrable but not necessarily square integrable over $\mathbb{R}^d$. Let us take a look at the $L_1(\mu_{\mathfrak{D}})$ error of u by

$$\begin{aligned}\|u(\tau,\cdot)-\tilde{u}_{\mathfrak{S}}(\tau,\cdot)\|_{L_1(\mu_{\mathfrak{D}})} &= \int_{\mathbb{R}^d}\left|\int_{\mathbb{R}^d}(g(\mathbf{x}+\mathbf{y})-g_{\mathfrak{S}}(\mathbf{x}+\mathbf{y}))k^\tau_{\mathfrak{D}}(\mathbf{y})\mathrm{d}\mathbf{y}\right|\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x})\\ &= \int_{\mathbb{R}^d}\left|\int_{\mathbb{R}^d}(g(\mathbf{y})-g_{\mathfrak{S}}(\mathbf{y}))k^\tau_{\mathfrak{D}}(\mathbf{y}-\mathbf{x})\mathrm{d}\mathbf{y}\right|\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x})\\ &\le \int_{\mathbb{R}^d}|g(\mathbf{y})-g_{\mathfrak{S}}(\mathbf{y})|\int_{\mathbb{R}^d}k^\tau_{\mathfrak{D}}(\mathbf{x}-\mathbf{y})\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x})\mathrm{d}\mathbf{y} = \|g-g_{\mathfrak{S}}\|_{L_1(\vartheta^\tau_{\mathfrak{D}})}\end{aligned}$$

with

$$\vartheta^\tau_{\mathfrak{D}}(\mathrm{d}\mathbf{y}) = \int_{\mathbb{R}^d} k^\tau_{\mathfrak{D}}(\mathbf{x}-\mathbf{y})\mu_{\mathfrak{D}}(\mathrm{d}\mathbf{x})\ . \tag{2.79}$$

In summary, we find that the $L_1(\mu_{\mathfrak{D}})$ error of u at time τ is bounded by the $L_1(\vartheta^\tau_{\mathfrak{D}})$ error of the initial condition, where $\vartheta^\tau_{\mathfrak{D}}$ is the convolution of the space weight with the density function $k^\tau_{\mathfrak{D}}$ of $\mathbf{X}(\tau)$, see (2.79).

All these approaches are not completely satisfactory, but—not surprisingly—have in common that a low error in the approximation of the initial condition leads to a low error of $u_{\mathfrak{S}}$ at time τ.

Interpretation as altered operator

Now we briefly argue that the ANOVA projections $Q^V_{\mathfrak{m}}, \mathfrak{m}\subset\mathfrak{D}$, applied to the initial condition g result in problems similar to (2.64), but with processes that are no longer Lévy. In [Rei12] it was already mentioned that the anchor ANOVA means replacing the stochastic process with zeros in the respective component.

Let us consider one term in (2.77), which for $\mathfrak{m}\in\mathfrak{S}$ is

$$\begin{aligned}\mathbb{E}[[Q^V_{\mathfrak{m}}g](\mathbf{x}_{\mathfrak{m}}+\mathbf{X}_{\mathfrak{m}}(\tau))] &= \int_{\mathbb{R}^{\mathfrak{m}}}[Q^V_{\mathfrak{m}}g](\mathbf{x}_{\mathfrak{m}}+\mathbf{y}_{\mathfrak{m}})\eta^\tau_{\mathfrak{m}}(\mathrm{d}\mathbf{y}_{\mathfrak{m}})\\ &= \int_{R^{\mathfrak{m}}}\int_{\mathbb{R}^{\mathfrak{D}\setminus\mathfrak{m}}} g((\mathbf{x}_{\mathfrak{m}}+\mathbf{y}_{\mathfrak{m}})\diamond\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}})\mu_{\mathfrak{D}\setminus\mathfrak{m}}(\mathrm{d}\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}})\eta^\tau_{\mathfrak{m}}(\mathrm{d}\mathbf{y}_{\mathfrak{m}})\\ &= \int_{R^{\mathfrak{D}}} g(\mathbf{x}_{\mathfrak{m}}+\mathbf{y}_{\mathfrak{m}}\diamond\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}})(\eta^\tau_{\mathfrak{m}}\otimes\mu_{\mathfrak{D}\setminus\mathfrak{m}})(\mathrm{d}(\mathbf{y}_{\mathfrak{m}}\diamond\mathbf{z}_{\mathfrak{D}\setminus\mathfrak{m}}))\ ,\end{aligned} \tag{2.80}$$

so in fact we convolute our initial condition g with the measure $\eta^\tau_{\mathfrak{m}}\otimes\mu_{\mathfrak{D}\setminus\mathfrak{m}}$, which is the distribution of our process $\mathbf{X}$ with the dimensions $\mathfrak{D}\setminus\mathfrak{m}$ replaced by Gaussian white noise (in case of Gaussian measures $\mu_i, i\in\mathfrak{D}\setminus\mathfrak{m}$) or by constants (in case of Dirac measures).

In (2.77), we wrote $\tilde{u}_{\mathfrak{S}}$ as a linear combination of low-dimensional projections, but in fact we

could also view it as a linear combination of high-dimensional problems in the fashion of (2.64) with a modified stochastic process as in (2.80). We will use this perspective to produce Monte Carlo simulation results in Section 6.4.

2.4 Common approximation schemes

The description of an approximation scheme consists of three simple steps:

1. The selection of a set $\mathfrak{H} \subset \mathcal{P}(\mathfrak{D})$ of subspaces $V_\mathfrak{m}, \mathfrak{m} \in \mathfrak{H}$, that need to be included in our approximation.

2. The computation of the closure

$$\mathfrak{S} = \operatorname{clos}(\mathfrak{H}) := \{\mathfrak{m} \subset \mathfrak{D} : \exists \mathfrak{n} \in \mathfrak{H} \quad \text{with} \quad \mathfrak{m} \subset \mathfrak{n}\}$$

   as the smallest set $\mathfrak{S}$ that satisfies both $\mathfrak{H} \subset \mathfrak{S}$ and the admissibility condition (2.59).

3. The computation of

$$c_\mathfrak{n} = \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{n}} (-1)^{\#\mathfrak{m} - \#\mathfrak{n}} \Big) \quad \text{for} \quad \mathfrak{n} \in \mathfrak{S} \tag{2.81}$$

   and

$$u_\mathfrak{S} = \sum_{\mathfrak{n} \in \mathfrak{S}} c_\mathfrak{n} Q_\mathfrak{n}^V u \, ,$$

   where $Q_\mathfrak{n}^V$ is one of the projection methods discussed in Section 2.3.

2.4.1 Truncation dimension

The truncation dimension is a concept from [CMO97]. Given a d-dimensional problem, we choose a truncation dimension d_t with $d_t \ll d$ assuming that the dimensions $d_t + 1, \ldots, d$ do not contribute much to the result. In fact this means that $\mathfrak{H}$ contains only one set

$$\mathfrak{H} = \{ \{1, \ldots, d_t\} \}$$

and $\mathfrak{S}$ is given by

$$\mathfrak{S} = \{\mathfrak{n} : \mathfrak{n} \subset \{1, \ldots, d_t\}\} \, .$$

It is not hard to see that the coefficients computed by (2.81) in fact look like this

$$c_\mathfrak{n} = \begin{cases} 1 & \text{for} \quad \mathfrak{n} = \{1, \ldots, d_t\} \, , \\ 0 & \text{else} \, , \end{cases}$$

and thus we get $u_\mathfrak{S} = Q^V_{\{1,\ldots,d_t\}} u$, which means we have to compute only one projection and the respective d_t-dimensional problem in (2.77). See Table 2.1 for a list of the involved sets $\mathfrak{n}$ for $d_t = 2$ and $d = 4$. All problems corresponding to the sets below the horizontal line do not need to be computed.

Table 2.1: Sets $\mathfrak{n} \in \mathfrak{S}$ for truncation dimension $d_t = 2$ in $d = 4$ dimensions

set $\mathfrak{n}$	active dimensions	coefficient $c_\mathfrak{n}$
$\{1,2\}$	■ ■ □ □	1
$\{1\}$	■ □ □ □	0
$\{2\}$	□ ■ □ □	0
$\{\}$	□ □ □ □	0

2.4.2 Superposition dimension

The concept of the superposition dimension [CMO97] is a bit more intricate. It means that we aim to include all interactions of at most d_s dimensions. Then, the set $\mathfrak{H}$ is given by

$$\mathfrak{H} = \{\ \{m_1, m_2, \dots, m_{d_s}\} : 1 \le m_1 < m_2 < \cdots < m_{d_s} \le d\ \}\ .$$

Computing $\mathrm{clos}(\mathfrak{H})$ results in

$$\mathfrak{S} = \{\mathfrak{n} \subset \mathfrak{D} : \#\mathfrak{n} \le d_s\}\ .$$

The coefficients are not trivial in this case. We reorder the sum in (2.81) with respect to $k := \#(\mathfrak{m} \setminus \mathfrak{n})$ and use a common identity for series of binomial coefficients

$$c_\mathfrak{n} = \sum_{k=0}^{d_s - \#\mathfrak{n}} (-1)^k \binom{d - \#\mathfrak{n}}{k} = (-1)^{d_s - \#\mathfrak{n}} \binom{d - \#\mathfrak{n} - 1}{d_s - \#\mathfrak{n}}\ . \tag{2.82}$$

Lets have a look at an example with $d = 4$ and $d_s = 2$. Then it holds that

$$c_\mathfrak{n} = \begin{cases} (+1)\binom{1}{0} = 1 & \text{for} \quad \#\mathfrak{n} = 2\ , \\ (-1)\binom{2}{1} = -2 & \text{for} \quad \#\mathfrak{n} = 1\ , \\ (+1)\binom{3}{2} = 3 & \text{for} \quad \#\mathfrak{n} = 0 \Leftrightarrow \mathfrak{n} = \{\}\ . \end{cases}$$

See Table 2.2 for a list of the involved sets.

2.4.3 Combination of truncation and superposition dimension

We now present a mixture of the two approaches, where for fixed d_t and d_s with $d_t + d_s \le d$ all subspaces

$$\mathfrak{H} = \{\ \{1, \dots, d_t\} \cup \{m_1, \dots, m_{d_s}\} : d_t < m_1 < \cdots < m_{d_s} \le d\}$$

are included. The closure $\mathfrak{S} = \mathrm{clos}(\mathfrak{H})$ is given by

$$\mathfrak{S} = \{\ \mathfrak{u} \cup \mathfrak{m} : \mathfrak{u} \subset \{1, \dots, d_t\}, \mathfrak{m} \subset \{d_t + 1, \dots, d\}, \#\mathfrak{m} \le d_s\}\ .$$

Table 2.2: Sets $\mathfrak{n} \in \mathfrak{S}$ for superposition dimension $d_s = 2$ in $d = 4$ dimensions

set $\mathfrak{n}$	active dimensions	coefficient $c_\mathfrak{n}$
$\{1,2\}$	■ ■ □ □	1
$\{1,3\}$	■ □ ■ □	1
$\{1,4\}$	■ □ □ ■	1
$\{2,3\}$	□ ■ ■ □	1
$\{2,4\}$	□ ■ □ ■	1
$\{3,4\}$	□ □ ■ ■	1
$\{1\}$	■ □ □ □	-2
$\{2\}$	□ ■ □ □	-2
$\{3\}$	□ □ ■ □	-2
$\{4\}$	□ □ □ ■	-2
$\{\}$	□ □ □ □	3

On the coefficients $c_{\mathfrak{u}\cup\mathfrak{m}} \in \mathfrak{S}$ with $\mathfrak{u} \subset \{1,\dots d_t\}$ and $\mathfrak{m} \subset \{d_t+1,\dots,d\}, \#\mathfrak{m} \le d_s$, this has the following effect

$$c_{\mathfrak{u}\cup\mathfrak{m}} = \begin{cases} \sum_{k=0}^{s-\#\mathfrak{m}} (-1)^k \binom{d-\#\mathfrak{m}}{k} & \text{for} \quad \mathfrak{u} = \{1,\dots,d_t\}\,, \\ 0 & \text{for} \quad \mathfrak{u} \subsetneq \{1,\dots,d_t\}\,. \end{cases} \tag{2.83}$$

To see that $c_{\mathfrak{u}\cup\mathfrak{m}} = 0$ for $\mathfrak{u} \subsetneq \{1,\dots,d_t\}$, we have to pick an $i \in \{1,\dots,d_t\}$ with $i \notin \mathfrak{u}$ and evaluate (2.81)

$$\begin{aligned} c_{\mathfrak{u}\cup\mathfrak{m}} &= \sum_{\substack{\mathfrak{p}\cup\mathfrak{n}\in\mathfrak{S} \\ \mathfrak{p}\supset\mathfrak{u},\mathfrak{n}\supset\mathfrak{m}}} (-1)^{|\mathfrak{p}|+\#\mathfrak{n}-|\mathfrak{u}|-\#\mathfrak{m}} \\ &= \sum_{\substack{\mathfrak{p}\cup\mathfrak{n}\in\mathfrak{S} \\ i\notin\mathfrak{p},\mathfrak{p}\supset\mathfrak{u},\mathfrak{n}\supset\mathfrak{m}}} (-1)^{|\mathfrak{p}|+\#\mathfrak{n}-|\mathfrak{u}|-\#\mathfrak{m}} + \sum_{\substack{\mathfrak{p}\cup\mathfrak{n}\in\mathfrak{S} \\ i\in\mathfrak{p},\mathfrak{p}\supset\mathfrak{u},\mathfrak{n}\supset\mathfrak{m}}} (-1)^{|\mathfrak{p}|+\#\mathfrak{n}-|\mathfrak{u}|-\#\mathfrak{m}} \\ &= \sum_{\substack{\mathfrak{p}\cup\mathfrak{n}\in\mathfrak{S} \\ i\notin\mathfrak{p},\mathfrak{p}\supset\mathfrak{u},\mathfrak{n}\supset\mathfrak{m}}} (-1)^{|\mathfrak{p}|+\#\mathfrak{n}-|\mathfrak{u}|-\#\mathfrak{m}} - \sum_{\substack{\mathfrak{p}\cup\mathfrak{n}\in\mathfrak{S} \\ i\notin\mathfrak{p},\mathfrak{p}\supset\mathfrak{u},\mathfrak{n}\supset\mathfrak{m}}} (-1)^{|\mathfrak{p}|+\#\mathfrak{n}-|\mathfrak{u}|-\#\mathfrak{m}} = 0\,. \end{aligned} \tag{2.84}$$

For the sake of readability, we tacitly assumed that $\mathfrak{p} \subset \{1,\dots,d_t\}$ and $\mathfrak{n} \subset \{d_t+1,\dots,d\}$ in (2.84). Note that the same idea was already used in the proof of Lemma 2.13. Essentially (2.83) means that the first d_t dimensions are always included and we get coefficients (2.82) applied to the dimensions $\{d_t+1,\dots,d\}$ with superposition dimension d_s. This explains why this is the combination of the truncation and the superposition dimension. Table 2.3 shows the

simple case of $d_t = 1$ and $d_s = 1$ in $d = 4$. Three two-dimensional and one one-dimensional subproblems have to be computed there. In Table 2.4, we see the same setting with superposition dimension $d_s = 2$ and in Table 2.5 with truncation dimension $d_t = 2$.

Table 2.3: Sets $\mathfrak{n} \in \mathfrak{S}$ for truncation dimension $d_t = 1$ and superposition dimension $d_s = 1$ in $d = 4$ dimensions

set $\mathfrak{n}$	active dimensions	coefficient $c_{\mathfrak{n}}$
$\{1,2\}$	■ ■ □ □	1
$\{1,3\}$	■ □ ■ □	1
$\{1,4\}$	■ □ □ ■	1
$\{1\}$	■ □ □ □	-2
$\{2\}$	□ ■ □ □	0
$\{3\}$	□ □ ■ □	0
$\{4\}$	□ □ □ ■	0
$\{\}$	□ □ □ □	0

Relation to other examples in the literature

The idea of approximating a solution u by a fixed truncation dimension and improving the accuracy of the result by "corrective terms" dates back to [Rei04, RW07]. There, a Brownian motion and a diagonal covariance structure is assumed, and the derivative of u with respect to the eigenvalues λ_i of the covariance matrix leads to

$$u(\mathbf{x},t) = u^{(1)}(\mathbf{x},t) + \sum_{j=2}^{d} \left(u^{(1,j)}(\mathbf{x},t) - u^{(1)}(\mathbf{x},t)\right) + \mathcal{O}(\|\lambda - \lambda^{(1)}\|^2) \,. \tag{2.85}$$

In [Rei12] we find that the general idea is related to "parameter bumping" known in financial industry practice, and that corrective terms up to second order look like

$$u(\lambda) = u(0) + \sum_{i=2}^{d} (u(\lambda_i) - u(0)) + \sum_{i \neq j} (u(\lambda_i,\lambda_j) - u(\lambda_i) - u(\lambda_j) + u(0)) + \mathcal{O}(\|\lambda\|_2^3) \,. \tag{2.86}$$

Without introducing the used notation, the similarity of (2.85) and (2.86) to the combination formula (2.55) is obvious. In [SGW13] a proof is given that explains the dimensionality reduction of the associated PDEs by showing that the marginal of a heat kernel is the heat kernel of a lower-dimensional problem. In [RW13] we find a discussion of more complicated stochastic processes than the Brownian motion and numerical results for exotic options like Bermudian swaptions and Ratchet floors.

Table 2.4: Sets $\mathfrak{n} \in \mathfrak{S}$ for truncation dimension $d_t = 1$ and superposition dimension $d_s = 2$ in $d = 4$ dimensions

set $\mathfrak{n}$	active dimensions	coefficient $c_{\mathfrak{n}}$
$\{1,2,3\}$	■ ■ ■ □	1
$\{1,2,4\}$	■ ■ □ ■	1
$\{1,3,4\}$	■ □ ■ ■	1
$\{1,2\}$	■ ■ □ □	-1
$\{1,3\}$	■ □ ■ □	-1
$\{1,4\}$	■ □ □ ■	-1
$\{1\}$	■ □ □ □	1
$\{2,3\}$	□ ■ ■ □	0
$\{2,4\}$	□ ■ □ ■	0
$\{3,4\}$	□ □ ■ ■	0
$\{2\}$	□ ■ □ □	0
$\{3\}$	□ □ ■ □	0
$\{4\}$	□ □ □ ■	0
$\{\}$	□ □ □ □	0

Table 2.5: Sets $\mathfrak{n} \in \mathfrak{S}$ for truncation dimension $d_t = 2$ and superposition dimension $d_s = 1$ in $d = 4$ dimensions

set $\mathfrak{n}$	active dimensions	coefficient $c_{\mathfrak{n}}$
$\{1,2,3\}$	■ ■ ■ □	1
$\{1,2,4\}$	■ ■ □ ■	1
$\{1,2\}$	■ ■ □ □	-1
$\{1,3\}$	■ □ ■ □	0
$\{1,4\}$	■ □ □ ■	0
$\{1\}$	■ □ □ □	0
$\{2,3\}$	□ ■ ■ □	0
$\{2,4\}$	□ ■ □ ■	0
$\{2\}$	□ ■ □ □	0
$\{3\}$	□ □ ■ □	0
$\{4\}$	□ □ □ ■	0
$\{\}$	□ □ □ □	0

We see that there is already a substantial basis in the literature tackling the problem of high-dimensional problems. Our contribution is the rigorous description of the ANOVA decomposition for general measures μ_i and unit functions $\gamma_i, i = 1, \dots, d$, in Section 2.2, which we apply in different ways to the problem (2.64) as described in Section 2.3. We observe that the resulting problems only depend on lower- and moderate-dimensional marginals of Lévy processes, which can be computed easily using Proposition 2.3.

3 Discretization of the moderate-dimensional subproblems

In this chapter, we want to discuss the discretization of (2.30), where the infinitesimal generator (2.31)-(2.32) is based on the process $\mathbf{X}$ with triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$. Sections 3.1 to 3.5 strongly rely on [Win09]. In Section 3.6, we discuss the special case (2.42) based on $\mathbf{Y}$ and its lower-dimensional $\mathfrak{m}$-marginals $\mathbf{Y}_\mathfrak{m}$, and we state what problems we finally want to solve numerically.

3.1 Function spaces

Let $\mathcal{D}$ be an open subset of $\mathbb{R}^d$ with a piecewise smooth boundary, and let $L_2(\mathcal{D})$ be the usual class of square-integrable functions on $\mathcal{D}$ with norm $\|f\|^2_{L_2(\mathcal{D})} := \int_\mathcal{D} (f(\mathbf{x}))^2 \mathrm{d}\mathbf{x}$. Then, we can define the scalar product

$$(u, v)_{H^m(\mathcal{D})} := \sum_{|\boldsymbol{\alpha}|_1 \leq m} (\partial^{\boldsymbol{\alpha}} u, \partial^{\boldsymbol{\alpha}} v) \tag{3.1}$$

for all functions in $L_2(\mathcal{D})$ that have square-integrable weak derivatives $\partial^{\boldsymbol{\alpha}} f = \partial^{\alpha_1} \cdots \partial^{\alpha_d} f$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$, up to order $m \in \mathbb{N}$, see [Bra07a]. This gives rise to the definition of the Hilbert space

$$H^m(\mathcal{D}) := \{f \in L_2(\mathcal{D}) : \|f\|_{H^m(\mathcal{D})} < \infty\}$$

with inner product (3.1) and norm $\|f\|_{H^m(\mathcal{D})} := (f, f)^{1/2}_{H^m(\mathcal{D})}$. The functions in $H^m(\mathcal{D})$ are of *isotropic* smoothness, and we can define Sobolev spaces with *mixed* smoothness by the scalar product

$$(u, v)_{H^m_{\mathrm{mix}}(\mathcal{D})} := \sum_{|\boldsymbol{\alpha}|_\infty \leq m} (\partial^{\boldsymbol{\alpha}} u, \partial^{\boldsymbol{\alpha}} v) \tag{3.2}$$

and the corresponding Hilbert space

$$H^m_{\mathrm{mix}}(\mathcal{D}) := \{f \in L_2(\mathcal{D}) : \|f\|_{H^m_{\mathrm{mix}}(\mathcal{D})} < \infty\}$$

with inner product (3.2) and norm $\|f\|_{H^m_{\mathrm{mix}}(\mathcal{D})} := (f, f)^{1/2}_{H^m_{\mathrm{mix}}(\mathcal{D})}$. This smoothness class is relevant when we deal with sparse grid discretizations in Chapter 4. For further information and combinations of isotropic and mixed smoothness, cf. [GK09].

We now define function classes with zero boundary conditions by

$$H^m_0(\mathcal{D}) := \{ f|_\mathcal{D} : f \in H^m(\mathbb{R}^d),\, f|_{\mathbb{R}^d \setminus \mathcal{D}} = 0\}$$

and

$$H^m_{0,\mathrm{mix}}(\mathcal{D}) := \{ f|_\mathcal{D} : f \in H^m_{\mathrm{mix}}(\mathbb{R}^d),\, f|_{\mathbb{R}^d \setminus \mathcal{D}} = 0\} \,.$$

We conclude this short section with a few remarks. Let us denote the space of infinitely differentiable functions by $C^\infty(\mathcal{D})$. We define the space of smooth functions with zero boundary conditions by

$$C_0^\infty(\mathcal{D}) := \{f \in C^\infty(\mathcal{D}) : \operatorname{supp} f \subsetneq \mathcal{D}\}$$

Then, $H^m(\mathcal{D}) \cap C^\infty(\mathcal{D})$ is dense in $H^m(\mathcal{D})$, and $\mathbb{C}_0^\infty(\mathcal{D})$ is dense in $H_0^m(\mathcal{D})$, i.e., $H_0^m(\mathcal{D})$ is the closure of $C_0^\infty(\mathcal{D})$ with respect to the norm $\|\cdot\|_{H^m(\mathcal{D})}$. This holds for $m \in \mathbb{N}$. Fourier representations of functions are needed to compute function norms as in (3.1) and (3.2) for non-integer orders m, but as we do not deal with pure jump processes as, e.g., in [Win09], we do not need these spaces.

3.2 Variational formulation

We introduce the bilinear form

$$\mathcal{E}(u,v) = \frac{1}{2}\sum_{i,j=1}^{d} q_{ij} \int_{\mathbb{R}^d} \frac{\partial u}{\partial x_i}(\mathbf{x}) \frac{\partial v}{\partial x_j}(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{3.3}$$

$$- \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \left(u(\mathbf{x}+\mathbf{z}) - u(\mathbf{x}) - \sum_{i=1}^{d} z_i \frac{\partial u}{\partial x_i}(\mathbf{x}) \right) v(\mathbf{x}) \mathrm{d}\mathbf{x} \nu(\mathrm{d}\mathbf{z}) \tag{3.4}$$

for functions $u, v \in C_0^\infty(\mathbb{R}^d)$. We assume that the covariance matrix $\mathbf{Q}$ is symmetric positive definite and that the drift of our process $\mathbf{X}$ is zero, i.e., $\boldsymbol{\theta} = \mathbf{0}$. Then, [Win09, Theorem 3.2.2] tells us that we can express (2.30) as the uniquely solvable variational problem:

Find $u \in L_2((0,T); H^1(\mathbb{R}^d)) \cap H^1((0,T); H^1(\mathbb{R}^d)^*)$ such that

$$\left\langle \frac{\partial u}{\partial \tau}, v \right\rangle_{H^1(\mathbb{R}^d)^*, H^1(\mathbb{R}^d)} + \mathcal{E}(u,v) = 0 \quad \text{for} \quad \tau \in (0,T) \quad \forall\, v \in H^1(\mathbb{R}^d) \tag{3.5}$$

with $u(0,\cdot) = g \in L_2(\mathbb{R}^d)$. In (3.5), $\langle \cdot, \cdot \rangle_{H^1(\mathbb{R}^d)^*, H^1(\mathbb{R}^d)}$ denotes the duality pairing. The requirement $g \in L_2(\mathbb{R}^d)$ can be softened after we have localized our domain.

3.3 Localization

Prior to an effective discretization of (3.5) in space we restrict ourselves to a finite domain

$$\mathcal{D} = [-\zeta, \zeta]^d \subset \mathbb{R}^d\,. \tag{3.6}$$

Essentially, this means that instead of computing the quantity

$$u(\tau, \mathbf{x}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})]$$

as in (2.21), we compute

$$u_{\mathcal{D}}(\tau, \mathbf{x}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x}) \mathbb{1}_{\{\iota_{\mathcal{D}} > \tau\}}]\,,$$

where $\iota_{\mathcal{D}}$ is the stopping time of $\mathbf{X}(\tau)$ reaching $\mathbb{R}^d \setminus \mathcal{D}$. This truncation can be understood as an approximation of the price of the option by that of a barrier option, i.e., as an option that has no payoff if the underlying process $\mathbf{X}$ leaves $\mathcal{D}$ at least once before maturity.

Given that the payoff function (2.20) grows only polynomially

$$g(\log \mathbf{s}) = h(\mathbf{s}) \lesssim \left(\sum_{i=1}^{d} s_i + 1 \right)^q \qquad \forall\, \mathbf{s} \in \mathbb{R}^d_{\geq 0}$$

and that the marginal Lévy measures ν_i have densities $k_i : \mathbb{R} \to \mathbb{R}_{\geq 0}$ that satisfy

$$k_i(z) \lesssim \begin{cases} e^{-\beta_i^- |z|} & \text{for} \quad z < -1 \,, \\ e^{-\beta_i^+ z} & \text{for} \quad z > 1 \,, \end{cases} \tag{3.7}$$

where $\beta_i^+, \beta_i^- > q$ holds, [Win09, Theorem 3.3.2] states that the solution of the localized problem converges pointwise exponentially to the solution of the original problem, i.e.

$$|u(\tau, \mathbf{x}) - u_{\mathcal{D}}(\tau, \mathbf{x})| \lesssim e^{-\gamma_1 \zeta + \gamma_2 \|\mathbf{x}\|_\infty} \,. \tag{3.8}$$

In (3.8), the constants γ_1 and γ_2 satisfy $0 < \gamma_1 < \min(\min_{i=1}^d \beta_i^+, \min_{i=1}^d \beta_i^-) - q$ and $\gamma_2 = \gamma_1 + q$. The exponential decrease in the pointwise error is also discussed in [CV05]. We see that a fast decay (3.7) of all marginal Lévy measures implies a fast decay of the pointwise error.

It is possible to balance the domain size and the decay of the Lévy measure instead of using a hypercube domain (3.6). In some of our numerical experiments it turned out useful to use an adapted domain

$$\mathcal{D} = (\alpha_1^-, \alpha_1^+) \times \cdots \times (\alpha_d^-, \alpha_d^+) \tag{3.9}$$

with

$$\alpha_i^- = c_i - \zeta \cdot \sigma(X_i(T)) \,, \tag{3.10}$$
$$\alpha_i^+ = c_i + \zeta \cdot \sigma(X_i(T)) \,, \tag{3.11}$$

where $\mathbf{c} = (c_1, \ldots, c_d)$ is our point of interest, i.e., we evaluate $u(\tau, \mathbf{x})$ mostly where $\mathbf{x}$ is in the proximity of $\mathbf{c}$, and $\sigma(X_i(T)), i = 1, \ldots, d$, denote the standard deviations of the components of our stochastic process until maturity T. The particular choice in (3.9) should result in the same error convergence rates as (3.8) but with an improved constant factor.

As many quadrature rules and discretization choices are typically given for $(0,1)^d$, we need an affine linear scaling $\mathcal{S} : \mathcal{D} \to (0,1)^d$ with

$$\mathcal{S} : \mathbf{x} \mapsto \left(\frac{x_1 - \alpha_1^-}{\alpha_1^+ - \alpha_1^-}, \ldots, \frac{x_d - \alpha_d^-}{\alpha_d^+ - \alpha_d^-} \right) \tag{3.12}$$

and

$$\mathcal{S}^{-1} : \mathbf{x} \mapsto \left(\alpha_1^- + x_1(\alpha_1^+ - \alpha_1^-), \ldots, \alpha_d^- + x_d(\alpha_d^+ - \alpha_d^-) \right) \,.$$

So, let us assume that $u, v \in H_0^1(\Omega^d)$ with $\Omega = (0,1)$. Then, $u(\mathcal{S}\,\cdot)$ and $v(\mathcal{S}\,\cdot)$ are functions on the domain $\mathcal{D}$. If we denote their zero extension on $\mathbb{R}^d \setminus \mathcal{D}$ by $\tilde{\mathcal{S}}u$ and $\tilde{\mathcal{S}}v$, we see the continuity

and coercivity of the bilinear form

$$\mathcal{E}_{\Omega^d}(u,v) := \mathcal{E}(\tilde{\mathcal{S}}u, \tilde{\mathcal{S}}v) \ ,$$

which is defined on $H_0^1(\Omega^d)$. Now the problem we want to solve reads:

Find $u \in L_2((0,T); H_0^1(\Omega^d)) \cap H^1((0,T); H_0^1(\Omega^d)^*)$ such that

$$\left\langle \frac{\partial u}{\partial \tau}, v \right\rangle_{H^1(\Omega^d)^*, H^1(\Omega^d)} + \mathcal{E}_{\Omega^d}(u,v) = 0 \quad \text{for} \quad \tau \in (0,T) \quad \forall\, v \in H_0^1(\Omega^d) \tag{3.13}$$

with $u(0,\cdot) = g_{\Omega^d} \in L_2(\Omega^d)$, where $\tilde{S} g_{\Omega^d} \equiv g$ on $\mathcal{D}$.

3.4 Space and time discretization

We need a discretization of (3.13) in space and time. Typically space is discretized first, which results in the method of lines. Let us assume a discretization space

$$V_N = \operatorname{span}\{\phi_i : i = 1, \ldots, N\}$$

with N degrees of freedom. Then, the Galerkin projection of (3.13) on V_N results in the problem:

Find $u_N \in C^1([0,T]; V_N)$ such that

$$\left\langle \frac{\partial u_N}{\partial \tau}, v_N \right\rangle_{L_2(\Omega^d)} + \mathcal{E}_{\Omega^d}(u_N, v_N) = 0 \quad \text{for} \quad \tau \in (0,T) \qquad \forall\, v_N \in V_N \tag{3.14}$$

with $u_N(0,\cdot) = g_{\Omega^d,N}$, where $g_{\Omega^d,N}$ is an approximation to g_{Ω^d} in V_N.

For the time discretization, we subdivide the interval $[0,T]$ into $M+1$ equidistant time-steps

$$t_k = k\Delta t \ , \qquad k = 0, \ldots, M \ ,$$

with $\Delta t = \frac{T}{M}$ and apply the well-known θ-scheme with $\theta = 1$ (implicit Euler) or $\theta = \frac{1}{2}$ (Crank-Nicolson).[1] So we finally have to solve in every time step $k = 1, \ldots, M$ the sequence of $H^1(\Omega^d)$-elliptic variational problems:

Given $u_N^{(k-1)}$, find $u_N^{(k)} \in V_N$, such that for all $v_N \in V_N$

$$\Delta t^{-1}(u_N^{(k)}, v_N)_{L_2(\Omega^d)} + \theta \mathcal{E}_{\Omega^d}(u_N^{(k)}, v_N) = \Delta t^{-1}(u_N^{(k-1)}, v_N)_{L_2(\Omega^d)} - (1-\theta)\mathcal{E}_{\Omega^d}(u_N^{(k-1)}, v_N) \tag{3.15}$$

with $u_N^{(0)} = g_{\Omega^d,N}$. We can easily cast the problem (3.15) in the typical form of variational problems

$$a(u_N^{(k)}, v_N) = F^{(k-1)}(v_N) \qquad \forall\, v_N \in V_N \tag{3.16}$$

[1] More sophisticated space-time discretizations are available [SS00, GOV05], but they are beyond the scope of this work.

by setting the bilinear form to

$$a(u_N^{(k)}, v_N) = \Delta t^{-1}(u_N^{(k)}, v_N)_{L_2(\Omega^d)} + \theta \mathcal{E}_{\Omega^d}(u_N^{(k)}, v_N) \tag{3.17}$$

and the right-hand side to

$$F^{(k-1)}(v_N) = \Delta t^{-1}(u_N^{(k-1)}, v_N)_{L_2(\Omega^d)} - (1-\theta)\mathcal{E}_{\Omega^d}(u_N^{(k-1)}, v_N) \ , \tag{3.18}$$

which, in the k-th time step, results in a system of linear equations

$$\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}^{(k-1)} \tag{3.19}$$

with the stiffness matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$

$$(\mathbf{A})_{ij} = a(\phi_j, \phi_i) \quad \text{for} \quad i,j = 1, \dots, N \ ,$$

the vector representation $\mathbf{x}^{(k)} = (x_1^{(k)}, \dots, x_N^{(k)})$ of

$$u^{(k)} = \sum_{j=1}^{N} x_j^{(k)} \phi_j$$

and right-hand side

$$\mathbf{b}_i^{(k-1)} = F^{(k-1)}(\phi_i) \ .$$

Sophisticated discretization spaces and the preconditioning of (3.19) will be discussed in Chapter 4 and Chapter 5, respectively.

3.5 Convergence rates

In this section we describe some standard convergence results that can be found in [Tho06, EG04]. Let $\mathcal{T}_h$ be a regular conforming partition of Ω^d, where h denotes the uniform mesh-width. Then, for V_N that contain polynomials of maximum degree $p-1$ on $\mathcal{T}_h$, we obtain the order of accuracy p, and the best approximation rate we can expect for a function $u \in H^r(\Omega^d)$ with $r \leq p$ in terms of h is

$$\inf_{u_h \in V_N} \|u - u_h\|_{H^s(\Omega^d)} \lesssim h^{r-s} \|u\|_{H^r(\Omega^d)} \ ,$$

where $0 \leq s < r$. The same rate can be expected for the discrete solution of Poisson's equation when $s = 0, 1$ and $1 \leq r \leq p$. Note that the convergence rate is expressed in terms of h, but $h \sim N^{-1/d}$ implies $N \sim h^{-d}$, so the number of degrees of freedom N grows exponentially with the dimension. Sparse grids, which will be discussed in Chapter 4, offer a remedy if the necessary regularity assumptions on u are met.

We now turn to convergence results for time-dependent problems. A simple parabolic problem

with solution u and the space discrete solution u_N like in (3.14) results in the estimate

$$\|u_N(\tau) - u(\tau)\|_{L_2(\Omega^d)} \leq \|g_{\Omega^d,N} - g_{\Omega^d}\|_{L_2(\Omega^d)} + Ch^r \left(\|g_{\Omega^d}\|_{H^r(\Omega^d)} + \int_0^\tau \left\| \tfrac{\partial u(s,\cdot)}{\partial \tau} \right\|_{H^r(\Omega^d)} \mathrm{d}s \right)$$

for $\tau \geq 0$, which means that we get the approximation error of the initial condition and a term that depends on the initial condition and the solution, both converging with rate r.

Now, we state a result with discretized time as in (3.15) with $\theta = 1$

$$\begin{aligned} \|u(t_k) - u_N^{(k)}\|_{L_2(\Omega^d)} \leq & Ch^r \left(\|g_{\Omega^d}\|_{H^r(\Omega^d)} + \int_0^{t_k} \left\| \tfrac{\partial u(s,\cdot)}{\partial \tau} \right\|_{H^r(\Omega^d)} \mathrm{d}s \right) \\ & + \Delta t \int_0^{t_k} \left\| \tfrac{\partial^2 u(s,\cdot)}{\partial \tau^2} \right\|_{L_2(\Omega^d)} \mathrm{d}s \,, \end{aligned}$$

so we have first-order convergence in time. The Crank-Nicolson scheme with $\theta = \frac{1}{2}$ typically results in second-order convergence with respect to Δt and we get

$$\begin{aligned} \|u(t_k) - u_N^{(k)}\|_{L_2(\Omega^d)} \leq & Ch^r \left(\|g_{\Omega^d}\|_{H^r(\Omega^d)} + \int_0^{t_k} \left\| \tfrac{\partial u(s,\cdot)}{\partial \tau} \right\|_{H^r(\Omega^d)} \mathrm{d}s \right) \\ & + C\Delta t^2 \int_0^{t_k} \left(\left\| \tfrac{\partial^3 u(s,\cdot)}{\partial \tau^3} \right\|_{L_2(\Omega^d)} + \left\| \Delta \tfrac{\partial^2 u(s,\cdot)}{\partial \tau^2} \right\|_{L_2(\Omega^d)} \right) \mathrm{d}s \,. \end{aligned}$$

To summarize the results: The $L_2(\Omega^d)$ error of our discrete result $u_N^{(M)}$ after M time steps compared to the true solution $u(T,\cdot)$ decreases with rate r with respect to the mesh width h and with first-order in time for $\theta = 1$ and with second-order in time for $\theta = \frac{1}{2}$.

3.6 Further simplifications and summary

In this chapter, we stated some well-known results regarding the discretization of the BKE. Now we become more specific what problem we will finally solve in our numerical experiments in Chapter 6. For a given process $\mathbf{X}$ with triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$, we can always diagonalize the covariance matrix and end up with a process $\mathbf{Y}(\tau) = \mathbf{B}^T \mathbf{X}(\tau)$. Then we need to consider (2.42) with the infinitesimal generator (2.31) and (2.32) of the process $\mathbf{Y}(\tau)$. This amounts to solving the equation

$$\frac{\partial v(\tau, \mathbf{y})}{\partial \tau} = \sum_{i=1}^d \sigma_i^2 \frac{\partial^2 v}{\partial y_i^2}(\tau, \mathbf{y}) + \sum_{i=1}^d (\mathbf{B}^T \boldsymbol{\theta})_i \frac{\partial v}{\partial y_i}(\tau, \mathbf{y}) \tag{3.20}$$

$$+ \int_{\mathbb{R}^d} \left(v(\tau, \mathbf{y} + \mathbf{z}) - v(\tau, \mathbf{y}) - \sum_{i=1}^d z_i \frac{\partial v}{\partial y_k}(\tau, \mathbf{y}) \right) \nu_{\mathbf{B}}(\mathrm{d}\mathbf{z}) \tag{3.21}$$

on $\mathbf{y} \in \mathbb{R}^d, \tau \in (0, T)$ with a given initial condition $v(0, \mathbf{y}) = g_{\mathbf{B}}(\mathbf{y})$.

Remark 3.1. The weak form of the right-hand side of (3.20) and (3.21) does not yet result in the Dirichlet form (3.3) and (3.4). There, zero drift was assumed. We can realize this by a simple change of variables. Instead of the process $\mathbf{Y}(\tau) = \mathbf{B}^T \mathbf{X}(\tau)$ and the variable transform $\mathbf{y} = \mathbf{B}^T \mathbf{x}$

we deal with the process $\mathbf{Y}(\tau) = \mathbf{B}^T\mathbf{X}(\tau) - \tau\mathbf{B}^T\boldsymbol{\theta}$ and the variable transform $\mathbf{y} = \mathbf{B}^T\mathbf{x} + \tau\mathbf{B}^T\boldsymbol{\theta}$, respectively.

So far, we have assumed $E[|\mathbf{X}(\tau)|] < \infty$ and (2.26), which allowed us to use the truncation function $\mathcal{T}(\mathbf{z}) = 1$. That setup still covers infinite activity models. However, in our numerical experiments in Chapter 6, we only work with finite activity models. This means that instead of the condition (2.3), we have finite activity

$$\lambda := \nu_{\mathbf{B}}(\mathbb{R}^d) = \nu(\mathbb{R}^d) < \infty \, . \tag{3.22}$$

This allows us to come up with a simple form of (3.20) and (3.21):

$$\frac{\partial v(\tau,\mathbf{y})}{\partial \tau} = \sum_{i=1}^{d} \sigma_i^2 \frac{\partial^2 v}{\partial y_i^2}(\tau,\mathbf{y}) - \lambda v(\tau,\mathbf{y}) + \int_{\mathbb{R}^d} v(\tau,\mathbf{y}+\mathbf{z})\nu_{\mathbf{B}}(\mathrm{d}\mathbf{z}) \tag{3.23}$$

$$+ \sum_{i=1}^{d} \left((\mathbf{B}^T\boldsymbol{\theta})_i - \int_{\mathbb{R}} z_i \nu_{\mathbf{B}}(\mathrm{d}\mathbf{z}) \right) \frac{\partial v}{\partial y_i}(\tau,\mathbf{y}) \quad \text{on} \quad \mathbf{y} \in \mathbb{R}^d, \tau \in (0,T) \, . \tag{3.24}$$

Considering

$$\int_{\mathbb{R}^d} \mathbf{z}\nu_{\mathbf{B}}(\mathrm{d}\mathbf{z}) = \int_{\mathbb{R}^d} \mathbf{B}^T\mathbf{z}\nu(\mathrm{d}\mathbf{z}) = \mathbf{B}^T \int_{\mathbb{R}^d} \mathbf{z}\nu(\mathrm{d}\mathbf{z}) \, ,$$

we can remove the convection term in (3.24) by using the transformation

$$\mathbf{y} = \mathbf{B}^T\mathbf{x} + \tau\mathbf{B}^T \left(\boldsymbol{\theta} - \int_{\mathbb{R}^d} \mathbf{z}\nu(\mathrm{d}\mathbf{z}) \right)$$

instead of the original transformation $\mathbf{y} = \mathbf{B}^T\mathbf{x}$.

We now summarize all the transformations we have used so far. Under the assumption of finite activity (3.22), the BKE (2.27) and (2.28) can be transformed to the PIDE

$$\frac{\partial v}{\partial \tau}(\tau,\mathbf{y}) = \frac{1}{2}\sum_{i=1}^{d} \sigma_i^2 \frac{\partial^2 v}{\partial y_i^2}(\tau,\mathbf{y}) - \lambda v(\tau,\mathbf{y}) + \int_{\mathbb{R}^d} v(\tau,\mathbf{y}+\mathbf{z})\nu_{\mathbf{B}}(\mathrm{d}\mathbf{z}) \quad \text{on} \quad \mathbf{y} \in \mathbb{R}^d, \tau \in (0,T) \tag{3.25}$$

by setting $v(\tau,\mathbf{y}) = e^{(T-t)r}V(t,\mathbf{s})$ with $\tau = T - t$ and

$$\mathbf{y} = \mathbf{B}^T \left(\mathbf{x} + \tau \left(\boldsymbol{\theta} - \int_{\mathbb{R}^d} \mathbf{z}\nu(\mathrm{d}\mathbf{z}) \right) \right) = \mathbf{B}^T \left(\log \mathbf{s} + \tau(r + \boldsymbol{\theta} - \int_{\mathbb{R}^d} \mathbf{z}\nu(\mathrm{d}\mathbf{z})) \right) \, .$$

The final condition (2.29) becomes an initial condition with

$$v(0,\mathbf{y}) = u(0,\mathbf{B}\mathbf{y}) = V(T,\exp(\mathbf{B}\mathbf{y})) = h(\exp(\mathbf{B}\mathbf{y})) = g(\mathbf{B}\mathbf{y}) = g_{\mathbf{B}}(\mathbf{y}) \, ,$$

see Theorem 2.10. From now on, we refer to equation (3.25) only. The corresponding bilinear form is

$$\mathcal{E}(u,v) = \frac{1}{2}\int_{\mathbb{R}^d} \sum_{i=1}^{d} \sigma_i^2 \frac{\partial u(\mathbf{x})}{\partial x_i} \frac{\partial v(\mathbf{x})}{\partial x_j} \mathrm{d}\mathbf{x} + \lambda(u,v)_{L_2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} u(\tau,\mathbf{x}+\mathbf{z})\nu(\mathrm{d}\mathbf{z})v(\mathbf{x})\mathrm{d}\mathbf{x} \, , \tag{3.26}$$

which is to be used in (3.5).

The final steps are the localization, transformation to Ω^d and the discretization of space and time. In the end, we compute $g_{\Omega^d,N}$ as the $L_2(\Omega^d)$ approximation of the initial condition $g_{\Omega^d}(\mathbf{y}) = g(S^{-1}\mathbf{y})$ in our space V_N, and then solve (3.16) on the basis of (3.26) M times. At this point we explicitly state the resulting bilinear form $a(\cdot,\cdot)$ from (3.17)

$$\begin{aligned} a(u,v) =& \theta \sum_{i=1}^{d} \frac{\sigma_i^2}{2(\alpha_i^+ - \alpha_i^-)^2} \int_{\Omega^d} \frac{\partial u(\mathbf{x})}{\partial x_i} \frac{\partial v(\mathbf{x})}{\partial x_i} \mathrm{d}\mathbf{x} + (\theta\lambda + \Delta t^{-1}) \int_{\Omega^d} u(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x} \\ & - \theta \prod_{i=1}^{d} (\alpha_i^+ - \alpha_i^-) \int_{\Omega^d} \int_{\Omega^d} u(\mathbf{x}+\mathbf{z}) \nu_{\mathbf{B}}(\mathcal{S}^{-1}(\mathrm{d}\mathbf{z}))\, v(\mathbf{x}) \mathrm{d}\mathbf{x} \end{aligned} \tag{3.27}$$

with right hand-side (3.18)

$$\begin{aligned} F^{(k-1)}(v) =& (1-\theta) \sum_{i=1}^{d} \frac{\sigma_i^2}{2(\alpha_i^+ - \alpha_i^-)^2} \int_{\Omega^d} \frac{\partial u^{(k-1)}(\mathbf{x})}{\partial x_i} \frac{\partial v(\mathbf{x})}{\partial x_i} \mathrm{d}\mathbf{x} \\ & + ((1-\theta)\lambda + \Delta t^{-1}) \int_{\Omega^d} u^{(k-1)}(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x} \\ & - (1-\theta) \prod_{i=1}^{d} (\alpha_i^+ - \alpha_i^-) \int_{\Omega^d} \int_{\Omega^d} u^{(k-1)}(\mathbf{x}+\mathbf{z}) \nu_{\mathbf{B}}(\mathcal{S}^{-1}(\mathrm{d}\mathbf{z}))\, v(\mathbf{x}) \mathrm{d}\mathbf{x} \,. \end{aligned} \tag{3.28}$$

Having clarified how we discretize the BKE, we now briefly summarize the methods presented so far:

- We are interested in the function $V(t,\mathbf{s})$ from (2.1) based on the process $\mathbf{S}$. We know $V(t,\mathbf{s})$ satisfies the BKE (2.27) and (2.28).
- Due to some straightforward transformations described in Section 2.1, we can express our problem in terms of $v(\tau,\mathbf{y})$ based on the process $\mathbf{Y}$, see (2.23).
- Now, for low- and moderate-dimensional d, we can discretize the corresponding BKE (2.42) directly using the methods described in this chapter.
- If the dimension d is too high to solve the BKE directly:
  - We use the ANOVA approximation technique and obtain a superposition of moderate-dimensional subproblems (2.77) based on the marginals of $\mathbf{Y}$. From Proposition 2.3 we know that the marginals $\mathbf{Y}_{\mathfrak{m}}, \mathfrak{m} \in \mathfrak{S}$, have triplets $(\boldsymbol{\Sigma}_{\mathfrak{m}}, (\mathbf{B}^T\boldsymbol{\theta})_{\mathfrak{m}}, (\nu_{\mathbf{B}})_{\mathfrak{m}})$.
  - Now, for every $\mathfrak{m} \in \mathfrak{S}$, we can solve the $\#\mathfrak{m}$-dimensional PIDE (2.42) using the techniques described in this chapter. Then, we combine the results using the summation in (2.77).

In the next chapter, we present the sparse grid discretizations to be used for V_N, and in Chapter 5 we deal with the preconditioning of an abstract variational problem (3.16) discretized by generalized sparse grids.

4 Sparse grid spaces and fast operator application

In this chapter, we recall generalized sparse grid spaces. First we start with a little motivation based on the multiplication method by John Napier (1550–1617), see Figure 4.1. Obviously, both input values are given with an accuracy of only three decimal places and there is no point in computing a result which has a higher accuracy than that. Thus, the intermediate results in the gray shaded area do not contribute to the final result apart from a small carry of $+1 \cdot 10^{-3}$ and therefore essentially do not need to be computed.

2	.	3	7	4	×	1	.	5	4	4
			2	3	7	4				
+			1	1	8	7	0			
+					9	4	9	6		
+						9	4	9	6	
=		3	.	6	6	5	4	5	6	

Figure 4.1: Multiplication of two decimal numbers

Let us assume that two numbers $a = \sum_{i=0}^{p} a_i \cdot 10^{-i}$ and $b = \sum_{j=0}^{p} b_j \cdot 10^{-j}$ with precision p are given and that we can only perform the multiplication of single digits. Then, we can rearrange the multiplication

$$a \cdot b = \left(\sum_{i=0}^{p} a_i \cdot 10^{-i} \right) \cdot \left(\sum_{j=0}^{p} b_j \cdot 10^{-j} \right) \tag{4.1}$$

$$= \sum_{i,j=0}^{p} a_i \cdot b_j \cdot 10^{-i-j}$$

$$= \sum_{k=0}^{2p} \left(\sum_{\substack{i+j=k \\ i,j \geq 0}} a_i \cdot b_j \right) \cdot 10^{-k} \,. \tag{4.2}$$

As we are only interested in a precision of up to the p-th digit, we can essentially discard any summands with $k > p$ in (4.2), and instead of performing $(p+1)^2$ multiplications in (4.1), we

only need

$$1 + \cdots + (p+1) = \frac{(p+1)(p+2)}{2}$$

digit multiplication operations. The same idea can be applied to products with more than two factors with an even greater reduction in the number of digit multiplications.

In this somewhat artificial example, numbers are represented in a multiscale system of different powers of 10, and combinations of less significant digits can be discarded. The same idea applied to the tensorization of a multiscale basis for multi-dimensional function approximation is the principle of the sparse grid approach. There, the computational costs typically increase by a factor of 2 between two levels in the multiscale basis, so the gain in efficiency by discarding finer scales is even larger than in our simple multiplication example, where the computational costs do not increase for the less significant digits. The sparse grid approach has been used in a myriad of relevant applications, among them radiative transfer equations [WHS08], high-dimensional numerical integration in finance [GG03, GH10b], the Hamilton–Jacobi–Bellmann equation [BGGK12], data mining [GGT01, GH09] and time-series prediction [BG12], the Schrödinger-equation in quantum chemistry [Yse05, Ham10], the pricing of basket options [BHPS11] and PDEs with stochastic input data [BNT10, JR08, NTW08].

In Section 4.1, we give the definition of a generalized sparse grid space and state some approximation rates. In Section 4.2, we present a redundant but natural discretization of generalized sparse grid spaces via a generating system. Finally, in Section 4.3, we describe two matrix-vector multiplication algorithms for operator matrices that stem from generalized sparse grid discretizations of variational problems.

4.1 Definition

The building blocks of a d-dimensional sparse grid discretization are sequences of dense finite-dimensional subspaces

$$V_0^{(p)} \subset V_1^{(p)} \subset V_2^{(p)} \subset \ldots \subset V^{(p)}, \qquad p = 1, \ldots, d\,, \tag{4.3}$$

on the unit interval $\Omega = (0,1)$, equipped with a scalar product $(\cdot,\cdot)_{V^{(p)}}$. Typically, the spaces $V_l^{(p)}, l \in \mathbb{N}$, in (4.3) are discretizations with dyadically refined linear or higher-order splines with $n_l = \Theta(2^l)$ basis functions on level l. Note that often $l = 1$ is the first non-trivial level, but for our purposes it is useful to start from $l = 0$. In the following, we consider linear spline spaces spanned by $n_l = 2^{l+1} - 1$ hat functions

$$\phi_{l,i}(x) = \max(1 - 2^{l+1}\,|x - x_{l,i}|\,, 0)\,, \qquad i = 1, \ldots, n_l\,, \tag{4.4}$$

which are centered at the points of an equidistant mesh $x_{l,i} := 2^{-l-1}i$. See Figure 4.2 for an illustration of the basis functions on the levels $l = 0, \ldots, 3$. If desired, we can define boundary functions $\tilde{\phi}_{l,0} := \phi_{l,0}|_{[0,1]}$ and $\tilde{\phi}_{l,1} := \phi_{l,n_l+1}|_{[0,1]}$.

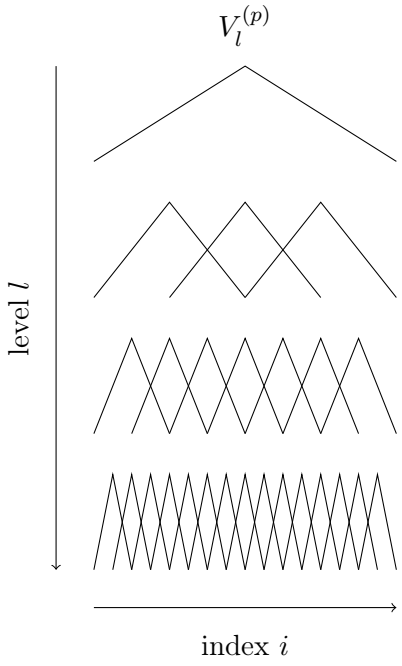

Figure 4.2: Multiscale basis of linear splines

By tensorization, we obtain the spaces

$$V_{\mathbf{l}} = \bigotimes_{p=1}^{d} V_{l_i}^{(p)},$$

where $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$ denotes a level multi-index, and $V_{\mathbf{l}}$ is a function space on the d-dimensional unit cube $\Omega^d := (0,1)^d$ equipped with the tensor scalar product $(\cdot,\cdot)_V$. Then, $V_{\mathbf{l}}$ is the space of piecewise d-linear functions on an (in general) anisotropic full grid with

$$n_{\mathbf{l}} := \#V_{\mathbf{l}} = \prod_{p=1}^{d} (2^{l_p+1} - 1) = \Theta(2^{|\mathbf{l}|_1})$$

degrees of freedom. It is spanned by the functions

$$\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = \phi_{l_1,i_1}(x_1) \cdots \phi_{l_d,i_d}(x_d) \tag{4.5}$$

with $\mathbf{x} = (x_1, \ldots, x_d) \in \Omega^d$ and

$$\mathbf{i} \in \chi_{\mathbf{l}} := \{\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^d : 1 \le j_p \le n_{l_p}, p = 1, \ldots, d\} \,. \tag{4.6}$$

The space

$$V = \sum_{\mathbf{l} \in \mathbb{N}^d} V_{\mathbf{l}}$$

is equal to the underlying Sobolev space $H_0^1(\Omega^d)$ up to completion with the H^1-norm, see [BG04]. Note that domains different to Ω^d can be treated by a possibly non-linear transform to Ω^d and dealing with the emerging variable coefficient functions, cf. [Ach03].

Remark 4.1. Note that the multilevel spaces $V_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}^d$, are not the same as the ANOVA subspaces $V_{\mathfrak{m}}, \mathfrak{m} \subset \mathfrak{D}$, used in Chapter 2 even though the basic idea is similar: The spaces $V_{\mathfrak{m}}$ can be understood as two-level spaces where a dimension $i \in \mathfrak{D}$ is either *active* ($i \in \mathfrak{m}$) or *inactive* ($i \in \mathfrak{D} \setminus \mathfrak{m}$).

For our numerical computation, we have to resort to a finite-dimensional subset of V. To this end, we use an index set $\mathcal{I} \subset \mathbb{N}^d$ with $\#\mathcal{I} < \infty$ whose elements $\mathbf{l} \in \mathcal{I}$ determine which anisotropic full grid spaces $V_{\mathbf{l}}$ need to be included in the discretization space

$$V_{\mathcal{I}} := \sum_{\mathbf{l} \in \mathcal{I}} V_{\mathbf{l}} \,. \tag{4.7}$$

Since $V_{\mathbf{k}} \subset V_{\mathbf{l}}$ whenever $\mathbf{k} \leq \mathbf{l}$ componentwise, we always silently assume that $\mathcal{I}$ satisfies the monotonicity condition

$$\mathbf{l} \in \mathcal{I}, \mathbf{k} \leq \mathbf{l} \Rightarrow \mathbf{k} \in \mathcal{I} \,. \tag{4.8}$$

This setup allows for dimension-adaptivity, whereas space-adaptivity is not possible since the subspaces $V_{\mathbf{l}}$ are either fully included or excluded by $\mathbf{l} \in \mathcal{I}$ or $\mathbf{l} \notin \mathcal{I}$, respectively.

Which kind of $\mathcal{I}$ we choose now depends on the target accuracy and on the smoothness of the function class we want to approximate. For example, the full grid space with index set

$$\mathcal{I} = \mathcal{F}_J^d := \{\mathbf{l} \in \mathbb{N}^d : |\mathbf{l}|_\infty \leq J\} \tag{4.9}$$

has the approximation property

$$\inf_{v \in V_{\mathcal{F}_J^d}} \|u - v\|^2_{H^s(\Omega^d)} \lesssim 2^{-2(r-s)J} \|u\|^2_{H^r(\Omega^d)}$$

with rate[1] $r - s$ and $u \in H_0^r(\Omega^d)$. Its number of degrees of freedom grows by $\mathcal{O}(2^{dJ})$. Thus, the accuracy as function of the degrees of freedom deteriorates exponentially with rising d, which resembles the well-known 'curse of dimensionality', cf. [Bel61].

Assuming additional mixed smoothness $u \in H^r_{0,\text{mix}}(\Omega^d)$, the sparse grid index set

$$\mathcal{I} = \mathcal{S}_J^d := \{\mathbf{l} \in \mathbb{N}^d : |\mathbf{l}|_1 \leq J\} \tag{4.10}$$

circumvents this problem to some extent, see Figure 4.3 for an illustration. The rate of the best approximation

$$\inf_{v \in V_{\mathcal{S}_J^d}} \|u - v\|^2_{H^s(\Omega^d)} \leq c 2^{-2(r-s)J} \|u\|^2_{H^r_{\text{mix}}(\Omega^d)}$$

in dependence of J is the same[2] as for the full grid space, i.e. $r - s$, but the number of degrees of freedom now only grows by $\mathcal{O}(2^J J^{d-1})$. This is a substantial improvement in comparison to

[1] This holds for a range of parameters $0 \leq s < r \leq p$ with p being the order of the spline of the space construction. In our case of linear splines $p = 2$ holds.

[2] For $s < 0$ the order of approximation deteriorates slightly.

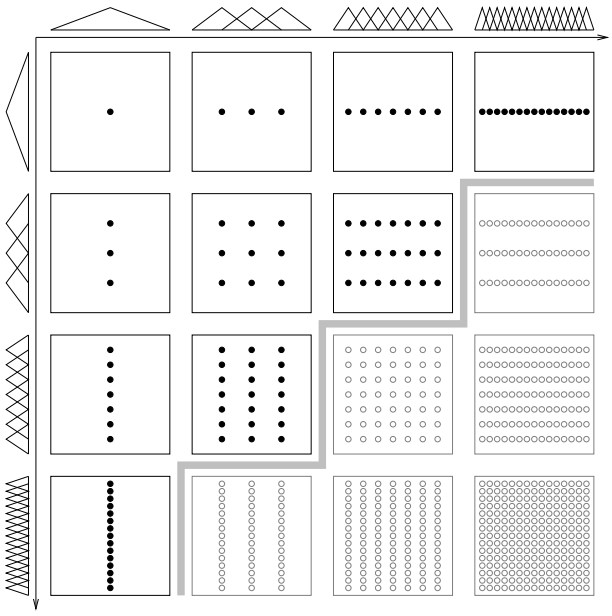

Figure 4.3: Two-dimensional tensor product of the ladder of dyadically refined linear spline spaces up to level $J = 3$. The subspaces of the regular sparse grid discretization are depicted in solid black

the full grid case. Note that without the stronger assumption of mix-regularity, our sparse grid approximation rate in terms of J would deteriorate to $\frac{r-s}{d}$. For further details, see [GK09].

Anisotropic sparse grids can be defined by a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ and the corresponding index set

$$\mathcal{I}_{\boldsymbol{\alpha}} := \{\mathbf{l} \in \mathbb{N}^d : \sum_{p=1}^{d} \alpha_p l_p \leq J\} \, . \tag{4.11}$$

Note that the regular sparse grid (4.10) is a special case of (4.11) with $\alpha_1 = \cdots = \alpha_d = 1$. Anisotropic sparse grids are useful to compensate for varying smoothness classes and costs in different dimensions, see [GH13a, GH13b].

Depending on the error norm and the regularity of the function to approximate, the construction of $\mathcal{I}$ and $V_{\mathcal{I}}$ can be optimized even further, cf. [BG99, GK09]. Moreover, it is possible to adapt the index set $\mathcal{I}$ a-posteriori to a given function by means of a proper error estimation and successive refinement procedure. This approach results in adaptively refined sparse grids, see e.g. [Feu10, GG03]. The results in this thesis are applicable to all constructions of $\mathcal{I}$ as long as (4.8) is satisfied.

4.2 Generating system discretization of variational problems

In what follows we assume H is a Hilbert space and $a(u, v)$ an H-elliptic symmetric positive definite bilinear form that is well-defined on every $V_{\mathbf{l}}$ with $\mathbf{l} \in \mathbb{N}^d$. Furthermore, it holds that H is the $a(\cdot, \cdot)$-closure of $\operatorname{span}\{V_{\mathbf{l}} : \mathbf{l} \in \mathbb{N}^d\}$. We now discuss the discretization of the variational

problems as in (3.16): Find $u \in H$ with

$$a(u, v) = F(v) \qquad \forall\, v \in H\ , \tag{4.12}$$

where F is a bounded linear functional on H.

For discretization level J, we set $\mathbf{J} := (J, \ldots, J)$ and then the weak problem (4.12) discretized on the isotropic full grid space $V_{\mathbf{J}}$ leads to the system

$$\mathbf{A_J x_J} = \mathbf{b_J} \tag{4.13}$$

of $n_{\mathbf{J}} = (n_J)^d$ linear equations with

$$\mathbf{A_J} \in \mathbb{R}^{n_{\mathbf{J}} \times n_{\mathbf{J}}}, (\mathbf{A_J})_{\mathbf{i},\mathbf{j}} = a(\phi_{\mathbf{J},\mathbf{j}}, \phi_{\mathbf{J},\mathbf{i}})$$

and

$$\mathbf{x_J}, \mathbf{b_J} \in \mathbb{R}^{n_{\mathbf{J}}}\ , (\mathbf{b_J})_{\mathbf{i}} = F(\phi_{\mathbf{J},\mathbf{i}})$$

for $\mathbf{i}, \mathbf{j} \in \chi_{\mathbf{J}}$, see (4.6).

Usually, the system (4.13) is solved using an iterative method. If the bilinear form $a(\cdot, \cdot)$ is the weak form of a local operator, i.e.,

$$\operatorname{supp} \phi_{\mathbf{J},\mathbf{i}} \cap \operatorname{supp} \phi_{\mathbf{J},\mathbf{j}} = \emptyset \Rightarrow a(\phi_{\mathbf{J},\mathbf{i}}, \phi_{\mathbf{J},\mathbf{j}}) = 0\ , \tag{4.14}$$

the system matrix of (4.13) is inherently sparse. To be more specific, for our tensor product discretization of (4.12) on $V_{\mathbf{J}}$, there are typically no more than 3^d non-zero entries in the rows of $\mathbf{A_J}$. This means that we can compute the matrix-vector product $\mathbf{A_J x_J}$ in $\mathcal{O}(n_{\mathbf{J}})$ floating point operations, however, for bilinear forms as in (3.27) this is not easily possible and we need complicated algorithms to exploit the structure of $\mathbf{A_J}$.

Furthermore, the condition number of the system matrix $\mathbf{A_J}$ is an issue. For the Laplacian in weak form $a(u, v) = (\nabla u, \nabla v)_{L_2(\Omega^d)}$, the matrix possesses a condition number that is of the order $\mathcal{O}(2^{2J})$. Thus, classical iterative solution methods for (4.13) like the Jacobi method, the steepest descent approach or the conjugate gradient technique converge slower for rising values of J. The same is true for the Gauss-Seidel and the SOR methods.

One remedy is to resort to multigrid methods or to include coarser scales in the discretization, as we do in the next subsection. The inclusion of multiple coarser scales is also the key to discretize (4.12) on the generalized sparse grid space $V_{\mathcal{I}}$ from (4.7), which we discuss in Subsection 4.2.2.

4.2.1 Generating system approach

For certain problems, we can deal with the deteriorating condition number of $\mathbf{A_J}$ with $\mathbf{J} = (J, \ldots, J)$ by a multigrid method or a multilevel preconditioner. Then, the number of iterations necessary to obtain a prescribed accuracy is bounded independently of J, cf. [Hac85, Xu92, Bra07b, BL11]. To this end, besides the grid and the basis functions on the finest scale J, also the grids and basis functions on all coarser isotropic scales are included in the iterative process,

i.e. the multiscale generating system

$$\bigcup_{l=0}^{J} \{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{l} = (l,\dots,l) \quad \text{and} \quad \mathbf{i} \in \chi_{\mathbf{l}}\}$$

is employed. Note that there is work that relates classical multigrid theory to multiplicative iterative algorithms operating on such a generating system [Gri94b, Gri94a]. Furthermore, the BPX-preconditioner [BPX90] can be identified with one step of the additive Jacobi iteration. For the discretized Laplacian, both methods guarantee asymptotically optimal convergence rates that are independent of J.

We follow a different route here that allows us to achieve robustness with respect to the coefficients of a certain class of PDEs and even a favorable d-dependence of the condition number, see Chapter 5. To this end, we include *all* coarser isotropic and anisotropic scales $\mathbf{l} \in \mathcal{F}_J^d$ from (4.9), i.e., the functions

$$\bigcup_{\mathbf{l}\in\mathcal{F}_J^d}^{J} \{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \chi_{\mathbf{l}}\} \,. \tag{4.15}$$

Then, the weak problem (4.12) discretized with (4.15) leads to the enlarged system

$$\mathbf{A}_{\mathcal{F}_J^d}\mathbf{x}_{\mathcal{F}_J^d} = \mathbf{b}_{\mathcal{F}_J^d} \tag{4.16}$$

of linear equations, with matrix $\mathbf{A}_{\mathcal{F}_J^d}$ of size $N_{\mathcal{F}_J^d} \times N_{\mathcal{F}_J^d}$ and vectors $\mathbf{x}_{\mathcal{F}_J^d}, \mathbf{b}_{\mathcal{F}_J^d}$ of size $N_{\mathcal{F}_J^d}$, where

$$N_{\mathcal{F}_J^d} := \sum_{\mathbf{l}\in\mathcal{F}_J^d} n_{\mathbf{l}} \,.$$

Due to the dyadic refinement and a geometric series argument, we get that

$$\begin{aligned}\sum_{\mathbf{l}\in\mathcal{F}_J^d} n_{\mathbf{l}} = \sum_{\mathbf{l}\in\mathcal{F}_J^d}\prod_{p=1}^{d} n_{l_p} = \prod_{p=1}^{d}\sum_{l_p=0}^{J} n_{l_p} = \prod_{p=1}^{d}\sum_{l_p=0}^{J}(2^{l_p+1}-1) = \prod_{p=1}^{d}\left(2^{J+2}-2-J-1\right) \\ < \prod_{p=1}^{d} 2n_J = 2^d n_{\mathbf{J}} \,.\end{aligned} \tag{4.17}$$

That means we have, apart from a dimension-dependent constant, essentially the same amount of degrees of freedom as a non-redundant representation of $V_{\mathbf{J}}$. The matrix $\mathbf{A}_{\mathcal{F}_J^d}$ is block-structured with blocks $(\mathbf{A}_{\mathcal{F}_J^d})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_{\mathbf{l}}\times n_{\mathbf{k}}}$ for $\mathbf{l},\mathbf{k} \in \mathcal{F}_J^d$,where

$$((\mathbf{A}_{\mathcal{F}_J^d})_{\mathbf{l},\mathbf{k}})_{\mathbf{i},\mathbf{j}} = a(\phi_{\mathbf{k},\mathbf{j}}, \phi_{\mathbf{l},\mathbf{i}}) \quad \text{for} \quad \mathbf{i} \in \chi_{\mathbf{l}}, \mathbf{j} \in \chi_{\mathbf{k}}$$

and the right-hand side vector $\mathbf{b}_{\mathcal{F}_J^d}$ consists of blocks $(\mathbf{b}_{\mathcal{F}_J^d})_{\mathbf{l}} \in \mathbb{R}^{n_{\mathbf{l}}}$, $\mathbf{l} \in \mathcal{F}_J^d$, with

$$((\mathbf{b}_{\mathcal{F}_J^d})_{\mathbf{l}})_{\mathbf{i}} = F(\phi_{\mathbf{l},\mathbf{i}}) \quad \text{for} \quad \mathbf{i} \in \chi_{\mathbf{l}} \,.$$

Since $V_{\mathbf{k}} \subset V_{\mathbf{l}}$ whenever $\mathbf{k} \leq \mathbf{l}$, the union (4.15) is a generating system and not a basis. Note that the non-unique representation of functions results in a non-trivial kernel of the system matrix $\mathbf{A}_{\mathcal{F}_J^d}$, which is thus not invertible. But the system (4.16) is nevertheless solvable since the right-hand side $\mathbf{b}_{\mathcal{F}_J^d}$ lies in the range of the system matrix. A solution can be generated by any semi-convergent iterative method [Kaa88, BP94]. Many convergence results, e.g., for the steepest descent or conjugate gradient method, also apply to the semi-definite case, cf. [Gri94b]. There, the usual condition number κ is no longer defined, but the *generalized* condition number $\tilde{\kappa}$, i.e. the ratio of the largest and the smallest *non-zero* eigenvalue, is now decisive for the speed of convergence.

Of course, at some point, we need to be able to transform the non-unique solution $\mathbf{x}_{\mathcal{F}_J^d}$ of (4.16) to the unique solution $\mathbf{x}_{\mathbf{J}}$ of (4.13). To this end, we assume to have matrices $\mathbf{I}_{l,k} \in \mathbb{R}^{n_l \times n_k}$, which represent one-dimensional restrictions from level k to level l for $k > l$, prolongations from level k to level l for $k < l$ and the identity matrix for $l = k$. Note here that for $k \neq l \pm 1$ the $\mathbf{I}_{k,l}$ can be expressed as just a product of successive 2-level restrictions and prolongations, respectively, i.e. we have

$$\mathbf{I}_{l,k} = \mathbf{I}_{l,l+1} \cdots \mathbf{I}_{k-1,k} \quad \text{for} \quad k > l \quad \text{and} \quad \mathbf{I}_{l,k} = \mathbf{I}_{l,l-1} \cdots \mathbf{I}_{k+1,k} \quad \text{for} \quad k < l \; . \tag{4.18}$$

Naturally, the multi-variate case is obtained by the product construction

$$\mathbf{I}_{\mathbf{l},\mathbf{k}} = \bigotimes_{p=1}^{d} \mathbf{I}_{l_p,k_p} \; . \tag{4.19}$$

Then, for $\mathbf{l}, \mathbf{k} \in \mathcal{F}_J^d$, we can express any block $(\mathbf{A}_{\mathcal{F}_J^d})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_{\mathbf{l}} \times n_{\mathbf{k}}}$ and any part $(\mathbf{b}_{\mathcal{F}_J^d})_{\mathbf{l}}$ as

$$(\mathbf{A}_{\mathcal{F}_J^d})_{\mathbf{l},\mathbf{k}} = \mathbf{I}_{\mathbf{l},\mathbf{J}} \mathbf{A}_{\mathbf{J}} \mathbf{I}_{\mathbf{J},\mathbf{k}} \quad \text{and} \quad (\mathbf{b}_{\mathcal{F}_J^d})_{\mathbf{l}} = \mathbf{I}_{\mathbf{l},\mathbf{J}} \mathbf{b}_{\mathbf{J}} \; , \tag{4.20}$$

respectively, where $\mathbf{J}$ again describes the isotropic level on the finest scale.

Furthermore, let us define the rectangular block-structured matrix $\mathbf{S}_{\mathcal{F}_J^1} \in \mathbb{R}^{n_J \times (\sum_{l=0}^{J} n_l)}$ by

$$\mathbf{S}_{\mathcal{F}_J^1} := (\, \mathbf{I}_{J,1} \mid \ldots \mid \mathbf{I}_{J,J} \,) \; .$$

Then, we can express the block-structured matrix $\mathbf{S}_{\mathcal{F}_J^d}$ as

$$\mathbf{S}_{\mathcal{F}_J^d} = \bigotimes_{p=1}^{d} \mathbf{S}_{\mathcal{F}_J^1} \; ,$$

and with (4.20) we obtain

$$\mathbf{A}_{\mathcal{F}_J^d} = \mathbf{S}_{\mathcal{F}_J^d}^T \mathbf{A}_{\mathbf{J}} \mathbf{S}_{\mathcal{F}_J^d} \quad \text{and} \quad \mathbf{b}_{\mathcal{F}_J^d} = \mathbf{S}_{\mathcal{F}_J^d}^T \mathbf{b}_{\mathbf{J}} \; .$$

As a result, we see that $\mathbf{x}_{\mathbf{J}} = \mathbf{S}_{\mathcal{F}_J^d} \mathbf{x}_{\mathcal{F}_J^d}$ solves (4.13), if $\mathbf{x}_{\mathcal{F}_J^d}$ is any solution to (4.16). Note that we will never set up the matrices $\mathbf{S}_{\mathcal{F}_J^d}$ and $\mathbf{S}_{\mathcal{F}_J^d}^T$ in our implementation, but compute their application to vectors by a straightforward algorithm in $\mathcal{O}(d \cdot N_{\mathcal{F}_J^d})$ floating point operations using (4.18) and (4.19).

4.2.2 Generalized sparse grid

So far, we have described how to include all coarser scales of an isotropic discretization, which is useful for preconditioning, as we will see in Chapter 5. Furthermore, the multilevel structure is also a natural way to represent generalized sparse grids with index sets $\mathcal{I}$. Discretizing the weak problem (4.12) on the generalized sparse grid $V_\mathcal{I}$ with the generating system

$$\Phi_\mathcal{I} = \bigcup_{\mathbf{l}\in\mathcal{I}} \{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \chi_\mathbf{l}\} \tag{4.21}$$

now leads to the equation

$$\mathbf{A}_\mathcal{I}\mathbf{x}_\mathcal{I} = \mathbf{b}_\mathcal{I} \,. \tag{4.22}$$

Here, the matrix $\mathbf{A}_\mathcal{I}$ is block-structured with blocks $(\mathbf{A}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l}\times n_\mathbf{k}}$ for $\mathbf{l},\mathbf{k} \in \mathcal{I}$, where

$$((\mathbf{A}_\mathcal{I})_{\mathbf{l},\mathbf{k}})_{\mathbf{i},\mathbf{j}} = a(\phi_{\mathbf{k},\mathbf{j}}, \phi_{\mathbf{l},\mathbf{i}}) \quad \text{for} \quad \mathbf{i} \in \chi_\mathbf{l}, \mathbf{j} \in \chi_\mathbf{k}$$

and the right-hand side vector $\mathbf{b}_\mathcal{I}$ consists of blocks $(\mathbf{b}_\mathcal{I})_\mathbf{l} \in \mathbb{R}^{n_\mathbf{l}}$, $\mathbf{l} \in \mathcal{I}$, with

$$((\mathbf{b}_\mathcal{I})_\mathbf{l})_\mathbf{i} = F(\phi_{\mathbf{l},\mathbf{i}}) \quad \text{for} \quad \mathbf{i} \in \chi_\mathbf{l} \,.$$

Similar to the full grid generating system case (4.16), the non-unique representation in the generalized sparse grid generating system (4.21) results in a non-trivial kernel of $\mathbf{A}_\mathcal{I}$. Thus, $\mathbf{A}_\mathcal{I}$ is not invertible. But, again, the system (4.22) is solvable since the right-hand side $\mathbf{b}_\mathcal{I}$ lies in the range of the system matrix and a solution can be generated by any semi-convergent iterative method.

We now describe the enlarged sparse grid system (4.22) as a submatrix and a subvector of the enlarged full grid system (4.16). To this end, let use choose the minimal J, such that

$$\mathcal{I} \subset \mathcal{F}_J^d \,.$$

Like in (4.20), we can express the blocks of $\mathbf{A}_\mathcal{I}$ and $\mathbf{b}_\mathcal{I}$ with respect to (4.13) by

$$(\mathbf{A}_\mathcal{I})_{\mathbf{l},\mathbf{k}} = \mathbf{I}_{\mathbf{l},\mathbf{J}}\mathbf{A}_\mathbf{J}\mathbf{I}_{\mathbf{J},\mathbf{k}} \quad \text{and} \quad (\mathbf{b}_\mathcal{I})_\mathbf{l} = \mathbf{I}_{\mathbf{l},\mathbf{J}}\mathbf{b}_\mathbf{J}$$

for $\mathbf{l},\mathbf{k} \in \mathcal{I}$. Now, we can express our sparse grid operator matrix by

$$\mathbf{A}_\mathcal{I} = \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}\mathbf{A}_{\mathcal{F}_J^d}\mathbf{R}^T_{\mathcal{I},\mathcal{F}_J^d} = \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}\mathbf{S}^T_{\mathcal{F}_J^d}\mathbf{A}_\mathbf{J}\mathbf{S}_{\mathcal{F}_J^d}\mathbf{R}^T_{\mathcal{I},\mathcal{F}_J^d} \,, \tag{4.23}$$

and our right-hand side by

$$\mathbf{b}_\mathcal{I} = \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}\mathbf{b}_{\mathcal{F}_J^d} = \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}\mathbf{S}^T_{\mathcal{F}_J^d}\mathbf{b}_\mathbf{J} \,, \tag{4.24}$$

where $\mathbf{R}_{\mathcal{I},\mathcal{F}_J^d} \in \mathbb{R}^{N_\mathcal{I}\times N_{\mathcal{F}_J^d}}$ is a rectangular block-structured matrix with $N_\mathcal{I} := \sum_{\mathbf{l}\in\mathcal{I}} n_\mathbf{l}$ and

$$(\mathbf{R}_{\mathcal{I},\mathcal{F}_J^d})_{\mathbf{l},\mathbf{k}} = \begin{cases} \mathbf{I}_{\mathbf{l},\mathbf{l}} & \text{for} \quad \mathbf{k} = \mathbf{l} \,, \\ 0 & \text{else} \,, \end{cases} \tag{4.25}$$

for $\mathbf{l} \in \mathcal{I}, \mathbf{k} \in \mathcal{F}_J^d$. Note that $\mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}^T \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d} \in \mathbb{R}^{N_{\mathcal{F}_J^d} \times N_{\mathcal{F}_J^d}}$ is a diagonal scaling matrix in the enlarged full grid system which simply sets all vector blocks to zero that belong to $\mathbf{l} \in \mathcal{F}_J^d \setminus \mathcal{I}$, and $\mathbf{R}_{\mathcal{I},\mathcal{F}_J^d} \mathbf{R}_{\mathcal{I},\mathcal{F}_J^d}^T \in \mathbb{R}^{N_\mathcal{I} \times N_\mathcal{I}}$ is simply the identity matrix on $\mathbb{R}^{N_\mathcal{I}}$.

Equations (4.23) and (4.24) have been stated for theoretical purposes only, and we of course avoid the full grid systems $\mathbf{A}_{\mathcal{F}_J^d}$ and $\mathbf{A}_\mathbf{J}$ in our implementation. A proof that the number of degrees of freedom of our generating system has always the same asymptotics in J as the size of a non-redundant system is given in Section 5.1. In the regular sparse grid case $\mathcal{I} = \mathcal{S}_J^d$, see (4.10), and we thus get $N_{\mathcal{S}_J^d} = \mathcal{O}(2^J J^{d-1})$ in J, see [BG04].

4.3 Operator application

If the bilinear form $a(\cdot,\cdot)$ is local in the sense of (4.14), the isotropic full grid operator matrix $\mathbf{A}_\mathbf{J}$ from (4.13) is inherently sparse and can be applied to vectors with a computational complexity that is linear in the number of degrees of freedom $n_\mathbf{J} = n_J^d$. Due to considerable overlap between the different subspaces of the generating system discretization, the matrix $\mathbf{A}_\mathcal{I}$ is far less sparse and exhibits the so called finger structure. An additional challenge is that the index set $\mathcal{I}$ in general does not have a tensor product structure like

$$\mathcal{F}_J^d = \{0,\dots,J\} \times \cdots \times \{0,\dots,J\} \subset \mathbb{N}^d$$

but can take any complicated shape. However, assuming that the block matrices

$$\mathbf{A}_{\mathbf{l},\mathbf{k}} := (\mathbf{A}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l} \times n_\mathbf{k}} \quad \text{for} \quad \mathbf{l},\mathbf{k} \in \mathcal{I}$$

admit a representation as a sum of tensor products, we can still efficiently execute the matrix-vector multiplication with the help of sophisticated algorithms. In this section, we present two different algorithms that also work for non-local operators.

We assume a PIDE operator of second order with constant coefficients

$$(Au)(\mathbf{x}) = -\sum_{i,j=1}^d a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j}(\mathbf{x}) + \sum_{i=1}^d b_i \frac{\partial u}{\partial x_i}(\mathbf{x}) + cu(\mathbf{x}) + \int_{\mathbb{R}^d} u(\mathbf{x}+\mathbf{y})\nu(\mathrm{d}\mathbf{y}) \,, \tag{4.26}$$

where the density of the absolutely continuous measure ν admits a representation as sum of products of univariate densities $K^{(r,p)} : \mathbb{R} \to \mathbb{R}_{\geq 0}, r = 1,\dots,R_\nu, p = 1,\dots,d$ with

$$\nu(\mathrm{d}\mathbf{y}) = \sum_{r=1}^{R_\nu} \prod_{p=1}^d K^{(r,p)}(y_p)\, \mathrm{d}\mathbf{y} \,. \tag{4.27}$$

Note that we might have to extent the function u to the whole space $\mathbb{R}^d$, e.g., by $u_{|\mathbb{R}^d \setminus \Omega^d} = 0$, in order to evaluate the integral in (4.26). The weak form $a(u,v) = (Au,v)_{L_2(\Omega^d)}$ of (4.26) is

given by

$$a(u,v) = \sum_{i,j=1}^{d} a_{ij}\Big(\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_j}\Big)_{L_2(\Omega^d)} + \sum_{i=1}^{d} b_i\Big(\frac{\partial u}{\partial x_i}, v\Big)_{L_2(\Omega^d)} + c(u,v)_{L_2(\Omega^d)} \tag{4.28}$$
$$+ \sum_{r=1}^{R_\nu} \int_{\Omega^d}\int_{\mathbb{R}^d} u(\mathbf{x}+\mathbf{y})\Big(\prod_{p=1}^{d} K^{(r,p)}(y_p)\Big)\mathrm{d}\mathbf{y}\cdot v(\mathbf{x})\mathrm{d}\mathbf{x}\,.$$

Let $\mathbf{A}_\mathcal{I}$ be the system matrix of (4.28) discretized on $V_\mathcal{I}$ with a generating system. Then, we can express its blocks for $\mathbf{l},\mathbf{k}\in\mathcal{I}$ in the form

$$\mathbf{A}_{\mathbf{l},\mathbf{k}} = \sum_{\substack{i,j=1\\ i\neq j}}^{d} a_{ij}\mathbf{H}^{(\triangleright,i)}_{l_i,k_i}\otimes\mathbf{H}^{(\triangleleft,j)}_{l_j,k_j}\otimes\bigotimes_{\substack{p=1\\ p\neq i,j}}^{d}\mathbf{M}^{(p)}_{l_p,k_p} + \sum_{i=1}^{d} a_{ii}\mathbf{H}^{(\bowtie,i)}_{l_i,k_i}\otimes\bigotimes_{\substack{p=1\\ p\neq i}}^{d}\mathbf{M}^{(p)}_{l_p,k_p} \tag{4.29}$$
$$+ \sum_{i=1}^{d} b_i\mathbf{H}^{(\triangleright,i)}_{l_i,k_i}\otimes\bigotimes_{\substack{p=1\\ p\neq i}}^{d}\mathbf{M}^{(p)}_{l_p,k_p} + c\bigotimes_{p=1}^{d}\mathbf{M}^{(p)}_{l_p,k_p} + \sum_{r=1}^{R_\nu}\bigotimes_{p=1}^{d}\mathbf{K}^{(r,p)}_{l_p,k_p} \tag{4.30}$$

with the matrices

$$(\mathbf{H}^{(\bowtie,p)}_{l,k})_{ij} = \Big(\frac{\partial\phi_{k,j}}{\partial x}, \frac{\partial\phi_{l,i}}{\partial x}\Big)_{L_2(\Omega)}, (\mathbf{H}^{(\triangleright,p)}_{l,k})_{ij} = \Big(\frac{\partial\phi_{k,j}}{\partial x}, \phi_{l,i}\Big)_{L_2(\Omega)},$$
$$(\mathbf{H}^{(\triangleleft,p)}_{l,k})_{ij} = \Big(\phi_{k,j}, \frac{\partial\phi_{l,i}}{\partial x}\Big)_{L_2(\Omega)}, (\mathbf{M}^{(p)}_{l,k})_{ij} = (\phi_{k,j},\phi_{l,i})_{L_2(\Omega)} \quad \text{and}$$
$$(\mathbf{K}^{(r,p)}_{l,k})_{ij} = \Big(\int_{\mathbb{R}}\phi_{k,j}(\,\cdot+y)K^{(r,p)}(y)\mathrm{d}y, \phi_{l,i}\Big)_{L_2(\Omega)}, \quad r=1,\dots,R_\nu, \tag{4.31}$$

for $1\le i\le n_l, 1\le j\le n_k$, where the index p in the superscripts denotes the dimension $p=1,\dots,d$ the respective matrix is applied to. This example demonstrates that a PIDE operator as in (4.26) produces blocks $A_{\mathbf{l},\mathbf{k}}, \mathbf{l},\mathbf{k}\in\mathcal{I}$, that can be written as a sum of tensor product matrices. Sometimes not all requirements are met: In the case of non-constant coefficient functions, it is possible to approximate the coefficient functions by another sparse grid [Ach03]. If the integration measure ν does not admit the sum of tensor products structure (4.27), it is still possible to exploit compressibility of the operator matrix, cf. [KK02, Rei10], but this is a topic outside the scope of this thesis.

In the following, we present two efficient matrix-vector multiplication methods for the system matrix $\mathbf{A}_\mathcal{I}$ whose blocks can be written as a sum of tensor products. For notational simplicity, we move from our example (4.29)–(4.30) to an abstract setting, in which we write

$$\mathbf{A}_{\mathbf{l},\mathbf{k}} = \sum_{r=1}^{R_A}\mathbf{A}^{(r,1)}_{l_1,k_1}\otimes\cdots\otimes\mathbf{A}^{(r,d)}_{l_d,k_d}\,, \tag{4.32}$$

where the matrices $\mathbf{A}_{l,k}^{(r,p)} \in \mathbb{R}^{n_l \times n_k}$ satisfy

$$\mathbf{A}_{l,k}^{(r,p)} = \begin{cases} \mathbf{A}_{l,l}^{(r,p)} \mathbf{I}_{l,k} & \text{for} \quad l \geq k \,, \\ \mathbf{I}_{k,l} \mathbf{A}_{k,k}^{(r,p)} & \text{for} \quad l < k \end{cases} \tag{4.33}$$

for all $r = 1, \ldots, R_A$ and $p = 1, \ldots, d$.

The first method we present is asymptotically slower but easy to implement and parallelize, the second one is linear in the degrees of freedom but involves a large constant in the costs and is quite sophisticated. In both cases, we need to apply the non-hierarchical matrices $\mathbf{A}_{l,l}^{(r,p)}$, which can be done in $\mathcal{O}(n_l)$ for local operators. The application of non-local operators needs $\mathcal{O}(n_l^2)$ operations if treated naively, $\mathcal{O}(n_l \log n_l)$ using the circular convolution theorem or only $\mathcal{O}(n_l)$ in special cases, e.g., if some recurrence formula is employed, see Section 6.2. For the following algorithms, we assume that an application in linear runtime is possible.

4.3.1 Single space matrix-vector multiplication algorithm

We start with a simple algorithm that has its advantages with respect to parallelization, but is asymptotically slower than the one we present in the next subsection. We assume that $r = R_A = 1$ in (4.32) and drop the index r. Operators with $R_A > 1$ can be treated by repeatedly applying the following algorithm and summing over the results.

Given the block-structured vector $\mathbf{x}_\mathcal{I} = (\mathbf{x}_\mathbf{l})_{\mathbf{l} \in \mathcal{I}} \in \mathbb{R}^{N_\mathcal{I}}$, we want to compute $\mathbf{y}_\mathcal{I} = \mathbf{A}_\mathcal{I} \mathbf{x}_\mathcal{I} \in \mathbb{R}^{N_\mathcal{I}}$. This translates into

$$\mathbf{y}_\mathbf{l} = \sum_{\mathbf{k} \in \mathcal{I}} \mathbf{w}_{\mathbf{l},\mathbf{k}} \quad \text{with} \quad \mathbf{w}_{\mathbf{l},\mathbf{k}} := \mathbf{A}_{\mathbf{l},\mathbf{k}} \mathbf{x}_\mathbf{k}, \mathbf{k} \in \mathcal{I}, \tag{4.34}$$

for all $\mathbf{l} \in \mathcal{I}$. We compute (4.34) by summing over $\mathbf{k} \in \mathcal{I}$ in an outer loop and computing the respective $\mathbf{w}_{\mathbf{l},\mathbf{k}} = \mathbf{A}_{\mathbf{l},\mathbf{k}} \mathbf{x}_\mathbf{k}$ for all $\mathbf{l} \in \mathcal{I}$ by the single space matrix-vector multiplication. The tensor product structure of the blocks $A_{\mathbf{l},\mathbf{k}}$ is relevant, as it helps to avoid computations in the much larger space $V_{\max(\mathbf{l},\mathbf{k})}$ with $n_{\max(\mathbf{l},\mathbf{k})}$ degrees of freedom, where $(\max(\mathbf{l},\mathbf{k}))_i = \max(l_i, k_i), i = 1, \ldots, d$. To see this, we have to realize that (4.32) in combination with (4.33) results in

$$\begin{aligned} \mathbf{A}_{\mathbf{l},\mathbf{k}} =& \mathbf{A}_{l_1,k_1}^{(1)} \otimes \cdots \otimes \mathbf{A}_{l_d,k_d}^{(d)} \\ =& \bigotimes_{p:l_p<k_p} \mathbf{I}_{k_p,l_p}^{(p)} \mathbf{A}_{k_p,k_p}^{(p)} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{A}_{l_q,l_q}^{(q)} \mathbf{I}_{l_q,k_q}^{(q)} \\ =& \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}_{l_p,l_p}^{(p)} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{A}_{l_q,l_q}^{(q)} \mathbf{I}_{l_q,k_q}^{(q)} \Big) \cdot \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}_{l_p,k_p}^{(p)} \mathbf{A}_{k_p,k_p}^{(p)} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{I}_{k_q,k_q}^{(q)} \Big) \,. \end{aligned} \tag{4.35}$$

So, when computing $\mathbf{w}_{\mathbf{l},\mathbf{k}} = \mathbf{A}_{\mathbf{l},\mathbf{k}} \mathbf{x}_\mathbf{k}, \mathbf{l} \in \mathcal{I}$, we can apply (4.35) in two steps. Note that

$$\Big(\bigotimes_{p:l_p<k_p} \mathbf{I}_{l_p,k_p}^{(p)} \mathbf{A}_{k_p,k_p}^{(p)} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{I}_{k_q,k_q}^{(q)} \Big) \mathbf{x}_\mathbf{k} \in \mathbb{R}^{n_{\min(\mathbf{l},\mathbf{k})}} \,,$$

thus the product structure helps to take the "shortcut" via the space indexed $V_{\min(\mathbf{l},\mathbf{k})}$, where

$\min(\mathbf{l},\mathbf{k})$ is defined analogously to $\max(\mathbf{l},\mathbf{k})$. This idea has been introduced in a matrix-vector multiplication algorithm in the context of PDEs with stochastic input data [ST03], see also [Zei11], but the following modified algorithm produces a smaller log-factor in the runtime complexity.

Now, we carry this principle further and compute all $\mathbf{w}_{\mathbf{l},\mathbf{k}} = \mathbf{A}_{\mathbf{l},\mathbf{k}}\mathbf{x}_{\mathbf{k}}, \mathbf{l} \in \mathcal{I}$, simultaneously in an efficient way. To this end, we use the slightly modified representation

$$\mathbf{A}_{\mathbf{l},\mathbf{k}} = \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,l_p} \otimes \bigotimes_{q:l_q\geq k_q} \mathbf{A}^{(q)}_{l_q,l_q} \Big) \cdot \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,k_p} \mathbf{A}^{(p)}_{k_p,k_p} \otimes \bigotimes_{q:l_q\geq k_q} \mathbf{I}^{(q)}_{l_q,k_q} \Big) . \tag{4.36}$$

The single space matrix-vector multiplication is done in two steps based on the representation (4.36). Algorithm 1 describes the propagation phase, which is similar to a depth-first search, and computes

$$\mathbf{z}_{\mathbf{l},\mathbf{k}} = \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,k_p} \mathbf{A}^{(p)}_{k_p,k_p} \otimes \bigotimes_{q:l_q\geq k_q} \mathbf{I}^{(q)}_{l_q,k_q} \Big) \mathbf{x}_{\mathbf{k}}$$

for all $\mathbf{l} \in \mathcal{I}$. Algorithm 2 describes the application phase, where the matrix-vector multiplication is carried out for all dimensions which have so far only been prolongated. In fact

$$\mathbf{w}_{\mathbf{l},\mathbf{k}} = \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,l_p} \otimes \bigotimes_{q:l_q\geq k_q} \mathbf{A}^{(q)}_{l_q,l_q} \Big) \mathbf{z}_{\mathbf{l},\mathbf{k}}$$

for all $\mathbf{l} \in \mathcal{I}$. Then the summation $\mathbf{y}_{\mathbf{l}} = \sum_{\mathbf{k}\in\mathcal{I}} \mathbf{w}_{\mathbf{l},\mathbf{k}}, \mathbf{l} \in \mathcal{I}$ yields our result vector $\mathbf{y}_{\mathcal{I}}$.

As both Algorithms 1 and 2 have a runtime complexity of $\mathcal{O}(N_{\mathcal{I}})$, we can compute all $\mathbf{w}_{\mathbf{l},\mathbf{k}} = \mathbf{A}_{\mathbf{l},\mathbf{k}}\mathbf{x}_{\mathbf{k}}, \mathbf{l} \in \mathcal{I}$, in linear time with respect to the number of degrees of freedom. Summing over all $\mathbf{k} \in \mathcal{I}$ computes (4.34) for all $\mathbf{l} \in \mathcal{I}$ in $\mathcal{O}(\#\mathcal{I} \cdot N_{\mathcal{I}})$ floating point operations, which is typically log-linear in $N_{\mathcal{I}}$. There is no exponential dependence on the dimension and the algorithm is easily parallelizable with respect to $\mathbf{k}$ with up to $\#\mathcal{I}$ processes. This algorithm is especially useful for setting up the auxiliary system of linear equations needed for the OptiCom, see Chapter 5.

In the following subsection, we present an algorithm with a computational complexity that is strictly linear in the number of degrees of freedom, but it cannot be easily parallelized and its computational complexity exhibits an exponential dependence on the dimension in most cases.

4.3.2 Unidirectional principle for non-local operators

The unidirectional principle for differential operators [BZ96, Bun92b, Zei11] is a sophisticated algorithm that realizes the matrix-vector multiplication for generalized sparse grid problems in a computational complexity that is strictly linear in the number of degrees of freedom. Now, we present a generalization from [GH13c], where it is no longer necessary to specifically tailor the TopDown/BottomUp algorithms to the operator in use. Moreover, it can be employed with non-local operators like integro-differential operators. A related abstraction of the unidirectional principle for multilevel discretizations has been presented in [Zei11].

With the same assumptions as in the previous subsection, we want to compute $\mathbf{y}_{\mathcal{I}} = \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}$,

Algorithm 1 Propagation phase of single space matrix-vector multiplication

Input: multi index $\mathbf{k}$, index set $\mathcal{I}$ and subspace vector $\mathbf{x}_\mathbf{k}$

Output: $\mathbf{z}_{\mathbf{l},\mathbf{k}} = \left(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,k_p} \mathbf{A}^{(p)}_{k_p,k_p} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{I}^{(q)}_{l_q,k_q} \right) \mathbf{x}_\mathbf{k}, \mathbf{l} \in \mathcal{I}$

Mark each $\mathbf{l}$ in $\mathcal{I}$ as "not visited"
$\mathbf{z}_{\mathbf{k},\mathbf{k}} \leftarrow \mathbf{x}_\mathbf{k}$ ▷ Start with subspace vector $\mathbf{k}$
Mark $\mathbf{k}$ as "visited"
for all $\mathbf{l} \in \mathcal{I}$ **do**
    if $\mathbf{l}$ is marked as "not visited" **then**
        computeSubspaceVec($\mathbf{l}$)
    end if
end for
return $\mathbf{z}_{\mathbf{l},\mathbf{k}}, \mathbf{l} \in \mathcal{I}$

function computeSubspaceVec($\mathbf{l}$)
    if $\mathbf{l}$ is marked as "visited" **then**
        return
    end if
    for $p = 1, \ldots, d$ **do** ▷ Since $\mathbf{l} \neq \mathbf{k}$, there is at least one p with $l_p \neq k_p$
        if $l_p > k_p$ **then**
            $\mathbf{l}' \leftarrow \mathbf{l} - \mathbf{e}_p$ ▷ Decrease p-th entry by 1
            computeSubspaceVec($\mathbf{l}'$) ▷ $\mathbf{z}_{\mathbf{l}',\mathbf{k}}$ is now set
            $\mathbf{z}_{\mathbf{l},\mathbf{k}} \leftarrow \left(\mathbf{I}^{(p)}_{l_p,l_p-1} \otimes \bigotimes_{q \neq p} \mathbf{I}^{(q)}_{l_q,l_q} \right) \mathbf{z}_{\mathbf{l}',\mathbf{k}}$ ▷ Prolongate from lower subspace
            Mark $\mathbf{l}$ as "visited"
            return
        else if $l_p = k_p - 1$ **then**
            $\mathbf{l}' \leftarrow \mathbf{l} + \mathbf{e}_p$ ▷ Increase p-th entry by 1
            computeSubspaceVec($\mathbf{l}'$) ▷ $\mathbf{z}_{\mathbf{l}',\mathbf{k}}$ is now set
            $\mathbf{z}_{\mathbf{l},\mathbf{k}} \leftarrow \left((\mathbf{I}^{(p)}_{l_p,k_p} \mathbf{A}^{(p)}_{k_p,k_p}) \otimes \bigotimes_{q \neq p} \mathbf{I}^{(q)}_{l_q,l_q} \right) \mathbf{z}_{\mathbf{l}',\mathbf{k}}$ ▷ Apply matrix $\mathbf{A}^{(p)}_{k_p,k_p}$ and restrict result
            Mark $\mathbf{l}$ as "visited"
            return
        else if $l_p < k_p - 1$ **then**
            $\mathbf{l}' \leftarrow \mathbf{l} + \mathbf{e}_p$ ▷ Increase p-th entry by 1
            computeSubspaceVec($\mathbf{l}'$) ▷ $\mathbf{z}_{\mathbf{l}',\mathbf{k}}$ is now set
            $\mathbf{z}_{\mathbf{l},\mathbf{k}} \leftarrow \left(\mathbf{I}^{(p)}_{l_p,l_p+1} \otimes \bigotimes_{q \neq p} \mathbf{I}^{(q)}_{l_q,l_q} \right) \mathbf{z}_{\mathbf{l}',\mathbf{k}}$ ▷ Restrict result from higher subspace
            Mark $\mathbf{l}$ as "visited"
            return
        end if
    end for
end function

Algorithm 2 Application phase of single space matrix-vector multiplication

Input: multi index $\mathbf{k}$, index set $\mathcal{I}$ and $\mathbf{z}_{\mathbf{l},\mathbf{k}} = \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,k_p} \mathbf{A}^{(p)}_{k_p,k_p} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{I}^{(q)}_{l_q,k_q} \Big) \mathbf{x}_{\mathbf{k}}, \mathbf{l} \in \mathcal{I}$

Output: $\mathbf{w}_{\mathbf{l},\mathbf{k}} = \Big(\bigotimes_{p=1}^{d} \mathbf{A}^{(p)}_{l_p,k_p} \Big) \mathbf{x}_{\mathbf{k}}, \mathbf{l} \in \mathcal{I}$

1: **for all** $\mathbf{l}$ in $\mathcal{I}$ **do**
2: $\quad \mathbf{w}_{\mathbf{l},\mathbf{k}} \leftarrow \Big(\bigotimes_{p:l_p<k_p} \mathbf{I}^{(p)}_{l_p,l_p} \otimes \bigotimes_{q:l_q \geq k_q} \mathbf{A}^{(q)}_{l_q,l_q} \Big) \mathbf{z}_{\mathbf{l},\mathbf{k}}$
3: **end for**

i.e.,

$$\mathbf{y}_{\mathbf{l}} = \sum_{\mathbf{k} \in \mathcal{I}} \big(\mathbf{A}^{(1)}_{l_1,k_1} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_p} \big) \mathbf{x}_{\mathbf{k}} \tag{4.37}$$

for all $\mathbf{l} \in \mathcal{I}$.

First, we start with the one-dimensional case of $\mathcal{I} = \mathcal{F}^1_J$ and split the sum (4.37) in two parts

$$\begin{aligned} \mathbf{y}_l &= \sum_{k=0}^{J} \mathbf{A}_{l,k} \mathbf{x}_k \\ &= \sum_{k=0}^{l} \mathbf{A}_{l,k} \mathbf{x}_k + \sum_{k=l+1}^{k} \mathbf{A}_{l,k} \mathbf{x}_k \ . \end{aligned} \tag{4.38}$$

Both sums in (4.38) need to be treated separately. As already indicated, the representation (4.18) of prolongations and restrictions is a very efficient tool to transport intermediate results. The computation of the first sum in (4.38) can be done using the TopDown algorithm.

TopDown algorithm

The TopDown algorithm uses a simple recursive relation to compute

$$\bar{\mathbf{y}}_l := \sum_{k=0}^{l} \mathbf{A}_{l,k} \mathbf{x}_k \tag{4.39}$$

for all $0 \leq l \leq J$. First, we define vectors

$$\mathbf{z}_l := \sum_{k=0}^{l} \mathbf{I}_{l,k} \mathbf{x}_k \ ,$$

for $l = 0, \ldots, J$ that satisfy the recursive relationship

$$\mathbf{z}_l = \sum_{k=0}^{l} \mathbf{I}_{l,k} \mathbf{x}_k = \sum_{k=0}^{l-1} \mathbf{I}_{l,l-1} \mathbf{I}_{l-1,k} \mathbf{x}_k + \mathbf{x}_l = \mathbf{I}_{l,l-1} \mathbf{z}_{l-1} + \mathbf{x}_l$$

for $l \geq 1$. Now, (4.39) can be expressed by

$$\bar{\mathbf{y}}_l = \sum_{k=0}^{l} \mathbf{A}_{l,l}\mathbf{I}_{l,k}\mathbf{x}_k = \mathbf{A}_{l,l}\sum_{k=0}^{l} \mathbf{I}_{l,k}\mathbf{x}_k = \mathbf{A}_{l,l}\mathbf{z}_l \ . \tag{4.40}$$

As long as the prolongations $\mathbf{I}_{l,l-1}$ and the application of the matrices $\mathbf{A}_{l,l}$ work in linear time, the TopDown algorithm is of optimal order. The whole procedure is described in pseudocode in Algorithm 3.

Algorithm 3 TopDown algorithm

Input: Matrices $\mathbf{A}_{l,l}$ and $\mathbf{x}_l$ for $l = 0, \ldots, J$
Output: $\bar{\mathbf{y}}_l = \sum_{k=0}^{l} \mathbf{A}_{l,k}\mathbf{x}_k, l = 0, \ldots, J$

```
z_0 ← x_0
ȳ_0 ← A_{0,0} z_0
for all l = 1, ..., J do
    z_l ← I_{l,l-1} z_{l-1} + x_l
    ȳ_l ← A_{l,l} z_l
end for
```

BottomUp algorithm

The BottomUp algorithm computes

$$\underline{\mathbf{y}}_l := \sum_{k=l+1}^{J} \mathbf{A}_{l,k}\mathbf{x}_k \tag{4.41}$$

for all $l = 0, \ldots, J-1$, and we define $\underline{\mathbf{y}}_J := \mathbf{0}$. The recursive relationship

$$\begin{aligned} \underline{\mathbf{y}}_l &= \sum_{k=l+2}^{J} \mathbf{A}_{l,k}\mathbf{x}_k + \mathbf{A}_{l,l+1}\mathbf{x}_{l+1} \\ &= \mathbf{I}_{l,l+1}\Big(\sum_{k=(l+1)+1}^{J} \mathbf{A}_{l+1,k}\mathbf{x}_k + \mathbf{A}_{l+1,l+1}\mathbf{x}_{l+1}\Big) \\ &= \mathbf{I}_{l,l+1}\Big(\underline{\mathbf{y}}_{l+1} + \mathbf{A}_{l+1,l+1}\mathbf{x}_{l+1}\Big) \ . \end{aligned} \tag{4.42}$$

holds for $l = J-1, \ldots, 0$. Clearly, all $\underline{\mathbf{y}}_l$ can be precalculated in linear time provided that the restrictions $\mathbf{I}_{l,l+1}$ and the application of the matrices $\mathbf{A}_{l,l}$ work in linear time. The pseudocode is given in Algorithm 4.

Algorithm 4 BottomUp algorithm

Input: Matrices $\mathbf{A}_{l,l}$ and $\mathbf{x}_l$ for $l = 0, \ldots, J$

Output: $\underline{\mathbf{y}}_l = \sum_{k=l+1}^{J} \mathbf{A}_{l,k}\mathbf{x}_k, l = 0, \ldots, J$

1: $\underline{\mathbf{y}}_J \leftarrow 0$
2: **for all** $l = J-1, \ldots, 0$ **do**
3: $\quad \underline{\mathbf{y}}_l \leftarrow I_{l,l+1}\left(\underline{\mathbf{y}}_{l+1} + A_{l+1,l+1}\mathbf{x}_{l+1}\right)$
4: **end for**

Multi-dimensional case

The multi-dimensional case can be reduced to the recursive application of the one-dimensional algorithms TopDown and BottomUp. To this end, we split and rearrange the sum

$$\mathbf{y_l} = \sum_{\mathbf{k}\in\mathcal{I}} \big(\mathbf{A}^{(1)}_{l_1,k_1} \otimes \mathbf{A}^{(2)}_{l_2,k_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_d}\big)\mathbf{x_k} \qquad \forall\, \mathbf{l} \in \mathcal{I}\ .$$

So, for all $\mathbf{l} \in \mathcal{I}$, we need to compute

$$\begin{aligned}
\mathbf{y_l} \;=\; & \sum_{\substack{k_1 \le l_1 \text{with} \\ (k_1,l_2,\ldots,l_d)\in\mathcal{I}}} (\mathbf{A}^{(1)}_{l_1,k_1} \otimes \mathbf{I}^{(2)}_{l_2,l_2} \otimes \cdots \otimes \mathbf{I}^{(d)}_{l_d,l_d}) \cdot && (4.43)\\
& \sum_{\substack{(k_2,\ldots,k_d) \text{ with} \\ (k_1,k_2,\ldots,k_d)\in\mathcal{I}}} \left(\mathbf{I}^{(1)}_{k_1,k_1} \otimes \mathbf{A}^{(2)}_{l_2,k_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_d}\right) \mathbf{x}_{(k_1,k_2,\ldots,k_d)} && (4.44)\\
+ & \sum_{\substack{(k_2,\ldots,k_d) \text{ with} \\ (l_1,k_2,\ldots,k_d)\in\mathcal{I}}} \left(\mathbf{I}^{(1)}_{l_1,l_1} \otimes \mathbf{A}^{(2)}_{l_2,k_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_d}\right) \cdot && (4.45)\\
& \sum_{\substack{k_1 > l_1 \text{ with} \\ (k_1,k_2,\ldots,k_d)\in\mathcal{I}}} (\mathbf{A}^{(1)}_{l_1,k_1} \otimes \mathbf{I}^{(2)}_{k_2,k_2} \otimes \cdots \otimes \mathbf{I}^{(d)}_{k_d,k_d})\mathbf{x}_{(k_1,k_2,\ldots,k_d)}\ . && (4.46)
\end{aligned}$$

Now, (4.43) resembles the application of the one-dimensional TopDown algorithm, and (4.46) resembles the application of the one-dimensional BottomUp algorithm. The sums (4.44) and (4.45) are the result of a recursive application of the multi-dimensional algorithm with the first dimension left unchanged. The monotonicity condition (4.8) ensures that intermediate results can be stored with the same complexity as $\mathbf{x}_\mathcal{I}$ and the final result $\mathbf{y}_\mathcal{I}$. Specifically, we know that

$$(l_1, \ldots, l_d) \in \mathcal{I}, k_1 \le l_1 \Rightarrow (k_1, l_2, \ldots, l_d) \in \mathcal{I}$$

in (4.43) and

$$(k_1, \ldots, k_d) \in \mathcal{I}, k_1 > l_1 \Rightarrow (l_1, k_2, \ldots, k_d) \in \mathcal{I}$$

in (4.45), which means that intermediate results can be represented as generalized sparse grid functions with the same index set $\mathcal{I}$. See Algorithm 5 for a description in pseudocode, where we use the shorthand notation

$$\mathbf{k}' \cup \{k_p\} = (k_1, \ldots, k_d)$$

for $\mathbf{k}' = (k_1, \ldots, k_{p-1}, k_{p+1}, \ldots, k_d) \in \mathbb{N}^{d-1}$.

Algorithm 5 Unidirectional principle

Input: Index set $\mathcal{I}$, Matrices $\mathbf{A}^{(p)}_{\mathbf{l},\mathbf{k}}, \mathbf{l}, \mathbf{k} \in \mathcal{I}$ and $p = 1, \ldots, d$, vector $\mathbf{x}_\mathcal{I}$

Output: $\mathbf{y}_\mathcal{I} = (\mathbf{y}_\mathbf{l})_{\mathbf{l}\in\mathcal{I}}$ with $\mathbf{y}_\mathbf{l} = \sum_{\mathbf{k}\in\mathcal{I}} \left(\mathbf{A}^{(1)}_{l_1,k_1} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_p}\right)\mathbf{x}_\mathbf{k}$ for all $\mathbf{l} \in \mathcal{I}$

$\mathbf{y}_\mathcal{I} \leftarrow$ UniDir$(0, \mathbf{x}_\mathcal{I})$
function UniDir(p, $\mathbf{x}_\mathcal{I}$) ▷ p is active dimension, $\mathbf{x}_\mathcal{I}$ the input vector
  if $p = d$ **then**
    return ▷ Nothing to do
  end if
  $\mathbf{z}_\mathcal{I} \leftarrow \mathbf{x}_\mathcal{I}$ ▷ Copy input vector
  if $p \neq 0$ **then**
    for all $\mathbf{k}' \in \mathbb{N}^{d-1}$ for which $\exists k_p : \mathbf{k}' \cup \{k_p\} \in \mathcal{I}$ **do**
      BottomUp$((\mathbf{x}_\mathbf{k})_{\mathbf{k}:=\mathbf{k}'\cup\{k_p\}\in\mathcal{I}}, (A^{(p)}_{k_p,k_p})_{k_p:\mathbf{k}'\cup\{k_p\}\in\mathcal{I}})$ ▷ Apply (4.46) to $\mathbf{x}_\mathcal{I}$ in place
    end for
  end if
  UniDir$(p+1, \mathbf{x}_\mathcal{I})$ ▷ Recursive call (4.45)
  UniDir$(p+1, \mathbf{z}_\mathcal{I})$ ▷ Recursive call (4.44)
  if $p \neq 0$ **then**
    for all $\mathbf{k}' \in \mathbb{N}^{d-1}$ for which $\exists k_p : \mathbf{k}' \cup \{k_p\} \in \mathcal{I}$ **do**
      TopDown$((\mathbf{z}_\mathbf{k})_{\mathbf{k}:=\mathbf{k}'\cup\{k_p\}\in\mathcal{I}}, (A^{(p)}_{k_p,k_p})_{k_p:\mathbf{k}'\cup\{k_p\}\in\mathcal{I}})$ ▷ Apply (4.43) to $\mathbf{z}_\mathcal{I}$ in place
    end for
  end if
  $\mathbf{x}_\mathcal{I} \leftarrow \mathbf{x}_\mathcal{I} + \mathbf{z}_\mathcal{I}$
end function

Note here that the cost complexity of the algorithm is only linear with respect to the degrees of freedom if the cost complexity of the univariate operator applications is linear. However, two recursive calls per dimension lead to 2^d calls of the function. This exponential dependence of the computational complexity on the dimension is undesirable, but can be avoided in special cases, see [Feu05]. Note that the parallelization of the unidirectional principle is far from trivial due to the involved recursive nature of the algorithm, see [HSB12].

5 Preconditioning of high-dimensional elliptic equations

In this chapter, we develop additive (sometimes coined "parallel") Schwarz preconditioners for generalized sparse grid discretizations of symmetric H-elliptic variational problems

$$a(u, v) = F(v) \quad \forall\, v \in H\,, \tag{5.1}$$

where H is the $\|\cdot\|_a := a(\cdot,\cdot)^{\frac{1}{2}}$-closure of $\operatorname{span}\{V_{\mathbf{l}} : \mathbf{l} \in \mathbb{N}^d\}$ and F is a bounded linear functional on H. The main focus of this chapter is on generic H^t- or $H^{t,l}_{\text{mix}}$-elliptic problems. Ultimately, we will apply our preconditioner to variational problems (3.16) that arise in the course of solving the BKE, even though they can be asymmetric. Our preconditioners still work well in these cases. Note that by our choice of subspaces in the additive Schwarz setting, we obtain methods that are in fact preconditioners for the system of linear equations (4.22), which is the generalized sparse grid version of (3.19).

For isotropic full grid discretizations, an optimal iteration count which is independent of the number of degrees of freedom is typically achieved by multiplicative multigrid methods [Yse93, BL11, Hac85, Gri94b], the additive BPX preconditioner [BPX90, Osw92, Osw94] or wavelet-based methods. In the sparse grid case, the condition number is more difficult to reduce than in the regular full grid case. For example, already for a straightforward regular sparse grid discretization, cf. [GO94], a simple diagonal scaling similar to the case of the BPX-preconditioner does *not* result in asymptotically bounded condition numbers in dimensions $d \geq 3$. Here, more complicated basis functions like prewavelets offer a solution [GO95b].

The main idea is to apply a subspace correction method to (5.1), where the subspaces solvers are based on auxiliary bilinear forms on the anisotropic full grid spaces that compose the sparse grid. Their relative scaling is at our disposal and amounts to a diagonal scaling of the operator matrix of the discretized system (4.22). We heavily rely on a norm equivalence based on orthogonal complement spaces. First we show that, for a certain class of second-order PDEs, the norm equivalence constants are independent of the space dimension and the diffusion coefficients. Then, based on the norm equivalence, we infer quasi-optimal scaling factors for our full grid spaces by a Linear Program (LP). This approach closely follows the lines of [GHO15]. We prove that $\mathcal{O}(J^{d-2})$ is a lower bound for the condition number for *any* positive diagonal scaling.

This motivates the use of partially negative scaling factors, and we present an algebraic transformation that results in optimal condition numbers. In fact, we even observe *falling* condition numbers with *rising* dimension for the Laplace operator discretized by a regular sparse grid. We also present a method closely related to a prewavelet approach which results in exactly the same condition numbers as the algebraic transformation, but only needs positive scalings and produces symmetrizable matrices, see also [GH14b].

If a norm equivalence is not available, we can still employ a non-linear variable preconditioner [JN99] that has been referred to as OptiCom in the case of sparse grids in data mining [Heg03, Gar06, HGC07]. We describe an efficient implementation based on the single space matrix-vector multiplication presented in Chapter 4. This reduces the typically quadratic costs of the OptiCom to log-linear with respect to the degrees of freedom if the associated bilinear form $a(\cdot,\cdot)$ is given as a sum of tensor products. All preconditioners allow a CG version, and with the exception of the variable preconditioner, possess a cost complexity that is only linear with respect to the degrees of freedom.

This chapter is organized as follows: In Section 5.1, we describe a well-known equivalence of norms for H^t-elliptic problems and prove that its norm equivalence constants are independent of the diffusion coefficients and the space dimension. In Section 5.2, we build the connection between the norm equivalence and our generalized sparse grid space via the theory of subspace splittings. In Section 5.3, we derive a Linear Program to find close-to-optimal positive scalings but prove the limits of this approach for H^t-elliptic problems. In Section 5.4, we admit negative values for the scaling of our generating system and thus obtain optimal condition numbers. The same condition numbers are realized by a block-diagonal preconditioner in Section 5.5. In Section 5.6, the non-linear variable preconditioner OptiCom is described. Section 5.7 contains experiments up to dimension $d = 10$ that support our theoretical findings.

5.1 Norm equivalences based on orthogonal subspaces

Introducing the orthogonal complement spaces $W_l^{(p)} = V_l^{(p)} \ominus_{V^{(p)}} V_{l-1}^{(p)}$ for $l \geq 1$, and setting $W_0^{(p)} := V_0^{(p)}$, we can write V as the $(\cdot,\cdot)_V$-orthogonal (from now on V-orthogonal) sum $V = \bigoplus_{\mathbf{l}\in\mathbb{N}^d} W_{\mathbf{l}}$ of the subspaces

$$W_{\mathbf{l}} = W_{l_1}^{(1)} \otimes \ldots \otimes W_{l_d}^{(d)}, \qquad \mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d \,.$$

Similarly, any finite-dimensional space $V_{\mathbf{l}} = V_{l_1}^{(1)} \otimes \ldots \otimes V_{l_d}^{(d)} \subset V$, $\mathbf{l} \in \mathbb{N}^d$, is the V-orthogonal sum $V_{\mathbf{l}} = \bigoplus_{\mathbf{k}\leq\mathbf{l}} W_{\mathbf{k}}$, where the inequality $\mathbf{k} \leq \mathbf{l}$ is meant componentwise. Furthermore, it is easy to verify that

$$V_{\mathcal{I}} = \bigoplus_{\mathbf{k}\in\mathcal{I}} W_{\mathbf{k}} \tag{5.2}$$

can be written as V-orthogonal sum of the subspaces $W_{\mathbf{k}}$ with $\mathbf{k} \in \mathcal{I}$.

Based on (5.2), we can estimate the redundancy we create by using a generating system instead of a basis. Let us assume that for our spaces $V_l^{(p)}, l \in \mathbb{N}, p = 1, \ldots, d$,

$$c \cdot n_{l-1} \leq n_l \quad \Leftrightarrow \quad n_{l-1} \leq \tfrac{1}{c} \cdot n_l$$

holds for $l \geq 1$, e.g., $c = 2$ for dyadically refined linear splines. Then,

$$\dim W_l = n_l - n_{l-1} \geq (1 - \tfrac{1}{c}) n_l$$

and

$$\dim V_{\mathcal{I}} = \sum_{\mathbf{k}\in\mathcal{I}} \dim W_{\mathbf{k}} \geq (1-\tfrac{1}{c})^d \sum_{\mathbf{k}\in\mathcal{I}} n_{\mathbf{k}} \Rightarrow N_{\mathcal{I}} \leq (1-\tfrac{1}{c})^{-d} \dim V_{\mathcal{I}} \,, \tag{5.3}$$

which means we can limit the number of degrees of freedom in our generating system $N_{\mathcal{I}}$ by the dimension of $V_{\mathcal{I}}$ times a factor $(1-\frac{1}{c})^{-d}$. It is not hard to see that this factor is relatively sharp, and thus, for $c = 2$, we have a factor of about 2^d more degrees of freedom in our generating system than needed in the sparse grid space. This result coincides with the somewhat simpler calculation (4.17).

The basis for many further considerations in this thesis is the assumption of a set of fixed positive weights $\beta_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d$, and a norm equivalence

$$\lambda_{\min} \sum_{\mathbf{k}\in\mathbb{N}^d} \beta_{\mathbf{k}} \|w_{\mathbf{k}}\|_V^2 \leq \|u\|_a^2 \leq \lambda_{\max} \sum_{\mathbf{k}\in\mathbb{N}^d} \beta_{\mathbf{k}} \|w_{\mathbf{k}}\|_V^2 \,, \tag{5.4}$$

where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$ and the $w_{\mathbf{k}} \in W_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d$, denote the components of the unique V-orthogonal decomposition of $u \in H$, i.e., $u = \sum_{\mathbf{k}\in\mathbb{N}^d} w_{\mathbf{k}}$. The estimate (5.4) is assumed to be sharp, which means that the *norm equivalence constants* $\lambda_{\min}$ and $\lambda_{\max}$ are given by

$$\lambda_{\min} := \inf_{0\neq u\in H} \frac{\|u\|_a^2}{\sum_{\mathbf{k}\in\mathbb{N}^d} \beta_{\mathbf{k}} \|w_{\mathbf{k}}\|_V^2} \quad \text{and} \quad \lambda_{\max} := \sup_{0\neq u\in H} \frac{\|u\|_a^2}{\sum_{\mathbf{k}\in\mathbb{N}^d} \beta_{\mathbf{k}} \|w_{\mathbf{k}}\|_V^2} \,.$$

From now on, we use the symbol $\simeq$ to indicate such an equivalence, and call $\kappa = \lambda_{\max}/\lambda_{\min}$ *condition number* of the splitting.

The described setup is motivated by the discretization of H^t-elliptic problems by generalized sparse grid spaces $V_{\mathcal{I}}$ over the d-dimensional unit cube. The restriction on t is $|t| < r + 3/2$ if we use C^r spline spaces of fixed degree $m \geq r+1$ over dyadic partitions of step size 2^{-l} as the building blocks $V_l^{(p)}$. Then, the equivalence of norms (5.4) has the form

$$\|u\|_{H^t}^2 \simeq \sum_{\mathbf{k}\in\mathbb{N}^d} 2^{2t|\mathbf{k}|_\infty} \|w_{\mathbf{k}}\|_{L_2(\Omega^d)}^2, \qquad |\mathbf{k}|_\infty = \max_{i=1,\ldots,d} k_i \,, \tag{5.5}$$

where the $w_{\mathbf{k}} \in W_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d$, denote the L_2-orthogonal components of the function $u \in H^t$, see [Osw94] for this kind of results. A more general result for $H^{t,l}_{\text{mix}}$-elliptic problems is stated in [GK09].

We now single out the case of linear spline spaces for H^1-elliptic problems and take a look at the norm equivalence constants for a special set of weights $(\beta_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}^d}$. Due to Jackson- and Bernstein-inequalities [Dah96, Osw94] for dyadically refined linear spline spaces $(V_l^{(p)})_{l=1}^\infty, p = 1, \ldots, d$, we have a norm equivalence

$$\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial x}\right)_{L_2(\Omega)} \simeq \sum_{l\in\mathbb{N}} \beta_l \|w_l\|_{L_2(\Omega)}^2 \tag{5.6}$$

for $u \in H_0^1(\Omega)$ with $\beta_l = 2^{2l}$, $u = \sum_{l\in\mathbb{N}} w_l$, where $w_l \in W_l, l \in \mathbb{N}$, and $0 < \lambda_{\min}^{(1)} \leq \lambda_{\max}^{(1)} < \infty$. Of course, (5.6) holds also for $u \in V_J^{(p)}, p = 1, \ldots, d$, with norm equivalence constants $\lambda_{\min}^{(1),J} \geq \lambda_{\min}^{(1)}$ and $\lambda_{\max}^{(1),J} \leq \lambda_{\max}^{(1)}$ for all $J \in \mathbb{N}$. The next theorem shows that this norm equivalence carries

over to the d-dimensional case with exactly the same norm equivalence constants. This is a generalization of the proof given in [GH14b].

Theorem 5.1. *For $u \in H_0^1(\Omega^d)$, $\alpha_p > 0, p = 1, \dots, d$ and*

$$a(u,v) = \sum_{p=1}^{d} \alpha_p \left(\frac{\partial u}{\partial x_p}, \frac{\partial v}{\partial x_p} \right)_{L_2(\Omega^d)} , \tag{5.7}$$

it holds that

$$a(u,u) \simeq \sum_{\mathbf{l} \in \mathbb{N}^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} \Big) \|w_{\mathbf{l}}\|^2_{L_2(\Omega^d)} \quad \textit{for} \quad u = \sum_{\mathbf{l} \in \mathbb{N}^d} w_{\mathbf{l}} \quad \textit{with} \quad w_{\mathbf{l}} \in W_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}^d , \tag{5.8}$$

where the constants $\lambda_{min}^{(d)}$ and $\lambda_{max}^{(d)}$ associated with (5.8) are the same as in (5.6) i.e. $\lambda_{min}^{(d)} = \lambda_{min}^{(1)}$ and $\lambda_{max}^{(d)} = \lambda_{max}^{(1)}$. For functions $u \in V_{\mathcal{F}_J^d} = V_{\mathbf{J}} \subset H_0^1(\Omega^d)$, the corresponding equivalence constants $\lambda_{max}^{(d),J}$ and $\lambda_{min}^{(d),J}$ of (5.8) are equal to $\lambda_{max}^{(1),J}$ and $\lambda_{min}^{(1),J}$, respectively, for $J \in \mathbb{N}$.

Proof. We start with the case $u \in V_{\mathcal{F}_J^d}, J \in \mathbb{N}$. In Chapter 4, the functions $(\phi_{J,i})_{i=1}^{n_J}$ formed a basis for the spaces $V_J^{(p)}, p = 1, \dots, d$. Of course, there also exists an L_2-orthonormal basis $(\psi_{J,i})_{i=1}^{n_J}$ of V_J. Furthermore, we need the orthogonal decomposition $(\omega_{l,i})_{l=1}^{J}$ of $\psi_{J,i} \in V_J$ for all $i = 1, \dots, n_J$ with $\omega_{l,i} \in W_l, l = 1, \dots, J$ and

$$\psi_{J,i} = \sum_{l=1}^{J} \omega_{l,i} \, .$$

Next, analogously to (4.5), we define

$$\psi_{\mathbf{J},\mathbf{i}}(\mathbf{x}) = \psi_{J,i_1}(x_1) \cdots \psi_{J,i_d}(x_d) \quad \text{and} \quad \omega_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = \omega_{l_1,i_1}(x_1) \cdots \omega_{l_d,i_d}(x_d)$$

for $\mathbf{x} = (x_1, \dots, x_d)$, $\mathbf{i} \in \chi_{\mathbf{J}}$ and $\mathbf{l} \in \mathcal{F}_J^d$. This opens a direct way to find orthogonal decompositions of functions $u = \sum_{\mathbf{i} \in \chi_{\mathbf{J}}} z_{\mathbf{i}} \psi_{\mathbf{J},\mathbf{i}} \in V_{\mathcal{F}_J^d}$ by

$$u = \sum_{\mathbf{i} \in \chi_{\mathbf{J}}} z_{\mathbf{i}} \sum_{\mathbf{l} \in \mathcal{F}_J^d} \omega_{\mathbf{l},\mathbf{i}} = \sum_{\mathbf{l} \in \mathcal{F}_J^d} \sum_{\mathbf{i} \in \chi_{\mathbf{J}}} z_{\mathbf{i}} \omega_{\mathbf{l},\mathbf{i}} = \sum_{\mathbf{l} \in \mathcal{F}_J^d} w_{\mathbf{l}}$$

with

$$w_{\mathbf{l}} = \sum_{\mathbf{i} \in \chi_{\mathbf{J}}} z_{\mathbf{i}} \omega_{\mathbf{l},\mathbf{i}} \in W_{\mathbf{l}} \tag{5.9}$$

for all $\mathbf{l} \in \mathcal{F}_J^d$.

Now, we show that the norm equivalence (5.8) holds for any $u \in V_{\mathcal{F}_J^d}$ with the constants

$\lambda_{\max}^{(1),J}$ and $\lambda_{\min}^{(1),J}$ from (5.6). It holds that

$$a(u,u) = \sum_{p=1}^{d} \alpha_p \Big(\frac{\partial}{\partial x_p} \sum_{\mathbf{i}\in\chi_{\mathbf{J}}} z_{\mathbf{i}} \psi_{\mathbf{J},\mathbf{i}}, \frac{\partial}{\partial x_p} \sum_{\mathbf{j}\in\chi_{\mathbf{J}}} z_{\mathbf{j}} \psi_{\mathbf{J},\mathbf{j}} \Big)_{L_2(\Omega^d)} \tag{5.10}$$

$$= \sum_{p=1}^{d} \alpha_p \sum_{\mathbf{i}\in\chi_{\mathbf{J}}} \sum_{\mathbf{j}\in\chi_{\mathbf{J}}} \Big(\frac{\partial}{\partial x_p} z_{\mathbf{i}} \psi_{J,i_p}, \frac{\partial}{\partial x_p} z_{\mathbf{j}} \psi_{J,j_p} \Big)_{L_2(\Omega)} \prod_{\substack{q=1 \\ q\neq p}}^{d} (\psi_{J,i_q}, \psi_{J,j_q})_{L_2(\Omega)} \tag{5.11}$$

$$= \sum_{p=1}^{d} \sum_{\substack{\mathbf{i}'=\mathbf{i}\setminus\{i_p\} \\ \mathbf{i}\in\chi_{\mathbf{J}}}} \Big(\frac{\partial}{\partial x_p} \sum_{i_p=1}^{n_J} z_{\mathbf{i}'\cup\{i_p\}} \psi_{J,i_p}, \frac{\partial}{\partial x_p} \sum_{j_p=1}^{n_J} z_{\mathbf{i}'\cup\{j_p\}} \psi_{J,j_p} \Big)_{L_2(\Omega)} . \tag{5.12}$$

We obtain (5.11) by repeated applications of the distributive law and by using the product structure of the L_2-scalar product. Then, the orthonormal basis property of the $(\psi_{J,i})_{i=1}^{n_J}$ cancels all terms for $i_q \neq j_q, q \neq p$ and we get (5.12). Here we use the notation

$$\begin{aligned} \mathbf{i} \quad &\setminus \quad \{i_p\} = (i_1, \ldots, i_{p-1}, i_{p+1}, \ldots, i_d) \quad \text{and} \\ \mathbf{i}' \quad &\cup \quad \{i_p\} = (i_1, \ldots, i_{p-1}, i_p i_{p+1}, \ldots, i_d) . \end{aligned}$$

We can apply the one-dimensional norm equivalence (5.6) to (5.12) and obtain the upper bound

$$\cdots \leq \sum_{p=1}^{d} \alpha_p \lambda_{\max}^{(1),J} \sum_{\substack{\mathbf{i}'=\mathbf{i}\setminus\{i_p\} \\ \mathbf{i}\in\chi_{\mathbf{J}}}} \sum_{l_p=1}^{J} 2^{2l_p} \Big(\sum_{i_p=1}^{n_J} z_{\mathbf{i}'\cup\{i_p\}} \omega_{l_p,i_p}, \sum_{j_p=1}^{n_J} z_{\mathbf{i}'\cup\{j_p\}} \omega_{l_p,j_p} \Big)_{L_2(\Omega)} \tag{5.13}$$

$$= \lambda_{\max}^{(1),J} \sum_{p=1}^{d} \alpha_p \sum_{\mathbf{i}\in\chi_{\mathbf{J}}} \sum_{\mathbf{j}\in\chi_{\mathbf{J}}} \sum_{l_p=1}^{J} 2^{2l_p} (z_{\mathbf{i}} \omega_{l_p,i_p}, z_{\mathbf{j}} \omega_{l_p,j_p})_{L_2(\Omega)} \cdot \prod_{\substack{q=1 \\ q\neq p}}^{d} (\psi_{J,i_q}, \psi_{J,j_q})_{L_2(\Omega)} \tag{5.14}$$

$$= \lambda_{\max}^{(1),J} \sum_{p=1}^{d} \alpha_p \sum_{\mathbf{i}\in\chi_{\mathbf{J}}} \sum_{\mathbf{j}\in\chi_{\mathbf{J}}} \sum_{\mathbf{l}\in\mathcal{F}_J^d} 2^{2l_p} (z_{\mathbf{i}} \omega_{l_p,i_p}, z_{\mathbf{j}} \omega_{l_p,j_p})_{L_2(\Omega)} \cdot \prod_{\substack{q=1 \\ q\neq p}}^{d} (\omega_{l_q,i_q}, \omega_{l_q,j_q})_{L_2(\Omega)} \tag{5.15}$$

$$= \lambda_{\max}^{(1),J} \sum_{\mathbf{l}\in\mathcal{F}_J^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} \Big) \Big(\sum_{\mathbf{i}\in\chi_{\mathbf{J}}} z_{\mathbf{i}} \omega_{\mathbf{l},\mathbf{i}}, \sum_{\mathbf{j}\in\chi_{\mathbf{J}}} z_{\mathbf{j}} \omega_{\mathbf{l},\mathbf{j}} \Big)_{L_2(\Omega^d)} . \tag{5.16}$$

In (5.13) and (5.14), we used the distributive law again and reintroduced the terms we dropped previously. In (5.15), we replaced the ψ_{J,i_q} and ψ_{J,j_q} by the decompositions $\sum_{l_q=1}^{J} \omega_{l_q,i_q}$ and $\sum_{l_q=1}^{J} \omega_{l_q,j_q}$, respectively. Then, in (5.16), we recombined the product of d one-dimensional L_2-scalar products to one d-dimensional L_2-scalar product. Note that the lower bound with $\lambda_{\min}$ can be proven in the same way. Now, in combination with (5.9), we know that (5.8) is a norm equivalence with constants $\lambda_{\max}^{(d),J} \leq \lambda_{\max}^{(1),J}$ and $\lambda_{\min}^{(d),J} \geq \lambda_{\min}^{(1),J}$.

It is almost trivial to prove the sharpness of the estimates, i.e. to show that indeed $\lambda_{\max}^{(d),J} = \lambda_{\max}^{(1),J}$ and $\lambda_{\min}^{(d),J} = \lambda_{\min}^{(1),J}$. Choose the functions $u_{\max}^{(1),J} \in V_J^{(p)}, p = 1, \ldots, d$, for which the

maximum in (5.6) is attained and plug the multivariate function

$$u(\mathbf{x}) = \prod_{p=1}^{d} u_{\max}^{(1),J}(x_p)$$

into (5.10). This results in an equality instead of an upper bound in (5.13). The $\lambda_{\min}^{(d),J}$-case can be shown analogously.

We can now easily extend this result to any function $u \in H_0^1(\Omega^d)$ by a density argument and

$$\lambda_{\max}^{(d)} = \lim_{J\to\infty} \lambda_{\max}^{(d),J} = \lim_{J\to\infty} \lambda_{\max}^{(1),J} = \lambda_{\max}^{(1)} .$$

Again, the $\lambda_{\min}^{(d)}$-case works analogously. □

Theorem 5.1 is important to understand why the condition numbers of sparse grid discretizations of the Laplacian decrease with rising dimension. We will discuss this effect in Subsection 5.7.1. The following Theorem also includes a reaction term.

Theorem 5.2. *Let $\lambda_{min}^{(1)}$ and $\lambda_{max}^{(1)}$ be the norm equivalence constants from (5.6) for functions $u \in H_0^1(\Omega)$. For $u \in H_0^1(\Omega^d)$, $\alpha_p > 0, p = 1, \ldots, d$, $\gamma \geq 0$ and*

$$a(u,v) = \sum_{p=1}^{d} \alpha_p \left(\frac{\partial u}{\partial x_p}, \frac{\partial v}{\partial x_p} \right)_{L_2(\Omega^d)} + \gamma(u,v)_{L_2(\Omega^d)} , \tag{5.17}$$

it holds that

$$a(u,u) \eqsim \sum_{\mathbf{l}\in\mathbb{N}^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} + \frac{2\gamma}{\lambda_{max}^{(1)} + \lambda_{min}^{(1)}} \Big) \|w_{\mathbf{l}}\|_{L_2(\Omega^d)}^2 \quad \textit{for} \quad u = \sum_{\mathbf{l}\in\mathbb{N}^d} w_{\mathbf{l}} \tag{5.18}$$

with $w_{\mathbf{l}} \in W_{\mathbf{l}}, \mathbf{l} \in \mathbb{N}^d$. The norm equivalence constants $\lambda_{min}^{(d)}$ and $\lambda_{max}^{(d)}$ of (5.18) satisfy $\lambda_{min}^{(d)} \geq \lambda_{min}^{(1)}$ and $\lambda_{max}^{(d)} \leq \lambda_{max}^{(1)}$.

Proof. The proof consists of an application of Theorem 5.1 and a simple inequality

$$\begin{aligned}
a(u,u) &\leq \lambda_{\max}^{(1)} \sum_{\mathbf{l}\in\mathbb{N}^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} \Big) \|w_{\mathbf{l}}\|_{L_2(\Omega^d)}^2 + \gamma(u,v)_{L_2(\Omega^d)} \\
&= \lambda_{\max}^{(1)} \sum_{\mathbf{l}\in\mathbb{N}^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} \Big) \|w_{\mathbf{l}}\|_{L_2(\Omega^d)}^2 + \lambda_{\max}^{(1)} \frac{\gamma}{\lambda_{\max}^{(1)}} \sum_{\mathbf{l}\in\mathbb{N}^d} \|w_{\mathbf{l}}\|_{L_2(\Omega^d)}^2 \\
&\leq \lambda_{\max}^{(1)} \sum_{\mathbf{l}\in\mathbb{N}^d} \Big(\sum_{p=1}^{d} \alpha_p 2^{2l_p} + \frac{2\gamma}{\lambda_{\max}^{(1)} + \lambda_{\min}^{(1)}} \Big) \|w_{\mathbf{l}}\|_{L_2(\Omega^d)}^2 .
\end{aligned}$$

The lower bound is proven analogously using $\frac{\gamma}{\lambda_{\min}^{(1)}} \geq \frac{2\gamma}{\lambda_{\min}^{(1)} + \lambda_{\max}^{(1)}}$. □

What we have shown is that the norm equivalence constants of (5.4) for any bilinear operator in the form (5.7) and the weights $\beta_{\mathbf{l}} = \sum_{p=1}^{d} \alpha_p 2^{2l_p}, \mathbf{l} \in \mathbb{N}^d$, are equal to $\lambda_{\min}^{(1)}$ and $\lambda_{\max}^{(1)}$ from (5.6) and thus independent of the diffusion coefficients and even the dimension. This results also covers bilinear forms with a reaction term (5.17). In [DSS09], a similar result regarding the robustness and dimension-independence of condition numbers was proven in the context of the best N-term approximation using $L_2(0,1)$-orthonormal wavelets. The norm equivalence (5.8) can also be found in, e.g., [GK09] or in [GO95b] for the two-dimensional case, but there no special attention was paid to the dimension-independence of the equivalence constants. In the next sections we assume a norm equivalence in the general form (5.4) and use it as the basis to derive an optimal diagonal scaling for our generating system.

5.2 The theory of subspace splittings applied to sparse grids

If a basis of V-orthonormal wavelets is available, we can use the norm equivalence (5.4) to precondition the sparse grid operator matrix based on the bilinear form $a(\cdot,\cdot)$ and obtain an upper bound $\kappa := \lambda_{\max}/\lambda_{\min}$ for the resulting condition numbers. However, we shun the effort of finding and implementing such a basis and stick with our generating system. The question is now what diagonal scaling preconditioners are available to realize a condition number as close as possible to κ.

In Subsection 5.2.1, we recall some facts about subspace correction methods [Xu92, Osw94] for solving a variational problem (5.1) in a Hilbert space H. We slightly depart from this standard theory and introduce scaling parameters for the subspace solvers in Subsection 5.2.2. We apply the theory of subspace correction methods to our generalized sparse grids in Subsection 5.2.3 and discuss the implementation in a generating system in Subsection 5.2.4. This section and the following ones closely follow the lines of [GHO15].

5.2.1 Subspace splitting theory

Let $H_i, i \in I$, be auxiliary Hilbert spaces with an at most countable index set I (if we discuss algorithmic issues, we silently assume that I is finite). Each H_i carries a symmetric H_i-elliptic bilinear form $b_i(u_i, v_i)$, and writing $\{H_i; b_i\}$ indicates that we use this bilinear form as scalar product on H_i. In general, the H_i, $i \in I$, are not assumed to be subspaces of H, and in order to relate them with H, we define bounded linear embedding operators $R_i : H_i \to H$. We call the formal decomposition

$$\{H; a\} = \sum_{i \in I} R_i \{H_i, b_i\} \tag{5.19}$$

a stable space splitting if for any $u \in H$ there is at least one H-converging representation of the form

$$u = \sum_{i \in I} R_i v_i, \qquad v_i \in H_i, \quad i \in I\,, \tag{5.20}$$

and

$$0 < \lambda_{\min} := \inf_{u \in H} \frac{a(u,u)}{|||u|||^2} \;\le\; \lambda_{\max} := \sup_{u \in H} \frac{a(u,u)}{|||u|||^2} < \infty\,, \tag{5.21}$$

where

$$|||u|||^2 := \inf_{v_i \in H_i:\ u=\sum_{i\in I} R_i v_i} \sum_{i\in I} b_i(v_i, v_i)\ . \tag{5.22}$$

The constants $\lambda_{\min}$ and $\lambda_{\max}$ are called lower and upper stability constants, and $\kappa = \lambda_{\max}/\lambda_{\min}$ is called the condition number of the space splitting (5.19), respectively. It is easy to see that frames and fusion frames [CK04, Osw09] are special cases of this definition, where $a(\cdot,\cdot) = (\cdot,\cdot)_H$, the H_i are closed subspaces of H, the scalar products $b_i(\cdot,\cdot) = w_i \cdot (\cdot,\cdot)_V$ are modified by weights $w_i > 0$, and the R_i denote the natural embeddings $H_i \subset H$ for $i \in I$. In the frame case, the H_i are one-dimensional spaces and are spanned by individual frame elements, see also [KOPT13]. Specific examples of stable space splittings related to sparse grid discretizations will be given in the Subsection 5.2.3.

To formally define the core subspace correction methods associated with a stable space splitting (5.19), we define the adjoint operators $R_i^* : H \to H_i$ by

$$b_i(R_i^* u, v_i) = a(u, R_i v_i) \qquad \forall\ v_i \in H_i\ , \tag{5.23}$$

and set $T_i := R_i R_i^* : H \to H$, $i \in I$, and $P = \sum_{i\in I} T_i$.

The additive Schwarz method for (5.19) (also called *parallel* or *asynchronous subspace correction method*) is then given by the iteration

$$u^{(m+1)} = u^{(m)} + \omega \sum_{i\in I} T_i e^{(m)} = u^{(m)} + \omega P e^{(m)}, \quad m = 0, 1, \dots\ , \tag{5.24}$$

where the single relaxation parameter $\omega > 0$ is to be chosen appropriately and $e^{(m)} := u - u^{(m)}$ denotes the current error. The essential work to be done is to compute all $u_i^{(m)} := R_i^* e^{(m)} \in H_i$ by solving the subproblems

$$\begin{aligned} b_i(u_i^{(m)}, v_i) &= a(u - u^{(m)}, R_i v_i) \\ &= F(R_i v_i) - a(u^{(m)}, R_i v_i) \quad \forall\ v_i \in H_i, \quad i \in I\ . \end{aligned} \tag{5.25}$$

The theoretically best value of ω is given by $\omega^* = 2/(\lambda_{\min} + \lambda_{\max})$, since the operator $P = \sum_{i\in I} T_i$ satisfies the identity

$$a(Pu, u) = \sum_{i\in I} b_i(R_i^* u, R_i^* u) = |||Pu|||^2\ .$$

Together with (5.21), this implies that $\lambda_{\min}(P) = \lambda_{\min}$, $\lambda_{\max}(P) = \lambda_{\max}$, and $\kappa(P) = \kappa$. Thus, the best possible error reduction factor in the energy norm for the linear iteration (5.24) is given by

$$\rho := \inf_{\omega>0} \|\mathrm{Id} - \omega P\|_a = \|\mathrm{Id} - \omega^* P\|_a = 1 - \frac{2}{1+\kappa}\ . \tag{5.26}$$

This simple result has appeared in many papers, see [Xu92, Osw94, GO95a].

Since in practice the value ω^* is hardly accessible, one often determines in each iteration the

value

$$\omega^{(m)} = \frac{a(Pe^{(m)}, e^{(m)})}{a(Pe^{(m)}, Pe^{(m)})} , \tag{5.27}$$

which corresponds to finding $u^{(m+1)}$ by solving the minimization problem

$$\phi(u^{(m)} + \omega^{(m)} Pe^{(m)}) \to \min_{\omega^{(m)}} \qquad \text{with} \qquad \phi(u) := \frac{1}{2} a(u,u) - F(u) ,$$

or, equivalently, by minimizing the energy error

$$\|u - u^{(m)} - \omega^{(m)} Pe^{(m)}\|_a^2 \to \min_{\omega^{(m)}} \tag{5.28}$$

with respect to the parameter $\omega^{(m)} > 0$. The iterative method with the parameter choice $\omega^{(m)}$ from (5.27) can also be interpreted as steepest descent method for the quadratic minimization problem associated with the linear variational problem

$$a(Pu, v) = f(Pv) \qquad \forall\, v \in H , \tag{5.29}$$

which is a preconditioned version of (5.1). Consequently, since

$$\|e^{(m+1)}\|_a = \|u - (u^{(m)} + \omega^{(m)} Pe^{(m)})\|_a \le \inf_{\omega > 0} \|\mathrm{Id} - \omega P\|_a \|e^{(m)}\|_a = \rho \|e^{(m)}\|_a ,$$

this method is as good as any linear method (5.24). Note furthermore that the conjugate gradient method can be realized by minimizing the energy error

$$\|u - u^{(m)} - \omega^{(m)} Pe^{(m)} - \eta^{(m)} (u^{(m)} - u^{(m-1)})\|_a^2 \to \min_{\omega^{(m)}, \eta^{(m)}} \tag{5.30}$$

in every iteration step [GL96].

An alternative to the above additive Schwarz method is the *multiplicative Schwarz method* (or *synchronous subspace correction method*), where in the n-th step only one index $i = i^{(n)} \in I$ is picked, the corresponding subproblem is solved and used to immediately update the iterate according to

$$u^{(n+1)} = u^{(n)} + \omega T_{i^{(n)}} e^{(n)}, \qquad n = 0, 1, \dots . \tag{5.31}$$

Here, various rules for choosing the next subproblem index $i^{(n)}$ (cyclic deterministic rules, random choices, greedy pick, and their combinations), and block updates (intermediate between the additive and multiplicative versions) have been proposed, see for example [GO12].

The convergence theory of multiplicative Schwarz methods is a bit more intricate. Generally speaking, they often are slightly faster than additive Schwarz methods (to achieve a fair comparison, usually $\#I$ steps of the multiplicative method are combined into one step). A potential drawback is that the multiplicative method is less straightforward for parallelization. Since our focus is on the choice of scaling parameters in additive Schwarz methods, we do not want to go into detail but refer to the literature, see [Xu92, Gri94a, Gri94b, Osw94, GO95a, XZ02, Osw09, GO12].

5.2.2 Introducing multiple scaling factors

From now on we slightly depart from the above theory and ask the question if the introduction of individual scaling parameters ω_i offers additional improvements. To this end, we keep the basic setup of given auxiliary problems in $\{H_i, b_i\}$, and consider the family of iterations

$$u^{(m+1)} = u^{(m)} + \sum_{i\in I} \omega_i T_i e^{(m)} = u^{(m)} + P_{\boldsymbol{\omega}} e^{(m)}, \quad m = 0, 1, \dots , \tag{5.32}$$

where $P_{\boldsymbol{\omega}} = \sum_{i\in I} \omega_i T_i$ and $\boldsymbol{\omega}$ stands from now on for a set of scaling parameters $(\omega_i)_{i\in I}$. Convergence of this linear iterative method is guaranteed if and only if $P_{\boldsymbol{\omega}}$ is strictly positive definite ($\lambda_{\min}(P_{\boldsymbol{\omega}}) > 0$) and $\lambda_{\max}(P_{\boldsymbol{\omega}}) < 2$. We note that enforcing the upper bound $\lambda_{\max}(P_{\boldsymbol{\omega}}) < 2$ is not a major concern. For this, we can, at little extra work, determine an additional relaxation parameter using the steepest descent approach, and guarantee an error reduction of at least $1 - 2/(1 + \kappa(P_{\boldsymbol{\omega}}))$. Thus, the question about guaranteeing best convergence rates for (5.32) is essentially equivalent to determining positive definite $P_{\boldsymbol{\omega}}$ with (close to) minimal condition numbers. To this end, let Ω denote the family of all parameter sets $\boldsymbol{\omega}$ such that $P_{\boldsymbol{\omega}}$ is bounded and strictly positive definite. Then, the optimal error reduction rate ρ^* can be expressed as

$$\rho^* = \inf_{\boldsymbol{\omega}\in\Omega} \|\mathrm{Id} - P_{\boldsymbol{\omega}}\|_a = 1 - \frac{2}{1+\kappa^*}, \qquad \kappa^* := \inf_{\boldsymbol{\omega}\in\Omega} \kappa(P_{\boldsymbol{\omega}}) \, . \tag{5.33}$$

Note that Ω may contain parameter sets with some negative or zero ω_i. To cover such situations, the theory of subspace correction methods based on stable space splittings (5.19) is not of immediate help, as it can only deal with the case $\omega_i > 0$. Indeed, the operator P associated with the space splitting (5.19) becomes $P_{\boldsymbol{\omega}}$ if the auxiliary bilinear forms $b_i(u_i, v_i)$ are replaced by their weighted versions $\omega_i^{-1} \cdot b_i(u_i, v_i)$, $i \in I$. Thus, whenever the space splitting is stable, and $0 < \omega_{\min} \le \omega_i \le \omega_{\max} < \infty$, the space splitting with these modified auxiliary scalar products is also stable according to (5.21), and satisfies

$$\frac{\omega_{\min}}{\omega_{\max}} \kappa(P) \le \kappa(P_{\boldsymbol{\omega}}) \le \frac{\omega_{\max}}{\omega_{\min}} \kappa(P) \, .$$

For $\omega_i > 0$ and finite I, this rough estimate guarantees $\boldsymbol{\omega} \in \Omega$, but does not help in minimizing $\kappa(P_{\boldsymbol{\omega}})$, nor in dealing with sets $\boldsymbol{\omega} \in \Omega$ that contain some negative ω_i.

A priori choice of scaling parameters

We want to find a set $\boldsymbol{\omega}^* \in \Omega$ that realizes or at least comes close to the optimal error reduction rate (5.33), i.e. $\rho^* = \rho(P_{\boldsymbol{\omega}^*})$ and $\kappa^* = \kappa(P_{\boldsymbol{\omega}^*})$, respectively.

Even though we do not believe that this leads to a practically useful approach in this generality, we mention that the problem of finding $\boldsymbol{\omega}^*$ can be formulated as semi-definite program (if H is finite dimensional, and I is finite). To this end, we set

$$\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & -\mathrm{Id} \end{pmatrix}, \quad \mathbf{A}_i = \begin{pmatrix} -T_i & 0 \\ 0 & T_i \end{pmatrix}, \quad \mathbf{A}' = \begin{pmatrix} \mathrm{Id} & 0 \\ 0 & 0 \end{pmatrix} \, .$$

Then minimizing the value of λ with respect to the vector of variables $(\boldsymbol{\omega}, \lambda)$ subject to the

constraint that

$$\mathbf{A} + \sum_{i \in I} \omega_i \mathbf{A}_i + \lambda \mathbf{A}' \geq 0 \tag{5.34}$$

is a semi-definite program [VB96]. It computes the set of weights $\boldsymbol{\omega}^*$ that realizes the optimal error reduction rate (5.33) under the constraint that the 2×2-system on the left-hand side of (5.34) is positive semi-definite. If $(\tilde{\boldsymbol{\omega}}, \tilde{\lambda})$ is a minimizer of (5.34), we have

$$\kappa^* = \tilde{\lambda}, \qquad \omega_i^* = \frac{2\tilde{\omega}_i}{1 + \tilde{\lambda}}, \quad i \in I \, .$$

For related problems, we refer the reader to [KOPT13] and the references therein. In this chapter, we tackle this problem for sparse grid discretizations in a less general but more efficient way by incorporating knowledge on the tensor product structure of sparse grid spaces and a norm equivalence (5.4).

5.2.3 Subspace splittings for generalized sparse grids

In Subsection 5.2.1, we recalled the general theory of additive subspace correction methods and introduced multiple scaling parameters in Subsection 5.2.2. We now become more specific and put the theory in the context of generalized sparse grid spaces. First, note that the norm equivalence (5.4) is in fact a stable subspace splitting

$$\{H, a(\cdot,\cdot)\} = \sum_{\mathbf{k} \in \mathbb{N}^d} \{W_{\mathbf{k}}, \beta_{\mathbf{k}}(\cdot,\cdot)_V\} \tag{5.35}$$

with a finite condition number denoted by κ^W. Note that in (5.35) we have a decomposition of H into a direct sum of V-orthogonal *sub*spaces, which allows us to omit the trivial embedding operators $R_{\mathbf{k}}^W : W_{\mathbf{k}} \to H$ that correspond to the R_i in (5.19) and (5.20).

Of course, for any generalized sparse grid space $V_{\mathcal{I}} \subset H$, (5.4) implies an associated stable subspace splitting

$$\{V_{\mathcal{I}}, a(\cdot,\cdot)\} = \sum_{\mathbf{k} \in \mathcal{I}} \{W_{\mathbf{k}}, \beta_{\mathbf{k}}(\cdot,\cdot)_V\} \tag{5.36}$$

with condition numbers $\kappa_{\mathcal{I}}^W$ uniformly bounded by κ^W. Thus, if $\#\mathcal{I} < \infty$, we arrive at subspace correction methods for solving (5.1) on generalized sparse grid spaces $H = V_{\mathcal{I}}$ with convergence rates that are uniform with respect to $\mathcal{I}$. The computational cost per step of these optimally converging methods essentially depend on the involved subproblem solvers $T_{\mathbf{k}}^W : V_{\mathcal{I}} \to W_{\mathbf{k}}$. Since $W_{\mathbf{k}} \subset H$, we can write (5.23) for $b_i(\cdot,\cdot) = (\cdot,\cdot)_V$ directly in the form

$$(T_{\mathbf{k}}^W u, w_{\mathbf{k}})_V = a(u, w_{\mathbf{k}}) \qquad \forall\, w_{\mathbf{k}} \in W_{\mathbf{k}} \, .$$

We still have to account for the $\beta_{\mathbf{k}}$-weights in (5.36), which is done in the update step of the resulting subspace correction method

$$u^{(m+1)} = u^{(m)} + \tau \sum_{\mathbf{k} \in \mathcal{I}} \omega_{\mathbf{k}}^W T_{\mathbf{k}}^W e^{(m)}, \qquad m = 0, 1, \ldots \tag{5.37}$$

by setting $\omega_{\mathbf{k}}^W = \beta_{\mathbf{k}}^{-1}, \mathbf{k} \in \mathcal{I}$. Here, an appropriately selected relaxation parameter $\tau > 0$ guarantees convergence rates uniformly in $\mathcal{I}$. Note that methods like steepest descent or conjugate gradients determine $\tau^{(m)}$ automatically in every iteration step m.

However, computing with the spaces $W_{\mathbf{k}}$ is often not as convenient as computing with the original anisotropic full grid spaces $V_{\mathbf{l}}$. This is why we now turn to the splitting

$$\{V_{\mathcal{I}}, a(\cdot,\cdot)\} = \sum_{\mathbf{l}\in\mathcal{I}} \{V_{\mathbf{l}}, \gamma_{\mathbf{l}}(\cdot,\cdot)_V\} \tag{5.38}$$

with positive weights $\gamma_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$, which only needs subspace solvers $T_{\mathbf{l}}^V : V_{\mathcal{I}} \to V_{\mathbf{l}} \subset V_{\mathcal{I}}$,

$$(T_{\mathbf{l}}^V u, v_{\mathbf{l}})_V = a(u, v_{\mathbf{l}}) \qquad \forall\, v_{\mathbf{l}} \in V_{\mathbf{l}}\,,$$

that operate on full grid subspaces. The subspace correction method, which is based on the V-splitting corresponding to (5.38), is then of the form

$$u^{(m+1)} = u^{(m)} + \tau \sum_{\mathbf{l}\in\mathcal{I}} \omega_{\mathbf{l}}^V T_{\mathbf{l}}^V e^{(m)}, \qquad m = 0, 1, \dots\,, \tag{5.39}$$

with $\omega_{\mathbf{l}}^V = \gamma_{\mathbf{l}}^{-1}$. The next subsection deals with the implementation of (5.39) in a generating system for given scaling factors $\omega_{\mathbf{l}}^V$. In the sections after that, we explore possibilities to determine optimal or close-to-optimal weighs $\gamma_{\mathbf{l}}^{-1}$ and the corresponding scaling factors $\omega_{\mathbf{l}}^V$, such that the resulting convergence rates are competitive with the rates of the W-splitting based method (5.37) for given $\beta_{\mathbf{k}}, \mathbf{k} \in \mathcal{I}$.

5.2.4 Implementation

In this section, we describe the implementation of the sparse grid subspace correction methods (5.39) and discuss the computational complexity of one iteration step. For representing elements in our generalized sparse grid space $V_{\mathcal{I}}$, we use the generating system $\Phi_{\mathcal{I}}$ from (4.21) with $N_{\mathcal{I}}$ degrees of freedom. We assume that the unidirectional principle from Subsection 4.3.2 is available, which allows us to compute the matrix-vector product

$$\mathbf{y}_{\mathcal{I}} = \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}} \quad \Longleftrightarrow \quad y_{\mathbf{l},\mathbf{i}} = \sum_{\mathbf{k}\in\mathcal{I}} \sum_{\mathbf{j}\in\chi_{\mathbf{k}}} a(\phi_{\mathbf{k},\mathbf{j}}, \phi_{\mathbf{l},\mathbf{i}}) x_{\mathbf{k},\mathbf{j}} \qquad \forall\, \mathbf{i} \in \chi_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$$

with a computational complexity that is linear in the number of degrees of freedom $N_{\mathcal{I}}$.

In the last subsection, we did not need the embedding operators $R_i : H_i \to H$ from (5.19) and (5.20), since the corresponding $R_{\mathbf{l}}^V : V_{\mathbf{l}} \to V_{\mathcal{I}}$ are simply the identity in the subspace case $V_{\mathbf{l}} \subset V_{\mathcal{I}}$ and could be dropped from the notation. In the matrix-vector setting, we need however the associated rectangular matrices $\mathbf{R}_{\mathcal{I},\mathbf{l}} \in \mathbb{R}^{N_{\mathcal{I}} \times n_{\mathbf{l}}}$ that act as the identity on coefficients that belong to $V_{\mathbf{l}}$, and pad all other entries with zeros. Note that $\mathbf{R}_{\mathcal{I},\mathbf{l}}^T \in \mathbb{R}^{n_{\mathbf{l}} \times N_{\mathcal{I}}}$ crops a generating system vector by extracting only the coefficients that belong to subspace $V_{\mathbf{l}}$.

Now, let $\mathbf{x}_{\mathcal{I}}^{(m)} \in \mathbb{R}^{N_{\mathcal{I}}}$ be the vector representing the current iterate $u^{(m)}$ in the generating system $\Phi_{\mathcal{I}}$. The corresponding residual is then given by

$$\mathbf{r}_{\mathcal{I}}^{(m)} = \mathbf{b} - \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}^{(m)} = \mathbf{A}_{\mathcal{I}}\mathbf{e}_{\mathcal{I}}^{(m)}\,,$$

where $\mathbf{e}_{\mathcal{I}}^{(m)} := \mathbf{x}_{\mathcal{I}} - \mathbf{x}_{\mathcal{I}}^{(m)}$ is the current error vector. The auxiliary variational problem (5.25) for the space $V_{\mathbf{l}}$ translates into

$$\mathbf{M}_{\mathbf{l}}(\mathbf{x}_{\mathcal{I}}^{(m)})_{\mathbf{l}} = \mathbf{R}_{\mathcal{I},\mathbf{l}}^T(\mathbf{b}_{\mathcal{I}} - \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}^{(m)}) = \mathbf{R}_{\mathcal{I},\mathbf{l}}^T\mathbf{r}_{\mathcal{I}}^{(m)} \,, \tag{5.40}$$

where $\mathbf{M}_{\mathbf{l}} \in \mathbb{R}^{n_{\mathbf{l}} \times n_{\mathbf{l}}}$ is the mass matrix on the subspace $V_{\mathbf{l}}$ with

$$(\mathbf{M}_{\mathbf{l}})_{\mathbf{i},\mathbf{j}} = (\phi_{\mathbf{l},\mathbf{i}}, \phi_{\mathbf{l},\mathbf{j}})_V \qquad \forall\, \mathbf{i},\mathbf{j} \in \chi_{\mathbf{l}} \,,$$

which in the linear spline case and $(\cdot,\cdot)_V = (\cdot,\cdot)_{L_2(\Omega^d)}$ is simply a d-fold Kronecker-product of tridiagonal matrices (or band-limited matrices in the higher-order case), and can therefore be inverted with computational costs of $\mathcal{O}(n_{\mathbf{l}})$ operations. Having solved for $\mathbf{x}_{\mathbf{l}}^{(m)}$, the vector representation of $u_{\mathbf{l}}^{(m)} = T_{\mathbf{l}}^V e^{(m)}$ is simply $\mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}$. Obviously, these tasks can be performed for all $\mathbf{l} \in \mathcal{I}$ with total costs of $\mathcal{O}(N_{\mathcal{I}})$ operations. The topic of the following Sections 5.3 and 5.4 is how to choose scaling factors $\omega_{\mathbf{l}}^V, \mathbf{l} \in \mathcal{I}$. For now, we assume the $\omega_{\mathbf{l}}^V$ are given, and so we can set

$$\mathbf{x}_{\mathcal{I}}^{(m+1)} = \tau^{(m)} \sum_{\mathbf{l}\in\mathcal{I}} \omega_{\mathbf{l}}^V \mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)} + \eta^{(m)}(\mathbf{x}_{\mathcal{I}}^{(m)} - \mathbf{x}_{\mathcal{I}}^{(m-1)}) \tag{5.41}$$

with optimal $\tau^{(m)}$ and $\eta^{(m)} = 0$ for the steepest descent case, and optimal $\tau^{(m)}$ and $\eta^{(m)}$ for the conjugate gradient (CG) case. The computation of the optimal values $\tau^{(m)}$ or $(\tau^{(m)}, \eta^{(m)})$ associated with the minimization problems (5.28) and (5.30) leads to a 1×1 and 2×2 system of linear equations in the steepest descent and CG case, respectively. Setting up the system requires only a fixed number of matrix-vector multiplications, which can be computed with $\mathcal{O}(N_{\mathcal{I}})$ operations. The solution for $\tau^{(m)}$ or $(\tau^{(m)}, \eta^{(m)})$ can be done with costs of $\mathcal{O}(1)$, and the update step (5.41) is linear in the number of degrees of freedom $N_{\mathcal{I}}$. Altogether, we arrive at a complexity of the order $\mathcal{O}(N_{\mathcal{I}})$. This holds for any generalized sparse grid space.

5.3 Positive scalings for sparse grid subspace correction methods

In the following subsection, we present a method that explores the theory of stable subspace splittings and shows how to determine (positive) $\gamma_{\mathbf{l}}$-weights by solving a linear optimization problem such that the condition number of the splitting is in some sense minimal. Then, in Subsection 5.3.2, we show that H^t-elliptic problems and the best possible positive scaling lead to condition numbers of $\Theta(J^{d-2})$ for $J \to \infty$ for dimensions $d \geq 2$, which we can substantially improve by admitting negative scaling factors in the subsequent Section 5.4.

5.3.1 Formulation as Linear Program

Now, we want to determine weights $\gamma_{\mathbf{l}} > 0, \mathbf{l} \in \mathcal{I}$, such that the splitting number $\kappa_{\mathcal{I}}^V$ of (5.38) is small. To this end, we take a detour: Instead of estimating $\kappa_{\mathcal{I}}^V$ by comparing $a(u,u) = \|u\|_a^2$ and the squared splitting norm

$$|||u|||_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}},V}^2 = \inf_{u=\sum_{\mathbf{l}\in\mathcal{I}} u_{\mathbf{l}}} \sum_{\mathbf{l}\in\mathcal{I}} \gamma_{\mathbf{l}} \|u_{\mathbf{l}}\|_V^2 \tag{5.42}$$

associated with (5.38) directly, we concentrate on comparing $|||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I}}$ and the squared splitting norm

$$|||u|||^2_{\{\beta_\mathbf{k}\}_\mathcal{I},W} = \sum_{\mathbf{k}\in\mathcal{I}} \beta_\mathbf{k} \|w_\mathbf{k}\|^2_V \,, \qquad u = \sum_{\mathbf{k}\in\mathcal{I}} w_\mathbf{k}, \quad w_\mathbf{k} \in W_\mathbf{k} \tag{5.43}$$

associated with (5.36). We define

$$0 < \lambda^{WV}_{\min} := \inf_{u\in V_\mathcal{I}} \frac{|||u|||^2_{\{\beta_\mathbf{k}\}_\mathcal{I},W}}{|||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I},V}} \;\le\; \lambda^{WV}_{\max} := \sup_{u\in V_\mathcal{I}} \frac{|||u|||^2_{\{\beta_\mathbf{k}\}_\mathcal{I},W}}{|||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I},V}} < \infty \,,$$

and try to minimize $\kappa^{WV}_\mathcal{I} := \lambda^{WV}_{\max}/\lambda^{WV}_{\min}$ which is the relative condition number between the two splittings

$$\sum_{\mathbf{k}\in\mathcal{I}} \{W_\mathbf{k}, \beta_\mathbf{k}(\cdot,\cdot)_V\} \quad \text{and} \quad \sum_{\mathbf{l}\in\mathcal{I}} \{V_\mathbf{l}, \gamma_\mathbf{l}(\cdot,\cdot)_V\} \tag{5.44}$$

based on the same underlying generalized sparse grid space $V_\mathcal{I}$. Since

$$\frac{a(u,u)}{|||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I},V}} = \frac{a(u,u)}{|||u|||^2_{\{\beta_\mathbf{k}\}_\mathcal{I},W}} \cdot \frac{|||u|||^2_{\{\beta_\mathbf{k}\}_\mathcal{I},W}}{|||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I},V}} \,,$$

we get an upper and lower estimate of the condition number $\kappa^V_\mathcal{I}$ of (5.38) as

$$\max(\kappa^W_\mathcal{I}/\kappa^{WV}_\mathcal{I}, 1) \le \kappa^V_\mathcal{I} \le \kappa^W_\mathcal{I}\kappa^{WV}_\mathcal{I} \,. \tag{5.45}$$

Thus, under the assumption that an orthogonal splitting (5.36) (with a preferably low $\kappa^W_\mathcal{I}$) is available, the minimization of $\kappa^{WV}_\mathcal{I}$ results in tight upper and lower bounds for $\kappa^V_\mathcal{I}$.

In the remainder of this subsection, we express $\kappa^{WV}_\mathcal{I}$ in explicit form as a function of the parameters $\omega^V_\mathbf{l} = \gamma_\mathbf{l}^{-1} \ge 0$, $\mathbf{l} \in \mathcal{I}$, which can then be minimized using an LP. Note that we do not exclude the case $\omega^V_\mathbf{l} = 0$ (formally, this corresponds to $\gamma_\mathbf{l} = \infty$) for which the corresponding subspaces $V_\mathbf{l}$ in (5.42) and thus the corresponding operators $T^V_\mathbf{l}$ in (5.39) are "switched off". As a first step, we express the norm $|||u|||_{\{\gamma_\mathbf{l}\}_\mathcal{I},V}$ in terms of V-orthogonal decompositions into the subspaces $W_\mathbf{l}$.

Lemma 5.3. *For weights $\gamma_\mathbf{l}, \mathbf{l} \in \mathcal{I}$, and $\alpha_\mathbf{k}, \mathbf{k} \in \mathcal{I}$, with*

$$\alpha_\mathbf{k} := \Big(\sum_{\mathbf{l}\ge\mathbf{k}} \gamma_\mathbf{l}^{-1}\Big)^{-1} \quad \textit{for} \quad \mathbf{k} \in \mathcal{I} \,, \tag{5.46}$$

the norm (5.42) *is given by*

$$|||u|||_{\{\gamma_\mathbf{l}\}_\mathcal{I},V} = |||u|||_{\{\alpha_\mathbf{k}\}_\mathcal{I},W} \,. \tag{5.47}$$

Proof. We obtain the result by the following rearrangements

$$\begin{aligned} |||u|||^2_{\{\gamma_\mathbf{l}\}_\mathcal{I},V} &= \inf_{\substack{u_\mathbf{l}\in V_\mathbf{l}, \mathbf{l}\in\mathcal{I} \\ u=\sum_{\mathbf{l}\in\mathcal{I}} u_\mathbf{l}}} \sum\nolimits_{\mathbf{l}\in\mathcal{I}} \gamma_\mathbf{l} \|u_\mathbf{l}\|^2_V \\ &= \inf_{\substack{w_{\mathbf{l},\mathbf{k}}\in W_\mathbf{k}, \mathbf{l}\in\mathcal{I}, \mathbf{k}\le\mathbf{l} \\ u=\sum_{\mathbf{l}\in\mathcal{I},\mathbf{k}\le\mathbf{l}} w_{\mathbf{l},\mathbf{k}}}} \sum\nolimits_{\mathbf{l}\in\mathcal{I}} \gamma_\mathbf{l} \sum_{\mathbf{k}\le\mathbf{l}} \|w_{\mathbf{l},\mathbf{k}}\|^2_V \end{aligned}$$

$$= \inf_{\substack{w_{\mathbf{l},\mathbf{k}}\in W_{\mathbf{k}},\mathbf{k}\in\mathcal{I},\mathbf{l}\geq\mathbf{k} \\ u=\sum_{\mathbf{k}\in\mathcal{I},\mathbf{l}\geq\mathbf{k}} w_{\mathbf{l},\mathbf{k}}}} \sum_{\mathbf{k}\in\mathcal{I}}\sum_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}\|w_{\mathbf{l},\mathbf{k}}\|_V^2 \ ,$$

where we first replaced each $u_{\mathbf{l}}$ by its unique V-orthogonal decomposition $u_{\mathbf{l}} = \sum_{\mathbf{k}\leq\mathbf{l}} w_{\mathbf{l},\mathbf{k}}$ with $w_{\mathbf{l},\mathbf{k}} \in W_{\mathbf{k}}$, and then changed the order of summation. Note that $w_{\mathbf{k}} = \sum_{\mathbf{l}\geq\mathbf{k}} w_{\mathbf{l},\mathbf{k}}$ must hold for all $\mathbf{k} \in \mathcal{I}$. Thus the infimum and the summation over $\mathbf{k}$ commute, and we get

$$\begin{aligned} |||u|||^2_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}},V} &= \sum_{\mathbf{k}\in\mathcal{I}} \inf_{\substack{w_{\mathbf{l},\mathbf{k}}\in W_{\mathbf{l}},\mathbf{l}\geq\mathbf{k} \\ w_{\mathbf{k}}=\sum_{\mathbf{l}\geq\mathbf{k}} w_{\mathbf{l},\mathbf{k}}}} \sum\nolimits_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}\|w_{\mathbf{l},\mathbf{k}}\|_V^2 \\ &= \sum_{\mathbf{k}\in\mathcal{I}} \left(\sum\nolimits_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}^{-1}\right)^{-1} \|w_{\mathbf{k}}\|_V^2 \ . \end{aligned} \tag{5.48}$$

The equality (5.48) follows from solving a quadratic minimization problem, see also [GO95b]. Note that we can conclude that $\alpha_{\mathbf{l}} > 0$ for all $\mathbf{l} \in \mathcal{I}$ if $\gamma_{\mathbf{l}}$ is finite for at least all maximal subspaces $V_{\mathbf{l}}$ in $V_{\mathcal{I}}$ (i.e., those $V_{\mathbf{l}}$ for which $V_{\mathbf{l}} \subset V_{\mathbf{k}} \subset V_{\mathcal{I}}$ implies $\mathbf{k} = \mathbf{l}$). But this must be true for any splitting (5.38), which concludes the proof. □

Lemma 5.3 says that the norm $|||\cdot|||_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}},V}$ is also the norm of a weighted space splitting of $V_{\mathcal{I}}$ into V-orthogonal subspaces $W_{\mathbf{k}}$ with the $\alpha_{\mathbf{k}}$-weights from (5.46). Since W-splittings are direct sum splittings this means we can easily compute the splitting condition number $\kappa_{\mathcal{I}}^{WV}$ between the V-splitting and the W-splitting in (5.44) by directly comparing $\beta_{\mathbf{k}}$ and $\alpha_{\mathbf{k}}$ for $\mathbf{k} \in \mathcal{I}$. The following theorem is therefore an immediate consequence of Lemma 5.3.

Theorem 5.4. *Let $\mathcal{I}$ be an index set, and consider the splittings in* (5.44) *with weights $(\beta_{\mathbf{k}} > 0)_{\mathbf{k}\in\mathcal{I}}$ and $(\gamma_{\mathbf{l}} > 0)_{\mathbf{l}\in\mathcal{I}}$, where $\gamma_{\mathbf{l}} < \infty$ or $\omega_{\mathbf{l}}^V = \gamma_{\mathbf{l}}^{-1} > 0$ for at least all maximal subspaces $V_{\mathbf{l}}$ in $V_{\mathcal{I}}$. Then, the best constants in the norm equivalence*

$$\lambda_{min}|||u|||^2_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}},V} \leq |||u|||^2_{\{\beta_{\mathbf{k}}\}_{\mathcal{I}},W} \leq \lambda_{max}|||u|||^2_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}},V} \ ,$$

valid for any $u = \sum_{\mathbf{k}\in\mathcal{I}} w_{\mathbf{k}} \in V_{\mathcal{I}}$, where $w_{\mathbf{k}} \in W_{\mathbf{k}}$, $\mathbf{k} \in \mathcal{I}$, are given by

$$\lambda_{max} = \max_{\mathbf{k}\in\mathcal{I}} \beta_{\mathbf{k}} \sum\nolimits_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}^{-1} \qquad \text{and} \qquad \lambda_{min} = \min_{\mathbf{k}\in\mathcal{I}} \beta_{\mathbf{k}} \sum\nolimits_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}^{-1} \ .$$

In particular, it holds that

$$\kappa_{\mathcal{I}}^{WV} = \frac{\max_{\mathbf{k}\in\mathcal{I}} \beta_{\mathbf{k}} \sum_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}^{-1}}{\min_{\mathbf{k}\in\mathcal{I}} \beta_{\mathbf{k}} \sum_{\mathbf{l}\geq\mathbf{k}} \gamma_{\mathbf{l}}^{-1}} \ . \tag{5.49}$$

Now, we want to minimize $\kappa_{\mathcal{I}}^{WV}$ with respect to $(\gamma_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}}$. To this end, we write $\kappa_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}^{WV}$ instead of $\kappa_{\mathcal{I}}^{WV}$ to indicate the dependence of the condition number on the set of $\gamma_{\mathbf{l}}$-weights and formulate the problem of finding the optimal weights and scaling parameters

$$(\gamma_{\mathbf{l}}^*)_{\mathbf{l}\in\mathcal{I}} := \arg\min_{(\gamma_{\mathbf{l}}>0)_{\mathbf{l}\in\mathcal{I}}} \kappa_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}^{WV} \quad \text{with} \quad \omega_{\mathbf{l}}^V = (\gamma_{\mathbf{l}})^{-1}, \mathbf{l} \in \mathcal{I}, \tag{5.50}$$

as the LP

$$\begin{aligned} \text{Minimize} \quad & \lambda \\ \text{subject to} \quad & \sum_{\mathbf{l}\geq\mathbf{k}} \omega_{\mathbf{l}}^V \geq \beta_{\mathbf{k}}^{-1} \qquad \forall\, \mathbf{k}\in\mathcal{I} \qquad (5.51) \\ \text{and} \quad & \beta_{\mathbf{k}}^{-1}\lambda - \sum_{\mathbf{l}\geq\mathbf{k}} \omega_{\mathbf{l}}^V \geq 0 \qquad \forall\, \mathbf{k}\in\mathcal{I} \qquad (5.52) \\ \text{and} \quad & \lambda \geq 0,\ \omega_{\mathbf{k}}^V \geq 0 \qquad \forall\, \mathbf{k}\in\mathcal{I}\,, \qquad (5.53) \end{aligned}$$

which can then be solved by one of the common LP algorithms. The vector of unknowns $\big(\lambda, (\omega_{\mathbf{l}}^V)_{\mathbf{l}\in\mathcal{I}}\big)$ of this LP is of size $\#\mathcal{I}+1$. As we will see, the parameter λ represents an upper bound for $\kappa^{WV}_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}$.

Let us check that the above LP is indeed minimizing $\kappa^{WV}_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}$. The inequality constraints (5.51) are equivalent to

$$\beta_{\mathbf{k}}\sum\nolimits_{\mathbf{l}\geq\mathbf{k}}\gamma_{\mathbf{l}}^{-1} \geq 1, \qquad \mathbf{k}\in\mathcal{I}\,,$$

which ensures that for any feasible vector the denominator of (5.49) is at least 1. Moreover, the inequalities (5.51) imply that for any $\mathbf{k}\in\mathcal{I}$ there is at least one $\mathbf{l}\geq\mathbf{k}$ in $\mathcal{I}$ such that $\gamma_{\mathbf{l}}<\infty$, i.e., feasible $\big(\lambda, (\omega_{\mathbf{l}}^V)_{\mathbf{l}\in\mathcal{I}}\big)$ create admissible weight sets such that $V_{\mathcal{I}} = \sum_{\substack{\mathbf{l}\in\mathcal{I}\\ \gamma_{\mathbf{l}}<\infty}} V_{\mathbf{l}}$. The other set of constraints (5.52) can be rewritten as

$$\beta_{\mathbf{k}}\sum\nolimits_{\mathbf{l}\geq\mathbf{k}}\gamma_{\mathbf{l}}^{-1} \leq \lambda\,, \qquad \mathbf{k}\in\mathcal{I}\,,$$

which implies that for any feasible vector we guarantee that $\lambda \geq \kappa^{WV}_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}$ according to (5.49). Altogether, this specifies a LP with non-empty feasibility set, and any optimal solution provides a set of weights $(\gamma_{\mathbf{l}}^*)_{\mathbf{l}\in\mathcal{I}}$ minimizing $\kappa^{WV}_{\{\gamma_{\mathbf{l}}\}_{\mathcal{I}}}$.

Now, we return to the H^t-elliptic model problems given in Section 5.1 and prove upper and lower bounds for the condition number.

5.3.2 Bounds for H^t-elliptic problems

In [GO94], a set of $\gamma_{\mathbf{l}}$-weights for a regular sparse grid space splitting (i.e., (5.38) with $\mathcal{I}=\mathcal{S}_J^d$ from (4.10)) was constructed which resulted in a condition number $\kappa^V_{\mathcal{S}_J^d} = \mathcal{O}(J^{d-2})$ for the special case of H^1-elliptic problems discretized by linear splines. Theorem 5.5 gives a similar result for general H^t-elliptic problems. Note that it is asymptotically sharp, i.e., $\kappa^V_{\mathcal{S}_J^d}$ possesses a matching lower bound.

Theorem 5.5. *Let $d\geq 2$, and consider the discretization of a H^t-elliptic problem with regular sparse grid spaces $V_{\mathcal{S}_J^d}$ based on C^r splines of degree $m\geq r+1$, where $0<t<r+3/2$. We denote the condition number of the splitting* (5.38) *with $\mathcal{I}=\mathcal{S}_J^d$ and the set of weights $(\gamma_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}}$ by $\kappa^V_{\{\gamma_{\mathbf{l}}\}_{\mathcal{S}_J^d}}$. For optimal $\gamma_{\mathbf{l}}$-weights, the condition number $\kappa^V_{\{\gamma_{\mathbf{l}}\}_{\mathcal{S}_J^d}}$ grows as $\Theta(J^{d-2})$ in J, i.e., it holds*

$$\max(cJ^{d-2},1) \leq \inf_{(\gamma_{\mathbf{l}}>0)_{\mathbf{l}\in\mathcal{S}_J^d}} \kappa^V_{\{\gamma_{\mathbf{l}}\}_{\mathcal{S}_J^d}} \leq CJ^{d-2}$$

with constants $0 < c < C < \infty$ *independently of* $J \geq 1$.

Proof. According to (5.45), it is sufficient to prove this result for the optimal value of $\kappa^{WV}_{\{\gamma_\mathbf{l}\}_{\mathcal{S}^d_J}}$ of (5.44), where the underlying weights $\beta_\mathbf{k} = 2^{2t|\mathbf{k}|_\infty}, \mathbf{k} \in \mathcal{S}^d_J$, originate from the well-known norm equivalence (5.4) for H^t. To obtain upper and lower bounds, we use the characterization of the optimal $\kappa^{WV}_{\{\gamma_\mathbf{l}\}_{\mathcal{S}^d_J}}$ via the LP. The value of λ associated with any feasible solution of the LP gives an upper bound for $\kappa^{WV}_{\{\gamma_\mathbf{l}\}_{\mathcal{S}^d_J}}$. We choose the set of weights $(\gamma_\mathbf{l})_{\mathbf{l}\in\mathcal{S}^d_J}$ from [GO94] to define the $\omega^V_\mathbf{l}$ as follows:

$$\omega^V_\mathbf{l} = \begin{cases} 2^{-2t|\mathbf{l}|_\infty}, & |\mathbf{l}|_1 = J\ , \\ 2^{-2tl}, & \mathbf{l} = (l,\ldots,l),\ \ l = 0,\ldots,\lfloor J/d \rfloor\ , \\ 0, & \text{otherwise}\ . \end{cases} \tag{5.54}$$

It is easy to check that, with this choice for the $\omega^V_\mathbf{l}$, the inequalities (5.51) are automatically satisfied since, for any $\mathbf{k} \in \mathcal{S}^d_J$, there is a $\mathbf{l} \in \mathcal{S}^d_J$ with $\mathbf{l} \geq \mathbf{k}$ such that $\omega^V_\mathbf{l} = \beta^{-1}_\mathbf{k}$. Indeed, if $k := |\mathbf{k}|_\infty \leq J/d$, then $\mathbf{l} = (k,\ldots,k) \in \mathcal{S}^d_J$ will do, if $k > J/d$ then we can find a $\mathbf{l} \geq \mathbf{k}$ with $|\mathbf{l}|_\infty = k$, and $|\mathbf{l}|_1 = J$ since $k \leq |\mathbf{k}|_1 \leq J$ and $dk > J$.

The inequalities in (5.52) are satisfied by determining a suitable λ. To this end, we have to bound the maximum of the quantities

$$\lambda_\mathbf{k} := \beta_\mathbf{k} \sum_{\mathbf{l}\in\mathcal{S}^d_J:\, \mathbf{l}\geq\mathbf{k}} \omega^V_\mathbf{l}, \qquad \mathbf{k} \in \mathcal{S}^d_J\ .$$

Since $\beta_\mathbf{k} = 2^{2t|\mathbf{k}|_\infty}$ depends on $k = |\mathbf{k}|_\infty$ only, the maximum of the $\lambda_\mathbf{k}$ with the same value k is obtained for $\mathbf{k} = (k, 0, \ldots, 0)$. Thus, for each $k = 0,\ldots,J$, we must ensure that

$$\lambda \geq \lambda_{(k,0,\ldots,0)} = \sum_{l=k}^{\lfloor J/d \rfloor} 2^{2t(k-l)} + \sum_{l=k}^{J} 2^{2t(k-l)} \sum_{\substack{\mathbf{l}\geq(k,0,\ldots,0) \\ |\mathbf{l}|_1=J,\, |\mathbf{l}|_\infty=l}} 1\ .$$

The first sum stems from the $\omega^V_\mathbf{l} > 0$ with $\mathbf{l} = (l,\ldots,l) \geq (k,0,\ldots,0)$ (if such indices exists in $\mathcal{S}^d_J$), and is uniformly bounded with respect to k and J since $t > 0$. The second sum stems from the remaining non-zero $\omega^V_\mathbf{l}$ with $\mathbf{l} \geq (k,0,\ldots,0)$ and $|\mathbf{l}|_1 = J$, ordered by their value $l := |\mathbf{l}|_\infty \geq k$. For a rough estimate of the counting sum, observe that

$$\{\mathbf{l} \geq (k,0,\ldots,0) :\ |\mathbf{l}|_1 = J,\, |\mathbf{l}|_\infty = l\} \subset \bigcup_{i=1}^{d} \{\mathbf{l} :\ l_i = l,\, \sum_{j\neq i} l_j = J - l\}\ .$$

Thus,

$$\sum_{\substack{\mathbf{l}\geq(k,0,\ldots,0) \\ |\mathbf{l}|_1=J,\, |\mathbf{l}|_\infty=l}} 1 \leq d \cdot \#\{\mathbf{n} \in \mathbb{N}^{d-1} :\ |\mathbf{n}|_1 = J - l\} \leq C' d J^{d-2}$$

with C' independent of l, k, and J. This shows that the first and second sum together, and

thus the maximum of the $\lambda_{\mathbf{k}}$, are bounded by CJ^{d-2}, with a constant C independent of k and J. As a consequence, we can always choose some $\lambda \leq CJ^{(d-2)}$ to arrive at a feasible vector $\big(\lambda, (\omega_{\mathbf{l}}^V)_{\mathbf{l}\in\mathcal{I}}\big)$. This gives the desired upper bound.

A matching lower bound can be found from considering the corresponding dual LP. To formulate it we use two vectors $\mathbf{y} = (y_{\mathbf{k}})_{\mathbf{k}\in\mathcal{S}_J^d}$ and $\mathbf{z} = (z_{\mathbf{k}})_{\mathbf{k}\in\mathcal{S}_J^d}$ associated to (5.51) and (5.52), respectively. The dual problem then reads

$$\text{Maximize} \qquad \sum_{\mathbf{k}\in\mathcal{S}_J^d} \beta_{\mathbf{k}}^{-1} y_{\mathbf{k}} \tag{5.55}$$

$$\text{subject to} \qquad \sum_{\mathbf{k}\in\mathcal{S}_J^d} \beta_{\mathbf{k}}^{-1} z_{\mathbf{k}} \leq 1 \tag{5.56}$$

$$\text{and} \qquad \sum_{\mathbf{k}\leq\mathbf{l}} y_{\mathbf{k}} \leq \sum_{\mathbf{k}\leq\mathbf{l}} z_{\mathbf{k}} \qquad \forall\, \mathbf{l}\in\mathcal{S}_J^d \tag{5.57}$$

$$\text{and} \qquad \mathbf{y}, \mathbf{z} \geq 0\ . \tag{5.58}$$

All we have to do is to find a feasible pair of vectors for this dual LP and to evaluate the target function on it. To this end, fix the smallest integer $k \geq J/2$ and set

$$z_{\mathbf{k}} = \begin{cases} 2^{2tk}, & \mathbf{k} = (k, 0, \ldots)\ , \\ 0, & \text{otherwise}\ , \end{cases} \qquad y_{\mathbf{k}} = \begin{cases} 2^{2tk}, & \mathbf{k} = (k, k_2, \ldots, k_d),\ |\mathbf{k}|_1 = J\ , \\ 0, & \text{otherwise}\ . \end{cases}$$

These vectors trivially fulfil (5.56) and (5.58). For (5.57) observe that $\sum_{\mathbf{k}\leq\mathbf{l}} y_{\mathbf{k}}$ contains exactly one non-zero term (namely $y_{\mathbf{k}} = 2^{2tk}$ if $\mathbf{k} = (k, k_2, \ldots, k_d)$ satisfies $|\mathbf{k}|_1 = J$ and equals $\mathbf{l}$). Since $(k, 0, \ldots, 0) \leq \mathbf{l}$ for such $\mathbf{l}$, we have

$$2^{2tk} = \sum_{\mathbf{k}\leq\mathbf{l}} y_{\mathbf{k}} = z_{(k,0,\ldots,0)} = \sum_{\mathbf{l}\leq\mathbf{k}} z_{\mathbf{l}}\ .$$

For all other $\mathbf{l}$, the inequality (5.57) is automatically valid since $\sum_{\mathbf{k}\leq\mathbf{l}} y_{\mathbf{k}} = 0$.

Now, the value of the target functional for this feasible pair of vectors is

$$\sum_{\mathbf{k}\in\mathcal{S}_J^d} \beta_{\mathbf{k}}^{-1} y_{\mathbf{k}} = \sum_{k_2+\ldots+k_d=J-k} 1 \geq c'(J-k)^{d-2} \geq cJ^{d-2}\ ,$$

where we have used the fact that, due to $k \geq J/2$ for all $\mathbf{k}$ with $y_{\mathbf{k}} = 2^{2tk} > 0$, we have $|\mathbf{k}|_\infty = k$, and thus $\beta_{\mathbf{k}}^{-1} y_{\mathbf{k}} = 1$. This proves the lower bound. □

For standard H^t-elliptic problems ($t > 0$) in dimensions $d \geq 3$, Theorem 5.5 shows that we cannot expect an optimal preconditioner derived from (5.38) using positive weights only. We investigate the possibility of negative scaling factors $\omega_{\mathbf{l}}^V$ in the following Section.

5.4 Unconstrained scalings for sparse grid subspace correction methods

In the last section, we have seen that for sparse grid discretizations of standard H^t-elliptic problems ($t > 0$) in dimensions $d \geq 3$, we cannot expect optimal convergence of the subspace correction method (5.39) using positive scaling factors $\omega_{\mathbf{l}}^V, \mathbf{l} \in \mathcal{I}$, only. In this section, we see that there is an algebraic transformation that allows us to obtain optimal convergence rates, i.e., the same rates we would expect from a W-splitting based method (5.37) given the norm equivalence (5.4). To this end, we have to circumvent the subspace splitting theory since partially negative scaling factors occur. We explicitly state for which sets of positive *and* negative scaling factors the resulting subspace splitting operator remains positive definite. First, we need the V-orthogonal projectors $Q_{\mathbf{l}}^V : V_{\mathcal{I}} \to V_{\mathbf{l}}$ and $Q_{\mathbf{k}}^W : V_{\mathcal{I}} \to W_{\mathbf{k}}$ with

$$\begin{aligned} (Q_{\mathbf{l}}^V u, v_{\mathbf{l}})_V &= (u, v_{\mathbf{l}})_V \qquad \forall\, v_{\mathbf{l}} \in V_{\mathbf{l}} \quad \text{and} \\ (Q_{\mathbf{k}}^W u, w_{\mathbf{k}})_V &= (u, w_{\mathbf{k}})_V \qquad \forall\, w_{\mathbf{k}} \in W_{\mathbf{k}}\,, \end{aligned}$$

respectively. The following simple lemma proves helpful for future considerations.

Lemma 5.6. *For $\mathbf{k} \leq \mathbf{l}$, we have $Q_{\mathbf{k}}^V T_{\mathbf{l}}^V = T_{\mathbf{k}}^V$ and $Q_{\mathbf{k}}^W T_{\mathbf{l}}^V = T_{\mathbf{k}}^W$.*

Proof. This immediately follows from

$$(Q_{\mathbf{k}}^V T_{\mathbf{l}}^V u, v_{\mathbf{k}})_V = (T_{\mathbf{l}}^V u, v_{\mathbf{k}})_V = a(u, v_{\mathbf{k}}) = (T_{\mathbf{k}}^V u, v_{\mathbf{k}})_V \qquad \forall\, v_{\mathbf{k}} \in V_{\mathbf{k}}\,,$$

and

$$(Q_{\mathbf{k}}^W T_{\mathbf{l}}^V u, w_{\mathbf{k}})_V = (T_{\mathbf{l}}^V u, w_{\mathbf{k}})_V = a(u, w_{\mathbf{k}}) = (T_{\mathbf{k}}^W u, w_{\mathbf{k}})_V \qquad \forall\, w_{\mathbf{k}} \in W_{\mathbf{k}}\,.$$

□

The following theorem gives a formula for rewriting the subspace correction scheme based on a W-splitting (5.37) as a subspace correction method based on a V-splitting (5.39).

Theorem 5.7. *We have*

$$\sum_{\mathbf{k} \in \mathcal{I}} \omega_{\mathbf{k}}^W T_{\mathbf{k}}^W = \sum_{\mathbf{l} \in \mathcal{I}} \omega_{\mathbf{l}}^V T_{\mathbf{l}}^V$$

if

$$\omega_{\mathbf{l}}^V = \sum_{\substack{\mathbf{e} \in \{0,1\}^d \\ \mathbf{l}+\mathbf{e} \in \mathcal{I}}} (-1)^{|\mathbf{e}|_1} \omega_{\mathbf{l}+\mathbf{e}}^W\,. \tag{5.59}$$

Proof. From the classical combination technique [GSZ92, BGRZ94, BGR94, HGC07] for projections based on tensor scalar products, we know that

$$Q_{\mathbf{k}}^W = \sum_{\substack{\mathbf{e} \in \{0,1\}^d \\ \mathbf{k}-\mathbf{e} \in \mathcal{I}}} (-1)^{|\mathbf{e}|_1} Q_{\mathbf{k}-\mathbf{e}}\,. \tag{5.60}$$

This carries over to the subspaces solvers by using Lemma 5.6 twice, i.e., we have

$$T_{\mathbf{k}}^W = Q_{\mathbf{k}}^W T_{\mathbf{k}}^V = \sum_{\substack{\mathbf{e}\in\{0,1\}^d \\ \mathbf{k}-\mathbf{e}\in\mathcal{I}}} (-1)^{|\mathbf{e}|_1} Q_{\mathbf{k}-\mathbf{e}}^V T_{\mathbf{k}}^V = \sum_{\substack{\mathbf{e}\in\{0,1\}^d \\ \mathbf{k}-\mathbf{e}\in\mathcal{I}}} (-1)^{|\mathbf{e}|_1} T_{\mathbf{k}-\mathbf{e}}^V \,.$$

Then we can easily write

$$\begin{aligned} \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^W T_{\mathbf{k}}^W &= \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^W \sum_{\substack{\mathbf{e}\in\{0,1\}^d \\ \mathbf{k}-\mathbf{e}\in\mathcal{I}}} (-1)^{|\mathbf{e}|_1} T_{\mathbf{k}-\mathbf{e}}^V \\ &= \sum_{\mathbf{l}\in\mathcal{I}} \Big(\sum_{\substack{\mathbf{e}\in\{0,1\}^d \\ \mathbf{l}+\mathbf{e}\in\mathcal{I}}} (-1)^{|\mathbf{e}|_1} \omega_{\mathbf{l}+\mathbf{e}}^W \Big) T_{\mathbf{l}}^V \,. \end{aligned}$$

□

Theorem 5.7 shows that our W-splitting based Schwarz operator can be expressed using the operators $T_{\mathbf{l}}^V$ only. A similar idea was already used in [BPX90] for the full grid case, i.e., when the underlying space splitting is based on spaces $V_l := V_{(l,\ldots,l)}$. Often, this case is benign because for scaling parameters $\omega_l^W = 2^{-2tl}$, typical for H^t-elliptic problems with $t > 0$, we have

$$\omega_l^V = \omega_l^W - \omega_{l+1}^W = 2^{-2lt} - 2^{-2t(l+1)} = 2^{-2l}(1 - 2^{-2t}) \simeq \omega_l^W \,,$$

which means that we do not need to form orthogonal complements at all, neither implicitly nor explicitly. In our sparse grid case, the differences involve however 2^d terms, and we generally cannot expect $\omega_{\mathbf{l}}^W \simeq \omega_{\mathbf{l}}^V$ to hold uniformly in $\mathbf{l}$. Moreover, negative $\omega_{\mathbf{l}}^V$ are possible.

If we set $\omega_{\mathbf{k}}^W = 1$ for $\mathbf{k} \in \mathcal{I}$, Theorem 5.7 yields the standard combination technique [GSZ92, BGRZ94, BGR94], a popular method for approximating sparse grid solutions of, e.g. partial differential equations. Our case differs in the respect that our subspace solvers are based on the auxiliary bilinear forms $(\cdot,\cdot)_V$ instead of $a(\cdot,\cdot)$. Furthermore, we have $\omega_{\mathbf{k}}^W = \beta_{\mathbf{k}}^{-1}$ with $\beta_{\mathbf{k}}$ that stem from the norm equivalence (5.4). Finally, we do not stop after one iteration step but converge into the true sparse grid solution.

As already mentioned in Subsection 5.2.2, it is crucial that the operator $P_{\boldsymbol{\omega}}$ is positive definite. This is guaranteed for positive weights, but Theorem 5.7 suggests that even partially negative weights may result in a positive definite operator. We characterize these $\boldsymbol{\omega} = (\omega_{\mathbf{l}}^V)_{\mathbf{l}\in\mathcal{I}}$ by the following theorem.

Theorem 5.8. *The operator $P_{\boldsymbol{\omega}} = \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^V T_{\mathbf{k}}^V$ is positive definite if and only if*

$$\sum_{\mathbf{k}\geq\mathbf{j}} \omega_{\mathbf{k}}^V > 0 \qquad \forall\, \mathbf{j} \in \mathcal{I} \,. \tag{5.61}$$

Proof. With Lemma 5.6, we get

$$P_{\boldsymbol{\omega}} = \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^V T_{\mathbf{k}}^V = \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^V \sum_{\mathbf{l}\leq\mathbf{k}} Q_{\mathbf{l}}^W T_{\mathbf{k}}^V = \sum_{\mathbf{k}\in\mathcal{I}} \omega_{\mathbf{k}}^V \sum_{\mathbf{l}\leq\mathbf{k}} T_{\mathbf{l}}^W = \sum_{\mathbf{l}\in\mathcal{I}} \Big(\sum_{\mathbf{k}\geq\mathbf{l}} \omega_{\mathbf{k}}^V \Big) T_{\mathbf{l}}^W$$

and thus

$$a(P_{\boldsymbol{\omega}}u,u)=\sum_{\mathbf{l}\in\mathcal{I}}\Big(\sum_{\mathbf{k}\geq\mathbf{l}}\omega_{\mathbf{k}}^V\Big)a(T_{\mathbf{l}}^W u,u)=\sum_{\mathbf{l}\in\mathcal{I}}\Big(\sum_{\mathbf{k}\geq\mathbf{l}}\omega_{\mathbf{k}}^V\Big)(T_{\mathbf{l}}^W u,T_{\mathbf{l}}^W u)_V\ .$$

Now, if the scaling factors $w_{\mathbf{l}}^V$ satisfy (5.61) and if $u\neq 0$, it follows directly that $a(P_{\boldsymbol{\omega}}u,u)>0$. Otherwise, if the condition (5.61) is violated for a $\mathbf{j}\in\mathcal{I}$, pick a $z_{\mathbf{j}}\in W_{\mathbf{j}}, z_{\mathbf{j}}\neq 0$ and compute $u\in W_{\mathbf{j}}\subset V_{\mathcal{I}}$ by

$$a(u,w_{\mathbf{j}})=(z_{\mathbf{j}},w_{\mathbf{j}})_V\qquad\forall\, w_{\mathbf{j}}\in W_{\mathbf{j}}\ .$$

Then $T_{\mathbf{k}}^W u=\delta_{\mathbf{kj}}w_{\mathbf{j}}$ and $a(P_{\boldsymbol{\omega}}u,u)=\Big(\sum_{\mathbf{k}\geq\mathbf{j}}\omega_{\mathbf{k}}^V\Big)(z_{\mathbf{j}},z_{\mathbf{j}})_V\leq 0$, which concludes the proof. □

With Theorem 5.8 we see that the positivity constraints (5.53) of our Linear Program from Subsection 5.3 are too restrictive and can be dropped. This is because the requirement (5.61) is already covered by (5.51). However, positive weights are an integral part of the derivation using the subspace splitting theory. Nonetheless, note that the set of scaling factors $(\omega_{\mathbf{l}}^V)_{\mathbf{l}\in\mathcal{I}}$ proposed in (5.59) for $\omega_{\mathbf{k}}^W:=\beta_{\mathbf{k}}^{-1},\mathbf{k}\in\mathcal{I}$, would be a solution to the Linear Program from Section 5.3 with $\lambda=1$ without the positivity constraints (5.53).

5.5 An orthogonal projection based preconditioner

In the previous section, we have learned that negative scaling factors can be used to effectively implement W-splitting based subspace correction methods using V-spaces only, and still lead to positive definite operators $P_{\boldsymbol{\omega}}$. However, the corresponding scaling matrix in the generating system representation is not positive definite and and cannot be symmetrized. In this section, we present a preconditioner from [GH14b] that follows a similar idea as in Section 5.4 but is positive definite in the generating system representation as well. We do not rely on the theory of subspace splittings anymore but give a self-contained linear algebra view on the topic.

We assume a norm equivalence (5.4) with norm equivalence constants $\lambda_{\min}$ and $\lambda_{\max}$. Note that for $u=\sum_{\mathbf{k}\in\mathcal{I}}w_{\mathbf{k}}\in V_{\mathcal{I}}\subset H$ with $w_{\mathbf{k}}\in W_{\mathbf{k}},\mathbf{k}\in\mathcal{I}$, it immediately follows that

$$\|u\|_a^2\simeq\sum_{\mathbf{k}\in\mathcal{I}}\beta_{\mathbf{k}}\|w_{\mathbf{k}}\|_V^2\ ,\tag{5.62}$$

with norm equivalence constants

$$\lambda_{\min}^{\mathcal{I}}:=\inf_{0\neq u\in V_{\mathcal{I}}}\frac{\|u\|_a^2}{\sum_{\mathbf{k}\in\mathbb{N}^d}\beta_{\mathbf{k}}\|w_{\mathbf{k}}\|_V^2}\geq\lambda_{\min}\quad\text{and}\quad\lambda_{\max}^{\mathcal{I}}:=\sup_{0\neq u\in V_{\mathcal{I}}}\frac{\|u\|_a^2}{\sum_{\mathbf{k}\in\mathbb{N}^d}\beta_{\mathbf{k}}\|w_{\mathbf{k}}\|_V^2}\leq\lambda_{\max}$$

and the inequality $\kappa_{\mathcal{I}}:=\lambda_{\max}^{\mathcal{I}}/\lambda_{\min}^{\mathcal{I}}\leq\kappa=\lambda_{\max}/\lambda_{\min}$ holds independently of $\mathcal{I}$. In the next subsection, we propose a matrix $\mathbf{C}_{\mathcal{I}}$ that can be applied cheaply to a vector and acts as a preconditioner on the system

$$\mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}=\mathbf{b}_{\mathcal{I}}\tag{5.63}$$

from (4.22). Of course, $\mathbf{C}_{\mathcal{I}}\mathbf{A}_{\mathcal{I}}$ still has a non-trivial null space due to the redundancy of the

generalized sparse grid generating system (4.21), but the generalized condition number satisfies

$$\tilde{\kappa}(\mathbf{C}_{\mathcal{I}}\mathbf{A}_{\mathcal{I}}) = \lambda_{\max}^{\mathcal{I}}/\lambda_{\min}^{\mathcal{I}} = \kappa_{\mathcal{I}} \ , \tag{5.64}$$

where $\lambda_{\max}^{\mathcal{I}}/\lambda_{\min}^{\mathcal{I}}$ are the norm equivalence constants from (5.62). Note that if $\mathcal{I} = \mathcal{F}_J^d$, this also results in a full grid preconditioner $\mathbf{C_J}$ on $\mathbf{A_J}$ by

$$\tilde{\kappa}(\mathbf{C}_{\mathcal{F}_J^d}\mathbf{A}_{\mathcal{F}_J^d}) = \tilde{\kappa}(\mathbf{C}_{\mathcal{F}_J^d}\mathbf{S}_{\mathcal{F}_J^d}^T\mathbf{A_J}\mathbf{S}_{\mathcal{F}_J^d}) = \kappa(\mathbf{S}_{\mathcal{F}_J^d}\mathbf{C}_{\mathcal{F}_J^d}\mathbf{S}_{\mathcal{F}_J^d}^T\mathbf{A_J}) \ ,$$

and we can deduce that $\mathbf{C_J} := \mathbf{S}_{\mathcal{F}_J^d}\mathbf{C}_{\mathcal{F}_J^d}\mathbf{S}_{\mathcal{F}_J^d}^T$. For general index sets $\mathcal{I}$, we do not achieve such a non-redundant representation, but semi-convergent iterative methods [Kaa88, BP94, Gri94b] are still applicable to the preconditioned system (5.63).

5.5.1 Construction

The norm equivalence (5.62) holds for orthogonal subspaces $W_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$. In order to make this result available to our discretization, which is based on the subspaces $V_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$, we can use an explicit orthogonalization operator.

Orthogonalization Operator

We now consider the whole multivariate sequence of subspaces $V_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$, which we denote as

$$\widehat{V}_{\mathcal{I}} = (V_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}} \ .$$

For $\widehat{u} = (u_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}}, \widehat{v} = (v_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}} \in \widehat{V}_{\mathcal{I}}$, we define the scalar product

$$(\widehat{u}, \widehat{v})_{\widehat{V}_{\mathcal{I}}} = \sum_{\mathbf{l}\in\mathcal{I}} (u_{\mathbf{l}}, v_{\mathbf{l}})_V \ .$$

Then, we define the operator $F_{\mathcal{I}} : \widehat{V}_{\mathcal{I}} \to \widehat{V}_{\mathcal{I}}$ by

$$F_{\mathcal{I}}\widehat{u} = (Q_{\mathbf{l}}^W u_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}} \ ,$$

where $Q_{\mathbf{l}}^W$ is again the V-orthogonal projection into $W_{\mathbf{l}}$. Note that we can rewrite (5.60) as

$$Q_{\mathbf{l}}^W = (Q_{V_{l_1}}^{(1)} - Q_{V_{l_1-1}}^{(1)}) \otimes \cdots \otimes (Q_{V_{l_d}}^{(d)} - Q_{V_{l_d-1}}^{(d)}) \ , \tag{5.65}$$

where the $Q_{V_l}^{(p)}, p = 1, \ldots, d$, denote the one-dimensional $V^{(p)}$-projections into the spaces $V_l^{(p)}$, see (4.3), for $l \in \mathbb{N}$ and $Q_{V_l}^{(p)} = 0$ for $l = -1$.

The operator $F_{\mathcal{I}}$ can be given in block-diagonal matrix form as $\mathbf{F}_{\mathcal{I}} : \mathbb{R}^{N_{\mathcal{I}}\times N_{\mathcal{I}}}$ with blocks $(\mathbf{F}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_{\mathbf{l}}\times n_{\mathbf{k}}}$ and

$$(\mathbf{F}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} = \begin{cases} \mathbf{Q}_{\mathbf{l}}^W & \text{for} \quad \mathbf{l} = \mathbf{k} \ , \\ 0 & \text{else} \end{cases} \tag{5.66}$$

for all $\mathbf{l}, \mathbf{k} \in \mathcal{I}$, where $\mathbf{Q}_{\mathbf{l}}^W \in \mathbb{R}^{n_{\mathbf{l}}\times n_{\mathbf{l}}}$ is the matrix representation of the operator $Q_{\mathbf{l}}^W$ restricted

to the subspace $V_{\mathbf{l}}$. Considering (5.65), the matrices $\mathbf{Q}_{\mathbf{l}}^W$ can be expressed by

$$\begin{aligned}\mathbf{Q}_{W_{\mathbf{l}}} =&(\mathbf{I}^{(1)}_{l_1,l_1} - \mathbf{I}^{(1)}_{l_1,l_1-1}(\mathbf{M}^{(1)}_{l_1-1})^{-1}\mathbf{I}^{(1)}{}_{l_1-1,l_1}\mathbf{M}^{(1)}_{l_1}) \otimes \dots \\ &\cdots \otimes (\mathbf{I}^{(d)}_{l_d,l_d} - \mathbf{I}^{(d)}_{l_d,l_d-1}(\mathbf{M}^{(d)}_{l_d-1})^{-1}\mathbf{I}^{(d)}_{l_d-1,l_d}\mathbf{M}^{(d)}_{l_d})\end{aligned} \tag{5.67}$$

where $\mathbf{M}_l^{(p)}, p = 1, \dots, d, l \in \mathbb{N}$, are the non-hierarchical isotropic mass matrices

$$(\mathbf{M}_l^{(p)})_{ij} = (\phi_{l,i}, \phi_{l,j})_{V^{(p)}} \ .$$

In (5.67), besides the simple two-level restrictions and prolongations, d applications of one-dimensional mass matrices and d applications of the inverse of one-dimensional mass matrices are employed. In the practically relevant cases of $V^{(p)} = L_2(\Omega)$ and polynomial splines, both operations can be executed cheaply since only band matrices are involved here.

Note that the matrix $\mathbf{F}_{\mathcal{I}}$ is block-diagonal but not symmetric since its blocks on the diagonal are not symmetric. This is remarkable as the corresponding operator $F_{\mathcal{I}} : \widehat{V}_{\mathcal{I}} \to \widehat{V}_{\mathcal{I}}$ is self-adjoint. In fact, the non-symmetry is a property of the matrix representation only.

For our preconditioner, we also need to apply $\mathbf{F}_{\mathcal{I}}^T$ efficiently. To obtain a favorable representation of $\mathbf{F}_{\mathcal{I}}^T$, we first consider the mapping

$$Z_{\mathcal{I}} : \mathbb{R}^{N_{\mathcal{I}}} \to \widehat{V}_{\mathcal{I}}$$

that maps a block-structured vector $\mathbf{x}_{\mathcal{I}} = (x_{\mathbf{l},\mathbf{i}})_{\mathbf{i}\in\chi_{\mathbf{l}},\mathbf{l}\in\mathcal{I}}$ of the enlarged generating system to the sequence of subspaces by

$$Z_{\mathcal{I}} : \mathbf{x}_{\mathcal{I}} \mapsto \Big(\sum_{\mathbf{i}\in\chi_{\mathbf{l}}} x_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}} \Big)_{\mathbf{l}\in\mathcal{I}} \ . \tag{5.68}$$

Note that $F_{\mathcal{I}}$ and $\mathbf{F}_{\mathcal{I}}$ are linked by $\mathbf{F}_{\mathcal{I}} = Z_{\mathcal{I}}^{-1} F_{\mathcal{I}} Z_{\mathcal{I}}$.

Lemma 5.9. *The adjoint $Z_{\mathcal{I}}^* : \widehat{V}_{\mathcal{I}} \to \mathbb{R}^{N_{\mathcal{I}}}$ of (5.68) is given by*

$$Z_{\mathcal{I}}^* : \widehat{u} \mapsto \mathbf{x}_{\mathcal{I}} \quad \textit{with} \quad \mathbf{x}_{\mathcal{I}} = ((u_{\mathbf{l}}, \phi_{\mathbf{l},\mathbf{i}})_V)_{\mathbf{i}\in\chi_{\mathbf{l}},\mathbf{l}\in\mathcal{I}} \quad \textit{for} \quad \widehat{u} = (u_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}} \ .$$

Proof. For any $\widehat{v} = (v_{\mathbf{l}})_{\mathbf{l}\in\mathcal{I}} \in \widehat{V}_{\mathcal{I}}$ and $\mathbf{x}_{\mathcal{I}} \in \mathbb{R}^{N_{\mathcal{I}}}$, we have

$$(Z_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}, \widehat{v})_{V_{\mathcal{I}}} = \sum_{\mathbf{l}\in\mathcal{I}} \Big(\sum_{\mathbf{i}\in\chi_{\mathbf{l}}} x_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}, v_{\mathbf{l}} \Big)_V = \sum_{\mathbf{l}\in\mathcal{I}} \sum_{\mathbf{i}\in\chi_{\mathbf{l}}} x_{\mathbf{l},\mathbf{i}}(v_{\mathbf{l}}, \phi_{\mathbf{l},\mathbf{i}})_V = (\mathbf{x}_{\mathcal{I}}, Z_{\mathcal{I}}^*\widehat{v})_{\ell^2} \ .$$

□

Now, having $Z_{\mathcal{I}}$ and $Z_{\mathcal{I}}^*$, we are able to give a computationally efficient representation of $\mathbf{F}_{\mathcal{I}}^T$.

Lemma 5.10. *It holds that*

$$\mathbf{F}_{\mathcal{I}}^T = \mathbf{G}_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}\mathbf{G}_{\mathcal{I}}^{-1} \ ,$$

where $\mathbf{G}_\mathcal{I} : \mathbb{R}^{N_\mathcal{I} \times N_\mathcal{I}}$ *is a block-diagonal matrix with blocks* $(\mathbf{G}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l} \times n_\mathbf{k}}$ *and*

$$(\mathbf{G}_\mathcal{I})_{\mathbf{l},\mathbf{k}} = \begin{cases} \mathbf{M}_\mathbf{l} & \text{for} \quad \mathbf{l} = \mathbf{k} \,, \\ 0 & \text{else} \end{cases}$$

for all $\mathbf{l}, \mathbf{k} \in \mathcal{I}$ *with the mass matrices* $\mathbf{M}_\mathbf{l} = \bigotimes_{p=1}^d \mathbf{M}_{l_p}^{(p)}$.

Proof. It holds that

$$(Z_\mathcal{I}^* Z_\mathcal{I} \mathbf{x}_\mathcal{I}, \widehat{\mathbf{y}}_{d,J})_{\ell^2} = (Z_\mathcal{I} \mathbf{x}_\mathcal{I}, Z_\mathcal{I} \mathbf{y}_\mathcal{I})_{V_\mathcal{I}} = \sum_{\mathbf{l} \in \mathcal{I}} \sum_{\mathbf{i},\mathbf{j} \in \chi_\mathbf{l}} x_{\mathbf{l},\mathbf{i}} (\phi_{\mathbf{l},\mathbf{i}}, \phi_{\mathbf{l},\mathbf{j}})_V y_{\mathbf{l},\mathbf{j}} = {\mathbf{x}_\mathcal{I}}^T \mathbf{G}_\mathcal{I} \mathbf{y}_\mathcal{I} \,,$$

and thus $Z_\mathcal{I}^* Z_\mathcal{I} = \mathbf{G}_\mathcal{I}$. Then, we can infer

$$(\mathbf{F}_\mathcal{I})^T = (Z_\mathcal{I}^{-1} F_\mathcal{I} Z_\mathcal{I})^* = Z_\mathcal{I}^* F_\mathcal{I} (Z_\mathcal{I}^{-1})^* = \mathbf{G}_\mathcal{I} (Z_\mathcal{I})^{-1} F_\mathcal{I} Z_\mathcal{I} \mathbf{G}_\mathcal{I}^{-1} = \mathbf{G}_\mathcal{I} \mathbf{F}_\mathcal{I} \mathbf{G}_\mathcal{I}^{-1} \,.$$

□

Note furthermore that the operator $F_\mathcal{I}$ is a projection, i.e. $F_\mathcal{I} F_\mathcal{I} = F_\mathcal{I}$. The same is true for $\mathbf{F}_\mathcal{I}$ since

$$\mathbf{F}_\mathcal{I} \mathbf{F}_\mathcal{I} = (Z_\mathcal{I})^{-1} F_\mathcal{I} Z_\mathcal{I} (Z_\mathcal{I})^{-1} F_\mathcal{I} Z_\mathcal{I} = (Z_\mathcal{I})^{-1} F_\mathcal{I} F_\mathcal{I} Z_\mathcal{I} = (Z_\mathcal{I})^{-1} F_\mathcal{I} Z_\mathcal{I} = \mathbf{F}_\mathcal{I} \,.$$

Finally, we need the following Lemma.

Lemma 5.11. *For a block-diagonal scaling matrix* $\mathbf{D}_\mathcal{I} \in \mathbb{R}^{N_\mathcal{I} \times N_\mathcal{I}}$ *with blocks* $(\mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l} \times n_\mathbf{k}}$ *for* $\mathbf{l}, \mathbf{k} \in \mathcal{I}$ *and*

$$(\mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{k}} = \begin{cases} c_\mathbf{l} \mathbf{I}_{\mathbf{l},\mathbf{l}} & \text{for} \quad \mathbf{l} = \mathbf{k} \,, \\ 0 & \text{else} \,, \end{cases}$$

the matrix $\mathbf{D}_\mathcal{I}$ *commutes with any other block-diagonal matrix* $\mathbf{B}_\mathcal{I} \in \mathbb{R}^{N_\mathcal{I} \times N_\mathcal{I}}$, *i.e. a block-structured matrix with blocks* $(\mathbf{B}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l} \times n_\mathbf{k}}$ *for* $\mathbf{l}, \mathbf{k} \in \mathcal{I}$, *where*

$$(\mathbf{B}_\mathcal{I})_{\mathbf{l},\mathbf{k}} = \begin{cases} \mathbf{B}_\mathbf{k} & \text{for} \quad \mathbf{l} = \mathbf{k} \,, \\ 0 & \text{else} \,, \end{cases}$$

and $\mathbf{B}_\mathbf{k} \in \mathbb{R}^{n_\mathbf{k} \times n_\mathbf{k}}$ *are general matrices.*

Proof. For ease of notation, we use Kronecker's δ in this short proof. It holds that

$$\begin{aligned} (\mathbf{D}_\mathcal{I} \mathbf{B}_\mathcal{I})_{\mathbf{l},\mathbf{k}} &= \sum_{\mathbf{m} \in \mathcal{I}} (\mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{m}} (\mathbf{B}_\mathcal{I})_{\mathbf{m},\mathbf{k}} = \sum_{\mathbf{m} \in \mathcal{I}} \delta_{\mathbf{l},\mathbf{m}} c_\mathbf{l} \delta_{\mathbf{m},\mathbf{k}} \mathbf{B}_\mathbf{k} = \delta_{\mathbf{l},\mathbf{k}} c_\mathbf{l} \mathbf{B}_\mathbf{k} \\ &= \delta_{\mathbf{l},\mathbf{k}} \mathbf{B}_\mathbf{l} c_\mathbf{k} = \sum_{\mathbf{m} \in \mathcal{I}} \delta_{\mathbf{l},\mathbf{m}} \mathbf{B}_\mathbf{m} \delta_{\mathbf{m},\mathbf{k}} c_\mathbf{m} = \sum_{\mathbf{m} \in \mathcal{I}} (\mathbf{B}_\mathcal{I})_{\mathbf{l},\mathbf{m}} (\mathbf{D}_\mathcal{I})_{\mathbf{m},\mathbf{k}} = (\mathbf{B}_\mathcal{I} \mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \,, \end{aligned}$$

and thus $\mathbf{D}_\mathcal{I} \mathbf{B}_\mathcal{I} = \mathbf{B}_\mathcal{I} \mathbf{D}_\mathcal{I}$. □

Obviously, Lemma 5.11 can be applied to, e.g., $\mathbf{B}_\mathcal{I} = \mathbf{F}_\mathcal{I}$ or $\mathbf{B}_\mathcal{I} = \mathbf{G}_\mathcal{I}$.

Preconditioner

Now we present the projection based preconditioner for the operator matrix $\mathbf{A}_\mathcal{I}$. To this end, the most important ingredient is the norm equivalence (5.4) with weights $(\beta_\mathbf{l})_{\mathbf{l}\in\mathcal{I}}$.

Theorem 5.12. *Let $\mathbf{D}_\mathcal{I} \in \mathbb{R}^{N_\mathcal{I}\times N_\mathcal{I}}$ be a diagonal block-structured scaling matrix with blocks $(\mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{n_\mathbf{l}\times n_\mathbf{k}}$ and*

$$(\mathbf{D}_\mathcal{I})_{\mathbf{l},\mathbf{k}} = \begin{cases} \beta_\mathbf{l}\mathbf{I}_{\mathbf{l},\mathbf{l}} & \text{for} \quad \mathbf{l}=\mathbf{k}\,, \\ 0 & \text{else} \end{cases}$$

for all $\mathbf{l},\mathbf{k}\in\mathcal{I}$. Then, the generalized condition number of the symmetric matrix

$$\mathbf{L}_\mathcal{I}^{-1}\mathbf{F}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{-1/2}\mathbf{A}_\mathcal{I}\mathbf{D}_\mathcal{I}^{-1/2}\mathbf{F}_\mathcal{I}\mathbf{L}_\mathcal{I}^{-T} \tag{5.69}$$

is $\kappa_\mathcal{I} = \lambda_{max}^\mathcal{I}/\lambda_{min}^\mathcal{I}$, where $\lambda_{min}^\mathcal{I}$ and $\lambda_{min}^\mathcal{I}$ are the norm equivalence constants of (5.62) and $\mathbf{L}_\mathcal{I}$ denotes the Cholesky factor of $\mathbf{G}_\mathcal{I}$, i.e. $\mathbf{G}_\mathcal{I} = \mathbf{L}_\mathcal{I}\mathbf{L}_\mathcal{I}^T$.

Proof. For any vector $\mathbf{x}_\mathcal{I} \in \mathrm{im}(\mathbf{F}_\mathcal{I}) \subset \mathbb{R}^{N_\mathcal{I}}$, we have

$$\begin{aligned}
\mathbf{x}_\mathcal{I}{}^T\mathbf{F}_\mathcal{I}^T\mathbf{A}_\mathcal{I}\mathbf{F}_\mathcal{I}\mathbf{x}_\mathcal{I} &= \mathbf{x}_\mathcal{I}{}^T\mathbf{A}_\mathcal{I}\mathbf{x}_\mathcal{I} && (5.70)\\
&= a\Big(\sum_{\mathbf{l}\in\mathcal{I}}\sum_{\mathbf{i}\in\chi_\mathbf{l}} x_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}, \sum_{\mathbf{l}\in\mathcal{I}}\sum_{\mathbf{i}\in\chi_\mathbf{l}} x_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}\Big) \\
&\simeq \sum_{\mathbf{l}\in\mathcal{I}} \beta_\mathbf{l}\Big\|\sum_{\mathbf{i}\in\chi_\mathbf{l}} x_{\mathbf{l},\mathbf{i}}\phi_{\mathbf{l},\mathbf{i}}\Big\|^2_{L_2(\Omega^d)} && (5.71)\\
&= \sum_{\mathbf{l}\in\mathcal{I}} \beta_\mathbf{l}(\mathbf{x}_\mathcal{I})_\mathbf{l}^T\mathbf{M}_\mathbf{l}(\mathbf{x}_\mathcal{I})_\mathbf{l} \\
&= \mathbf{x}_\mathcal{I}{}^T\mathbf{D}_\mathcal{I}\mathbf{G}_\mathcal{I}\mathbf{x}_\mathcal{I}\,. && (5.72)
\end{aligned}$$

In (5.70), we have used $\mathbf{F}_\mathcal{I}\mathbf{x}_\mathcal{I} = \mathbf{x}_\mathcal{I}$ and in (5.71), we have applied the norm equivalence (5.62). The levelwise summation of the mass matrix products was then expressed using the matrix $\mathbf{G}_\mathcal{I}$ in (5.72). In the following, we need the block-diagonal factor $\mathbf{L}_\mathcal{I}$ of the Cholesky decomposition

$$\mathbf{G}_\mathcal{I} = \mathbf{L}_\mathcal{I}\mathbf{L}_\mathcal{I}^T\,.$$

We set $\mathbf{y}_\mathcal{I} = \mathbf{L}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{1/2}\mathbf{x}_\mathcal{I}$ and obtain with $\mathbf{D}_\mathcal{I}\mathbf{G}_\mathcal{I} = \mathbf{D}_\mathcal{I}^{1/2}\mathbf{G}_\mathcal{I}\mathbf{D}_\mathcal{I}^{1/2}$, see Lemma 5.11, the equation

$$\mathbf{x}_\mathcal{I}{}^T\mathbf{D}_\mathcal{I}\mathbf{G}_\mathcal{I}\mathbf{x}_\mathcal{I} = \mathbf{x}_\mathcal{I}{}^T\mathbf{D}_\mathcal{I}^{1/2}\mathbf{L}_\mathcal{I}\mathbf{L}_\mathcal{I}^T\mathbf{D}^{1/2}\mathbf{x}_\mathcal{I} = \mathbf{y}_\mathcal{I}{}^T\mathbf{y}_\mathcal{I}\,.$$

Then, using the equivalence of (5.70) and (5.72), and $\mathbf{x}_\mathcal{I} = \mathbf{D}_\mathcal{I}^{-1/2}\mathbf{L}_\mathcal{I}^{-T}\mathbf{y}_\mathcal{I}$, we obtain the relation

$$\mathbf{y}_\mathcal{I}{}^T\mathbf{L}_\mathcal{I}^{-1}\mathbf{D}_\mathcal{I}^{-1/2}\mathbf{F}_\mathcal{I}^T\mathbf{A}_\mathcal{I}\mathbf{F}_\mathcal{I}\mathbf{D}_\mathcal{I}^{-1/2}\mathbf{L}_\mathcal{I}^{-T}\mathbf{y}_\mathcal{I} \simeq \mathbf{y}_\mathcal{I}{}^T\mathbf{y}_\mathcal{I} \tag{5.73}$$

for all $\mathbf{y}_\mathcal{I} \in \mathrm{im}(\mathbf{L}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{1/2}\mathbf{F}_\mathcal{I})$ with the same constants for the upper and lower bounds as in (5.62). With the commuting of the matrices $\mathbf{F}_\mathcal{I}$ and $\mathbf{D}_\mathcal{I}^{-1/2}$, see Lemma 5.11, the left-hand side of (5.73) leads to (5.69).

Finally, we have to show that no $\mathbf{v}_\mathcal{I} \in \mathbb{R}^{N_\mathcal{I}}$ with $\mathbf{v}_\mathcal{I} \perp \mathbf{y}_\mathcal{I}$ affects the spectrum. From the Fundamental Theorem of Linear Algebra, from $\mathbf{F}_\mathcal{I}^T = \mathbf{G}_\mathcal{I}\mathbf{F}_\mathcal{I}\mathbf{G}_\mathcal{I}^{-1}$, see Lemma 5.10, and from $\mathbf{F}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{1/2} = \mathbf{D}_\mathcal{I}^{1/2}\mathbf{F}_\mathcal{I}^T$ we know that

$$\begin{aligned}\mathbf{v}_\mathcal{I} \in \ker(\mathbf{F}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{1/2}\mathbf{L}_\mathcal{I}) &= \ker(\mathbf{D}_\mathcal{I}^{1/2}\mathbf{G}_\mathcal{I}\mathbf{F}_\mathcal{I}\mathbf{G}_\mathcal{I}^{-1}\mathbf{L}_\mathcal{I}) \\ &= \ker(\mathbf{D}_\mathcal{I}^{1/2}\mathbf{G}_\mathcal{I}\mathbf{F}_\mathcal{I}\mathbf{L}_\mathcal{I}^{-T}\mathbf{L}_\mathcal{I}^{-1}\mathbf{L}_\mathcal{I}) = \ker(\mathbf{F}_\mathcal{I}\mathbf{L}_\mathcal{I}^{-T}) \,. \end{aligned} \tag{5.74}$$

We dropped the matrix $\mathbf{D}_\mathcal{I}^{1/2}\mathbf{G}_\mathcal{I}$ from the kernel in (5.74), as it is a full-rank matrix and thus has no effect on the kernel. Obviously, if $\mathbf{v}_\mathcal{I} \in \ker(\mathbf{F}_\mathcal{I}\mathbf{L}_\mathcal{I}^{-T})$, then $\mathbf{v}_\mathcal{I}$ belongs to the kernel of the preconditioned system (5.69). This finally proves the theorem. □

As a result of Theorem 5.12, we can express our left preconditioner for $\mathbf{A}_\mathcal{I}$ as

$$\mathbf{C}_\mathcal{I} := \mathbf{D}_\mathcal{I}^{-1/2}\mathbf{F}_\mathcal{I}\mathbf{L}_\mathcal{I}^{-T}\mathbf{L}_\mathcal{I}^{-1}\mathbf{F}_\mathcal{I}^T\mathbf{D}_\mathcal{I}^{-1/2} = \mathbf{F}_\mathcal{I}\mathbf{D}_\mathcal{I}^{-1}\mathbf{G}_\mathcal{I}^{-1}\mathbf{F}_\mathcal{I}^T = \mathbf{D}_\mathcal{I}^{-1}\mathbf{F}_\mathcal{I}\mathbf{G}_\mathcal{I}^{-1} \,. \tag{5.75}$$

The representation (5.75) shows the close relation to the preconditioner with negative weights from Section 5.4. The subspace-wise inversion of the mass matrices by $\mathbf{G}_\mathcal{I}^{-1}$ is done in (5.40). Furthermore, in (5.41), the results are scaled with the factors $\omega_\mathbf{l}^V, \mathbf{l} \in \mathcal{I}$, that stem from the combination formula (5.59). For the orthogonal projection based preconditioner (5.75), the scaling is done by the simpler $\mathbf{D}_\mathcal{I}$, but the combination formula is implicitly contained in the projection operator (5.67). In spite of the similarities, having explicit orthogonal projections is useful since it leads to a symmetric preconditioned matrix (5.69), which cannot easily be achieved using the subspace correction method (5.39) with partially negative $\omega_\mathbf{l}^V, \mathbf{l} \in \mathcal{I}$. A second advantage is related to the fact that the unidirectional principle allows in certain cases to exploit orthogonality between subspaces, which reduces the dimension-dependence of the constant factor in the costs of the operator application [Feu05]. The same principle also works for vectors $\mathbf{x}_\mathcal{I} \in \mathrm{im}(\mathbf{F}_\mathcal{I})$ and the unidirectional principle presented in Subsection 4.3.2, but we do not pursue this route further.

5.5.2 Relation to prewavelets

The enlarged generating system introduced some additional difficulties like a non-trivial kernel of the operator matrix and the need for an orthogonalization operator $\mathbf{F}_\mathcal{I}$. This can be avoided in the first place if a direct discretization of the orthogonal subspaces $W_\mathbf{l}$ is available, which is just the case for so-called prewavelets and for wavelets. For the sake of completeness, we describe this case in order to show the close relation to the method presented in Subsection 5.5.1.

Let us first assume that we have basis functions $(\psi_{\mathbf{l},\mathbf{i}})_{\mathbf{i}\in\xi_\mathbf{l},\mathbf{l}\in\mathcal{I}}$ with

$$W_\mathbf{l} = \mathrm{span}\{\psi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \xi_\mathbf{l}\} \quad \text{for} \quad \mathbf{l} \in \mathcal{I} \,, \tag{5.76}$$

and $\bar{n}_\mathbf{l} := \#\xi_\mathbf{l}$. Note that we have V-orthogonality between different levels by definition, but we have not necessarily V-orthogonality within one level. The sparse grid system (4.22) with index set $\mathcal{I}$ and $\bar{N}_\mathcal{I} := \sum_{\mathbf{l}\in\mathcal{I}} \bar{n}_\mathbf{l}$ degrees of freedom reads

$$\bar{\mathbf{A}}_\mathcal{I}\bar{\mathbf{x}}_\mathcal{I} = \bar{\mathbf{b}}_\mathcal{I} \,,$$

where $\bar{\mathbf{A}}_{\mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}} \times \bar{N}_{\mathcal{I}}}$ with

$$(\bar{\mathbf{A}}_{\mathcal{I}})_{(\mathbf{l},\mathbf{i}),(\mathbf{k},\mathbf{j})} = a(\psi_{\mathbf{l},\mathbf{i}}, \psi_{\mathbf{k},\mathbf{j}}) \quad \text{for} \quad \mathbf{i} \in \xi_{\mathbf{l}}, \mathbf{j} \in \xi_{\mathbf{k}}, \mathbf{l}, \mathbf{k} \in \mathcal{I} \,, \tag{5.77}$$

and $\bar{\mathbf{b}}_{\mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}}}$ with

$$(\bar{\mathbf{b}}_{\mathcal{I}})_{\mathbf{l},\mathbf{i}} = (f, \psi_{\mathbf{l},\mathbf{i}})_V \quad \text{for} \quad \mathbf{i} \in \xi_{\mathbf{l}}, \mathbf{l} \in \mathcal{I} \,.$$

Note that the system matrix $\bar{\mathbf{A}}_{\mathcal{I}}$ is invertible, since the functions in (5.76) form a basis of $V_{\mathcal{I}}$ in contrast to the generating system (4.21).

Given a norm equivalence (5.62), it is quite easy to precondition the operator matrix $\bar{\mathbf{A}}_{\mathcal{I}}$. To this end, we need a diagonal scaling matrix $\bar{\mathbf{D}}_{\mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}} \times \bar{N}_{\mathcal{I}}}$ with blocks $(\bar{\mathbf{D}}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{\bar{n}_{\mathbf{l}} \times \bar{n}_{\mathbf{k}}}$ and

$$(\bar{\mathbf{D}}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} = \begin{cases} \beta_{\mathbf{l}} \bar{\mathbf{I}}_{\mathbf{l},\mathbf{l}} & \text{for} \quad \mathbf{l} = \mathbf{k} \,, \\ 0 & \text{else} \,, \end{cases} \tag{5.78}$$

where the $\bar{\mathbf{I}}_{\mathbf{l},\mathbf{l}} \in \mathbb{R}^{\bar{n}_{\mathbf{l}} \times \bar{n}_{\mathbf{l}}}$ denote identity matrices on the subspaces. Furthermore, we need the subspace-wise mass matrix $\bar{\mathbf{G}}_{\mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}} \times \bar{N}_{\mathcal{I}}}$ with blocks $(\bar{\mathbf{G}}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} \in \mathbb{R}^{\bar{n}_{\mathbf{l}} \times \bar{n}_{\mathbf{k}}}$, where

$$(\bar{\mathbf{G}}_{\mathcal{I}})_{\mathbf{l},\mathbf{k}} = \begin{cases} \bar{\mathbf{M}}_{\mathbf{l}} & \text{for} \quad \mathbf{l} = \mathbf{k} \,, \\ 0 & \text{else} \,. \end{cases}$$

Here, $\bar{\mathbf{M}}_{\mathbf{l}} \in \mathbb{R}^{\bar{n}_{\mathbf{l}} \times \bar{n}_{\mathbf{l}}}$ denotes the mass matrix

$$(\bar{\mathbf{M}}_{\mathbf{l}})_{\mathbf{i},\mathbf{j}} = (\psi_{\mathbf{l},\mathbf{i}}, \psi_{\mathbf{l},\mathbf{j}}) \quad \text{for} \quad \mathbf{i}, \mathbf{j} \in \xi_{\mathbf{l}} \,.$$

Then, we have the following theorem.

Theorem 5.13. *The condition number of the matrix*

$$\bar{\mathbf{D}}_{\mathcal{I}}^{-1} \bar{\mathbf{G}}_{\mathcal{I}}^{-1} \bar{\mathbf{A}}_{\mathcal{I}} \tag{5.79}$$

is $\kappa_{\mathcal{I}} = \lambda_{max}^{\mathcal{I}} / \lambda_{min}^{\mathcal{I}}$, *where* $\lambda_{min}^{\mathcal{I}}$ *and* $\lambda_{min}^{\mathcal{I}}$ *are the norm equivalence constants of* (5.62).

Proof. We translate the norm equivalence (5.62) into the matrix-vector setting for block-structured vectors $\bar{\mathbf{x}}_{\mathcal{I}} = ((\bar{x}_{\mathbf{l},\mathbf{i}})_{\mathbf{i} \in \xi_{\mathbf{l}}})_{\mathbf{l} \in \mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}}}$ and obtain

$$\begin{aligned} \bar{\mathbf{x}}_{\mathcal{I}}^T \bar{\mathbf{A}}_{\mathcal{I}} \bar{\mathbf{x}}_{\mathcal{I}} &= a\Big(\sum_{\mathbf{l} \in \mathcal{I}} \sum_{\mathbf{i} \in \xi_{\mathbf{l}}} \bar{x}_{\mathbf{l},\mathbf{i}} \psi_{\mathbf{l},\mathbf{i}}, \sum_{\mathbf{l} \in \mathcal{I}} \sum_{\mathbf{i} \in \xi_{\mathbf{l}}} \bar{x}_{\mathbf{l},\mathbf{i}} \psi_{\mathbf{l},\mathbf{i}}\Big) \\ &\simeq \sum_{\mathbf{l} \in \mathcal{I}} \Big(\sum_{p=1}^{d} 2^{2 l_p} \Big) \Big\| \sum_{\mathbf{i} \in \xi_{\mathbf{l}}} \bar{x}_{\mathbf{l},\mathbf{i}} \psi_{\mathbf{l},\mathbf{i}} \Big\|_{L_2(\Omega^d)}^2 \\ &= \sum_{\mathbf{l} \in \mathcal{I}} \Big(\sum_{p=1}^{d} 2^{2 l_p} \Big) \bar{\mathbf{x}}_{\mathbf{l}}^T \bar{\mathbf{M}}_{\mathbf{l}} \bar{\mathbf{x}}_{\mathbf{l}} \\ &= \bar{\mathbf{x}}_{\mathcal{I}}^T \bar{\mathbf{D}}_{\mathcal{I}} \bar{\mathbf{G}}_{\mathcal{I}} \bar{\mathbf{x}}_{\mathcal{I}} \,. \end{aligned} \tag{5.80}$$

This equivalence holds for all $\bar{\mathbf{x}}_{\mathcal{I}} \in \mathbb{R}^{\bar{N}_{\mathcal{I}}}$. As both $\bar{\mathbf{A}}_{\mathcal{I}}$ and $\bar{\mathbf{D}}_{\mathcal{I}} \bar{\mathbf{G}}_{\mathcal{I}}$ are symmetric positive

definite matrices, it can be easily shown that the upper and lower bounds of (5.80) are also the maximum and minimum eigenvalues of $\bar{\mathbf{G}}_{\mathcal{I}}^{-1}\bar{\mathbf{D}}_{\mathcal{I}}^{-1}\bar{\mathbf{A}}_{\mathcal{I}} = \bar{\mathbf{D}}_{\mathcal{I}}^{-1}\bar{\mathbf{G}}_{\mathcal{I}}^{-1}\bar{\mathbf{A}}_{\mathcal{I}}$. □

Note that prewavelets have been used frequently in the past as the basis functions of regular sparse grid discretizations [GO95b, Feu10], but mostly no special attention was paid to the dependence of the condition number on the dimension. Let us consider the weak form of the Laplacian $a(\cdot,\cdot) = (\nabla\cdot, \nabla\cdot)_{L_2(\Omega^d)}$ and $(\cdot,\cdot)_V = (\cdot,\cdot)_{L_2(\Omega^d)}$. A simple Jacobi-diagonal scaling of $\bar{\mathbf{A}}_{\mathcal{I}}$ is equivalent to replacing the subspace-wise inversion of the mass matrices $\bar{\mathbf{G}}_{\mathcal{I}}^{-1}$ in (5.79) by the identity and the $\bar{\mathbf{D}}_{\mathcal{I}}$ from (5.78) by $\mathrm{diag}(\bar{\mathbf{A}}_{\mathcal{I}})$. This seems reasonable since the resulting condition numbers are still bounded independently of $\mathcal{I}$ for $L_2(\Omega^d)$-stable basis functions. However, they seem to grow exponentially with the dimension, see [Feu05] for numerical results. This is clearly not the case for the system (5.79). Theorem 5.13 has shown that the condition number is the same as of the norm equivalence (5.62), which is level- and dimension-independent for the Laplacian according to Theorem 5.1.

5.5.3 Implementation

So far, we obtained a preconditioner that realizes a generalized condition number that matches the condition number $\kappa_{\mathcal{I}}$ of (5.62). The question is now how high its computational costs are. Remember that a perfect preconditioner would be the inverse of the operator matrix, but that involves way too many computations and in our case $\mathbf{A}_{\mathcal{I}}$ is not even invertible. In this subsection, we discuss the costs of the orthogonal projection based preconditioner from Subsection 5.5.1 and the wavelet-based approach from Subsection 5.5.2.

Orthogonal projection based approach

With (5.69) we now have a preconditioner that involves only a number of floating point operations linear in the number of degrees of freedom $N_{\mathcal{I}}$ of the enlarged system and that results in a condition number $\kappa_{\mathcal{I}}$ bounded from above by κ from the norm equivalence (5.4) independently of $\mathcal{I}$.

We now give a short discussion of the required matrix-vector multiplications and their costs, also with respect to the dimension d. As stated earlier, the orthogonal projection preconditioner is closely related to the subspace correction approach, and in fact the runtime complexity is the same. The application of the scaling matrix $\mathbf{D}_{\mathcal{I}}^{-1}$ is obviously possible with $\mathcal{O}(N_{\mathcal{I}})$ floating point operations. The matrix $\mathbf{G}_{\mathcal{I}}^{-1}$ is block-diagonal and can be implemented with an algorithm that works subspace by subspace. On every $V_{\mathbf{l}}, \mathbf{l} \in \mathcal{F}_J^d$, the mass matrix $\mathbf{M}_{\mathbf{l}} = \bigotimes_{p=1}^{d} \mathbf{M}_{l_p}^{(p)}$ must be inverted. As these matrices have Kronecker product structure, the inversion can be realized by the application of $(\mathbf{M}_{l_p}^{(p)})^{-1}$ to the dimension p for $p = 1, \ldots, d$, which was already noted in Subsection 5.2.4. We assume the functions $(\phi_{\mathbf{l},\mathbf{i}})_{\mathbf{i}\in\chi_{\mathbf{l}}}$ to be of finite element type (h-version with fixed polynomial degree) having local support. Consequently, the associated one-dimensional matrices $\mathbf{M}_{l_p}^{(p)}$ have band matrix structure with constant band size and are thus invertible with linear costs[1]. As a result, we have a cost of $\mathcal{O}(d \cdot n_{\mathbf{l}})$ on each subspace and obtain a cost

[1] Non-local basis functions (p-version) are likely to result in a Toeplitz-type matrix, which can be inverted in log-linear time.

complexity of $\mathcal{O}(d \cdot N_{\mathcal{I}})$ in total. The same argumentation holds for $\mathbf{F}_{\mathcal{I}}$, which has a somewhat more complicated form, see (5.66) and (5.67), but also works subspace by subspace, where we can again exploit a Kronecker product structure.

In total, we arrive at costs of $\mathcal{O}(d \cdot N_{\mathcal{I}})$ for our preconditioner. As before, we assume that the application of $\mathbf{A}_{\mathcal{I}}$ is possible using the unidirectional principle from Subsection 4.3.2 with a computational complexity that is linear in the number of degrees of freedom. However, typically the associated dimension-dependent constant in the costs is proportional to 2^d. This factor can however be reduced to d^2 in special cases like the Laplacian by exploiting the L_2-orthogonality between subspaces, see [Feu05] for a demonstration using prewavelets.

So far, we have expressed the computational effort with respect to the enlarged sparse grid system with $N_{\mathcal{I}}$ degrees of freedom. In case of dyadically refined spline spaces, we get a factor of about 2^d more degrees of freedom than actually needed, see (5.3). We consider this acceptable, because the number of degrees of freedom of regular sparse grids $N_{\mathcal{S}_J^d}$ is of the order $\mathcal{O}(2^J J^{d-1})$, which is exponential in d anyway. Moreover, the number of degrees of freedom of energy sparse grids is of the order $\mathcal{O}(2^J)$ in J, but which involves a constant that is also exponential in d, cf. [Gri06]. Note that it is possible to remove the redundancy of our multilevel discretization via the generating system (4.21) by using prewavelets. This seems to eliminate the 2^d-factor by construction, but this step introduces additional difficulties, among them the setup of the discrete right-hand side $F(v) = (f, v)_V$ for general functions f.

Prewavelet-approach

The system (5.79) gives us the preconditioner $\bar{\mathbf{C}}_{\mathcal{I}} := \bar{\mathbf{D}}_{\mathcal{I}}^{-1} \bar{\mathbf{G}}_{\mathcal{I}}^{-1}$ for the prewavelet operator matrix $\bar{\mathbf{A}}_{\mathcal{I}}$. At first sight, this approach looks simpler and more efficient than the more complicated discretizations $\mathbf{A}_{\mathcal{I}}$ using the enlarged generating system (4.21) and the associated preconditioner $\mathbf{C}_{\mathcal{I}}$ from (5.75) that needs an additional projection step. This can be explained by the prewavelet system $\{\psi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \xi_{\mathbf{l}}\}_{\mathbf{l} \in \mathcal{I}}$ forming a basis and therefore exhibiting no redundancies. Thus, by a factor of about 2^d less degrees of freedom are involved than for the corresponding generating system.

However, there are additional difficulties to be faced in the prewavelet approach, which should not be underestimated and may give the generating system method a practical advantage. First, prewavelets are less local than, e.g. the corresponding multilevel spline basis. Thus, the mass matrix inversions in $\bar{\mathbf{G}}_{\mathcal{I}}^{-1}$ become more involved. From a programming perspective, the more complicated basis functions and different types of prewavelet functions near the boundary make the application of the matrix $\bar{\mathbf{A}}_{\mathcal{I}}$ to a vector more difficult. The efficient application of $\bar{\mathbf{A}}_{\mathcal{I}}$ onto a vector is even more involved, since the unidirectional principle strongly relies on the nestedness of the subspaces. If this is no longer the case, the one-dimensional operators have to be tailored to the specific discretization [Feu05] or the algorithm must switch to a generating system anyway [Zei11].

Finally, the cost complexity of the setup of the right-hand side $\bar{\mathbf{b}}_{\mathcal{I}}$ is increased, as this is typically an integration task and the support of the prewavelets is larger by a factor exponentially depending on the dimension than those of the corresponding splines. Alternatively, the corresponding integrations are realized by the interpolation of the function f from (5.1) in our prewavelet sparse grid space and a subsequent multiplication by the mass matrix to account for the necessary numerical quadrature. As stated in [Feu05], for general functions f, this approach

requires the inclusion of boundary functions in the interpolation step (even if the solution u of our Poisson problem has homogeneous boundary conditions). Since the d-dimensional hypercube Ω^d has 2^d faces, an additional factor of the order 2^d enters the cost complexity for the setup of the right-hand side. The dependence of the cost complexity on the dimension d of other techniques for the assembly of the right-hand side for wavelets and prewavelets with sufficient accuracy, e.g. by the solution of an eigenvector-moment problem associated with the coefficients of the refinement equation [DM93], is unknown to us. We however believe that also these methods involve a factor of at least 2^d in the d-dimensional case due to the tensor product construction, so it is not possible to avoid it altogether.

In summary, the generating system approach from (5.69) can be seen as a simple form of implementation of the prewavelet approach and, indeed, both methods give exactly the same (generalized) condition numbers.

5.6 OptiCom-approach

The OptiCom delivers the best possible scaling (including negative values) in each step of the iteration and results in a convergence that is at least as good as any fixed choice of scaling parameters, unfortunately at the extra cost of setting up and solving an auxiliary system of linear equations in every iteration step. It is therefore not competitive if an explicit norm equivalence (5.4) is known. Nevertheless, the OptiCom does not take the detour via the W-splitting (5.36) and poses a lower bound on the convergence rate that we can achieve by optimizing fixed a priori weights. So the OptiCom can be used to check whether the fixed scalings obtained by other methods are close-to-optimal. In this section, we describe the general case, a CG version, the OptiCom in the special setting of sparse grid discretizations, and the efficient implementation, cf. [GHO15].

5.6.1 Definition

The OptiCom is a nonlinear iterative method which generalizes the steepest decent algorithm mentioned before, and provides a safe lower bound for the best possible error reduction factor ρ^* from (5.33). The method was introduced in [JN99] and used in the context of subspace correction methods for L_2-data approximation with sparse grids in [Heg03]. It was later called OptiCom [Gar06, HGC07]. In the following, we essentially recall some general results and observations from [JN99].

The update formula of OptiCom is the same as in (5.32), i.e.,

$$u^{(m+1)} = u^{(m)} + \sum_{i\in I} \omega_i^{(m)} T_i e^{(m)} = u^{(m)} + P_{\boldsymbol{\omega}^{(m)}} e^{(m)}, \quad m = 0, 1, \dots , \tag{5.81}$$

however, the parameter set $\boldsymbol{\omega}^{(m)} = (\omega_i^{(m)})_{i\in I}$ now depends on $u^{(m)}$: We obtain $\boldsymbol{\omega}^{(m)}$ by solving the quadratic minimization problem

$$\|u - u^{(m)} - \sum_{i\in I} \omega_i^{(m)} T_i e^{(m)}\|_a^2 \to \min_{\boldsymbol{\omega}^{(m)}} \tag{5.82}$$

in each iteration step. The OptiCom iteration converges at least as fast as any stationary additive Schwarz iteration, and thus provides a lower bound for the convergence rate of the latter. The following theorem can also be found in [JN99].

Theorem 5.14. *The error $e^{(m)} = u - u^{(m)}$ of the OptiCom iteration* (5.81) *with $\boldsymbol{\omega}^{(m)}$ from* (5.82) *for the space splitting (5.19) decays in energy norm according to*

$$\|e^{(m+1)}\|_a \leq \rho^* \|e^{(m)}\|_a, \qquad m \geq 0 \,,$$

where ρ^* is the optimal error reduction factor (5.33) for additive Schwarz methods based on the same space splitting.

Proof. As above, denote by $\tilde{\boldsymbol{\omega}}^{(m)}$ the solution of the minimization problem (5.82). Then $e^{(m+1)} = e^{(m)} - \sum_{i\in I} \tilde{\omega}_i^{(m)} T_i e^{(m)}$, and, for any fixed parameter set $\boldsymbol{\omega}$, we have

$$\begin{aligned} \|e^{(m+1)}\|_a &= \|u - u^{(m)} - \sum_{i\in I} \tilde{\omega}_i^{(m)} T_i e^{(m)}\|_a \leq \|e^{(m)} - \sum_{i\in I} \omega_i T_i e^{(m)}\|_a \\ &= \|(I - P_{\boldsymbol{\omega}}) e^{(m)}\|_a \leq \|I - P_{\boldsymbol{\omega}}\|_a \|e^{(m)}\|_a \,. \end{aligned}$$

It remains to take the infimum over all $\boldsymbol{\omega} \in \Omega$ to get the claimed bound for the error reduction factor of the OptiCom iteration. □

5.6.2 CG version

We now point to the CG version of OptiCom. It could further reduce the dependence of the convergence estimates on the condition of the additive Schwarz splitting from an average reduction factor per step of $(1-\mathcal{O}((\kappa^*)^{-1}))$ to $(1-\mathcal{O}((\kappa^*)^{-1/2}))$. Suppose that we have, starting from $u^{(0)} = u^{(-1)} = 0$, already computed $u^{(1)}, \ldots, u^{(m)}$, and that $\boldsymbol{\omega}^{(m)} = (\omega_i^{(m)})_{i\in I}$ and $\eta^{(m)}$ are to be determined as solutions of the slightly modified minimization problem

$$\|u - u^{(m)} - \sum_{i\in I} \omega_i^{(m)} T_i e^{(m)} - \eta^{(m)}(u^{(m)} - u^{(m-1)})\|_a^2 \to \min_{\boldsymbol{\omega}^{(m)}, \eta^{(m)}} \,. \tag{5.83}$$

Then,

$$u^{(m+1)} = u^{(m)} + \sum_{i\in I} \omega_i^{(m)} T_i e^{(m)} + \eta^{(m)}(u^{(m)} - u^{(m-1)}) \tag{5.84}$$

realizes a CG-OptiCom iteration step. As already mentioned in the context of (5.41) for fixed $\boldsymbol{\omega}$, solving the two-parameter ($\tau^{(m)}$ and $\eta^{(m)}$) minimization problem

$$\|u - u^{(m)} - \tau^{(m)} \sum_{i\in I} T_i e^{(m)} - \eta^{(m)}(u^{(m)} - u^{(m-1)})\|_a^2 \to \min_{\tau^{(m)}, \eta^{(m)}} \tag{5.85}$$

is equivalent to the usual PCG-iteration for solving (5.1) with a preconditioner derived from the additive Schwarz operator $P_{\boldsymbol{\omega}}$, thus the name CG-OptiCom. By including the parameter set $\boldsymbol{\omega}^{(m)}$ into the minimization (5.83) we incorporate the scaling of the subproblems. This makes the convergence analysis more difficult, since the preconditioner is no longer fixed but changes from iteration to iteration.

We now discuss the classical proof of conjugate gradients and why it does not work in the stated case. Normally, the convergence speed of the CG method applied to a generic symmetric positive definite matrix $\mathbf{A}$ is based on the fact, that the error $\mathbf{e}^{(m)}$ after m iteration steps is the $\|\cdot\|_{\mathbf{A}}$-minimal element from the space $\mathbf{e}^{(0)} + \mathcal{K}^{(m)}$ with

$$\begin{aligned} \mathcal{K}^{(m)} &= \operatorname{span}\{\mathbf{r}^{(0)}, \mathbf{A}\mathbf{r}^{(0)}, \ldots, \mathbf{A}^{m-1}\mathbf{r}^{(0)}\} \\ &= \operatorname{span}\{\mathbf{A}^1\mathbf{e}^{(0)}, \mathbf{A}^2\mathbf{e}^{(0)}, \ldots, \mathbf{A}^m\mathbf{e}^{(0)}\} , \end{aligned} \tag{5.86}$$

where $\mathbf{e}^{(0)}$ denotes the initial error and $\mathbf{r}^{(0)}$ denotes the initial residual. Because of (5.86), we can express our error $\mathbf{e}_m$ by

$$\mathbf{e}_m = (\mathbf{I} + \textstyle\sum_{i=1}^m \alpha_i \mathbf{A}^i)\mathbf{e}_0 ,$$

where the $\alpha_i, i = 1, \ldots, m$, are chosen such that the resulting $\|\mathbf{e}^{(m)}\|_{\mathbf{A}}$ is minimal. In fact, $\mathbf{e}^{(m)} = \mathcal{P}^m_{\boldsymbol{\alpha}}(\mathbf{A})\mathbf{e}_0$, where $\mathcal{P}^m_{\boldsymbol{\alpha}}$ is a polynomial of degree m with coefficients $1, \alpha_1, \ldots, \alpha_m$. With the help of Chebyshev polynomials and the appropriate rescaling to $(\lambda_{\min}(\mathbf{A}), \lambda_{\max}(\mathbf{A}))$, we get an estimate

$$\|\mathbf{e}^{(m)}\|_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right)^m \|\mathbf{e}^{(0)}\|_{\mathbf{A}} .$$

This is a worst case estimate, and we can expect faster convergence when the eigenvalues of $\mathbf{A}$ are not evenly distributed within the spectrum of $\mathbf{A}$. Unfortunately, the whole argument breaks down when $\mathbf{A}$ is scaled differently in every iteration step, e.g., by diagonal matrices $\mathbf{D}^{(i)}, i = 1, \ldots, m$, and we can no longer express the error $\mathbf{e}^{(m)}$ in terms of a polynomial $\mathcal{P}^m_{\boldsymbol{\alpha}}$ of $\mathbf{A}$ applied to the initial error $\mathbf{e}^{(0)}$. This difficulty was discussed in a slightly different setting in [KL07], and there it was shown that in the worst case, no faster convergence than steepest descent is reached. This is consistent with the proof of Theorem 5.14 (just set $\eta^{(m)} = 0$) that the CG-OptiCom has at least the same error reduction factor per step as the OptiCom. In [JN99] this and other CG versions of a variable preconditioner were presented, but no stronger convergence estimates could be proven. We tried to prove the CG convergence speed by showing that the error $\mathbf{e}^{(m)}$ for the CG-OptiCom would eventually come close to a CG method with a fixed diagonal scaling, but all attempts were in vain. Another approach worth mentioning is to take a different perspective at the CG method [KV13]. This view could be helpful if it was possible to show that the necessary assumptions are met by the OptiCom-scaled search directions. As this did not work out either, this problem can be clearly marked as open research. However, the numerical experiments in Subsection 5.7.3 suggest a still significant speed-up by using the CG-OptiCom (5.84) over the plain OptiCom, i.e. the update (5.81) with the parameter set from (5.82).

5.6.3 Application to sparse grids

We gave a general description of how to find an optimal set of scaling parameters in every iteration step by solving an auxiliary minimization problem. This description largely followed [JN99] and was not confined to sparse grids. We now discuss aspects that are specific to the sparse grid case. They are similar to those that arise in the context of subspace correction methods for L_2-data approximation with sparse grids in the so-called OptiCom [Heg03, Gar06, HGC07].

The minimization problem (5.82) reads in the context of sparse grid discretizations as

$$\|u - u^{(m)} - \sum_{\mathbf{l}\in\mathcal{I}} \omega_{\mathbf{l}}^{(m)} T_{\mathbf{l}}^V e^{(m)}\|_a^2 \to \min_{\boldsymbol{\omega}^{(m)}} \tag{5.87}$$

with $\boldsymbol{\omega}^{(m)} = (\omega_{\mathbf{l}}^{(m)})_{\mathbf{l}\in\mathcal{I}}$, and leads in every iteration step m to a new system

$$\tilde{\mathbf{A}}^{(m)}\boldsymbol{\omega}^{(m)} = \tilde{\mathbf{b}}^{(m)} \ , \tag{5.88}$$

of linear equations. Here, the system matrix $\tilde{\mathbf{A}}^{(m)} \in \mathbb{R}^{\#\mathcal{I}\times\#\mathcal{I}}$ is positive semi-definite with

$$(\tilde{\mathbf{A}}^{(m)})_{\mathbf{lk}} = a(T_{\mathbf{k}}^V e^{(m)}, T_{\mathbf{l}}^V e^{(m)})$$

for $\mathbf{l}, \mathbf{k} \in \mathcal{I}$, and the right-hand side $\tilde{\mathbf{b}}^{(m)} \in \mathbb{R}^{\#\mathcal{I}}$ is given by

$$(\tilde{\mathbf{b}}^{(m)})_{\mathbf{l}} = a(e^{(m)}, T_{\mathbf{l}}^V e^{(m)}) = F(T_{\mathbf{l}}^V e^{(m)}) - a(u^{(m)}, T_{\mathbf{l}}^V e^{(m)})$$

for $\mathbf{l} \in \mathcal{I}$. Recall that the $T_{\mathbf{l}}^V e^{(m)} \in V_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$, are available from the subproblem solves. In contrast to that, the CG version of OptiCom associated with the minimization problem (5.83) would enlarge the system (5.88) by the unknown $\eta^{(m)}$. This leads to one additional entry on the right-hand side and one additional row and column of the system matrix, which we indicate by the letter η. We have

$$\begin{aligned}
(\tilde{\mathbf{A}}^{(m)})_{\eta,\eta} &= a(u^{(m)} - u^{(m-1)}, u^{(m)} - u^{(m-1)}) \ , \\
(\tilde{\mathbf{b}}^{(m)})_{\eta} &= a(u - u^{(m)}, u^{(m)} - u^{(m-1)}) \ , \\
(\tilde{\mathbf{A}}^{(m)})_{\eta,\mathbf{l}} &= a(T_{\mathbf{l}}^V e^{(m)}, u^{(m)} - u^{(m-1)}) \ , \\
(\tilde{\mathbf{A}}^{(m)})_{\mathbf{l},\eta} &= a(u^{(m)} - u^{(m-1)}, T_{\mathbf{l}}^V e^{(m)})
\end{aligned}$$

for $\mathbf{l} \in \mathcal{I}$. In practice, we use a direct method to solve $\tilde{\mathbf{A}}^{(m)}(\boldsymbol{\omega}^{(m)}, \eta^{(m)}) = \tilde{\mathbf{b}}^{(m)}$. In order to avoid problems with possibly singular system matrices, we use a Tikhonov regularization with a very small regularization parameter.

After the solution of (5.88), we can perform our update step (5.41) with $\omega_{\mathbf{l}}^V = \omega_{\mathbf{l}}^{(m)}$ and $\tau^{(m)} = 1, \eta^{(m)} = 0$ in the steepest descent case, or with $\tau^{(m)} = 1$ and $\eta^{(m)}$ obtained from the solution of $\tilde{\mathbf{A}}^{(m)}(\boldsymbol{\omega}^{(m)}, \eta^{(m)}) = \tilde{\mathbf{b}}^{(m)}$ in the CG-OptiCom case.

Note that the setup and solution of the auxiliary problem (5.87) or (5.88), respectively, involves additional costs in every iteration step. If the number of scaling parameters in $\boldsymbol{\omega}^{(m)}$ is moderate compared to the total number of degrees of freedom, then the extra work of solving these linear problems can be tolerated. Of course, the extreme case would be the space splitting into one-dimensional spaces (this is the case of frame decompositions), which results in an "auxiliary" system (5.88) which is as large as the original problem. In the sparse grid case, though, the number of subspaces $\#\mathcal{I}$ and the total number of degrees of freedom is well-balanced, as we see in the cost discussion of the next subsection.

5.6.4 Implementation

Compared to approaches with a priori fixed weights, the OptiCom is more intricate and additionally requires the setup and solution of (5.88). We now discuss the additional computational cost associated with the use of OptiCom. It might be helpful to check Subsection 5.2.4 for the used notation. First note that

$$(\tilde{\mathbf{A}}^{(m)})_{\mathbf{lk}} = a(u_{\mathbf{k}}^{(m)}, u_{\mathbf{l}}^{(m)}) = \left\langle \mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}, \mathbf{A}_{\mathcal{I}}\mathbf{R}_{\mathcal{I},\mathbf{k}}\mathbf{x}_{\mathbf{k}}^{(m)} \right\rangle \tag{5.89}$$

for all $\mathbf{k}, \mathbf{l} \in \mathcal{I}$ and

$$\begin{aligned} (\tilde{\mathbf{b}}^{(m)})_{\mathbf{l}} &= F(u_{\mathbf{l}}^{(m)}) - a(u^{(m)}, u_{\mathbf{l}}^{(m)}) = \left\langle \mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}, \mathbf{b} \right\rangle - \left\langle \mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}, \mathbf{A}_{\mathcal{I}}\mathbf{u}^{(m)} \right\rangle \\ &= \left\langle \mathbf{R}_{\mathcal{I},\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}, \mathbf{r}^{(m)} \right\rangle \end{aligned} \tag{5.90}$$

for all $\mathbf{l} \in \mathcal{I}$. Thus, for *setting up* the matrix $\tilde{\mathbf{A}}^{(m)}$, we have to compute the matrix-vector products $\mathbf{A}_{\mathcal{I}}\mathbf{R}_{\mathcal{I},\mathbf{k}}\mathbf{x}_{\mathbf{k}}^{(m)}$ for every $\mathbf{k} \in \mathcal{I}$. Note that $\mathbf{R}_{\mathcal{I},\mathbf{k}}\mathbf{x}_{\mathbf{k}}^{(m)}$ is zero in any subspace $\mathbf{l} \neq \mathbf{k}$, and in that case the computational complexity of the single space matrix-vector multiplication, see Subsection 4.3.1, is linear in the number of degrees of freedom. We can also use the unidirectional principle from Subsection 4.3.2 to compute $\mathbf{A}_{\mathcal{I}}\mathbf{R}_{\mathcal{I},\mathbf{k}}\mathbf{x}_{\mathbf{k}}^{(m)}$, but due to its complicated recursive structure this choice turns out to be computationally more costly. As the next step, the entries $(\tilde{\mathbf{A}}^{(m)})_{\mathbf{lk}}$ are computed by the scalar product with $\mathbf{R}_{\mathbf{l}}\mathbf{x}_{\mathbf{l}}^{(m)}$ for all $\mathbf{l} \in \mathcal{I}$. As these scalar products need only to be evaluated for coefficients that belong to the subspace $V_{\mathbf{l}}$, the costs for all $\mathbf{l} \in \mathcal{I}$ together is $\mathcal{O}(N_{\mathcal{I}})$ operations. The same argument applies to the entries (5.90) of the right hand side. Thus, we arrive at costs of $\mathcal{O}(N_{\mathcal{I}})$ operations for every $\mathbf{k} \in \mathcal{I}$, and consequently $\mathcal{O}(\#\mathcal{I} \cdot N_{\mathcal{I}})$ operations in total for setting up the system (5.88). The same holds true for the CG version. Note that parallelization is straightforwardly possible.

Solving the system (5.88) by a direct method needs $\mathcal{O}(\#\mathcal{I}^3)$ operations. The subsequent update step (5.41) is again linear in $N_{\mathcal{I}}$. So, the total costs of one OptiCom iteration are $\mathcal{O}(\#\mathcal{I} \cdot N_{\mathcal{I}} + \#\mathcal{I}^3 + N_{\mathcal{I}})$ operations. Since for generalized sparse grid spaces $\#\mathcal{I} \ll N_{\mathcal{I}}$ holds, we can conclude that the total cost complexity of one iteration step is dominated by $\mathcal{O}(\#\mathcal{I} \cdot N_{\mathcal{I}})$. This is by a factor of $\#\mathcal{I}$ more expensive than the costs for the fixed a priori scalings.

We now become a little more specific and choose the regular sparse grid setting $\mathcal{I} = \mathcal{S}_J^d$ (4.10) for level J and dimension d. It is well-known that the dimension of a regular sparse grid space grows as $N_{\mathcal{S}_J^d} = \Theta(J^{d-1}2^J)$ and $\#\mathcal{S}_J^d = \Theta(J^d)$. This means that the cost for one OptiCom iteration is $\mathcal{O}(J^d \cdot J^{d-1}2^J) = \mathcal{O}(J^{2d-1}2^J)$, which is log-linear in $N_{\mathcal{S}_J^d}$, whereas the methods with fixed a priori scalings are only linear in $N_{\mathcal{S}_J^d}$.

Nevertheless, the availability of the single space matrix-vector multiplication and unidirectional principle is a great advantage in our situation, and allows a significant reduction of the computational complexity of one iteration step compared to previous applications of OptiCom for sparse grids. This is due to the fact that we can represent $\mathbf{A}_{\mathcal{I}}$ as a sum of Kronecker-product matrices. Otherwise, the $a(\cdot,\cdot)$ products of $u_{\mathbf{l}}$ and $u_{\mathbf{k}}$ with $\mathbf{l} \neq \mathbf{k}$ have to be computed after the embedding of $u_{\mathbf{l}}$ and $u_{\mathbf{k}}$ into the much larger subspace $V_{\max(\mathbf{l},\mathbf{k})}$ that contains both functions. This is the case for elliptic problems with non-product coefficient functions or in data mining [Gar06], where the data-based energy norm normally does not permit to exploit

tensor product structures. The resulting costs are rather quadratic instead of log-linear in the number of degrees of freedom. Note that the single space matrix-vector multiplication reduces the constant factor even further as it works without the complicated recursive structure of the unidirectional principle, see the experiments in Subsection 5.7.4.

5.7 Numerical experiments

In this section, we demonstrate aspects of the described methods by a model problem. We provide numerical experiments for the Laplace problem

$$-\Delta u = f \tag{5.91}$$

on the open unit cube $\Omega^d = (0,1)^d$ with $f \in L_2(\Omega^d)$, and zero boundary conditions on $\partial\Omega^d$. The weak formulation of (5.91) is a $H_0^1(\Omega^d)$-elliptic variational problem of the form (5.1), where

$$a(u,v) = \sum_{i=1}^{d} \Big(\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i}\Big)_{L_2(\Omega^d)}, \qquad F(v) = (f,v)_{L_2(\Omega^d)} \,. \tag{5.92}$$

For (5.92), a discretization with linear C^0-splines ($r = 0$, $m = 1$ in Theorem 5.5) is sufficient. More precisely, for the $V_l^{(p)}$ in (4.3) we use linear spline spaces defined over dyadic partitions of step-size $2^{-(l+1)}$ on $[0,1]$, with homogeneous boundary conditions at the boundary. Note here that, in order to avoid trivial subspaces for $l = 0$, the step-size associated with $V_l^{(p)}$ is chosen as $2^{-(l+1)}$ and not as 2^{-l}. See Section 4.1 for further details.

5.7.1 Relation full grid and sparse grid condition numbers

Theorem 5.1 states that for our model problem (5.92), the following norm equivalence

$$a(u,u) = \|u\|^2_{H_0^1(\Omega^d)} \simeq \sum_{\mathbf{k}\in\mathbb{N}^d} \Big(\sum_{i=1}^{d} 2^{2k_i}\Big) \|w_{\mathbf{k}}\|^2_{L_2}, \qquad w_{\mathbf{k}} = Q^W_{\mathbf{k}} u, \quad \mathbf{k} \in \mathbb{N}^d \tag{5.93}$$

holds for all function $u \in H_0^1(\Omega^d)$ with a condition number κ that is independent of the dimension d. Thus, we employ the weights

$$\beta_{\mathbf{k}} = \sum_{i=1}^{d} 2^{2k_i} \tag{5.94}$$

in the following experiments. Theorem 5.1 states furthermore that (5.93) holds for $u \in V_{\mathcal{F}_J^d}$ with a condition number $\kappa_{\mathcal{F}_J^d}$ independent of d. As $V_{\mathcal{S}_J^d} \subset V_{\mathcal{F}_J^d}$, we can deduce that

$$\kappa_{\mathcal{S}_J^d} \leq \kappa_{\mathcal{F}_J^d} = \kappa_{\mathcal{F}_J^1} \,. \tag{5.95}$$

Equation (5.95) tells us two things: Using our orthogonal projection based preconditioner (5.69) for a full grid multilevel discretization results in dimension-independent condition numbers and

that the corresponding sparse grid system with the same dimension and level results in a condition number $\kappa_{\mathcal{S}_J^d}$ that is smaller or equal than $\kappa_{\mathcal{F}_J^d}$.

Both propositions become obvious in Table 5.1. It shows the generalized condition numbers of the preconditioned systems (5.69) for $\mathcal{I} = \mathcal{F}_J^d$ and $\mathcal{I} = \mathcal{S}_J^d$ in the full and sparse grid case, respectively, for different dimensions d and levels J. We clearly observe that the full grid condition numbers are bounded from above by a constant independently of the level J. Moreover, they are perfectly independent of the dimension as our theory suggests. The sparse grid condition numbers are even smaller than the corresponding full grid ones for $d > 1$, which is in accordance with our subset argument (5.95). In fact, we even observe *decreasing* condition numbers $\kappa_{\mathcal{S}_J^d}$ with rising dimension d for a fixed level J. This effect is more obvious when considering the splitting condition numbers $\kappa^W_{\mathcal{S}_J^d}$ in the lower part of Table 5.2. Note that they coincide with the condition numbers $\kappa_{\mathcal{S}_J^d}$ of (5.93), and that the prewavelet approach from Subsection 5.5.2 also results in exactly the same condition numbers.

Table 5.1: Degrees of freedom (DOF) $N_{\mathcal{F}_J^d}$ and $N_{\mathcal{S}_J^d}$ and generalized condition numbers $\kappa_{\mathcal{F}_J^d}$ and $\kappa_{\mathcal{S}_J^d}$ of the preconditioned system (5.69) with a full and sparse grid discretization approach, respectively, of the Laplacian on the unit hypercube with linear splines

		DOF $N_{\mathcal{F}_J^d}$ and $N_{\mathcal{S}_J^d}$		condition numbers $\kappa_{\mathcal{F}_J^d}$ and $\kappa_{\mathcal{S}_J^d}$	
	level J	full grid	sparse grid	full grid	sparse grid
dim = 1	1	4	4	3.40	3.40
	2	11	11	4.67	4.67
	3	26	26	5.17	5.17
	4	57	57	5.84	5.84
	5	120	120	6.37	6.37
	6	247	247	6.80	6.80
	7	502	502	7.16	7.16
	8	1013	1013	7.47	7.47
	9	2036	2036	7.74	7.74
	10	4083	4083	7.96	7.96
	11	8178	8178	8.16	8.16
	12	16369	16369	8.33	8.33
dim = 2	1	16	7	3.40	2.99
	2	121	30	4.67	4.46
	3	676	102	5.17	5.06
	4	3249	303	5.84	5.65
	5	14400	825	6.37	6.20
dim = 3	1	64	10	3.40	2.71
	2	1331	58	4.67	4.28
	3	17576	256	5.17	5.00
dim = 4	1	256	13	3.40	2.51
	2	14641	95	4.67	4.12
dim = 5	1	1024	16	3.40	2.36

5.7.2 Splitting condition numbers

Now we compare the condition numbers of the operators $P_{\boldsymbol{\omega}}$ associated with the linear iterative methods for the regular sparse grid case $\mathcal{I} = \mathcal{S}_J^d$ (4.10) with different choices of parameter sets $\boldsymbol{\omega}$ discussed above (OptiCom and CG are nonlinear methods, and left out in this comparison). The results are shown in Table 5.2. We see that the LP-optimized scalings (5.50) lead to condition numbers that are improved by a factor of up to 3 compared to the AdHoc scalings (5.54) proposed in [GO94], which shows that weight optimization has a positive impact. However, for both scalings, the condition numbers are of the order $\Theta(J^{d-2})$, and we can clearly observe their growth for increasing J in dimensions $d \geq 3$. Furthermore, we see that the W-splitting (5.36) with the weights (5.94) leads to condition numbers that are bounded independently of J *and* d. Moreover, they even *decrease* with rising dimension for sparse grids. This condition number is realized by the partially negative scaling factors $\omega_{\mathbf{l}}^V$ that stem from the algebraic transformation (5.59) with $\omega_{\mathbf{k}}^W = \beta_{\mathbf{k}}^{-1}, \mathbf{k} \in \mathcal{S}_J^d$, and also by the orthogonal projection preconditioner (5.69).

5.7.3 Iteration counts

Now, we solve the test problem (5.91) with a random right-hand side f and a randomly initialized starting vector. We choose $J = 10$ and $d = 4$, and plot the reduction of the initial residual in the Euclidean norm against the iteration count. Figure 5.1 shows the convergence of the V-splitting based methods with AdHoc (5.54), LP-optimized (5.50) and algebraic (5.59) scaling factors as well as for the OptiCom. Both the steepest descent approach and the CG versions are considered. We observe that OptiCom is indeed always at least as good as any linear method, but the graphs also show that the potential gain from further optimizing the algebraic scaling factors is quite limited. Furthermore, the conjugate gradient approach works for all four methods, and roughly halves the number of necessary iterations (as it should). Furthermore, we see that the LP-optimized scalings are better than the AdHoc scalings, however, both methods converge slowly. We investigate this effect further in the following experiment, where we vary also J.

In Figure 5.2 we observe that the residual reduction needs more steps for higher values of J. However, we observe that the residual reduction rate ρ of the algebraic scalings appears to be bounded from above independently of J, whereas the convergence rate of the LP-optimized scalings deteriorates quickly. This is in full agreement with our theory, recall that, according to Theorem 5.5, the growth of the condition number of the underlying $P_{\boldsymbol{\omega}}$ is $\Theta(J^2)$ for LP-optimized scalings in dimension $d = 4$.

One final experiment concerns the dimension-dependence of the proposed scalings. Figure 5.3 shows the residual reduction for $J = 10$ in the dimensions $d = 1, \ldots, 4$. The convergence rates of the LP-optimized scalings deteriorate with dimension $d \geq 3$, whereas, as remarked earlier, the condition numbers of the W-splitting and thus the convergence rates of the algebraic scaling factors are bounded independently of the dimension. This is true for the specific, problem-dependent weights (5.94) for our model problem (5.91), and generalizes to problems with a similar sum of tensor products structure as the Laplacian. For other weights (such as the standard weights $\beta_{\mathbf{k}} = 2^{2|\mathbf{k}|_\infty}$ for H^1-problems), the d-independence is lost.

Table 5.2: Splitting condition numbers of the V-splittings with AdHoc- and LP-optimized scalings and of the W-splitting for different dimensions d and levels J

	V-splitting (5.38) condition number $\kappa^V_{\mathcal{S}^d_J}$ with AdHoc scalings (5.54)									
$d \setminus J$	1	2	3	4	5	6	7	8	9	10
1	2.86	3.87	4.74	5.47	6.06	6.55	6.95	7.29	7.58	7.83
2	3.13	5.08	8.12	8.07	10.81	10.37	11.69	11.14		
3	3.91	7.92	9.30	14.45	16.60	20.43				
4	4.86	12.33	18.84	26.53	38.05					
5	5.85	18.30	34.68	47.97						
6	6.85	26.38	59.33	94.59						
7	7.86	36.92	97.03	175.93						
8	8.87	50.16	153.26							
9	9.88	66.23	234.01							
10	10.88	85.28	345.58							

	V-splitting (5.38) condition number $\kappa^V_{\mathcal{S}^d_J}$ with LP-optimized scalings (5.50)									
$d \setminus J$	1	2	3	4	5	6	7	8	9	10
1	3.40	4.67	5.17	5.84	6.37	6.80	7.16	7.47	7.74	7.96
2	2.99	4.46	4.97	5.75	6.47	7.04	7.77	7.76		
3	2.42	4.05	6.69	8.72	12.31	14.62				
4	2.78	5.68	9.47	16.18	23.17					
5	3.43	7.85	14.58	26.09						
6	4.23	10.92	22.61	41.28						
7	5.08	15.02	34.36	67.90						
8	5.97	20.15	51.11							
9	6.89	26.27	73.92							
10	7.81	33.38	103.76							

	W-splitting (5.36) condition number $\kappa^W_{\mathcal{S}^d_J}$ with weights (5.94)									
$d \setminus J$	1	2	3	4	5	6	7	8	9	10
1	3.40	4.67	5.17	5.84	6.37	6.80	7.16	7.47	7.74	7.96
2	2.99	4.46	5.06	5.65	6.20	6.65	7.04	7.36		
3	2.71	4.28	5.00	5.49	6.06	6.53				
4	2.51	4.12	4.94	5.35	5.95					
5	2.36	3.97	4.88	5.23						
6	2.24	3.83	4.82	5.17						
7	2.15	3.71	4.77	5.15						
8	2.07	3.60	4.71							
9	2.00	3.50	4.66							
10	1.94	3.41	4.61							

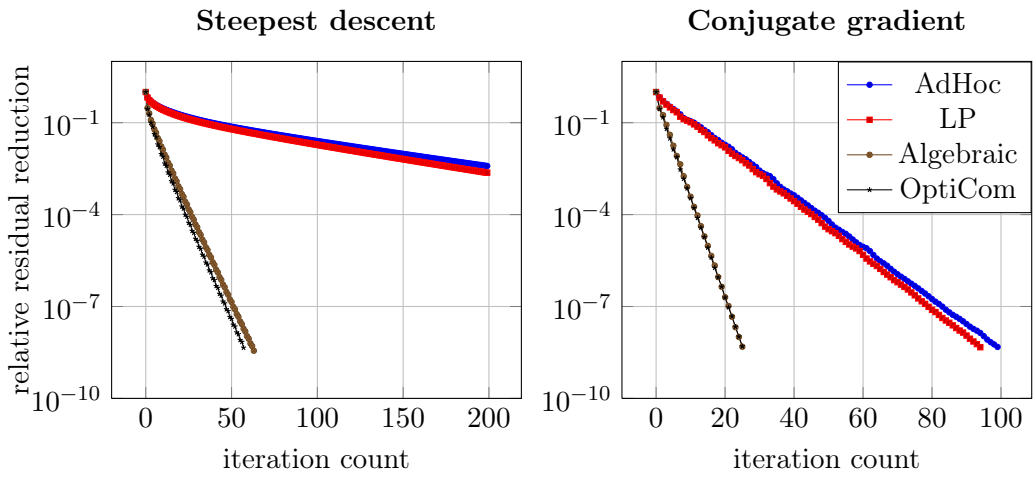

Figure 5.1: Residual reduction with AdHoc scalings, LP-optimized scalings, algebraic scalings, and for OptiCom with the steepest descent method (left) and the conjugate gradient version (right) on $J = 10$ in dimension $d = 4$

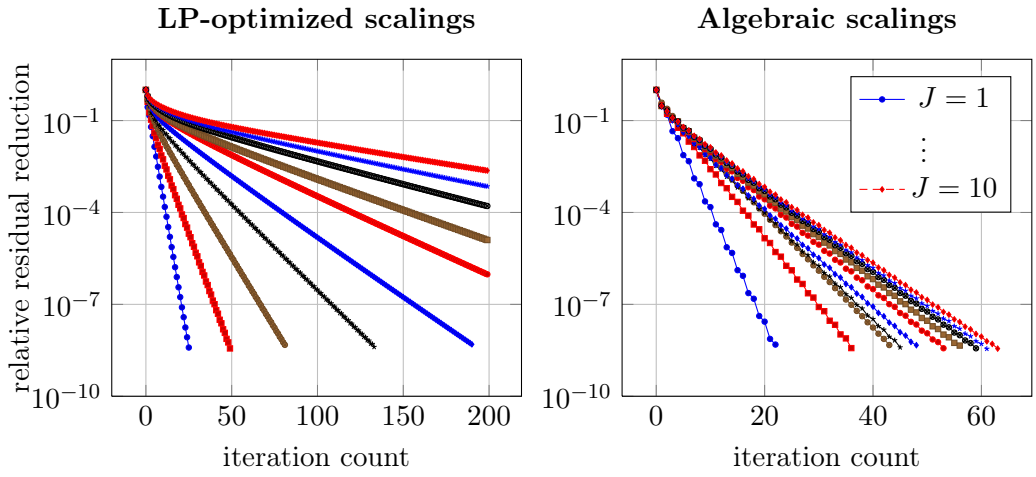

Figure 5.2: Residual reduction with LP-optimized scalings (left) and algebraic scalings (right) in dimension $d = 4$ for different values of J

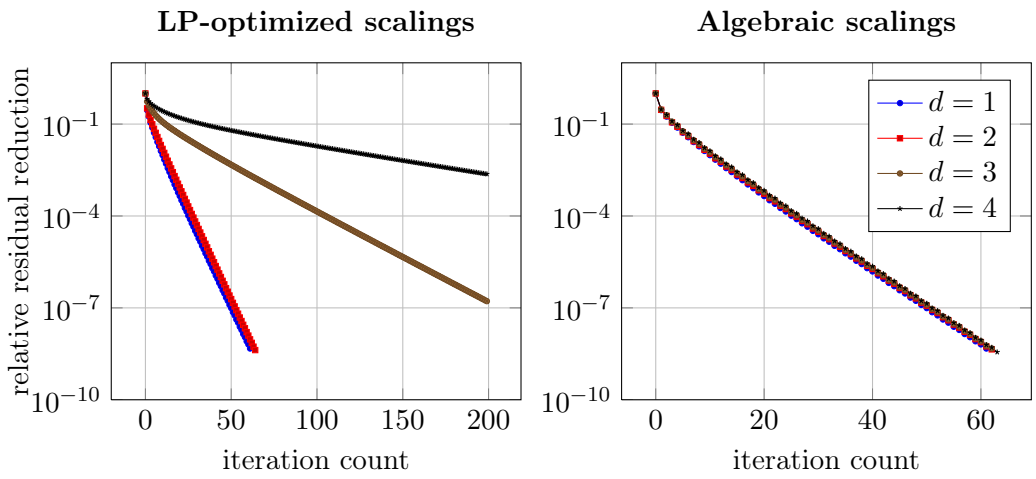

Figure 5.3: Residual reduction with LP-optimized scalings (left) and algebraic scalings (right) for $J = 10$ in dimensions $d = 1, \ldots, 4$

5.7.4 Impact of the single space matrix-vector multiplication

In this subsection, we carry out experiments to estimate the benefit from the single space matrix-vector multiplication presented in Subsection 4.3.1. To this end, we start with the Poisson problem, a random starting vector and a random right hand side, and consider the time necessary to set up the OptiCom auxiliary system (5.88). The time necessary for the direct solution of the system turned out to be negligible compared to the setup costs and is therefore left out. Figure 5.4 shows the setup times for $d = 3$ and various levels J, and Figure 5.5 shows the same for $d = 5$. The number of degrees of freedom is included for reference in both figures. Note that the time consumed and the number of degrees of freedom do not move in perfect lockstep as the runtime curves initially exhibit a higher slope than the number of degrees of freedom. This can be explained by the runtime estimate that contains the number of regular sparse grid subspaces $\#\mathcal{S}_J^d$ as an additional factor, which makes the setup times log-linear with respect to the number of degrees of freedom, see Subsection 5.6.4. We clearly observe that the use of the single space matrix-vector multiplication results in a downwards vertical shift of the runtime compared to the unidirectional principle for $d = 3$ and even more so for $d = 5$. This is due to the fact that we avoid the 2^d constant factor associated with the recursive structure of the unidirectional principle. Table 5.3 shows that the speedup factor roughly doubles if the dimension increases by 1, finally resulting in a very substantial saving in computational complexity for high dimensions.

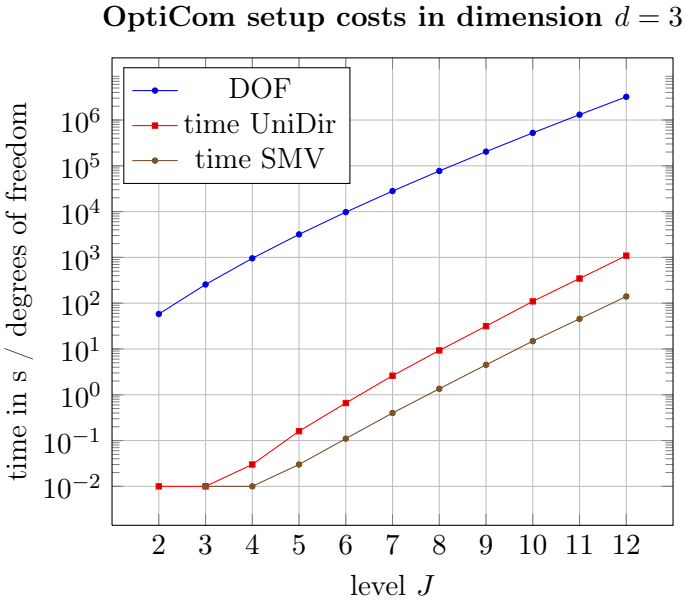

Figure 5.4: Time needed for the setup of the OptiCom auxiliary system (5.88) using the unidirectional principle (UniDir) and the single space matrix-vector multiplication (SMV) in dimension $d = 3$. The number of degrees of freedom (DOF) is given for reference

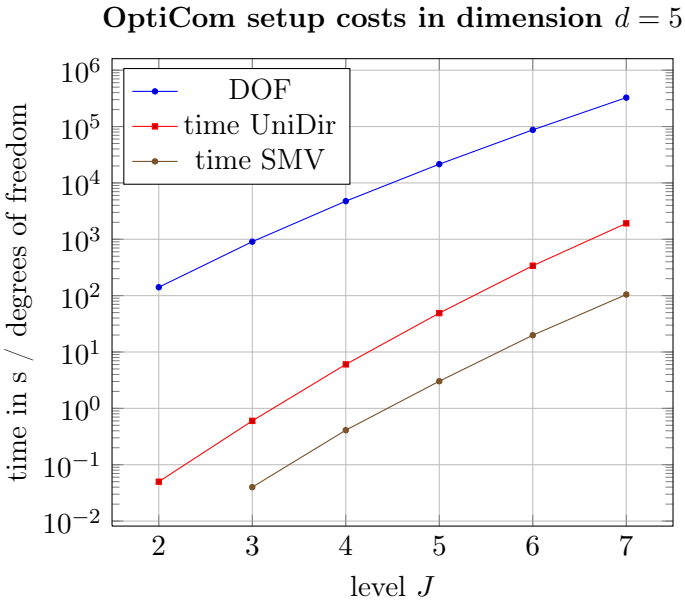

Figure 5.5: Time needed for the setup of the OptiCom auxiliary system (5.88) using the unidirectional principle (UniDir) and the single space matrix-vector multiplication (SMV) in dimension $d = 5$. The number of degrees of freedom (DOF) is given for reference

Table 5.3: Time needed for the setup of the OptiCom auxiliary system (5.88) using the unidirectional principle (UniDir) and the single space matrix-vector multiplication (SMV) for various dimensions and levels. The speedup factor is the ratio of both values

		setup times (s)		speedup
	level J	UniDir	SMV	factor
dim = 3	2	0.01	0.00	-
	3	0.01	0.01	1.0
	4	0.03	0.01	3.0
	5	0.16	0.03	5.3
	6	0.66	0.11	6.0
	7	2.61	0.40	6.5
	8	9.29	1.35	6.9
	9	31.51	4.50	7.0
	10	109.61	14.87	7.4
	11	345.52	45.46	7.6
	12	1092.40	140.11	7.8
dim = 4	2	0.01	0.00	-
	3	0.08	0.01	8.0
	4	0.58	0.06	9.7
	5	3.49	0.38	9.2
	6	19.10	1.89	10.1
	7	94.30	8.32	11.3
	8	419.95	35.20	11.9
	9	1634.70	154.14	10.6
dim = 5	2	0.05	0.00	-
	3	0.60	0.04	15.0
	4	6.03	0.41	14.7
	5	49.00	3.03	16.2
	6	339.31	19.86	17.1
	7	1917.10	104.76	18.3
dim = 6	2	0.23	0.01	23.0
	3	3.71	0.18	20.6
	4	49.57	2.00	24.8
	5	493.12	18.05	27.3
dim = 7	2	0.98	0.03	32.7
	3	20.32	0.48	42.3
	4	305.11	7.35	41.5
dim = 8	2	3.58	0.06	59.7
	3	89.73	1.38	65.0
dim = 9	2	12.40	0.10	124.0
	3	376.79	3.03	124.4
dim = 10	2	41.60	0.17	244.7
	3	1420.88	6.58	215.9

6 Numerical experiments with the Kou-model

In this chapter, we merge the methods we discussed so far and approximate the solution of a ten-dimensional BKE. A problem like this is way beyond the means of classical tensor product discretizations, and also a pure sparse grid approach may struggle due to 2^d constants in the runtime complexity of the operator application, see Subsection 4.3.2. However, the ANOVA approximation technique is able to bridge the gap between ten dimensional problems and moderate-dimensional problems sparse grids excel at.

We now give a more detailed account of the contents of this chapter: In Section 6.1, we introduce a multivariate generalization of Kou's jump-diffusion model [Kou02] with non-zero diffusion, finite activity and a Lévy measure that can be written as a sum of tensor products. Then, the BKE (2.27) and (2.28) can be transformed into the partial integro-differential equation (3.25). The constant coefficients[1] and the Lévy measure that can be written as a sum of tensor products[2] are necessary for the efficient matrix-vector multiplication algorithms described in Section 4.3, but also the preconditioning and the straightforward computation of marginal Lévy processes benefit from these assumptions. In Section 6.2, we introduce a recurrence formula for the Galerkin operator matrix based on the Kou model so that the application remains linear in the number of degrees of freedom altogether, even though the matrix itself is densely populated.

In the remaining sections we put our theoretical findings into practice. We explore the capabilities of our sparse grid PIDE solver and try various combinations of domain truncation parameters, preconditioners and model problems in Section 6.3. As our initial condition lacks the smoothness typically required by sparse grids, we are limited in the number of dimensions we can compute efficiently. That is where the ANOVA approximation technique comes into play. In Section 6.4, we state our example model in ten dimensions and try various ANOVA truncation schemes. We measure the approximation errors at different points of the solution using a Monte Carlo approach, which is a slow but robust method in high dimensions. In this way, we can identify promising parameter sets for our combined ANOVA-PIDE approach. In Section 6.5, we investigate the balance of the ANOVA approximation error and the PIDE discretization error with respect to the truncation dimension d_t and the number of degrees of freedom of the discretization. Then, we finally combine the ANOVA approximation and our PIDE solver and obtain a full ANOVA-PIDE approach. In doing so, we can tackle our ten-dimensional European option pricing problem based on the Kou model. We measure the ANOVA approximation errors and PIDE discretization errors at 100 randomly selected points

[1] The matrix-vector multiplication algorithms can be generalized quite easily so that they work with coefficient functions with product structure, but general coefficient functions require much more work [Ach03].

[2] For pure jump processes with infinite activity and more complex dependence structures like Lévy copulas, see [Win09, RSW10].

in order to assess the quality of our numerical solution. For the right choice of the truncation dimension d_t and the number of degrees of freedom of the discretization, we get a high level of accuracy in the region of $\pm$ one standard deviation around our anchor point.

6.1 The model and problem setup

In this section, we describe the univariate Kou model and a multivariate generalization. We also depict the payoff functions we use as initial condition.

6.1.1 One-dimensional Kou model

Kou's jump-diffusion model [Kou02] assumes that the price of a security S fulfills the stochastic differential equation (SDE)

$$\frac{\mathrm{d}S(t)}{S(t-)} = \mu \mathrm{d}t + \sigma \mathrm{d}W(t) + \mathrm{d}\left(\sum_{j=1}^{N(t)} (\mathcal{Z}_j - 1)\right), \tag{6.1}$$

where t denotes the time, $S(t-)$ is the left-sided limit of $S(t)$, $W(t)$ is a standard Brownian motion, μ and σ are the usual constants for drift and volatility, $N(t)$ is a Poisson process with rate λ and $\{\mathcal{Z}_j\}_{j\in\mathbb{N}}$ denotes a sequence of jumps. These jumps are assumed to be independent and identically distributed with density

$$\varkappa_{\rho,\alpha,\beta}(v) = \begin{cases} (1-\rho)\alpha v^{\alpha-1} & \text{for} \quad v < 1\,, \\ \rho\beta v^{-\beta-1} & \text{for} \quad v \geq 1\,, \end{cases} \tag{6.2}$$

where ρ and $1-\rho$ denote the probabilities of jumping upwards and downwards, respectively, while $\alpha > 0$ and $\beta > 1$ are parameters that control the jump sizes. Note that

$$\mathbb{E}(\mathcal{Z}_j) = (1-\rho)\frac{\alpha}{\alpha+1} + \rho\frac{\beta}{\beta-1}\,.$$

The density of $\mathcal{Y}_j := \log(\mathcal{Z}_j), j \in \mathbb{N}$, is then given by

$$\kappa_{\rho,\alpha,\beta}(z) = \begin{cases} (1-\rho)\alpha e^{\alpha z} & \text{for} \quad z < 0\,, \\ \rho\beta e^{-\beta z} & \text{for} \quad z \geq 0\,, \end{cases} \tag{6.3}$$

which belongs to the asymmetric double exponential distribution. We define $\kappa_{\rho,\infty,\infty}(z) = \delta(z)$, which is the Dirac-delta in 0 and means no jumps occur at all. This is the natural limit for $\alpha, \beta \to \infty$ and will be useful in the multi-dimensional generalization of the model in Subsection 6.1.2. The expected value and variance are given by

$$\mathbb{E}(\mathcal{Y}_j) = \frac{\rho}{\beta} - \frac{1-\rho}{\alpha}\,, \quad \mathrm{var}(\mathcal{Y}_j) = \rho(1-\rho)\left(\frac{1}{\beta} + \frac{1}{\alpha}\right)^2 + \left(\frac{\rho}{\beta^2} + \frac{1-\rho}{\alpha^2}\right)\,.$$

In this model all random variables $W(t), N(t), \mathcal{Z}_j, j \in \mathbb{N}$, are assumed to be independent. The dynamics of S in the SDE (6.1) can then be given by

$$S(t) = S(0) \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W(t)\right) \cdot \prod_{j=1}^{N(t)} \mathcal{Z}_j.$$

See Figure 6.1 for 10 simulated sample paths with the parameter set $T = 0.5$, $S(0) = 100$, $\sigma = 0.16$, $r = 5.0\%$ and jump-term $\lambda = 1, \rho = 0.4, \alpha = 5, \beta = 10$ from [Kou02].

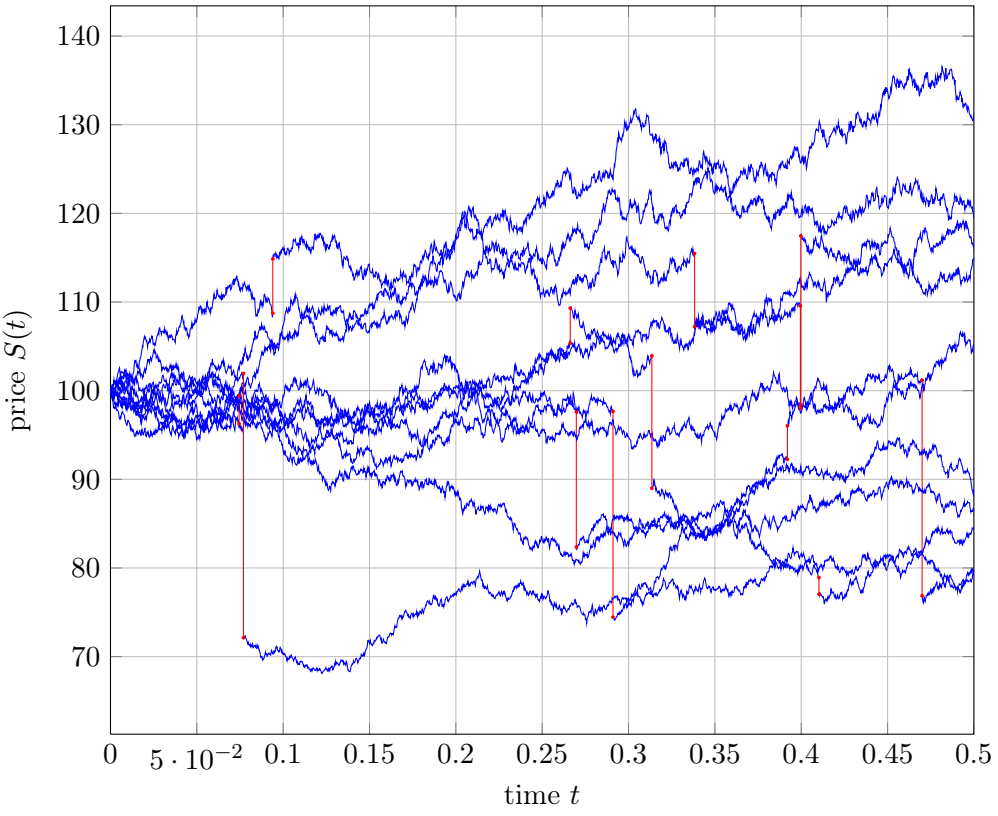

Figure 6.1: Ten sample paths of the univariate Kou-model with $T = 0.5$, $S(0) = 100$, $\sigma = 0.16$, $r = 5\%$ and jump-term $\lambda = 1, \rho = 0.4, \alpha = 5, \beta = 10$

6.1.2 Multi-dimensional case and dependence modelling

We now consider a d-dimensional price process $\mathbf{S} = (S_1, \ldots, S_d)$ with state space $\mathbb{R}^d$, where the components $S_i, i = 1, \ldots, d$, of $\mathbf{S}$ follow the dynamics

$$S_i(t) = S_i(0) \exp\left(\left(\mu_i - \frac{1}{2}\sigma_i^2\right) t + \sigma_i W_i(t)\right) \cdot \prod_{k=1}^{R} \prod_{j=1}^{N_k(t)} \mathcal{Z}_j^{(k,i)}, \tag{6.4}$$

where μ_i and σ_i denote drift and volatility constants, and for each of the R jump terms with $k = 1, \ldots, R$, we have the Poisson process N_k with rate λ_k and jumps $\mathcal{Z}_j^{(k,i)}, j \in \mathbb{N}$, for all components $i = 1, \ldots, d$. This is a slight generalization of the model proposed in [GH13c]. For a relatively recent publication on multivariate generalizations of univariate models, see [BB14].

Obviously, the diffusion part of the process $\log \mathbf{S}$ is decorrelated across the dimensions. In general, this assumption does not hold, but it can be achieved by the transformation discussed in Chapter 2, where $\mathbf{B}$ is chosen such that the covariance matrix $\mathbf{Q}$ is diagonalized to $\mathbf{\Sigma} = \mathbf{B}^T\mathbf{Q}\mathbf{B}$.

The jumps $\mathcal{Z}_j^{(k,i)}, k = 1, \ldots, R, i = 1, \ldots, d, j \in \mathbb{N}$, in (6.4) are all independent and identically distributed with density (6.2) with parameters $\rho_{k,i}, \alpha_{k,i}, \beta_{k,i}$. This, however, does not imply that the processes $S_i(t), i = 1, \ldots, d$, are independent. Since the jumps $\mathcal{Z}_j^{(k,i)}$ for $i = 1, \ldots, d$ share one Poisson process $N_k(t)$, jumps occur at the same time in all components, even though the jumps themselves are independent random variables. With these assumptions we can model market moving events that affect all processes, but also events that affect only single processes are possible. In the next subsection, we see that our R jump terms can be viewed as a low-rank approximation to more complex Lévy measures.

In Figure 6.2, we depict two sample processes with parameters

$$T = 0.5, r = 5.0\%, \mathbf{S}(0) = \begin{pmatrix} 100 \\ 100 \end{pmatrix}, \sigma_1 = \sigma_2 = 0.16$$

and three ($R = 3$) jump terms

$$\begin{bmatrix} k=1 & : & \lambda_1 = 4 & \rho_{1,1} = 0.4 & \alpha_{1,1} = 15 & \beta_{1,1} = 15 & \rho_{1,2} = 0.4 & \alpha_{1,2} = 15 & \beta_{1,2} = 15 \\ k=2 & : & \lambda_2 = 4 & \rho_{2,1} = 0.5 & \alpha_{2,1} = \infty & \beta_{2,1} = \infty & \rho_{2,2} = 0.4 & \alpha_{2,2} = 15 & \beta_{2,2} = 15 \\ k=3 & : & \lambda_3 = 4 & \rho_{3,1} = 0.4 & \alpha_{3,1} = 15 & \beta_{3,1} = 15 & \rho_{3,2} = 0.5 & \alpha_{3,2} = \infty & \beta_{3,2} = \infty \end{bmatrix},$$

where $\alpha_{2,1} = \beta_{2,1} = \infty$ and $\alpha_{3,2} = \beta_{3,2} = \infty$ indicate that no jumps occur in component $i = 1$ for $k = 2$ and component $i = 2$ for $k = 3$, respectively. Figure 6.2 shows two sample paths and their projections onto the time-independent (S_1, S_2)-plane. We note that there are jumps that affect both S_1 and S_2 (jump term $k = 1$) as well as axis-aligned jumps (jump terms $k \in \{2, 3\}$).

6.1.3 Representation of the multi-dimensional process as Lévy process

We now describe our d-dimensional generalization (6.4) in terms of the exponential Lévy model (2.17)

$$\mathbf{S}(t) = (S_1(0)\exp(rt + X_1(t)), \ldots, S_d(0)\exp(rt + X_d(t)) ,$$

where $(\mathbf{X}(t))_{0 \le t < \infty}$ is a $\mathbb{R}^d$-valued Lévy process with characteristic triplet $(\mathbf{Q}, \boldsymbol{\theta}, \nu)$. Then, the elements of covariance matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ satisfy

$$q_{ij} = \delta_{ij}\sigma_i^2$$

for all $i, j = 1, \ldots, d$, and the Lévy measure ν has finite activity and can be expressed by

$$\nu(\mathrm{d}\mathbf{z}) = \sum_{k=1}^{R} \lambda_k \bigotimes_{i=1}^{d} \kappa_{\rho_{k,i},\alpha_{k,i},\beta_{k,i}}(z_i)\mathrm{d}z_i . \tag{6.5}$$

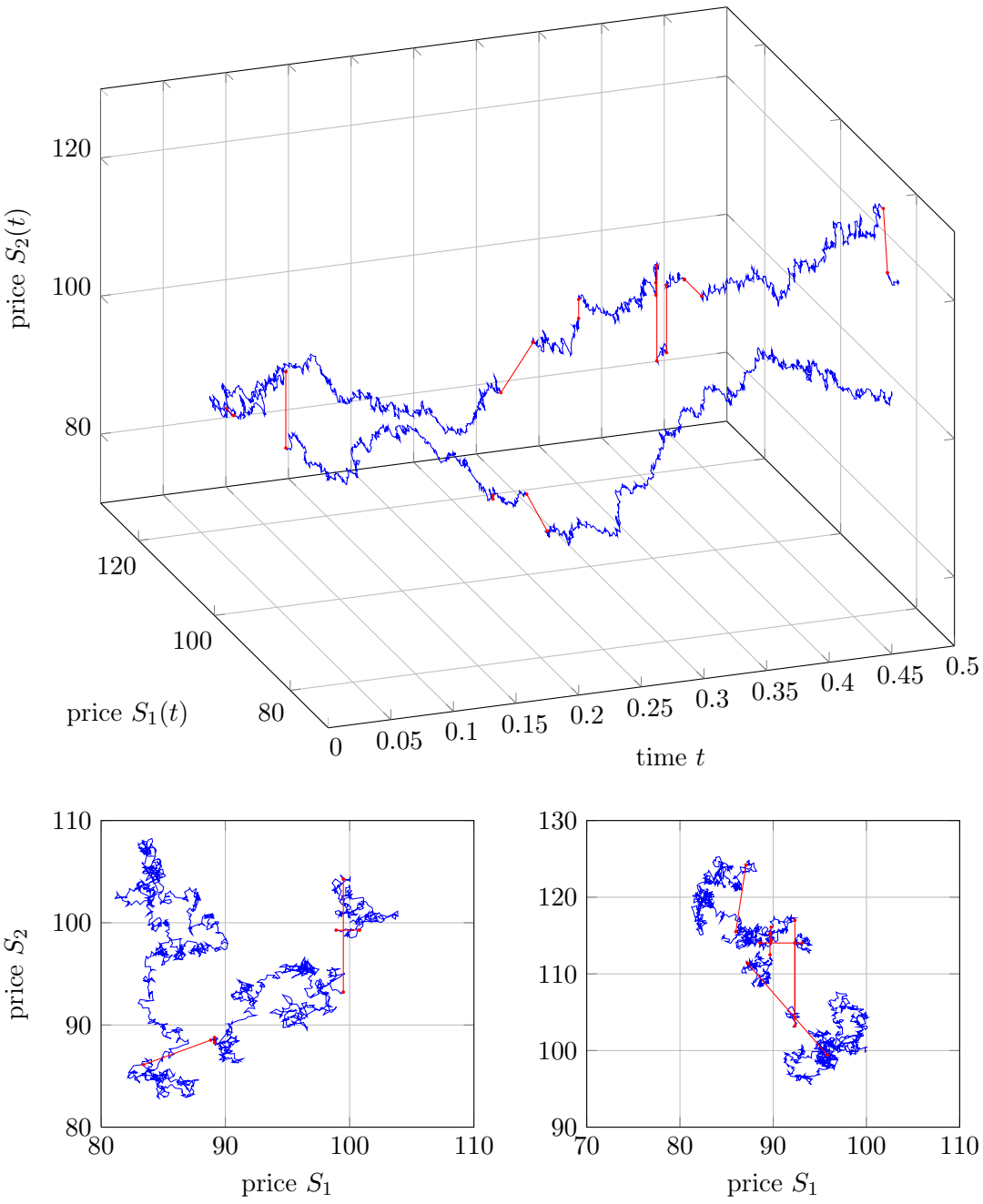

Figure 6.2: The upper plot shows two two-dimensional sample paths of a jump-diffusion process. The two lower plots show the projections of the processes on the (S_1, S_2)-plane, i.e., without time dependence

In (6.5), the k-th summand represents the jumps counted by the Poisson process $N_k(t)$ with rate λ_k, and the associated jump sizes $\mathcal{Z}_j^{(k,i)}, j \in \mathbb{N}$, in the components $i = 1, \ldots, d$ are distributed with densities $\kappa_{\rho_{k,i},\alpha_{k,i},\beta_{k,i}}$, see (6.3).

Note here that the sum of products in (6.5) can also be viewed as a low rank approximation [BM02] to a general finite Lévy measure. To pursue this idea further it might be necessary to enrich the discretization with other one-dimensional jump densities than the double exponential density $\kappa_{\rho,\alpha,\beta}$ and to investigate the approximation error for $R \to \infty$. Still, the jump model (6.5) we use is quite flexible[3], while being computationally advantageous as we will see in Section 6.2.

We adjust the drift $\boldsymbol{\theta}$ such that the martingale condition (2.18) is fulfilled, which means

$$\begin{aligned}
\theta_j &= -\frac{\sigma_j^2}{2} - \int_{\mathbb{R}^d} e^{z_j} - 1 - z_j \nu(\mathrm{d}\mathbf{z}) \\
&= -\frac{\sigma_j^2}{2} - \sum_{k=1}^{R} \lambda_k \int_{\mathbb{R}^d} (e^{z_j} - 1 - z_j)\, \kappa_{\rho_{k,j},\alpha_{k,j},\beta_{k,j}}(z_j)\mathrm{d}z_j \\
&= -\frac{\sigma_j^2}{2} - \sum_{k=1}^{R} \lambda_k \left(\rho_{k,j} \frac{\beta_{k,j}}{\beta_{k,j} - 1} + (1 - \rho_{k,j}) \frac{\alpha_{k,j}}{\alpha_{k,j} + 1} - 1 - \frac{\rho_{k,j}}{\beta_{k,j}} + \frac{1 - \rho_{k,j}}{\alpha_{k,j}} \right)
\end{aligned}$$

for $j = 1, \ldots, d$.

Now, we would like to compute the expected value and variance of $X_j(t), j = 1, \ldots, d$. Due to (2.4) with $\mathcal{T}(\mathbf{z}) = 1$ we know that

$$\mathbb{E}[X_j(t)] = t\theta_j \ ,$$

and using (2.6) we get

$$\begin{aligned}
\mathrm{var}(X_j(t)) &= t \left(q_{jj} + \int_{\mathbb{R}^d} z_j^2 \nu(\mathrm{d}\mathbf{z}) \right) \\
&= t \left(\sigma_j^2 + \sum_{k=1}^{R} \lambda_k \left(\frac{2\rho_{k,j}}{\beta_{k,j}^2} + \frac{2(1 - \rho_{k,j})}{\alpha_{k,j}^2} \right) \right) \ . \qquad (6.6)
\end{aligned}$$

In the previous subsection, we remarked that even though the jump sizes are all independent, the stochastic processes X_i (and thus the S_i) are not, because jumps occur simultaneously across all components. This is reflected very well in the characteristic function

$$\mathbb{E}\left(e^{\mathrm{i}\langle \boldsymbol{\xi}, \mathbf{X}(t) \rangle} \right) = e^{-t\psi(\boldsymbol{\xi})} \ ,$$

which cannot be written as a product in $\xi_1, \ldots, \xi_d$, since ψ from (2.2) with the jump measure ν from (6.5) does in general *not* decompose into a sum of one-dimensional functions over the parameters $\xi_1, \ldots, \xi_d$.

[3] In general it is not true that a more complex model is better than a simpler one, since in practice we still need to reliably estimate the parameters from real-world data, which gets more difficult the more parameters there are.

6.1.4 Payoff function

In this subsection, we discuss what kind of final condition h we focus on. A put option on a basket or an index would typically result in a payoff function

$$h(\mathbf{S}(T)) = \max\left(K - \sum_{i=0}^{d} S_i(T), 0\right) , \tag{6.7}$$

where K is referred to as the strike price. Now, we have used various transformations to arrive at a benign PIDE with constant coefficients (3.25) and our final condition h results in an initial condition

$$g_{\mathbf{B}}(\mathbf{x}) = h\left(\exp(\mathbf{B}\mathbf{x})\right) = \max\left(K - \sum_{i=1}^{d} \exp\left(\sum_{j=1}^{d} b_{ij}x_j\right), 0\right) , \tag{6.8}$$

where the exp-function in $\exp(\mathbf{B}\mathbf{x})$ is applied component-wise and $\mathbf{B} = (b_{ij})_{ij=1}^{d}$ is an orthonormal matrix used in (2.25). In our numerical experiments we start with a diagonal covariance matrix $\mathbf{Q}$ straightaway, i.e., $\mathbf{B} = \mathbf{I}$, since the resulting payoff functions are sufficiently challenging to be representative for other payoffs. See Figure 6.3 for an illustration of payoff functions of a standard put option and of a two-dimensional basket put option. Another reasoning for the choice $\mathbf{B} = \mathbf{I}$ is that our ten-dimensional model to be presented in Section 6.4 could be the result of an even higher-dimensional model, whose covariance matrix has already been diagonalized and truncated after ten dimensions.

At some point the practical relevance of our models has to be questioned using empirical data. For that, we recommend Kou's original paper [Kou02] and a recent publication [BB14] on multivariate generalizations. In this thesis, we restrict ourselves to the pricing of European options, which have no early-exercise features like American options. It is important to note that our payoff function lacks the H^2_{mix}-regularity necessary for achieving the optimal approximation rate of 2 with sparse grids based on linear splines as discussed in Section 4.1. However, it is well-known [Tho06] that the finite element solutions to parabolic problems can converge to full order due to the smoothing effect of the solution/propagation operator even when the initial data are nonsmooth, and thus sparse grids are a viable tool for option pricing, see also [Sch12]. Furthermore, in certain cases, the lower-dimensional terms of an ANOVA decomposition have a higher level of smoothness than the original function [GKS13], which works in our favor.

Depending on where we evaluate our solution $V(T, \mathbf{s})$ the terms "in the money", "at the money" and "out of the money" are used. These terms refer to the "inner value" of an option, i.e., the payoff we would get if the option would expire at the current state of the underlyings $\mathbf{S}(t) = (S_1(t), \ldots, S_d(t))$. In the money put options (6.7) have a positive inner value ($\sum_{j=1}^{d} S_j(t) < K$), at the money options have a inner value which is close to zero or just zero ($\sum_{j=1}^{d} S_j(t) \approx K$), and out of the money options have no inner value and are less likely to result in a non-zero payoff than at the money options ($\sum_{j=1}^{d} S_j(t) \gg K$). As the prices of out of the money options tend to be small, it is harder to achieve a high relative accuracy than for in the money options.

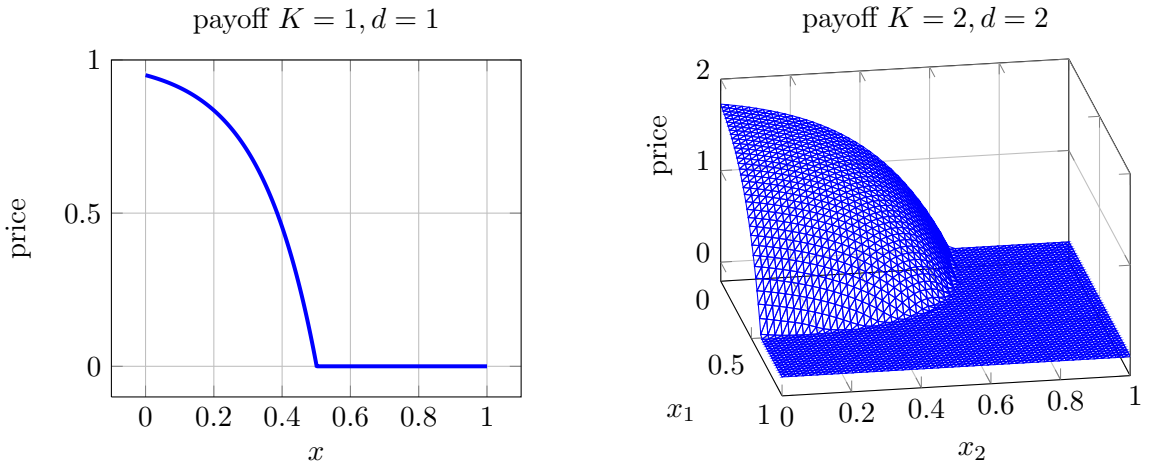

Figure 6.3: Payoff functions of one- (left) and two-dimensional (right) put options in log-coordinates rescaled to $[0, 1]$ and $[0, 1]^2$, respectively

6.2 Galerkin recurrence formula

In this section, we describe a Galerkin recurrence formula for the Kou model that can ultimately be combined with the fast matrix-vector multiplication algorithms from Section 4.3. This yields a sparse grid operator application with computational costs that are linear in the number of degrees of freedom in a generalized sparse grid. We introduced this method in [GH13c]. At the end of the section, we give a quick overview over alternative methods.

In Section 4.3, we have assumed that the matrices $\mathbf{A}^{(1)}_{l_1,k_1}, \ldots, \mathbf{A}^{(d)}_{l_d,k_d}$ from

$$\mathbf{A}_{\mathbf{l},\mathbf{k}} = \mathbf{A}^{(1)}_{l_1,k_1} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,k_d}$$

can be applied in linear time. This is possible for matrices that stem from differential operators, as they are inherently sparse for nodal basis functions. However, integral operators do not have this property and lead to densely populated matrices like (4.31).

There are multiple ways to deal with these matrices. In [GK09, Rei10] wavelet compression is used, which basically means that entries close to zero are discarded. In case of the Kou model, there also exists a recurrence formula that allows us to apply the integral operator for the finite difference case in linear time, cf. [Toi08]. We now derive a similar result for the Galerkin method.

Given the Lévy measure (6.5) we have to deal with operator matrices

$$(\mathbf{A}^{(p)}_{l,k})_{ij} = \int_{[0,1]} \int_{\mathbb{R}} \phi_{k_p,j}(x+z)\kappa_{\rho,\alpha,\beta}(z)\mathrm{d}z\, \phi_{l_p,i}(x)\mathrm{d}x \tag{6.9}$$

for $p = 1, \ldots, d$, where $\kappa_{\rho,\alpha,\beta}$ is an arbitrary Kou density function from (6.3) and $(\phi_{l_p,i})_{i=1}^{n_{l_p}}$ and $(\phi_{k_p,i})_{i=1}^{n_{k_p}}$ are sets of linear spline basis functions on levels l_p and k_p with mesh-widths $h_{l_p} = 2^{-l_p-1}$ and $h_{k_p} = 2^{-k_p-1}$, respectively. Due to the prolongations and restrictions (4.33)

we only need to consider square matrices (6.9) with $l = k$. For the sake of brevity, we drop the p-index in the following. Essentially, the necessary ingredient for our matrix-vector multiplication algorithms presented in Section 4.3 is to compute

$$\mathbf{v} = \mathbf{A}_l \mathbf{u} \ ,$$

where $\mathbf{v} = (v_1, \ldots, v_{n_l})$ with

$$v_i = \int_{[0,1]} \int_{\mathbb{R}} u(x+z) \kappa_{\rho,\alpha,\beta}(z) \mathrm{d}z \, \phi_{l,i}(x) \mathrm{d}x \quad \text{for} \quad i = 1, \ldots, n_l \ ,$$

and the coefficient vector $\mathbf{u} = (u_1, \ldots, u_{n_l})$ describes the function

$$u(x) = \sum_{j=1}^{n_l} u_j \phi_{l,j}(x) \ .$$

For future considerations it is important to note that

$$\phi_{l,i}(x) = \phi_{l,i+1}(x + h_l) \quad \text{for} \quad i = 1, \ldots, n_l - 1 \ . \tag{6.10}$$

Note furthermore that

$$\begin{aligned} (\mathbf{A}_l)_{ij} &= \int_{[0,1]} \int_{\mathbb{R}} \phi_{l,j}(x+z) \kappa_{\rho,\alpha,\beta}(z) \mathrm{d}z \, \phi_{l,i}(x) \mathrm{d}x \\ &= \int_{[0,1]} \int_{\mathbb{R}} \phi_{l,j}(z) \kappa_{\rho,\alpha,\beta}(z-x) \phi_{l,i}(x) \mathrm{d}z \mathrm{d}x \ . \end{aligned} \tag{6.11}$$

Obviously, the matrix $\mathbf{A}_l$ is dense but has Toeplitz structure: Applying (6.10) to both, $\phi_{l,i}$ and $\phi_{l,j}$ in (6.11) gives $(\mathbf{A})_{ij} = (\mathbf{A})_{i+1,j+1}$. This property of the matrix would allow us to execute the matrix-vector multiplication in $\mathcal{O}(n_l \log n_l)$ instead of $\mathcal{O}(n_l^2)$ using the circular convolution theorem.

A different approach that takes also the structure of $\kappa_{\rho,\alpha,\beta}$ into account achieves even a linear runtime complexity. To this end, let us assume that $i, j \in \mathbb{N}$ with $n_l \geq j \geq i + 2$. Then we know that the interior of the supports of $\phi_{l,i}$ and $\phi_{l,j}$ is disjoint and that $\phi_{j,i}(z) \neq 0, \phi_{l,i}(x) \neq 0$ implies $z > x$. This allows us to obtain

$$\begin{aligned} (\mathbf{A})_{ij} &= \int_{[0,1]} \int_{\mathbb{R}} \phi_{l,j}(z) \kappa_{\rho,\alpha,\beta}(z-x) \phi_{l,i}(x) \mathrm{d}z \mathrm{d}x \\ &= \int_{[0,1]} \int_{\mathbb{R}} \phi_{l,j}(z) \rho\beta e^{-\beta(z-x)} \phi_{l,i+1}(x + h_l) \mathrm{d}z \mathrm{d}x \\ &= \int_{[h_l, h_l+1]} \int_{\mathbb{R}} \phi_{l,j}(z) \rho\beta e^{-\beta(z-x+h_l)} \phi_{l,i+1}(x) \mathrm{d}z \mathrm{d}x \\ &= e^{-h_l\beta} (\mathbf{A}_l)_{i+1,j}. \end{aligned}$$

A similar argument gives $(\mathbf{A}_l)_{i,j} = e^{-h_l\alpha} (\mathbf{A}_l)_{i-1,j}$ for $1 \leq j \leq i - 2$. Now we can introduce the

splitting

$$v_i = v_i^{\leftarrow} + v_i^{\circ} + v_i^{\rightarrow}$$

with

$$v_i^{\leftarrow} = \sum_{j=1}^{i-2} (\mathbf{A}_l)_{i,j} u_j, \quad v_i^{\circ} = \sum_{j=i-1}^{i+1} (\mathbf{A}_l)_{i,j} u_j, \quad v_i^{\rightarrow} = \sum_{j=i+2}^{n_l} (\mathbf{A}_l)_{i,j} u_j \ .$$

With the recursive relationships

$$\begin{aligned} v_i^{\leftarrow} &= \sum_{j=1}^{i-3} (\mathbf{A}_l)_{ij} u_j + (\mathbf{A}_l)_{i,i-2} u_{i-2} \\ &= e^{-h_l \alpha} \sum_{j=1}^{(i-1)-2} (\mathbf{A}_l)_{i-1,j} u_j + (\mathbf{A}_l)_{i,i-2} u_{i-2} \\ &= e^{-h_l \alpha} v_{i-1}^{\leftarrow} + (\mathbf{A}_l)_{i,i-2} u_{i-2} \end{aligned}$$

for $i = 4, \ldots, n_l$ and

$$\begin{aligned} v_i^{\rightarrow} &= \sum_{j=i+3}^{n_l} (\mathbf{A}_l)_{ij} u_j + (\mathbf{A}_l)_{i,i+2} u_{i+2} \\ &= e^{-h_l \beta} \sum_{j=(i+1)+2}^{n_l} (\mathbf{A}_l)_{i+1,j} u_j + (\mathbf{A}_l)_{i,i+2} u_{i+2} \\ &= e^{-h_l \beta} v_{i+1}^{\rightarrow} + (\mathbf{A}_l)_{i,i+2} u_{i+2} \end{aligned}$$

for $i = 1, \ldots, n_l - 3$, all $v_i^{\leftarrow}$, v_i° and $v_i^{\rightarrow}$ can be precalculated in linear time. Altogether, we can compute the matrix-vector product $\mathbf{v} = \mathbf{A}_l \mathbf{u}$ in $\mathcal{O}(n_l)$ complexity even though the matrix $\mathbf{A}_l$ is dense. This approach also works with other basis functions than linear splines as long as they are positioned on an equidistant grid and have finite support.

So far, we have described the application of the integro-operator of the Kou model to a one-dimensional discretization on level l only. This approach now easily carries over to the multi-dimensional generating system case by the unidirectional principle with the dimension-recursive form of the algorithm (4.43)–(4.46), which exploits a given tensor product structure and requires only one-dimensional non-hierarchical applications of the Kou integral-operator in (4.40) and (4.42). This altogether allows for the application of the operator matrices in the discretized equation (3.19) in just linear time with respect to the number of degrees of freedom. This recurrence formula can also be employed in the context of the single space matrix-vector multiplication algorithm in (4.36), which is helpful for the OptiCom preconditioner.

Up to here, we have made lots of assumptions until we arrived at the Galerkin recurrence formula for the Kou model. At this point, we want to refer the reader to alternative matrix-vector multiplication methods if these requirements are not met. See Table 6.1 for a rough guidance, where we assume a discretization based on a regular sparse grid on level J in d dimensions with $\mathcal{O}(2^J J^{d-1})$ degrees of freedom.

Table 6.1: Overview over matrix–vector multiplication approaches

Assumption	Method	Complexity
Multivariate Kou model and sum of tensor product operators	Use the unidirectional principle from Subsection 4.3.2 and the recurrence formula from Subsection 6.2. The resulting computational complexity is linear in the number of degrees of freedom but the recursive structure of the unidirectional principle leads to a dimension-dependent constant of at least 2^d, which limits its use in high-dimensions. In certain instances, there are remedies [Feu05]	$\mathcal{O}(2^J J^{d-1})$
Sum of tensor product operators with a non-Kou integral kernel (4.27)	Use the unidirectional principle from Subsection 4.3.2 and employ the circular convolution theorem for the one-dimensional integral convolutions. This is straightforward and easy to implement, and the extra work only leads to an additional factor J in the runtime complexity. It may be worthwhile to check whether the constraint of having sums of tensor products can be dropped using a generalized sparse grid FFT [GH14a]	$\mathcal{O}(2^J J^d)$
No tensor product operators	Wavelet compression [GK09, Rei10]. Mathematically and computationally challenging and leads to a log-linear amount of non-zero entries of the system matrix with respect to the number of degrees of freedom	$\mathcal{O}(2^J J^{2(d-1)})$
Any operator, small J and high d	Naive approach using the full matrix. This results in a computational complexity that is quadratic in the number of degrees of freedom, but there is no additional factor of 2^d involved	$\mathcal{O}(2^{2J} J^{2(d-1)})$

6.3 Experiments with the PIDE solver

In this section, we perform several experiments with our PIDE solver. That means we combine the time and space discretization from Chapter 3, the sparse grid technique from Chapter 4, the preconditioning from Chapter 5, and the Galerkin recurrence formula from Section 6.2. We assume a low number of space dimensions, e.g., $d = 1, 2, 3$. Experiments with our PIDE solver for $d = 4$ will be presented in Section 6.5.

6.3.1 Toivanen option

In this subsection, we price an option with the parameters given in [Toi08]. Namely, we price a European put option with $\sigma = 0.2$, $r = 0.0\%$, $T = 0.2$, $S(0) = 1$, $K = 1$ and jump term $\lambda = 1$, $\rho = 0.5$, $\alpha = 2$, $\beta = 3$. We compute the reference value 0.04264848 by a Monte Carlo simulation with 2.0×10^{10} samples, which coincides with the price given in [Toi08] up to a relative error of $\sim 10^{-5}$. We do not expect any convergence of the discretization error below this accuracy, but we also need to consider other sources of error: The time discretization and the localization of the domain.

Our first set of experiments studies the effect of the domain truncation. In Section 3.3, we advocate to use a domain (3.9) that is adapted to the standard deviations of the components of $\mathbf{X}$. For several choices of the parameter ζ in (3.10) and (3.11) we measure the pointwise error at the money with respect to the discretization level. We use an implicit time discretization with $M = 2048$ steps and the orthogonal projection based preconditioner from Section 5.5. See Figure 6.4 for the resulting relative errors. It turns out that the localization error appears later and later the larger the domain is, but choosing a larger domain results in the need for a higher amount of degrees of freedom to achieve the same error. This shows that the domain should be large enough to reach a certain target accuracy but not larger than necessary. Note that here and in the following we refer to the domain immediately after localization but before transformation with (3.12) to the d-dimensional unit cube Ω^d.

6.3.2 Two-dimensional put option

In this subsection, we price a two-dimensional European put option based on our multivariate Kou model, and we examine the convergence rates for different kinds of errors.

Square domain at the money

The parameters we use in the first experiment are

$$T = 0.2, K = 2.0, r = 0.0\%, \sigma_1 = \sigma_2 = 0.2$$

with one ($R = 1$) jump term

$$\left[k = 1 \quad : \quad \lambda_1 = 0.3 \quad \rho_{1,1} = 0.5 \quad \alpha_{1,1} = 8 \quad \beta_{1,1} = 8 \quad \rho_{1,2} = 0.5 \quad \alpha_{1,2} = 8 \quad \beta_{1,2} = 8\right] .$$

We use a regular sparse grid in space and an implicit time discretization with $M = 128$ steps. Next to the pointwise error at the money for $\mathbf{S}(0) = (1.0, 1.0)$, we measure the L_2 error of the initial condition and the L_2 error of the solution at $\tau = T$. To this end, we use a tensorized

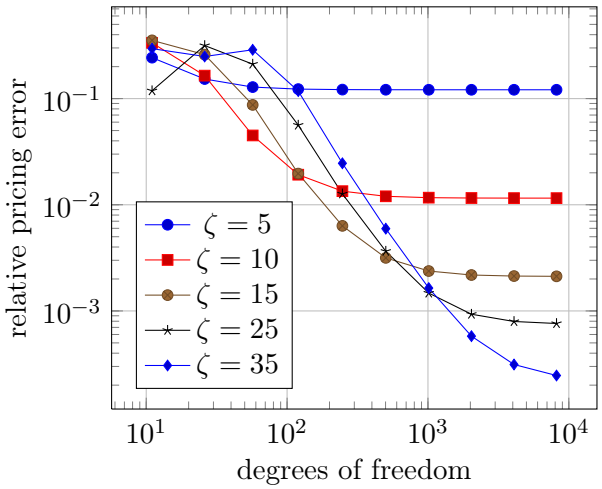

Figure 6.4: The relative pricing error versus the degrees of freedom for different domain sizes for our one-dimensional option

rectangle rule and 100 000 function evaluations. In order to exclude the influence of the domain size in log-space, we here compute the L_2 errors *after* transformation onto the unit square Ω^2 and also on the subset $(0.2, 0.8)^2 \subset \Omega^2$. The reference function for the initial condition is available analytically, and we use a finely resolved numerical solution on a full grid as reference solution. We obtain a reference value of 0.0540281 for the option price at the money using 2.0×10^{10} Monte Carlo samples.

In Figure 6.5 we see the $L_2(\Omega^2)$ approximation error of the initial condition, the $L_2(\Omega^2)$ error of the solution $u(T, \mathbf{x})$ and the relative error of the option price $V(0, \mathbf{s})$ with $\mathbf{s} = (1.0, 1.0)$, which is directly at the money. As our discretization spaces do not include boundary functions but our payoff function is non-zero on the boundary, we get only a meager rate of convergence for the initial condition. When we look at the L_2 error not on $(0, 1)^2$ but on $(0.2, 0.8)^2$, it improves a bit, but due to the non-differentiability at the kink of the payoff function, we do not achieve the full sparse grid convergence rate. Luckily, this is not necessary to obtain an accurate solution, since the error gets damped away by our implicit method. Evaluating the L_2 error at $\tau = T$ shows a convergence that comes close to the rate of 2, which is optimal and better than the rate of 1 we would expect for full grids in two dimensions. The relative error of the evaluation at the money decays fast and a bit erratically, but as we do not have much theory for single point evaluations this comes as no surprise.

Adapted domain at the money

In Section 3.3 we proposed to use a domain (3.9) that is adapted to the standard deviations of the components of $\mathbf{X}(T) = (X_1(T), \ldots, X_d(T))$ by (3.10) and (3.11). In order to have some empirical evidence that this choice is a good idea, we change the parameters from the previous subsection to

$$T = 0.2, K = 2.0, r = 0.0\%, \sigma_1 = 0.2, \sigma_2 = 0.05$$

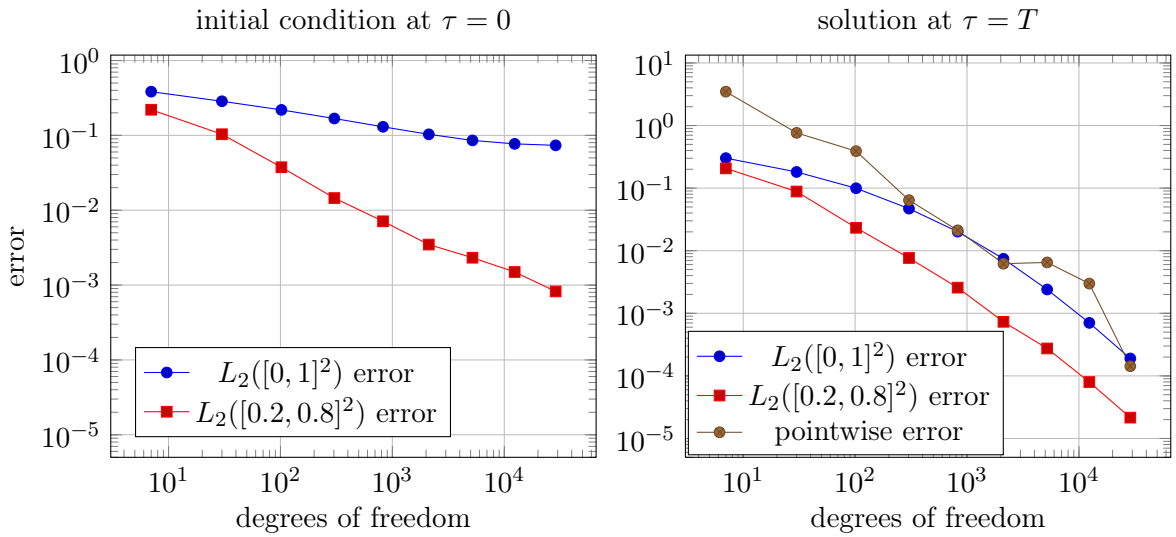

Figure 6.5: Two-dimensional option error convergence. The left plot shows the poor approximation of the initial condition. The right plot shows that we reach a reasonable error convergence at $\tau = T$ both in terms of the L_2 error approximation and the relative pricing error evaluated at the money.

with one ($R = 1$) jump term

$$\left[k = 1 \quad : \quad \lambda_1 = 0.3 \quad \rho_{1,1} = 0.5 \quad \alpha_{1,1} = 8 \quad \beta_{1,1} = 8 \quad \rho_{1,2} = 0.5 \quad \alpha_{1,2} = 32 \quad \beta_{1,2} = 32\right] ,$$

so that $\sigma(X_2(T))$ is a fourth of $\sigma(X_1(T))$. We now run two sets of experiments: In one of them the difference of the standard deviations of the first and second component of $\mathbf{X}$ is reflected in the domain size according to (3.10) and (3.11), in the other one we simply choose a square domain. In both cases we use $\zeta = 14$. Figure 6.6 shows the convergence of the approximation error of the initial condition. We observe that the adapted domain has a smaller L_2 error but exhibits the same convergence rate as the square domain. Note again that the L_2 errors are computed after transformation to the unit square, so the domain size does not directly affect the error. Figure 6.7 shows the pointwise error at $\mathbf{s} = (1.0, 1.0)$. Overall, we can conclude that the payoff function is easier to approximate on the adapted domain, while providing more accurate prices.

Adapted domain in the money

Now we want to evaluate our solution not on the kink but in a smooth region of the payoff function (6.8). To that end, we repeat the last experiment, but price an in the money option with $\mathbf{s} = (0.5, 0.5)$. As our localization is centered around our point of evaluation according to (3.10) and (3.11), the kink does no longer go straight through our domain of computation. See Figure 6.8 for the error convergence in that case. Even for the initial condition, we can observe approximation rates around 2 in the inner part of the domain $(0.2, 0.8)^2$. Also the error of the single point evaluation at $\mathbf{s} = (0.5, 0.5)$ converges fast. However, the convergence stops

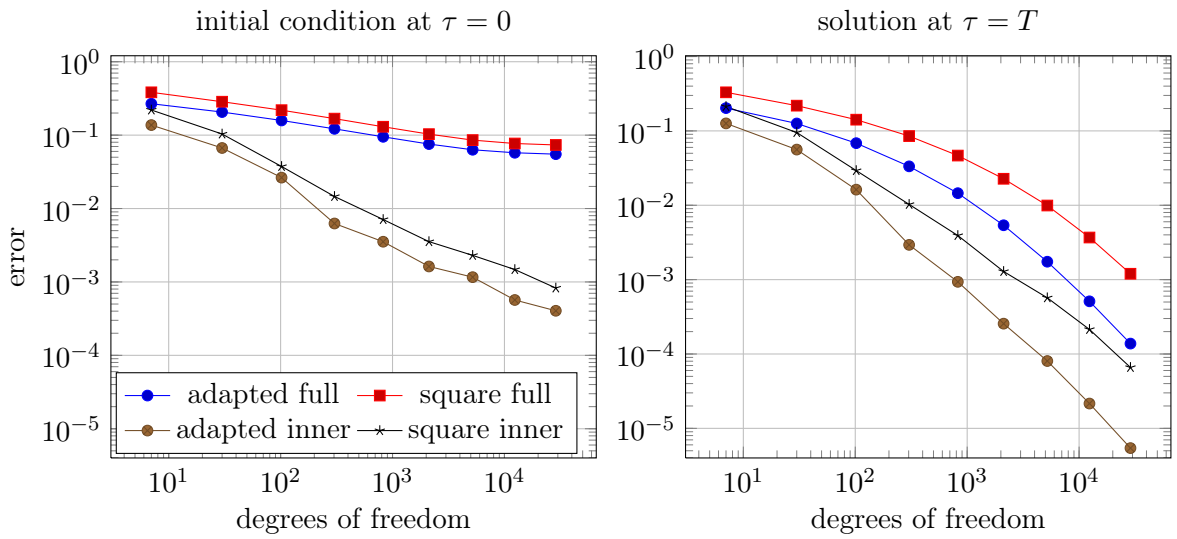

Figure 6.6: The left plot shows the $L_2((0,1)^2)$ ("full") and $L_2((0.2,0.8)^2)$ ("inner") approximation errors of the initial condition for a domain adapted to the standard deviations of X_1 and X_2 ("adapted") and a square domain ("square"). The right plot shows the same for the solution at $\tau = T$

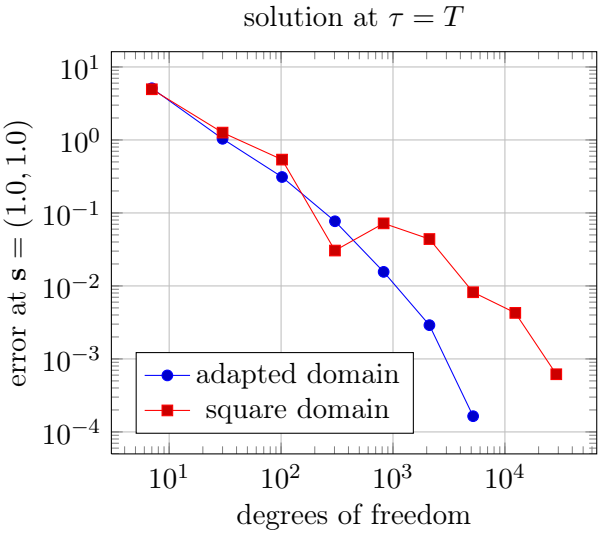

Figure 6.7: This plot shows the relative pointwise error at $\mathbf{s} = (1.0, 1.0)$ for a square domain and one which is adapted to the standard deviations of the components X_1 and X_2

at around $\sim 10^{-4}$, which is either due to the time discretization or the localization error. If we require a higher accuracy, we would have to choose a larger domain and to fully resolve the kink, which would again reduce the rate of convergence due to the non-differentiability. We can conclude that payoff functions like (6.8) are not optimal for sparse grid methods, but they are practically relevant so we stick to this example.

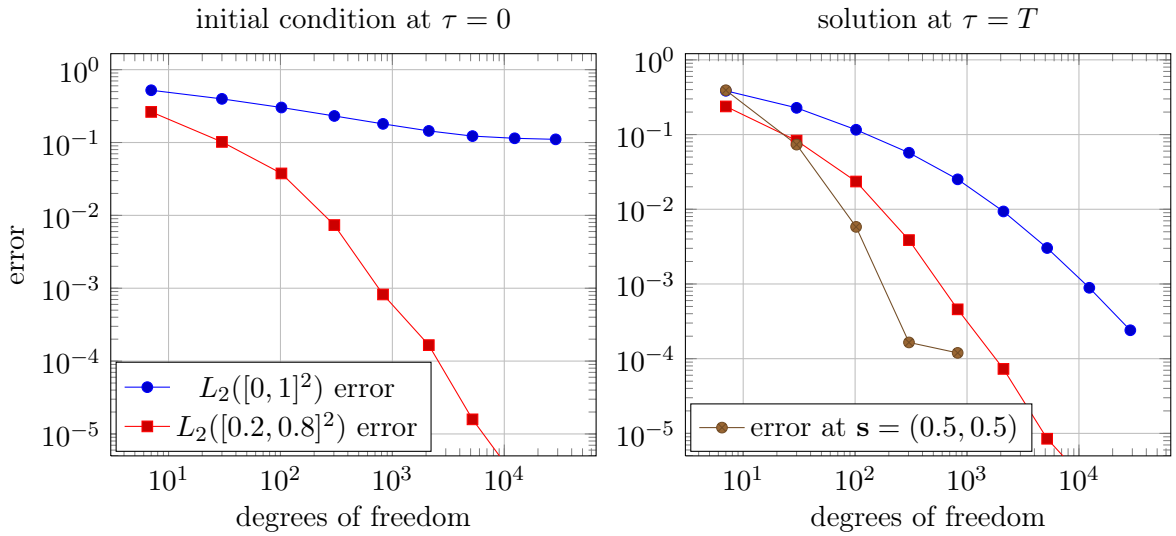

Figure 6.8: The left plot shows the $L_2((0,1)^2)$ and $L_2((0.2,0.8)^2)$ approximation errors of the initial condition. The right plot shows the same errors for the solution at $\tau = T$, and also the pointwise error at $\mathbf{s} = (0.5, 0.5)$

6.3.3 Preconditioning experiments

In this subsection we present experiments with the preconditioners discussed in Chapter 5.

Two-dimensional example

We consider the two-dimensional option from the previous subsection and choose a sparse grid with index set $\mathcal{I} = \mathcal{S}_8^2$ for discretization, i.e., a sparse grid with level $J = 8$ and dimension $d = 2$. For the purpose of this experiment we use only one time step, i.e., $M = 1$. Figure 6.9 shows the convergence of the residual for a preconditioner based on the algebraic scaling parameters discussed in Section 5.4 and the non-linear preconditioner based on the OptiCom from Section 5.6. We can see quite clearly that the OptiCom performs slightly better than the algebraic scaling parameters, but we have to keep in mind that the OptiCom is computationally more expensive. The CG versions of both methods seem to converge equally fast, which means the OptiCom method benefits less from the CG approach than the algebraic weights. This is consistent with the fact that the orthogonality of search directions breaks down in a CG method if the preconditioner changes from iteration step to iteration step. This problem has already been discussed in Subsection 5.6.2.

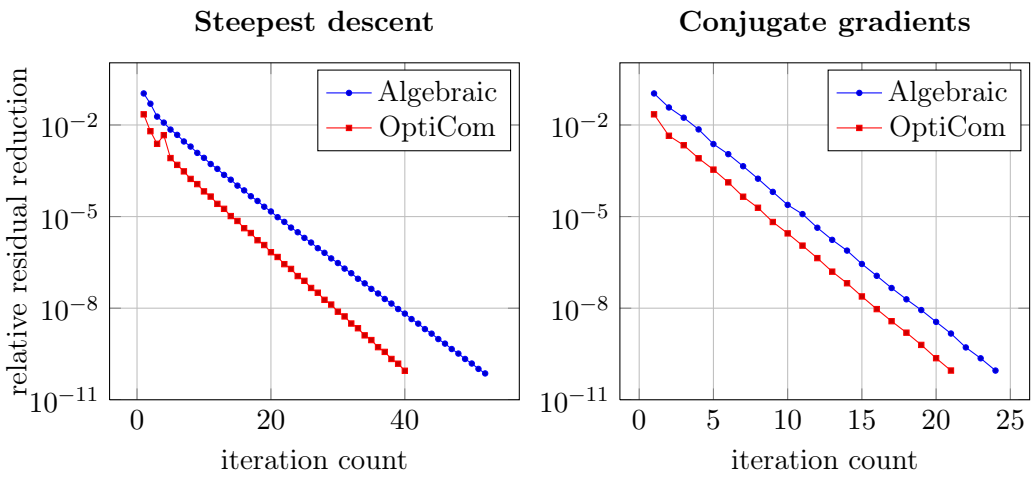

Figure 6.9: These plots show the convergence of the residual for the steepest descent (left) and conjugate gradient (right) method with algebraic scaling parameters and the non-linear OptiCom in our two-dimensional example problem

In Figure 6.10 we can see that there is a maximum of 8 subspaces in each dimension. This figure depicts the scaling parameters $\omega_{\mathbf{l}}^V, \mathbf{l} \in \mathcal{S}_8^2$. As implied by the formula (5.59) all maximal subspaces that are not contained in other spaces have positive scaling weights. Figure 6.11 shows the OptiCom scaling parameters in the first eight iteration steps. They appear a bit erratic, but the rule of positive scaling parameters on the outer diagonal still holds.

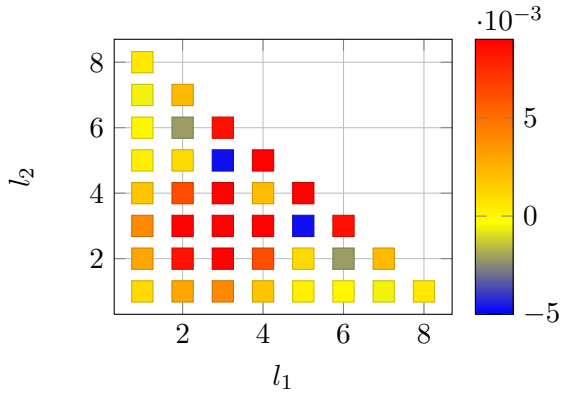

Figure 6.10: Visualization of the sparse grid index set $\mathcal{S}_2^8$, where the algebraic scaling parameters $\omega_{\mathbf{l}}^V, \mathbf{l} \in \mathcal{S}_8^2$, are color-coded. The parameters that exceed the range of the colorbar are clipped to the nearest value

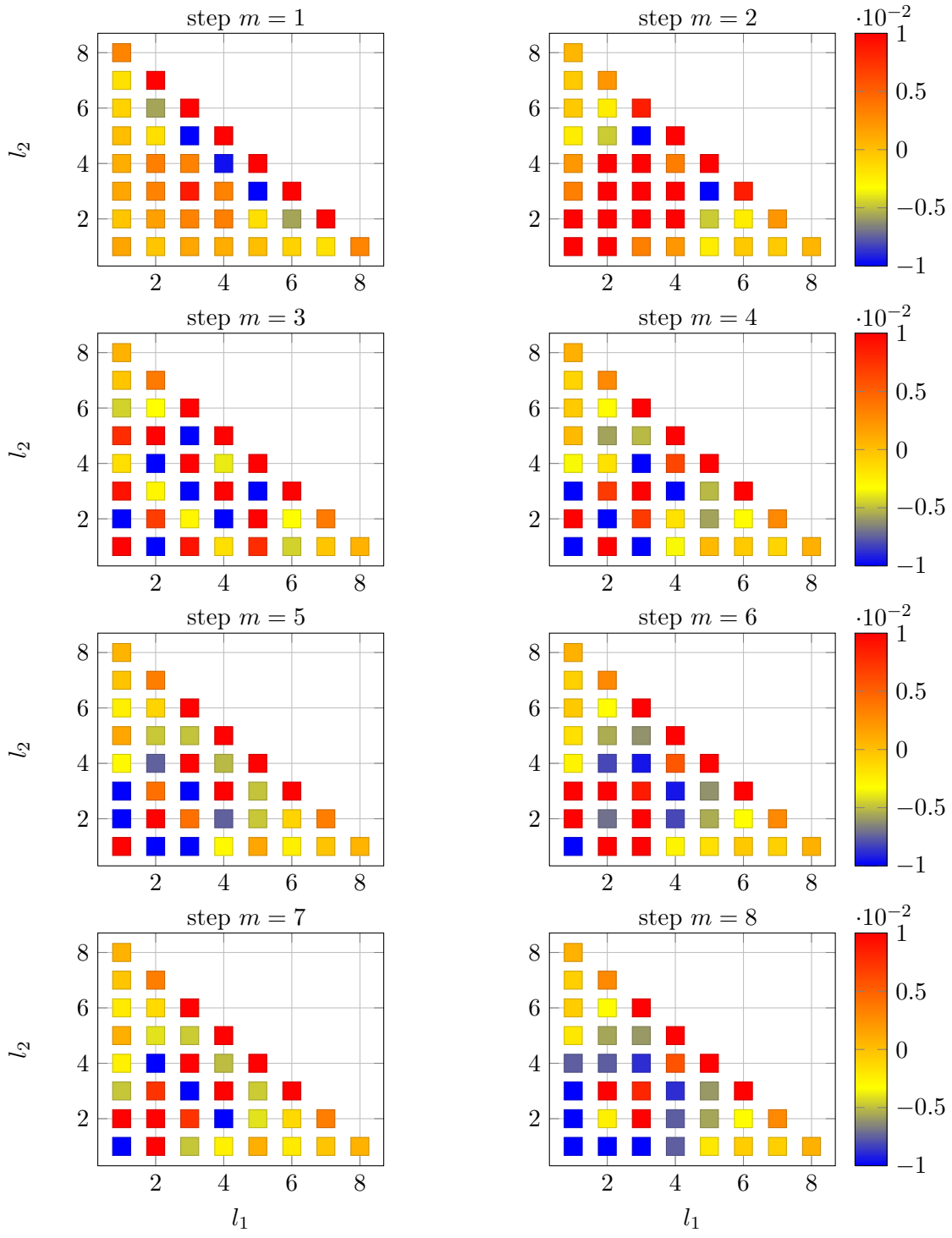

Figure 6.11: Color-coded OptiCom scaling parameters for our two-dimensional example in the iteration steps $m = 1, \ldots, 8$. The parameters that exceed the range of the colorbar are clipped to the nearest value

Three-dimensional example

Now we present a three-dimensional example. The parameters are

$$T = 0.5, K = 3.0, r = 0.0\%, \sigma_1 = 0.354372, \sigma_2 = 0.194484, \sigma_3 = 0.106735$$

and we have two ($R = 2$) jump-terms with

$$\begin{bmatrix} k=1 & : & \lambda_1 = 0.750798 & \rho_{1,1} = 0.290905 & \alpha_{1,1} = 5.662934 & \beta_{1,1} = 3.842384 \\ & & & \rho_{1,2} = 0.510828 & \alpha_{1,2} = 12.910091 & \beta_{1,2} = 7.272062 \\ & & & \rho_{1,3} = 0.892947 & \alpha_{1,3} = 19.776693 & \beta_{1,3} = 15.927953 \\ k=2 & : & \lambda_1 = 0.908148 & \rho_{1,1} = 0.902834 & \alpha_{1,1} = 3.158346 & \beta_{1,1} = 6.030037 \\ & & & \rho_{1,2} = 0.845751 & \alpha_{1,2} = 30.392669 & \beta_{1,2} = 9.085672 \\ & & & \rho_{1,3} = 0.377994 & \alpha_{1,3} = 17.501376 & \beta_{1,3} = 18.491728 \end{bmatrix} .$$

The parameters have been randomly generated and belong to an ANOVA subspace of the ten-dimensional example we deal with in Section 6.4. As opposed to the two-dimensional example the resulting bilinear form $a(\cdot,\cdot)$ is no longer symmetric, but the OptiCom still works. Instead of the energy based formulation (5.87) we can regard $P_{\boldsymbol{\omega}^{(m)}} e^{(m)} = \sum_{\mathbf{l}\in\mathcal{I}} \omega_{\mathbf{l}}^{(m)} T_{\mathbf{l}}^V e^{(m)}$ as the Galerkin projection of the error $e^{(m)}$ in iteration step m onto the space $\operatorname{span}\{T_{\mathbf{l}}^V e^{(m)} : \mathbf{l} \in \mathcal{I}\}$, i.e.,

$$a(P_{\boldsymbol{\omega}^{(m)}} e^{(m)}, T_{\mathbf{l}}^V e^{(m)}) = a(u - u^{(m)}, T_{\mathbf{l}}^V e^{(m)}) = F(T_{\mathbf{l}}^V e^{(m)}) - a(u^{(m)}, T_{\mathbf{l}}^V e^{(m)}) \qquad \forall\, \mathbf{l} \in \mathcal{I} .$$

We now choose the sparse grid index set $\mathcal{S}_6^3$ for the discretization of our three-dimensional space. In Figure 6.12, we see that the convergence of the residual starts off faster for the OptiCom than for the fixed weights, but both methods achieve roughly the same error reduction rate. This holds both for the steepest descent as well as for CG method. In Figure 6.13 we see, similarly to the two-dimensional case, that the scaling parameters are positive on the outer surface but partly negative beneath. In Figure 6.14, the scaling parameters found by the OptiCom method are depicted. They change from iteration step to iteration step, but appear slightly less erratic than the OptiCom scaling parameters in two-dimensions, see Figure 6.11.

In its current form, the sparse grid PIDE solver can tackle even four or five dimensional problems, but it has characteristics that inhibit its use in high dimensions. First, there is a 2^d constant in the runtime complexity of the operator application, see Subsection 4.3.2. Secondly, there is a another 2^d constant that stems from the redundancy of the generating system with respect to a basis, see (5.3). Thirdly, the payoff function (6.7) lacks the H^2_{mix} regularity typically required by sparse grids, which makes the approximation of the initial condition difficult. However, we can bridge the gap to ten-dimensional problems by the ANOVA approximation method discussed in Chapter 2. In the next section, we introduce a ten-dimensional model and compute ANOVA approximation errors using a simple Monte Carlo technique. A full ANOVA-PIDE approach will follow in Section 6.5

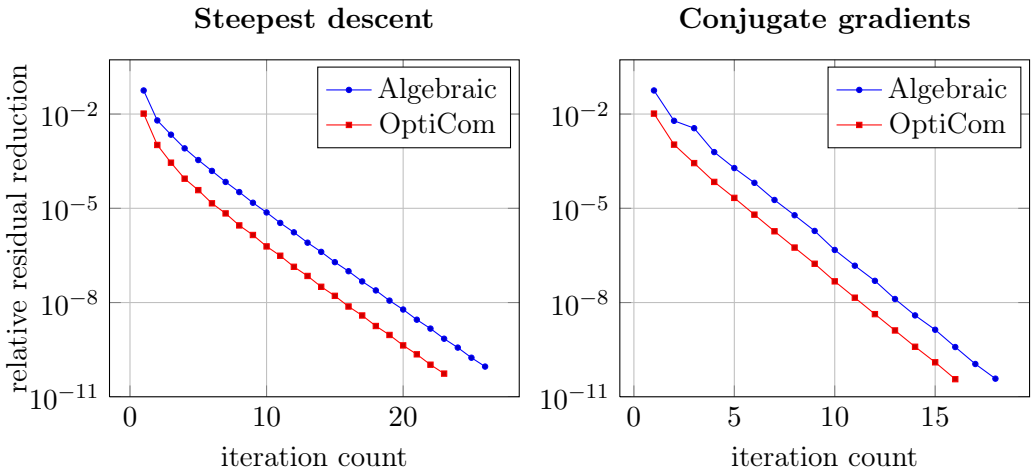

Figure 6.12: These plots show the convergence of the residual for the steepest descent (left) and conjugate gradient (right) method with algebraic scaling parameters and the non-linear OptiCom in our three-dimensional example problem

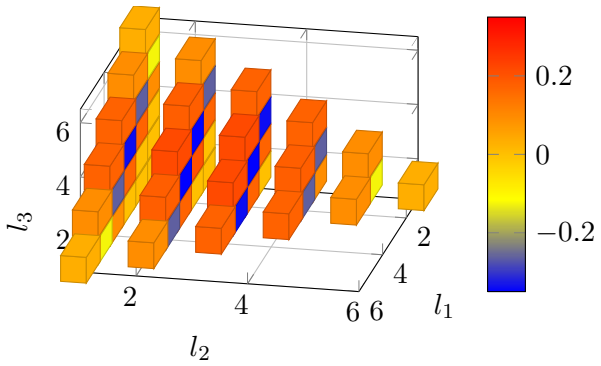

Figure 6.13: Visualization of the sparse grid index set $\mathcal{S}_3^6$, where the algebraic scaling parameters $\omega_{\mathbf{l}}^V, \mathbf{l} \in \mathcal{S}_3^6$, are color-coded. The parameters that exceed the range of the colorbar are clipped to the nearest value

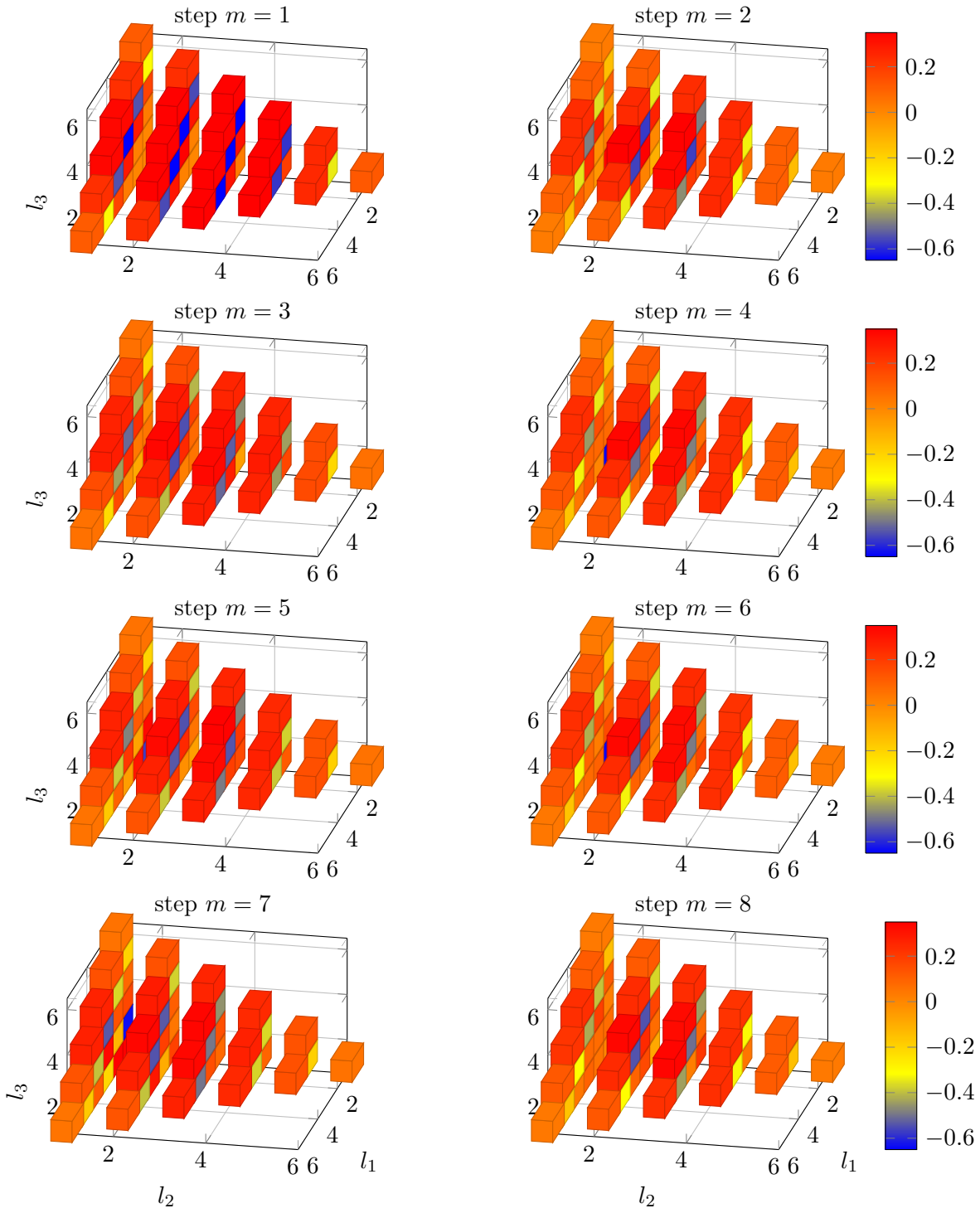

Figure 6.14: Color-coded OptiCom scaling parameters for our three-dimensional example in the iteration steps $m = 1, \ldots, 8$. The parameters that exceed the range of the colorbar are clipped to the nearest value

6.4 The ANOVA approximation for a ten-dimensional Kou model

In this section, we investigate the ANOVA approximation errors for a ten-dimensional jump-diffusion model and a Brownian motion model. We mainly consider models with the standard deviations

$$(\sigma(X_1(t)),\dots,\sigma(X_{10}(t))) = \sqrt{t}(0.501158, 0.275041, 0.150946, 0.082841, 0.045464, \\ 0.024951, 0.013694, 0.007515, 0.004124, 0.002264)\ , \tag{6.12}$$

which decay exponentially with the dimension. In the following, we assume a multivariate jump-diffusion process $\mathbf{X}$, where, according to (6.6), half of the variance implied by (6.12) stems from a diffusion process and the other half from two randomly chosen Kou jump-terms. In the end, we use the parameters

$$T = 0.5, K = 10.0, r = 0.0\%\ ,$$

the diffusion coefficients

$$(\sigma_1,\dots,\sigma_{10}) = (0.354372, 0.194484, 0.106735, 0.058577, 0.032148, \\ 0.017643, 0.009683, 0.005314, 0.002916, 0.001601)$$

and two ($R = 2$) jump terms with

$$\begin{bmatrix}
k=1 \ : \ \lambda_1 = 0.750798 & \rho_{1,1} = 0.290905 & \alpha_{1,1} = 5.662934 & \beta_{1,1} = 3.842384 \\
& \rho_{1,2} = 0.510828 & \alpha_{1,2} = 12.910091 & \beta_{1,2} = 7.272062 \\
& \rho_{1,3} = 0.892947 & \alpha_{1,3} = 19.776693 & \beta_{1,3} = 15.927953 \\
& \rho_{1,4} = 0.896293 & \alpha_{1,4} = 23.249837 & \beta_{1,4} = 30.704574 \\
& \rho_{1,5} = 0.125585 & \alpha_{1,5} = 50.429905 & \beta_{1,5} = 644.231495 \\
& \rho_{1,6} = 0.207243 & \alpha_{1,6} = 142.561945 & \beta_{1,6} = 56.620768 \\
& \rho_{1,7} = 0.051467 & \alpha_{1,7} = 176.407227 & \beta_{1,7} = 263.993395 \\
& \rho_{1,8} = 0.440810 & \alpha_{1,8} = 262.073528 & \beta_{1,8} = 591.184436 \\
& \rho_{1,9} = 0.029876 & \alpha_{1,9} = 586.033377 & \beta_{1,9} = 2017.337902 \\
& \rho_{1,10} = 0.456833 & \alpha_{1,10} = 1098.127950 & \beta_{1,10} = 1065.247317 \\
k=2 \ : \ \lambda_2 = 0.908148 & \rho_{2,1} = 0.902834 & \alpha_{2,1} = 3.158346 & \beta_{2,1} = 6.030037 \\
& \rho_{2,2} = 0.845751 & \alpha_{2,2} = 30.392669 & \beta_{2,2} = 9.085672 \\
& \rho_{2,3} = 0.377994 & \alpha_{2,3} = 17.501376 & \beta_{2,3} = 18.491728 \\
& \rho_{2,4} = 0.092217 & \alpha_{2,4} = 38.886723 & \beta_{2,4} = 16.366405 \\
& \rho_{2,5} = 0.653411 & \alpha_{2,5} = 52.861348 & \beta_{2,5} = 63.810803 \\
& \rho_{2,6} = 0.557841 & \alpha_{2,6} = 78.090436 & \beta_{2,6} = 205.701005 \\
& \rho_{2,7} = 0.361565 & \alpha_{2,7} = 164.325048 & \beta_{2,7} = 408.536213 \\
& \rho_{2,8} = 0.225055 & \alpha_{2,8} = 335.090239 & \beta_{2,8} = 507.977158 \\
& \rho_{2,9} = 0.406520 & \alpha_{2,9} = 553.133908 & \beta_{2,9} = 1006.039385 \\
& \rho_{2,10} = 0.468940 & \alpha_{2,10} = 901.915910 & \beta_{2,10} = 2992.378924
\end{bmatrix}.$$

Note that the three-dimensional example discussed in Subsection 6.3.3 is the $\{1,2,3\}$-marginal of this ten-dimensional model.

We focus on the ANOVA approximation method described in Subsection 2.3.3. That means

we approximate

$$u(\tau, \mathbf{x}) = \mathbb{E}[g(\mathbf{X}(\tau) + \mathbf{x})]$$

by

$$u_{\mathfrak{S}}(\tau, \mathbf{x}) = \sum_{\mathfrak{n} \in \mathfrak{S}} \Big(\sum_{\mathfrak{m} \in \mathfrak{S}, \mathfrak{m} \supset \mathfrak{n}} (-1)^{\#\mathfrak{m} - \#\mathfrak{n}} \Big) \mathbb{E}[[Q_{\mathfrak{n}}^V g](\mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}})] \,, \tag{6.13}$$

where $Q_{\mathfrak{n}}^V g$ either denotes

- the anchor ANOVA (see Example 2.15)

$$[Q_{\mathfrak{n}}^V g](\mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}}) = g(\log \mathbf{a}_{\mathfrak{D} \setminus \mathfrak{n}} + \mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}}) \tag{6.14}$$

  at anchor point $\mathbf{a} = (1.0, \ldots, 1.0)$, i.e., $\mu_i = \delta_{a_i}$ and $\gamma_i = 1$ for $i = 1, \ldots, d$,

- or the weighted ANOVA (see Example 2.16)

$$[Q_{\mathfrak{n}}^V g](\mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}}) = \int_{\mathbb{R}^{\mathfrak{D} \setminus \mathfrak{m}}} g(\mathbf{z}_{\mathfrak{D} \setminus \mathfrak{n}} + \mathbf{X}_{\mathfrak{n}}(\tau) + \mathbf{x}_{\mathfrak{n}}) \mu_{\mathfrak{D} \setminus \mathfrak{n}}(\mathrm{d}\mathbf{z}_{\mathfrak{D} \setminus \mathfrak{n}}) \tag{6.15}$$

  with $\gamma_i = 1$ for $i = 1, \ldots, d$ and Gaussian measures μ_i. In our experiments, we choose the Gaussian measures such that their first two moments match those of the respective $X_i(T)$. The idea behind that is that $\eta_{\mathfrak{m}}^{\tau} \otimes \mu_{\mathfrak{D} \setminus \mathfrak{m}}$ in (2.80) is supposed to be a close approximation to $\eta_{\mathfrak{D}}^T$.

In the following, we perform several experiments in order to gain some insight into the anchor ANOVA (6.14). To this end, we need to evaluate how large the resulting approximation error is at $\tau = T$. For the sake of brevity we drop the index τ from now on and define

$$e_{\mathfrak{S}}^{\mathrm{A}}(\mathbf{x}) := |u(\mathbf{x}) - u_{\mathfrak{S}}(\mathbf{x})| \,. \tag{6.16}$$

As u and $u_{\mathfrak{S}}$ in (6.16) are not known exactly, we have to resort to a numerical method to compute $e_{\mathfrak{S}}^{\mathrm{A}}(\mathbf{x})$. In this section, we use the robust albeit slow Monte Carlo simulation technique. We will employ a full ANOVA-PIDE approach in Section 6.5. Let us denote the Monte Carlo approximations of $u(\mathbf{x})$ and $u_{\mathfrak{S}}(\mathbf{x})$ with N sample runs by $u^N(\mathbf{x})$ and $u_{\mathfrak{S}}^N(\mathbf{x})$, respectively. In order to compute $u_{\mathfrak{S}}^N(\mathbf{x})$, we rely on the description of the ANOVA decomposition as an alteration of our integration measure as in (2.80), i.e.

$$\mathbb{E}[(Q_{\mathfrak{m}}^V g)(\mathbf{x}_{\mathfrak{m}} + \mathbf{X}_{\mathfrak{m}}(T))] = \int_{R^{\mathfrak{D}}} g(\mathbf{x}_{\mathfrak{m}} + \mathbf{y}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D} \setminus \mathfrak{m}}) (\eta_{\mathfrak{m}}^T \otimes \mu_{\mathfrak{D} \setminus \mathfrak{m}})(\mathrm{d}(\mathbf{y}_{\mathfrak{m}} \diamond \mathbf{z}_{\mathfrak{D} \setminus \mathfrak{m}})) \,. \tag{6.17}$$

So instead of computing first the projection $Q_{\mathfrak{m}}^V g$ and then the convolution with the marginal measure $\eta_{\mathfrak{m}}^T$, we interpret the expression $\mathbb{E}[[Q_{\mathfrak{m}}^V g](\mathbf{x}_{\mathfrak{m}} + \mathbf{X}_{\mathfrak{m}}(T))]$ from (6.17) as the convolution of g with the measure $\eta_{\mathfrak{m}} \otimes \mu_{\mathfrak{D} \setminus \mathfrak{m}}$ evaluated at $\mathbf{x}_{\mathfrak{m}}$. This seems to be a small change, but makes simulation a lot easier. See Algorithm 6 for the pseudocode of this method. Now, the triangle

Algorithm 6 ANOVA Monte Carlo-algorithm

Input: initial condition g, position $\mathbf{x}$,
measures $\mu_{\mathfrak{D}}$ and $\eta_{\mathfrak{D}}$,
number of simulations N,
ANOVA index set $\mathfrak{S}$
Output: approximations $r_{\mathfrak{m}} = \mathbb{E}[[Q^V_{\mathfrak{m}} g](\mathbf{x}_{\mathfrak{m}} + \mathbf{X}_{\mathfrak{m}}(T))]$ for all $\mathfrak{m} \in \mathcal{S}$

```
(r_m)_{m∈S} ← 0                                   ▷ Initialize return vector
for all n = 1, ..., N do
    draw y ~ η_D^T                                ▷ Sample from the Lévy process measure
    draw z ~ μ_D                                  ▷ Sample from the ANOVA measure
    for all m ∈ S do
        r_m ← r_m + g(x_m + y_m ◇ z_{D\m})        ▷ Evaluate initial condition
    end for
end for
for all m ∈ S do
    r_m ← r_m / N                                 ▷ Divide by the number of simulations
end for
```

inequality tells us that

$$\begin{aligned}\mathbb{E}[|u^N(\mathbf{x}) - u^N_{\mathfrak{S}}(\mathbf{x})|] &\leq \mathbb{E}\left[|u^N(\mathbf{x}) - u(\mathbf{x})| + |u(\mathbf{x}) - u_{\mathfrak{S}}(\mathbf{x})| + |u_{\mathfrak{S}}(\mathbf{x}) - u^N_{\mathfrak{S}}(\mathbf{x})|\right] \\ &\leq |u(\mathbf{x}) - u_{\mathfrak{S}}(\mathbf{x})| + \frac{\sigma(u^N(\mathbf{x})) + \sigma(u^N_{\mathfrak{S}}(\mathbf{x}))}{\sqrt{N}} \overset{N\to\infty}{\longrightarrow} e^A_{\mathfrak{S}}(\mathbf{x}) \,,\end{aligned} \tag{6.18}$$

where $\sigma(u^N(\mathbf{x}))$ and $\sigma(u^N_{\mathfrak{S}}(\mathbf{x}))$ denote the standard deviations of the Monte Carlo samples. So plotting the quantity

$$e^{\mathrm{A}}_{\mathfrak{S},N}(\mathbf{x}) := |u^N(\mathbf{x}) - u^N_{\mathfrak{S}}(\mathbf{x})|$$

for growing N will eventually reveal the ANOVA approximation error at point $\mathbf{x}$ with a convergence rate of, on average, $\frac{1}{2}$.

Since Monte Carlo simulations are able to evaluate $e^{\mathrm{A}}_{\mathfrak{S},N}$ only at predetermined points, we content ourselves with measuring the error at the money

$$\mathbf{s}^{\mathrm{at}} = (1.0, \ldots, 1.0) \,,$$

out of the money

$$\begin{aligned}\mathbf{s}^{\mathrm{out}} &= \mathbf{s}^{\mathrm{at}} + (\sigma(X_1(T)), \ldots, \sigma(X_d(T))) \\ &= (1.354372, 1.194484, 1.106735, 1.058577, 1.032148, \\ &\quad 1.017643, 1.009683, 1.005314, 1.002916, 1.001601)\end{aligned}$$

and in the money

$$\begin{aligned}\mathbf{s}^{\text{in}} &= \mathbf{s}^{\text{at}} - (\sigma(X_1(T)), \dots, \sigma(X_d(T))) \\ &= (0.645628, 0.805516, 0.893265, 0.941423, 0.967852, \\ &\qquad 0.982357, 0.990317, 0.994686, 0.997084, 0.998399)\ .\end{aligned}$$

Note that the transformation to the corresponding log-space coordinates $\mathbf{x}^{\text{at}}$, $\mathbf{x}^{\text{out}}$ and $\mathbf{x}^{\text{in}}$ is performed according to Subsection 3.6. Here and in the following, we only consider absolute errors. This makes error analysis much easier but we have to bear in mind that the same absolute accuracy is better in relative terms for in the money options with a high inner value than for relatively cheap out of the money options. In fact, we estimate the prices of our at, out, and in the money options using 1.0×10^9 Monte Carlo simulations at 0.1697311, 0.0187102 and 0.5817215, respectively.

In Figure 6.15, the quantity $e^{\text{A}}_{\mathfrak{G},N}(\mathbf{x})$ is depicted for our ten-dimensional jump-diffusion model evaluated at the money, i.e., $\mathbf{x} = \mathbf{x}^{\text{at}}$, and different numbers of samples N. The error bars indicate one empirical standard deviation of this stochastic quantity. We see that the error bars decline for higher values of N, and we end up with a rather accurate estimate of $e^{\text{A}}_{\mathfrak{G}}(\mathbf{x}^{\text{at}}) = |u(\mathbf{x}^{\text{at}}) - u_{\mathfrak{G}}(\mathbf{x}^{\text{at}})|$.

6.4.1 Comparison for models with different decay rates

Figure 6.15 shows the estimation of the ANOVA error $e^{\text{A}}_{\mathfrak{G}}(\mathbf{x}^{\text{at}}) = |u(\mathbf{x}^{\text{at}}) - u_{\mathfrak{G}}(\mathbf{x}^{\text{at}})|$ for our jump-diffusion model. We repeat this procedure for superposition dimensions $d_s = 0, 1, 2$ and truncation dimensions $d_t = 1, 2, \dots, 9 - d_s$ with $N = 1.0 \times 10^9$ samples and plot the estimated error $e^{\text{A}}_{\mathfrak{G}}(\mathbf{x}^{\text{at}})$. See Figure 6.16 for a comparison of the jump-diffusion model presented at the beginning of this section with a Brownian motion model with standard deviations (6.12). What we see is a smooth error convergence for the Brownian motion: The error appears to decrease exponentially with the truncation dimension d_t, and the slope of the curve, i.e., the exponential rate, is higher the higher the superposition dimension is. The jump-diffusion model, however, leads to more erratic results, but in principle increasing d_s and d_t improves the approximation.

Our choice of standard deviations (6.12) is somewhat arbitrary. It mainly depends on what kind of decay we can expect after diagonalizing our initial covariance matrix $\mathbf{Q}$. In Figure 6.17 we did the same computation for a process that has no decay at all with

$$\sigma_i = 0.189581 \qquad \forall\ i = 1, \dots, 10 \tag{6.19}$$

and a stronger decay than in (6.12) with

$$\begin{aligned}(\sigma_1, \dots, \sigma_{10}) = (&0.557467, 0.205081, 0.075445, 0.027755, 0.010210, \\ &0.003756, 0.001382, 0.000508, 0.000187, 0.000069)\ .\end{aligned} \tag{6.20}$$

Note that both sets of standard deviations (6.19) and (6.20) result in processes whose sum of variances $\sum_{i=1}^{d} \operatorname{var}(X_i(1))$ match those of our jump-diffusion example (6.12). This ensures a basic level of comparability. We see that the ANOVA approximation error declines faster when higher dimensions are less important. This is an important observation, as we typically observe

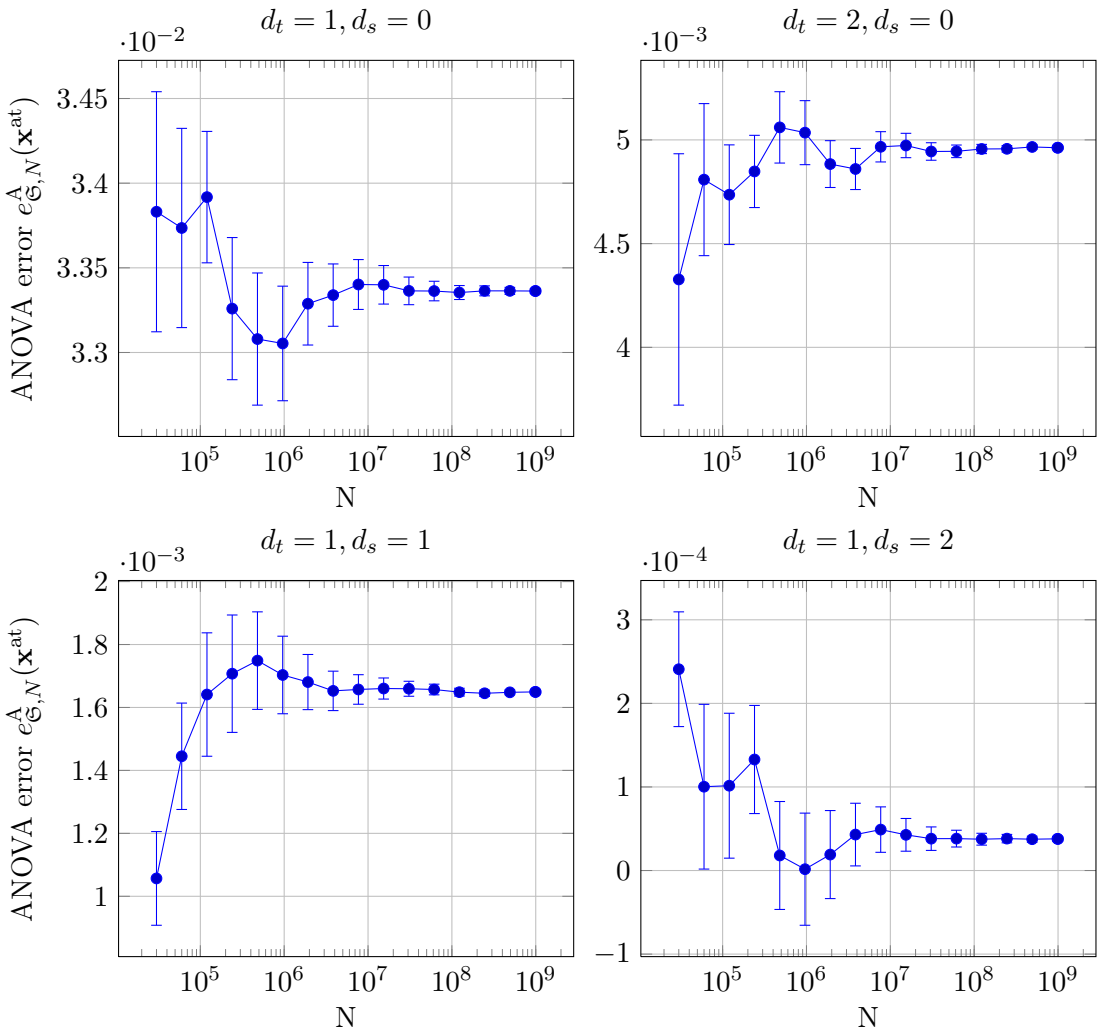

Figure 6.15: Estimated ANOVA error at the money for different parameters d_t and d_s and an increasing numbers of samples N. The error bars indicate one standard deviation of the estimator

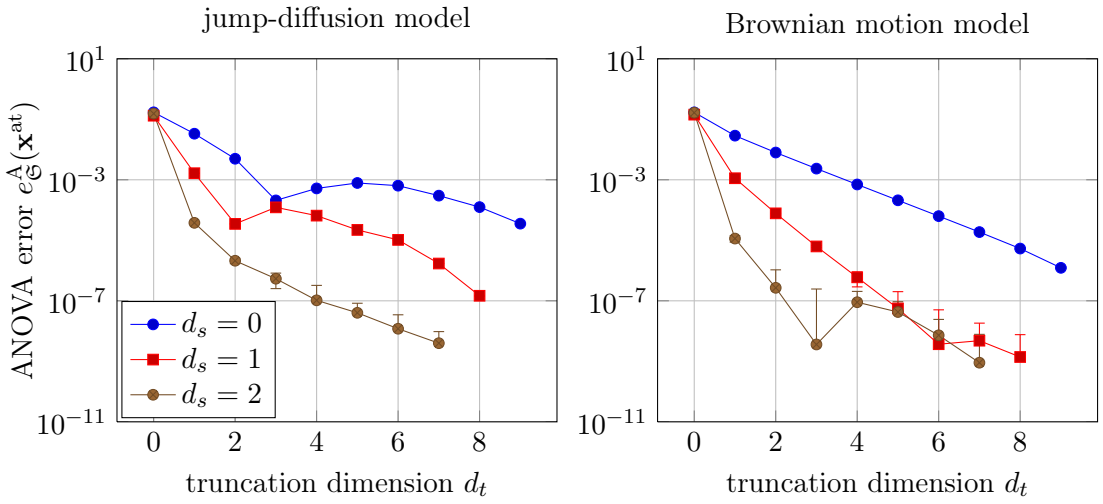

Figure 6.16: The jump-diffusion model and its estimated ANOVA approximation errors for different superposition dimensions d_s and truncation dimensions d_t compared to a similar model only based on the Brownian motion. Note that some of the error bars are small or too close to zero to be displayed

a decay in the importance of the dimensions after the covariance matrix has been diagonalized, see (2.25). As has been been noted in [RW13], a payoff function g which counteracts the decay of the spectrum of the covariance matrix is conceivable but unlikely.

6.4.2 Evaluation at, in and out of the money

So far, we have used the anchor ANOVA with an anchor point $\mathbf{a} = (1.0, \dots, 1.0)$ and we evaluated our error at the same position, i.e., $\mathbf{s}^{\mathrm{at}} = (1.0, \dots, 1.0)$, or to be precise, at the respective log-space coordinate $\mathbf{x}^{\mathrm{at}}$. When looking at an $\mathfrak{m}$-marginal, this amounts to replacing the processes $X_i, i \in \mathfrak{D} \setminus \mathfrak{m}$, by "processes" that remain constant, namely the respective components of the anchor point $\mathbf{a}$. This choice produces an error that gets worse when the point of evaluation $\mathbf{s}$ does not coincide with the anchor point $\mathbf{a}$. This is exactly what we want to examine in this set of experiments by evaluating the error also at $\mathbf{x}^{\mathrm{out}}$ and $\mathbf{x}^{\mathrm{in}}$.

Furthermore, we compare the anchor ANOVA (6.14) with the weighted ANOVA (6.15), see Figure 6.18. In Figure 6.16, we learned that error estimates of our jump-diffusion model are more erratic than those of the pure Brownian motion ones. So we repeat all experiments with a diffusion-only model in Figure 6.19. Note there that the weighted ANOVA at the money results in an error which is analytically zero as μ_i coincides with the distribution of the respective component of X_i. However, due to the Monte Carlo simulation, we still see a non-zero error.

The observation is that the weighted ANOVA, i.e., approximating X_i by a similar ANOVA measure, leads to an improvement when we evaluate at the same position as the anchor point. When we consider in the money or out of the money options, the difference between the left and right columns of Figures 6.18 and 6.19 is small. So we come up with the recommendation to use the anchor ANOVA when evaluating *anywhere else* than the anchor point, but to use, if

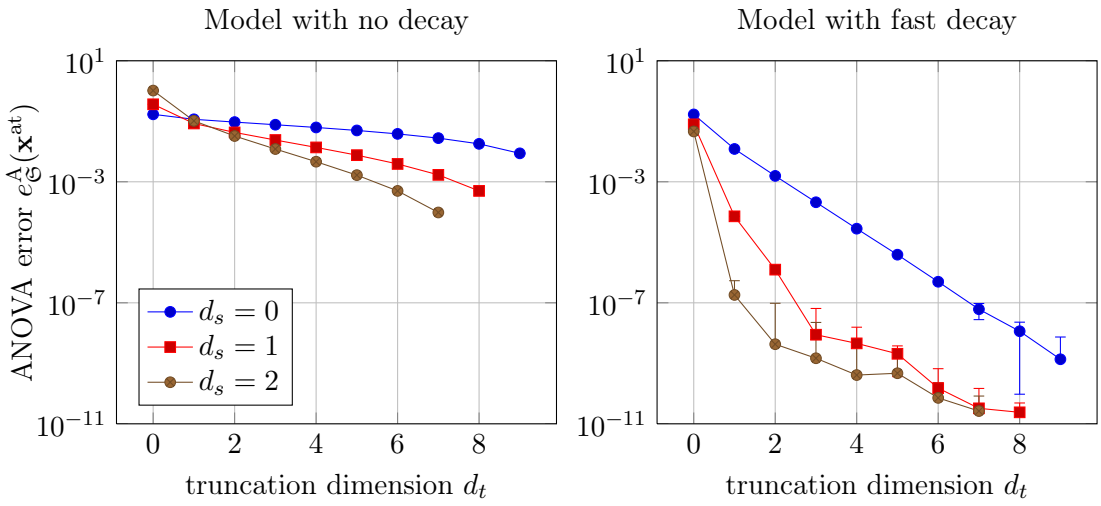

Figure 6.17: These plots show the estimated ANOVA approximation errors when a Brownian motion process has no decay (6.19) in the importance of its dimensions (left) and a fast decay (6.20) (right)

available, the weighted ANOVA when evaluating *at* the anchor point.

6.5 The ANOVA-PIDE approach for our ten-dimensional problem

In this section, we single out five promising anchor ANOVA approximations and apply them to the initial condition of the problem given in Section 6.4. Then, we use our PIDE solver to compute all emerging subproblems and recombine the results as in (6.13), i.e.,

$$\tilde{u}_{\mathfrak{G}}(\mathbf{x}) = \sum_{\mathfrak{n}\in\mathfrak{G}} \Big(\sum_{\mathfrak{m}\in\mathfrak{G},\mathfrak{m}\supset\mathfrak{n}} (-1)^{\#\mathfrak{m}-\#\mathfrak{n}} \Big) \tilde{u}_{\mathfrak{n}}(\mathbf{x}) \ ,$$

where $\tilde{u}_{\mathfrak{n}}(\mathbf{x})$ is the numerical solution of our PIDE (3.25) at point in time $\tau = T$ with initial condition $Q^V_{\mathfrak{n}} g$. In doing so, we obtain the numerical solution of a ten-dimensional PIDE.

Our solutions $\tilde{u}_{\mathfrak{n}}(\mathbf{x}), \mathfrak{n} \in \mathfrak{G}$, are subject to space discretization, time discretization and localization errors, and we quantify these by

$$e^{\mathrm{P}}_{\mathfrak{G}}(\mathbf{x}) := |\tilde{u}_{\mathfrak{G}}(\mathbf{x}) - u_{\mathfrak{G}}(\mathbf{x})| \ . \tag{6.21}$$

Note that (6.21) can only be evaluated in the localized domain (3.9). In the end we are interested in the total error

$$e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{G}}(\mathbf{x}) := |\tilde{u}_{\mathfrak{G}}(\mathbf{x}) - u(\mathbf{x})|$$

of our ANOVA-PIDE approach. A simple triangle equality

$$e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{G}}(\mathbf{x}) = |\tilde{u}_{\mathfrak{G}}(\mathbf{x}) - u_{\mathfrak{G}}(\mathbf{x}) + u_{\mathfrak{G}}(\mathbf{x}) - u(\mathbf{x})| \le e^{\mathrm{P}}_{\mathfrak{G}}(\mathbf{x}) + e^{\mathrm{A}}_{\mathfrak{G}}(\mathbf{x}) \tag{6.22}$$

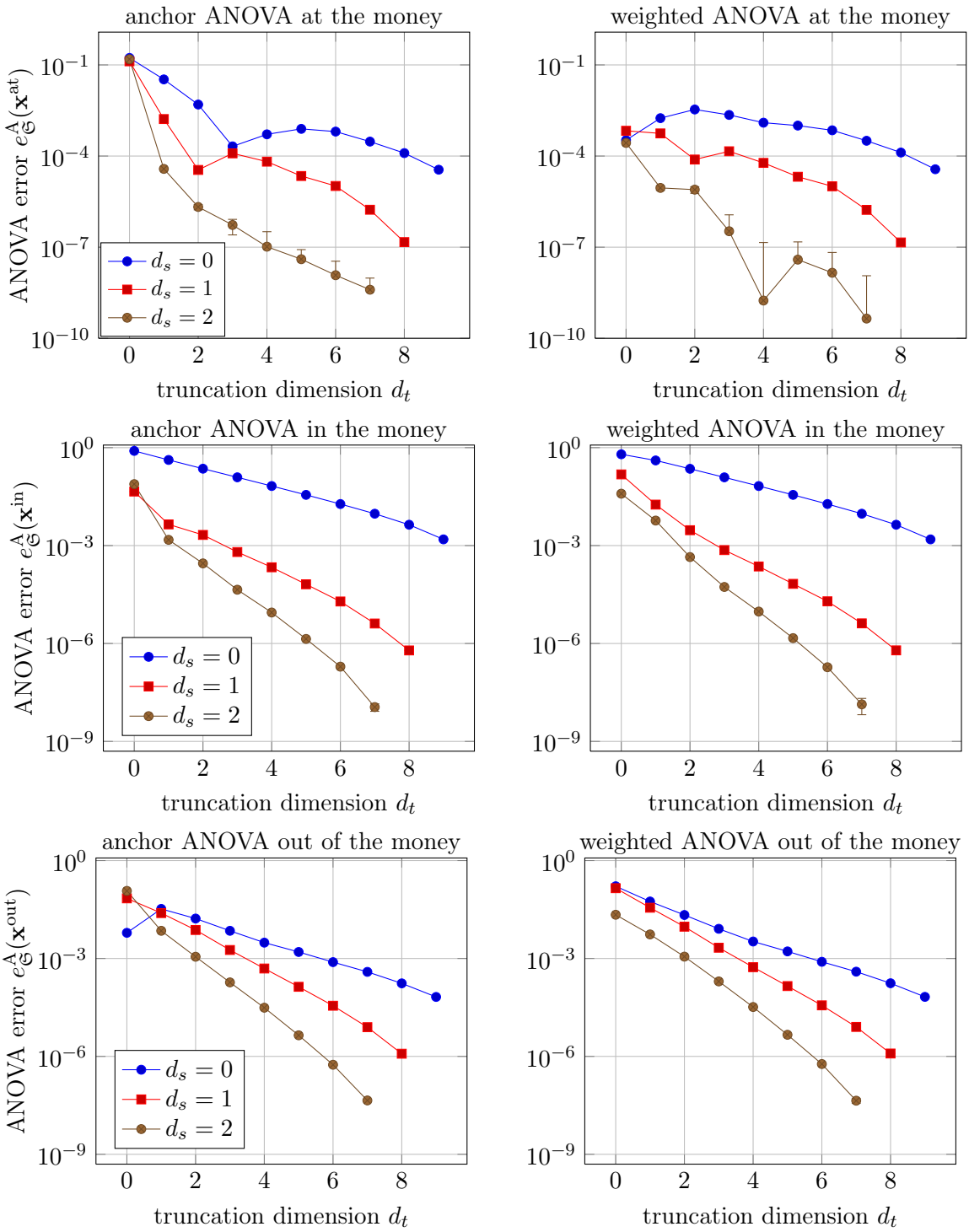

Figure 6.18: Estimated ANOVA approximation errors of our jump-diffusion example for options at, in and out of the money with the anchor and the weighted ANOVA

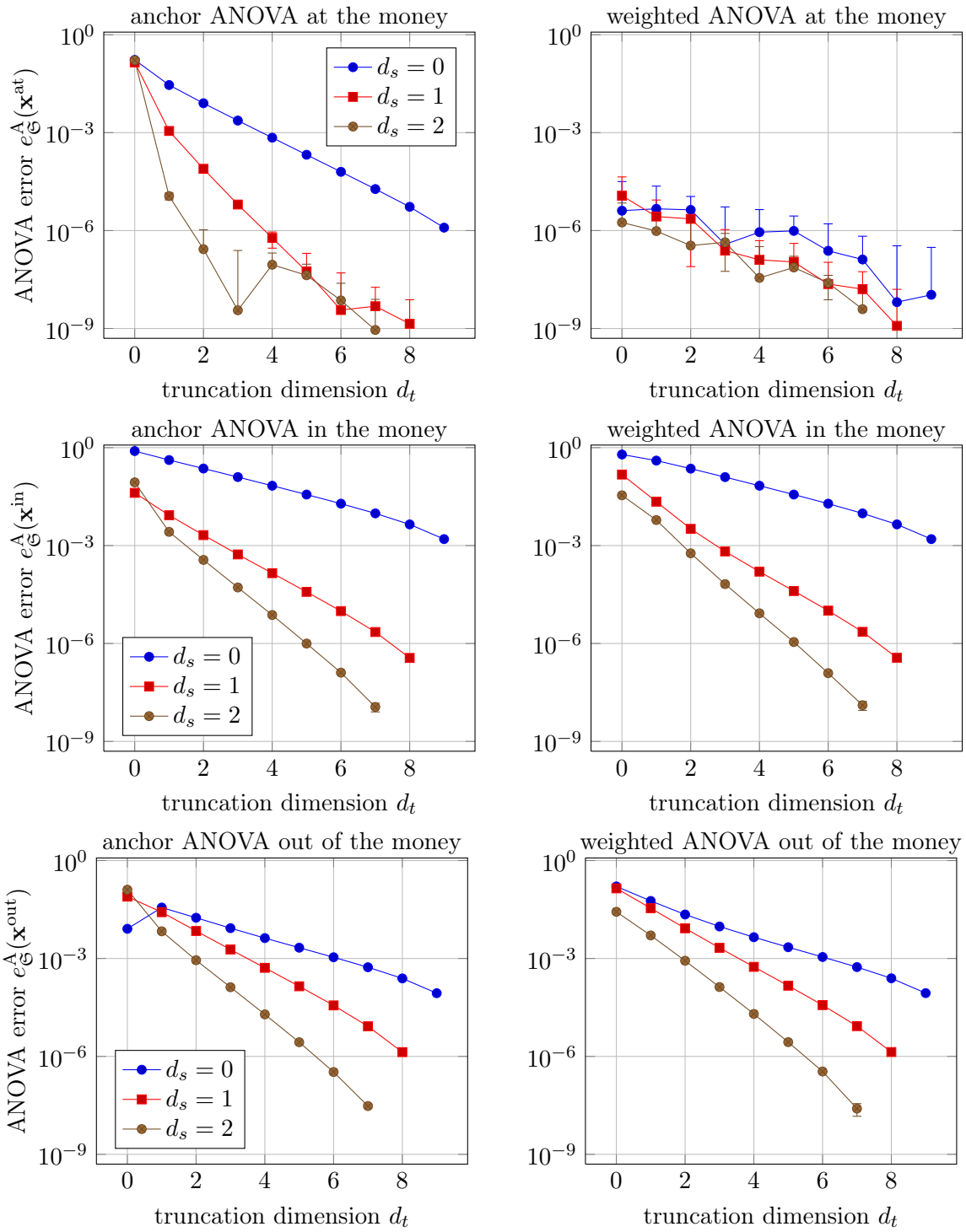

Figure 6.19: Estimated ANOVA approximation errors of our Brownian motion example for options at, in and out of the money with the anchor and the weighted ANOVA

reveals that we need to balance the ANOVA error $e_{\mathfrak{S}}^{\mathrm{A}}$ and discretization error $e_{\mathfrak{S}}^{\mathrm{P}}$ in order to achieve a low total error $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$.

We now give a back-of-the-envelope calculation and discussion of the ANOVA and PIDE errors for superposition dimension $d_s = 0$. Given Figure 6.18, we assume the relationship

$$\|e_{\mathfrak{S}}^{\mathrm{A}}\| \lesssim c_A^{-d_t} \tag{6.23}$$

with $c_A > 1$, an L_∞ or L_2 norm $\|\cdot\|$ and $\mathfrak{S} = \{\mathfrak{m} \subset \mathfrak{D} : \mathfrak{m} \subset \{1, \dots, d_t\}\}$. For the PIDE error we assume

$$\|e_{\mathfrak{S}}^{\mathrm{P}}\| \lesssim c_P^{d_t} N^{-r_P(d_t)} , \tag{6.24}$$

where N is the number of degrees of freedom, r_P is the rate of convergence, which is constant or decreases with rising d_t, and $c_P^{d_t}$ is a constant factor (with respect to N) that grows exponentially with the dimension. See [Gri06] for the case $c_P < 1$, but we typically assume $c_P \geq 1$.

Remark 6.1. Our computational effort stays linear in the number of degrees of freedom N. Therefore, we use it as a proxy for our computational costs. Note that the error estimate (6.24) does not account for time discretization and localization errors, which become relevant for $N \to \infty$. However, to prove our main point, this error estimate is sufficient.

We now want to balance the errors on the right-hand side of (6.22). We assume (6.23) and (6.24) to be sharp and thus we get

$$\|e_{\mathfrak{S}}^{\mathrm{P}}\| \simeq \|e_{\mathfrak{S}}^{\mathrm{A}}\| \Leftrightarrow c_P^{d_t} N^{-r_P(d_t)} \simeq c_A^{-d_t} \Leftrightarrow N \simeq (c_P c_A)^{\frac{d_t}{r_P(d_t)}} .$$

For a rate $r_P(d_t) = \frac{2}{d_t}$, which is common for function approximation with linear splines and full grids, or sparse grids when the required mix-regularity is lacking, this results in

$$N \simeq (c_P c_A)^{\frac{d_t^2}{2}} .$$

For a dimension-independent rate of 2 as we expect it for sparse grids and sufficient regularity of our solution, we get

$$N \simeq (c_P c_A)^{\frac{d_t}{2}} .$$

So in the ideal case the required number of degrees of freedom only grows by $(c_P c_A)^{\frac{d_t}{2}}$ instead of $(c_P c_A)^{\frac{d_t^2}{2}}$. Nonetheless, in both cases the required number of degrees of freedom grows exponentially with the dimension assuming $c_P c_A > 1$. Thus, independently of our rate being $r_P(d_t) = 2$ or $r_P(d_t) = \frac{2}{d_t}$, a sensible choice for d_t is generally quite small. The interpretation is the following: An ANOVA approximation that includes high-dimensional terms has a high-level of exactness, and thus the subproblems need to be solved with the same or greater accuracy to reach the desired target accuracy. So with growing d_t the error norm $\|e_{\mathfrak{S}}^{\mathrm{P}}\|$ needs to decay exponentially. If this is not possible, say the required N becomes too large, it is preferable in terms of the error $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ to use a coarser ANOVA approximation with a smaller d_t instead.

A calculation for superposition $d_s = 1$ and the set

$$\mathfrak{S} = \{\, \mathfrak{u} \cup \{k\} : \mathfrak{u} \subset \{1, \dots, d_t\}, k \in \{d_t + 1, \dots, d\}\}$$

reveals a similar result. With the coefficients from (2.82) it holds that

$$e^{\mathrm{P}}_{\mathfrak{S}}(\mathbf{x}) = |\tilde{u}_{\mathfrak{S}}(\mathbf{x}) - u_{\mathfrak{S}}| = \left| \sum_{k=d_t+1}^{d} \tilde{u}_{\{1,\dots,d_t,k\}}(\mathbf{x}) - (d-d_t-1)\tilde{u}_{\{1,\dots,d_t\}}(\mathbf{x}) - u_{\mathfrak{S}}(\mathbf{x}) \right|$$
$$\leq \sum_{k=d_t+1}^{d} \left| \tilde{u}_{\{1,\dots,d_t,k\}}(\mathbf{x}) - u_{\{1,\dots,d_t,k\}}(\mathbf{x}) \right| + (d-d_t-1) \left| \tilde{u}_{\{1,\dots,d_t\}}(\mathbf{x}) - u_{\{1,\dots,d_t\}}(\mathbf{x}) \right|$$

and thus

$$\|e^{\mathrm{P}}_{\mathfrak{S}}\| \leq (d-d_t) c_P^{d_t+1} N^{-r_P(d_t+1)} + (d-d_t-1) c_P^{d_t} N^{-r_P(d_t)} \leq 2(d-d_t) c_P^{d_t+1} N^{-r_P(d_t+1)} ,$$

so apart from the factor $2(d-d_t)$ the result we get in terms of our PIDE error is close to (6.24) with dimension d_t+1. Figure 6.18 suggests that (6.23) also holds for $d_s > 0$, but with a higher constant c_A. Thus, the situation for $d_s = 1$ is similar to $d_s = 0$, and the number of degrees of freedom needed for balancing $e^{\mathrm{A}}_{\mathfrak{S}}$ and $e^{\mathrm{P}}_{\mathfrak{S}}$ grows exponentially with d_t as in the last paragraph.

Note that an elaborate error discussion for the ANOVA approximation in the context integration tasks can be found in [GH10b].

In the following, we conduct several experiments with different truncation and superposition dimensions (and the corresponding index sets $\mathfrak{S}$) in order to back our considerations empirically. As preparation, we sample 100 points $(\mathbf{s}_i)_{i=1}^{100}$ uniformly from the cube

$$\begin{aligned} &(s_1^{\mathrm{in}}, s_1^{\mathrm{out}}) \times \cdots \times (s_d^{\mathrm{in}}, s_d^{\mathrm{out}}) \\ =&(1.0 - \sigma(X_1(T)), 1.0 + \sigma(X_1(T))) \times \cdots \times (1.0 - \sigma(X_d(T)), 1.0 + \sigma(X_d(T))) . \end{aligned} \tag{6.25}$$

We regard the 100 randomly chosen points as representative for the domain $(s_1^{\mathrm{in}}, s_1^{\mathrm{out}}) \times \cdots \times (s_d^{\mathrm{in}}, s_d^{\mathrm{out}})$, and we compute their corresponding log-space coordinates $(\mathbf{x}_i)_{i=1}^{100}$ according to Section 3.6. We estimate reference values $u(\mathbf{x}_i)$ and $u_{\mathfrak{S}}(\mathbf{x}_i)$ by a Monte Carlo simulation with 1.0×10^9 samples. Then, we start with the actual experiments: We apply the ANOVA approximation to our ten-dimensional jump-diffusion example and solve the emerging subproblems numerically with our sparse grid PIDE method. Now, we can compute the PIDE error $e^{\mathrm{P}}_{\mathfrak{S}}(\mathbf{x}_i)$, the ANOVA approximation error $e^{\mathrm{A}}_{\mathfrak{S}}(\mathbf{x}_i)$ and the total error $e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{S}}(\mathbf{x}_i)$ for all $i = 1, \dots, 100$, and depict them graphically. As we cannot visualize a ten-dimensional space, we simply plot the errors against the distance $|\mathbf{s}_i - \mathbf{a}|_1$ to the anchor point $\mathbf{a} = (1.0, \dots, 1.0)$.

Note that in the previous experiments, we centered our domain around our point of interest and made the domain size dependent on the standard deviations of the stochastic process, see (3.10) and (3.11) in Section 3.3. As setting up the discrete initial condition is getting increasingly difficult for higher dimensions, we precompute it and reuse it for several experiments. This means we no longer use problem-dependent domains, but restrict ourselves to domains in the form $(-\zeta, \zeta)^d$.

In the following, we try various combinations of truncation and superposition dimensions to approximate our ten-dimensional model problem as discussed. In the subsection titles we mention only the computationally most expensive subproblems.

6.5.1 Approximation by one one-dimensional subproblem ($d_t = 1, d_s = 0$)

The case $d_t = 1, d_s = 0$ is the most simple approximation possible: Instead of solving a ten-dimensional PIDE we only compute the first dimension and regard the remaining dimensions as constant. Obviously, this choice results in a high ANOVA approximation error, but the resulting PIDE problem is quite simple. For this computation we choose the Crank-Nicholson method with $\theta = \frac{1}{2}$, 32 time-steps and a computational domain of $(-\zeta, \zeta)$ with $\zeta = 8$. The absolute error of the PIDE solver measured at point $s = 1.0$ is decreasing rapidly, see Figure 6.20.

In Figure 6.21 we see the quantities $e_{\mathfrak{S}}^{\mathrm{P}}$, $e_{\mathfrak{S}}^{\mathrm{A}}$ and $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ evaluated at our reference points $(\mathbf{x}_i)_{i=1}^{100}$ plotted against the distance to the anchor point. The marks of the total error $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ are mostly within the marks of the ANOVA errors $e_{\mathfrak{S}}^{\mathrm{A}}$, which shows that the PIDE error is negligible and the ANOVA approximation error is dominating the total error.

6.5.2 Approximation by nine two-dimensional subproblems ($d_t = 1, d_s = 1$)

In order to improve the accuracy of our result, we now choose $d_s = 1$. Then, next to the one-dimensional problem, we also have to solve nine two-dimensional problems. The PIDE error convergence plots are given in Figure 6.22. Note that here and in the following all computations of the same dimension are given in the same plot. The two-dimensional subspace solutions are computed with the same time discretization parameters as the one-dimensional ones and on the domain $(-\zeta, \zeta)^2$ with $\zeta = 8$. Again, we increase the number of degrees of freedom until the localization and time discretization errors appear.

In Figure 6.23 we see that the ANOVA approximation error $e_{\mathfrak{S}}^{\mathrm{A}}$ is significantly reduced, but it is still larger than the PIDE error $e_{\mathfrak{S}}^{\mathrm{P}}$. As a result, the total error of our ANOVA-PIDE approach $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ is mostly influenced by the ANOVA approximation error. This means we have to improve the ANOVA approximation to achieve a higher accuracy.

6.5.3 Approximation by eight three-dimensional subproblems ($d_t = 2, d_s = 1$)

We increase our truncation dimension and set $d_t = 2$. Now, we have to solve one two-dimensional problem and eight three-dimensional problems. The error convergence plots are given in Figure 6.24. For the three-dimensional computations, we choose a log-space domain of $(-\zeta, \zeta)^3$ with $\zeta = 4.5$. Choosing $\zeta = 4.5$ instead of $\zeta = 8$ as in the one- and two-dimensional subproblems leads to an earlier onset of convergence as discussed in Subsection 6.3.1, but also results in a higher final error: $\sim 10^{-4}$ instead of $\sim 10^{-5}$ as in the two-dimensional computations. Since we would not reach the accuracy of 10^{-5} on a larger domain anyway simply because the required number of degrees of freedom would be too large, it makes sense to settle with a reduced domain size.

In Figure 6.25, we see that the PIDE and ANOVA errors are well balanced. For the first time we are able to compute the solution of a ten-dimensional parabolic jump-diffusion equation with a considerable level of accuracy. We now try to improve the accuracy further by choosing an even better ANOVA approximation.

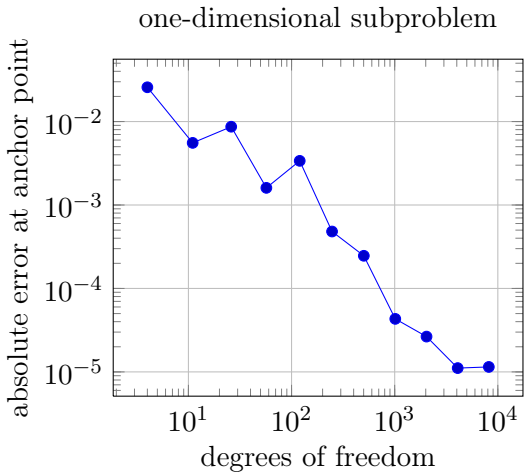

Figure 6.20: Absolute error of the PIDE solver for the one-dimensional subproblem that stems from the ANOVA approximation with $d_t = 1, d_s = 0$

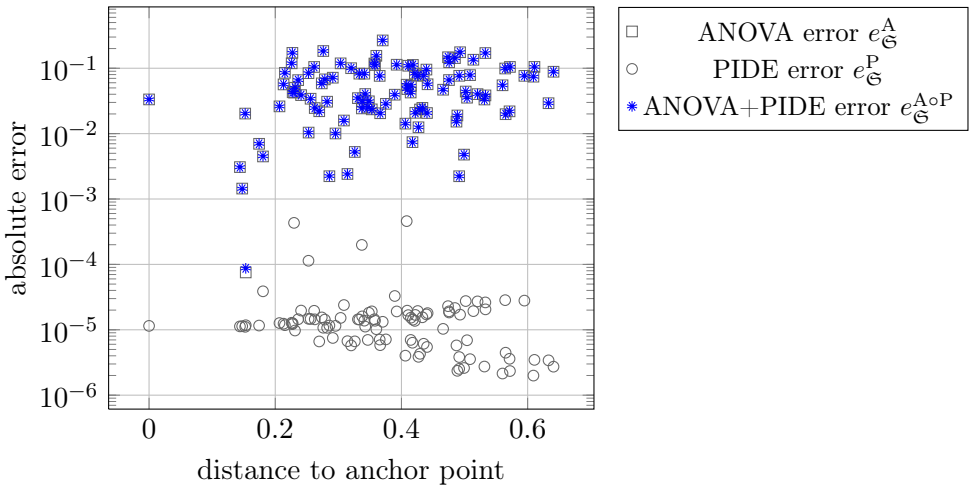

Figure 6.21: This plot shows the errors $e^{\mathrm{A}}_{\mathfrak{G}}$, $e^{\mathrm{P}}_{\mathfrak{G}}$ and $e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{G}}$ at 100 randomly sampled points $(\mathbf{x}_i)_{i=1}^{100}$ for the ANOVA approximation with $d_t = 1, d_s = 0$

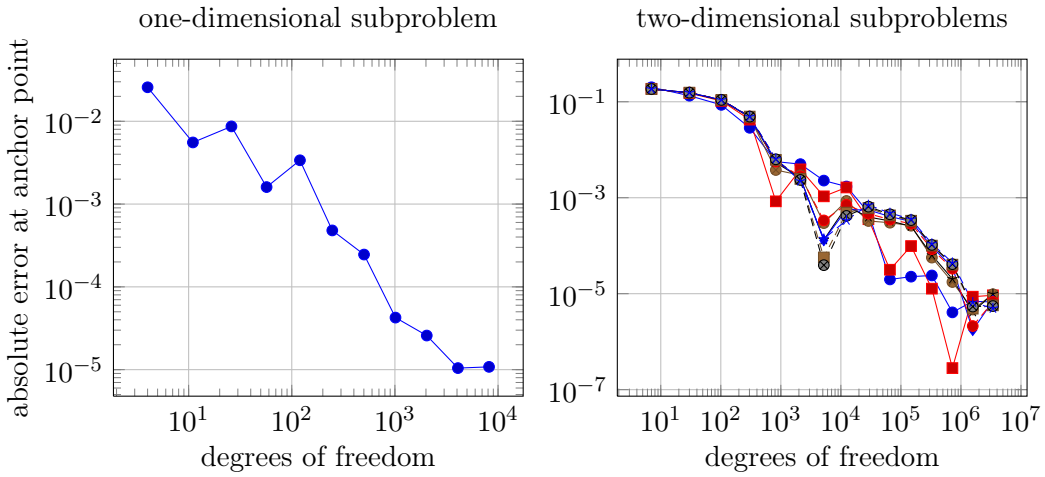

Figure 6.22: Absolute error of the PIDE solver for the one- and two-dimensional subproblems that stem from the ANOVA approximation with $d_t = 1, d_s = 1$

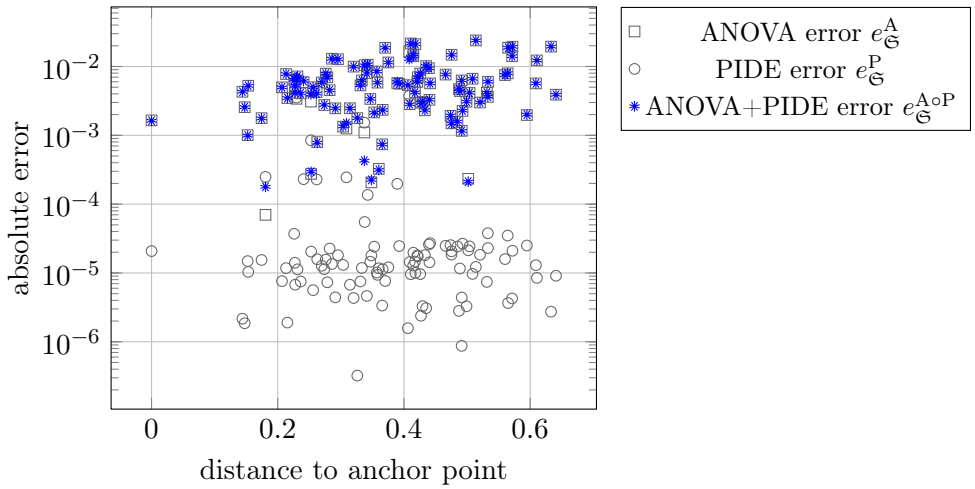

Figure 6.23: This plot shows the errors $e_{\mathfrak{S}}^{\mathrm{A}}$, $e_{\mathfrak{S}}^{\mathrm{P}}$ and $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ at 100 randomly sampled points $(\mathbf{x}_i)_{i=1}^{100}$ for the ANOVA approximation with $d_t = 1, d_s = 1$

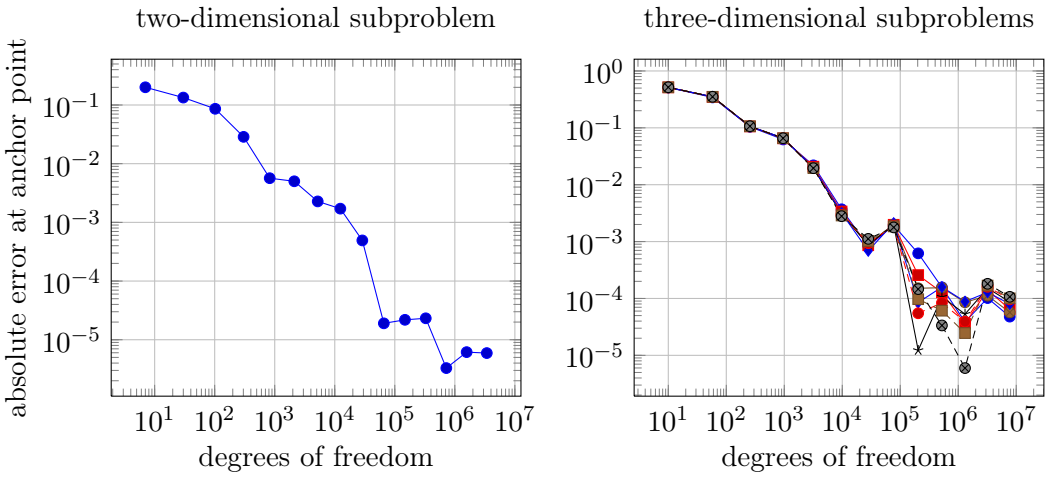

Figure 6.24: Absolute error of the PIDE solver for the two- and three-dimensional subproblems that stem from the ANOVA approximation with $d_t = 2, d_s = 1$

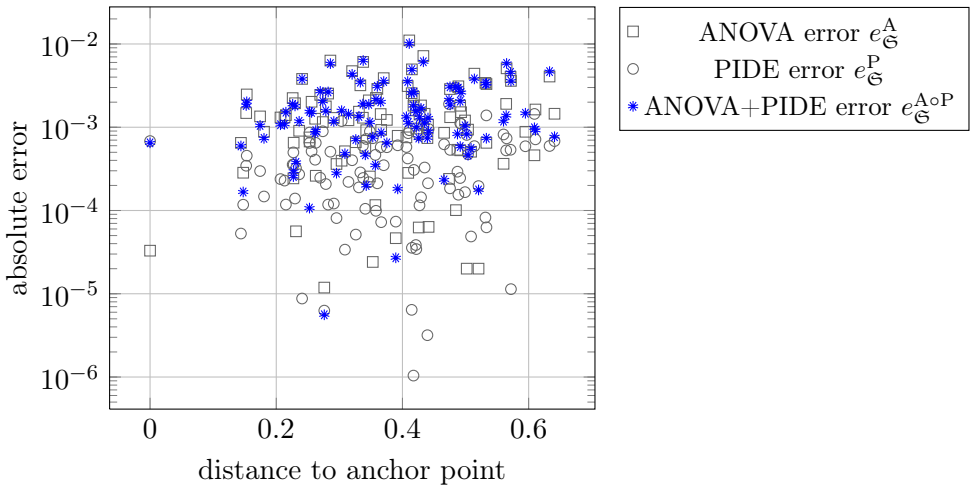

Figure 6.25: This plot shows the errors $e_{\mathfrak{G}}^{\mathrm{A}}$, $e_{\mathfrak{G}}^{\mathrm{P}}$ and $e_{\mathfrak{G}}^{\mathrm{A}\circ\mathrm{P}}$ at 100 randomly sampled points $(\mathbf{x}_i)_{i=1}^{100}$ for the ANOVA approximation with $d_t = 2, d_s = 1$

6.5.4 Approximation by 28 four-dimensional subproblems ($d_t = 2, d_s = 2$)

We now use superposition dimension $d_s = 2$. In addition to the two-dimensional problem and eight three-dimensional problems, we now have to solve 28 four-dimensional problems. This is especially challenging, since we have to achieve the same or even a higher level of accuracy for more complex problems. Even worse, we loose some accuracy by adding up a larger number of subspace solutions.

We compute the four-dimensional subproblems on the domain $(-\zeta, \zeta)^4$ with $\zeta = 4$. In the error convergence plots given in Figure 6.26 we see that the convergence rate for the four-dimensional subproblems is less than for the lower-dimensional ones, which comes as no surprise as our payoff function from Subsection 6.1.4 lacks the mix-regularity required by sparse grids.

In Figure 6.27, we see that the ANOVA errors are very low and the total error is dominated by the discretization error of our PIDE solver. Obviously, this choice of ANOVA approximation parameters d_t and d_s is not optimal, and we try to fix this in the next subsection.

6.5.5 Approximation by seven four-dimensional problems ($d_t = 3, d_s = 1$)

In the previous subsection we had an example of numerically challenging subproblems whose errors get amplified by adding up no less than 37 subspace solutions. It is preferable to lower the superposition dimension to $d_s = 1$ and to increase the truncation dimension to $d_t = 3$. This leads to one three-dimensional problem and seven four-dimensional problems only. The parameters for computation are all chosen as in the previous subsections. Figure 6.28 shows the convergence of the absolute PIDE error for the subproblems evaluated at the money. The two types of error at our randomly sampled points in Figure 6.29 are now more balanced, but still the PIDE error is dominating the total error.

6.5.6 Discussion

At the beginning of this section we conducted a rough error analysis and came to the conclusion that the number of degrees of freedom used in our PIDE solver needs to grow exponentially with the truncation dimension used in the ANOVA approximation. Then, we tried different choices of ANOVA approximation parameters and solved the resulting subproblems with our sparse grid PIDE solver. Our experiments support the theoretical finding: The case $d_t = 1, d_s = 0$, see Figure 6.21, had an extremely low PIDE error $e_{\mathfrak{S}}^{\mathrm{P}}$ and a high ANOVA approximation error, but the situation quickly reversed for the case $d_t = 2, d_s = 2$, see Figure 6.27, where the PIDE solver accuracy posed the bottleneck. In order to evaluate which choice is optimal, we summarize all total errors $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ by a mean regression in Figure 6.30. Obviously, the best choice is $d_t = 2, d_s = 1$, and leads to absolute errors in the region of 10^{-3} in the domain (6.25). Depending on whether we look at our in the money, out of the money and at the money prices this amounts to a relative accuracy of 0.6%, 5.3% and 0.17%, which is impressive considering that we are dealing with a ten-dimensional non-trivial solution.

Note that all subspaces and their discretizations involved in this computation can in theory be represented using a ten-dimensional generalized sparse grid. However, due to the 2^d constants in the computational complexity of the operator application, see Subsection 4.3.2, and the redundancy of the generating system, see Section 4.2, this is not advisable. With the ANOVA-PIDE approach, however, the same constants are limited to $2^{d_t+d_s}$.

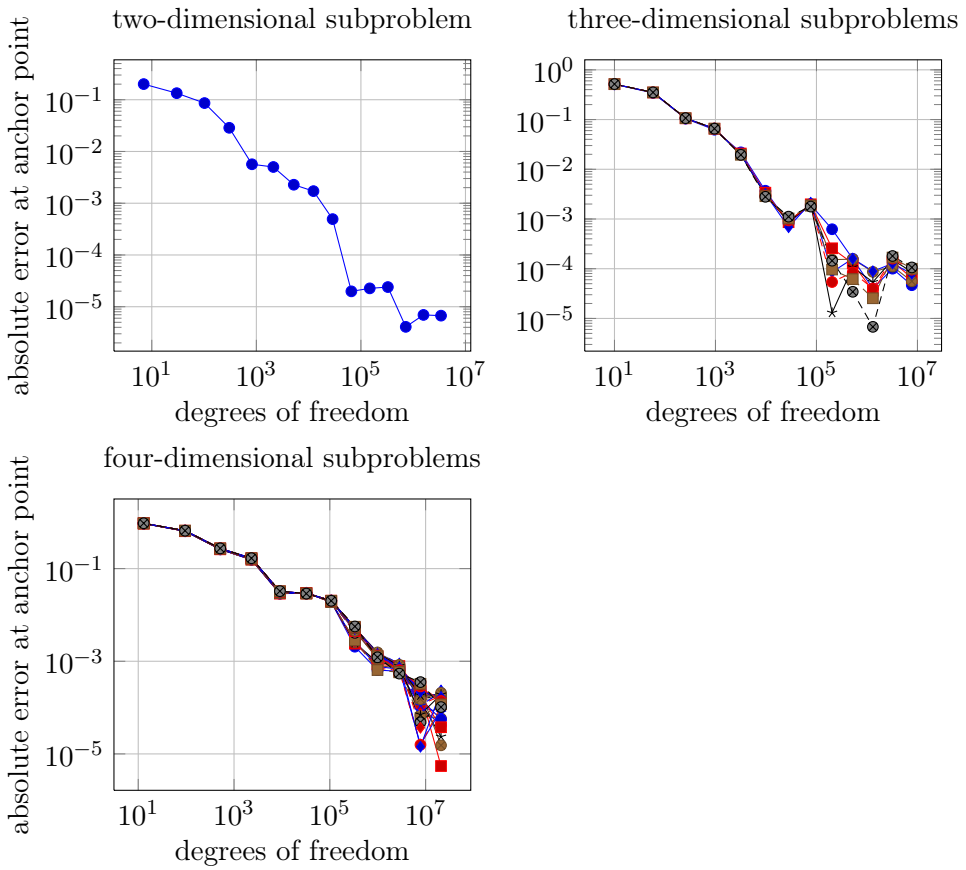

Figure 6.26: Absolute error of the PIDE solver for the two-, three- and four-dimensional subproblems that stem from the ANOVA approximation with $d_t = 2, d_s = 2$

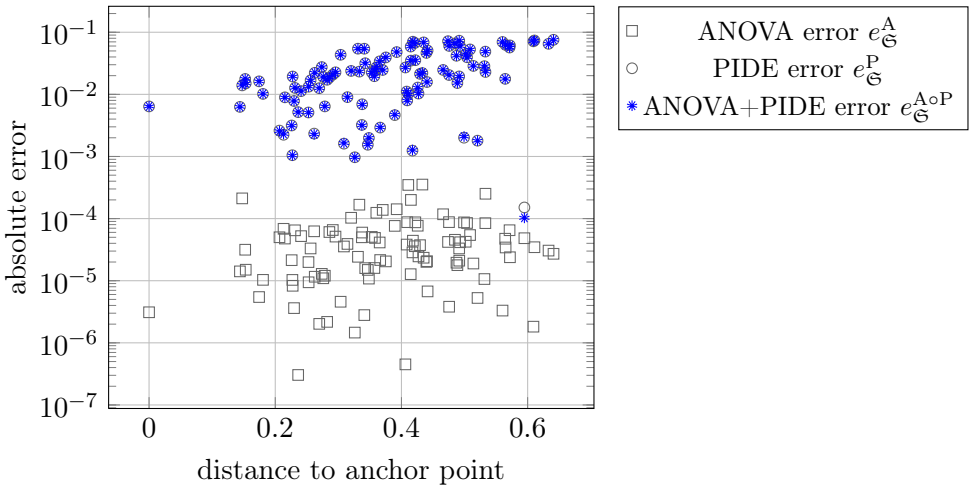

Figure 6.27: This plot shows the errors $e^{\mathrm{A}}_{\mathfrak{S}}$, $e^{\mathrm{P}}_{\mathfrak{S}}$ and $e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{S}}$ at 100 randomly sampled points $(\mathbf{x}_i)_{i=1}^{100}$ for the ANOVA approximation with $d_t = 2, d_s = 2$

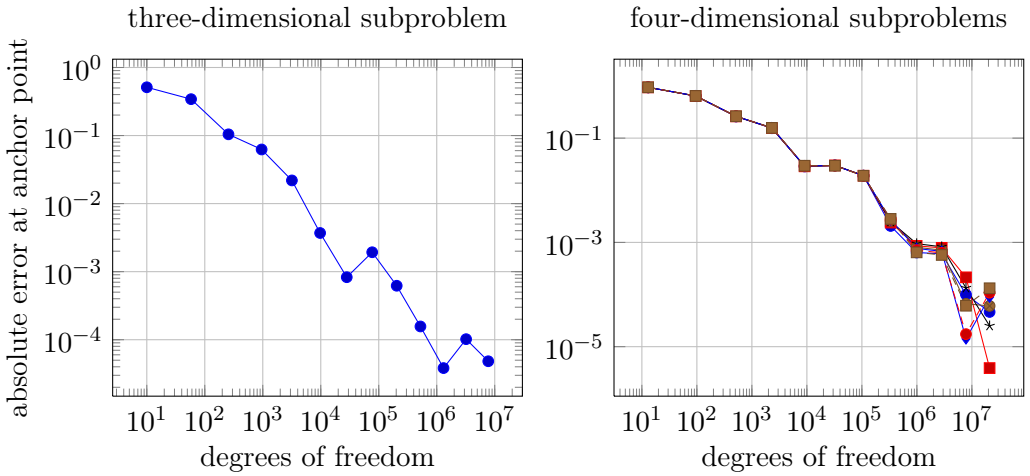

Figure 6.28: Absolute error of the PIDE solver for the three- and four-dimensional subproblems that stem from the ANOVA approximation with $d_t = 3, d_s = 1$

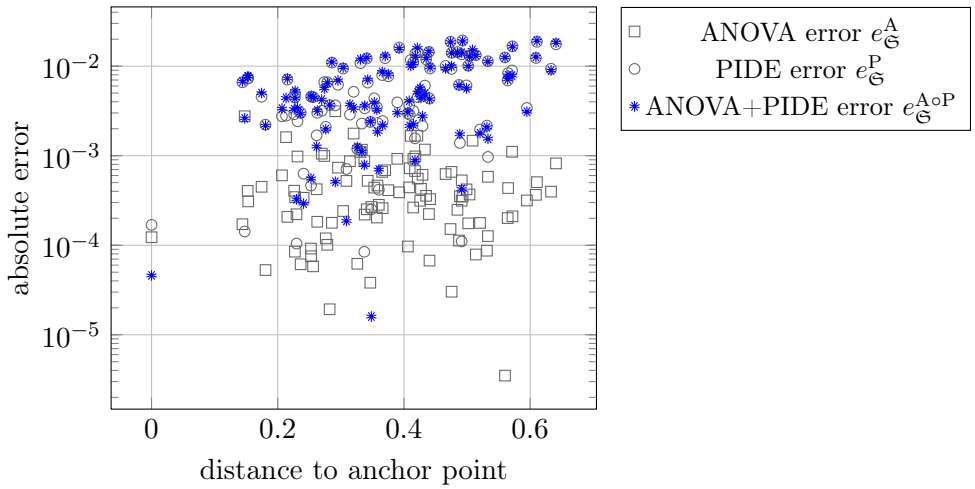

Figure 6.29: This plot shows the errors $e^{\mathrm{A}}_{\mathfrak{S}}$, $e^{\mathrm{P}}_{\mathfrak{S}}$ and $e^{\mathrm{A}\circ\mathrm{P}}_{\mathfrak{S}}$ at 100 randomly sampled points $(\mathbf{x}_i)_{i=1}^{100}$ for the ANOVA approximation with $d_t = 3, d_s = 1$

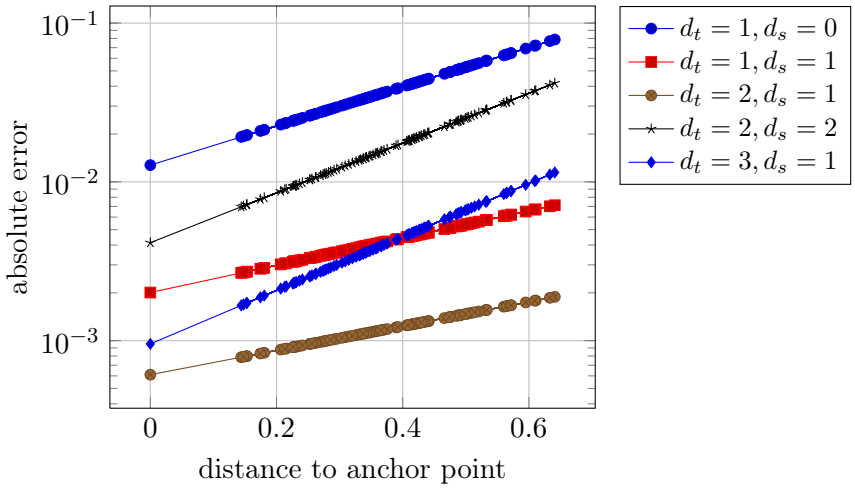

Figure 6.30: This plot shows the linear regression of the absolute errors $e_{\mathfrak{S}}^{\mathrm{A}\circ\mathrm{P}}$ for different ANOVA-PIDE approximation schemes

Is this necessarily the best method to solve the stated problem? It depends. The achieved accuracy is well within the means of Monte Carlo methods, but we have to bear in mind that we get the solution for different prices and times, whereas the Monte Carlo simulation can be evaluated only for one maturity and one price. So our solution remains valid and can be relied on after moderate changes of the underlyings or after some time passes. Monte Carlo simulations typically need to rerun in these cases. Moreover, Monte Carlo simulations are harder to adapt to American options and less useful to compute Greeks, i.e., derivatives of the solution with respect to its parameters, than PDE/PIDE approaches.

There is another possibility to circumvent the curse of dimensionality: For index options a straightforward approach could be to model the index price as a stochastic process and not every single one of its constituents, so that we essentially get a one-dimensional problem. This will work in many cases, but it is conceivable that this approach is not desired. If there is a number of options to be priced based on different subsets of underlyings, the option prices need to be consistent with each other, which is not guaranteed when all subindices are modeled separately.

We believe that the ANOVA-PIDE approach offers a great tradeoff between computational complexity and accuracy for this inherently challenging problem, and the case $d_t = 2, d_s = 1$ shows the potential of this approach.

7 Conclusion

In this final chapter we summarize the results of this thesis, we point to questions that have been left unanswered and discuss how this work can be extended.

7.1 Summary

This thesis dealt with the numerical solution of the high-dimensional backward Kolmogorov equation (1.1) and (1.2), i.e., the approximation of the expected value (1.3). Our approach was to first use the ANOVA approximation as a robust, simple and yet effective method for breaking the problem down to moderate-dimensional subproblems, and then to solve these using a sophisticated generalized sparse grid approach.[1]

We started with an introduction in Chapter 1. In Chapter 2, we gave a comprehensive description of the ANOVA decomposition and explored the interplay between different choices of one-dimensional measures (2.44), unit functions (2.46) and our problem at hand: In certain instances described in Subsections 2.3.1 and 2.3.2 we directly decomposed the solution of our problem, but these approaches either only worked in $\mathbb{R}^d$ without localization or were limited to the multivariate Brownian motion, respectively. Only the decomposition applied to the initial condition in Subsection 2.3.3 appeared feasible to us from a numerical perspective. By applying the ANOVA approximation to the initial condition, we introduced an error that is not subject to numerical convergence, i.e., the ANOVA approximation produces a modeling error. Therefore, a wise choice of the initial approximation is important. Then, in Chapter 3 we described the discretization of the moderate-dimensional subproblems in space and time. In Chapter 4 we discussed sparse grids as an alternative to discretizations based on regular tensor grids. We focused on generating systems, described the single space matrix-vector multiplication algorithm and gave a description of a variant of the unidirectional principle that works with non-local operators. In Chapter 5 we focused on several ways to precondition the resulting systems of linear equations. Most of the approaches were based on norm equivalences with L_2-orthogonal subspaces, and they were computed either explicitly or implicitly. They were similar to prewavelet approaches, but in our opinion easier to implement. Our presentation of the OptiCom as a preconditioner for PDEs and PIDEs was a novelty in the sparse grid context. We saw that this preconditioner is better than any a priori diagonal scaling but at additional costs. However, these costs were reduced drastically using the matrix-vector multiplication algorithm introduced in Subsection 4.3.1. In Chapter 6 we became specific about the problem we want to solve and introduced a multi-dimensional Kou model that resulted in a sum of tensor product operators. There, a recurrence formula previously only known for finite differences

[1] This concept is similar to data analysis, where the Principal Component Analysis is used as an initial and drastic dimensionality reduction, and the moderate-dimensional output is used to feed more sophisticated yet expensive methods, see [LV07].

could be used to apply this operator in linear runtime. We came up with a model problem and observed that the ANOVA approximation error decays exponentially with the truncation dimension. Typically the rate of decay is higher for a higher superposition dimension. A short error analysis lead to the conclusion that balancing the ANOVA and PIDE discretization errors requires an exponential growth of the number of degrees of freedom in our PIDE solver with respect to the truncation dimension. This was confirmed in our numerical experiments, in which the optimal balance was already achieved for $d_t = 2, d_s = 1$.

In summary, we can say we tackled a number of interesting problems. We made a considerable amount of assumptions about our jump-diffusion model, but in the end we were able for the first time to efficiently approximate the solution of a ten-dimensional BKE based on a generalization of the Kou model. This is well outside the means of classical tensor product methods, but also the sparse grid method benefits greatly from the initial application of the ANOVA approximation technique.

7.2 Outlook

There are numerous ways to complement or expand this thesis:

- The ANOVA approximation of functions by a sum of moderate-dimensional functions produces an additional modeling error that will not converge to zero.[2] However, new techniques like the iterated ANOVA from Subsection 2.2.2 offer the possibility of numerical convergence with low-dimensional functions only and should be looked into.

- Furthermore, a thorough error analysis for ANOVA approximations given different types of stochastic processes and initial conditions would be interesting.

- We could not give a proof that the CG version of the OptiCom-method in Subsection 5.6.2 exhibits the improved convergence properties we are used from CG approaches, even though the improvement was empirically present in all cases and only slightly smaller than for the a priori scaling parameters.

- We assumed a jump-diffusion model with several terms of independent jumps in all components. This proved helpful from an algorithmical perspective and can be regarded as a low rank-approximation to a general Lévy measure, see Subsection 6.1.3. This route could be pursued further to efficiently deal with general non-local operators.

- In future, we try to identify other suitable model problems for our method. This entails options with early exercise features and path dependent options, but also applications outside of financial mathematics.

[2] At least as long as we do not use all terms of the ANOVA decomposition (2.53), which would render the whole method pointless.

Bibliography

[AA00] L. Andersen and J. Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4(3):231–262, 2000.

[ACF10] A. Ammar, F. Chinesta, and A. Falcó. On the convergence of a greedy rank-one update algorithm for a class of linear systems. *Archives of Computational Methods in Engineering*, 17(4):473–486, 2010.

[Ach03] S. Achatz. Higher order sparse grid methods for elliptic partial differential equations with variable coefficients. *Computing*, 71(1):1–15, 2003.

[BB14] L. Ballotta and E. Bonfiglioli. Multivariate asset models using Lévy processes and applications. *The European Journal of Finance*, 2014. forthcoming.

[Bel61] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[BG99] H. Bungartz and M. Griebel. A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives. *J. Complexity*, 15:167–199, 1999.

[BG04] H. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.

[BG12] B. Bohn and M. Griebel. An adaptive sparse grid approach for time series predictions. In J. Garcke and M. Griebel, editors, *Sparse grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 1–30. Springer, 2012.

[BGGK12] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker. An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. *Journal of Scientific Computing*, 2012.

[BGR94] H. Bungartz, M. Griebel, and U. Rüde. Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems. *Comput. Methods Appl. Mech. Eng.*, 116:243–252, 1994.

[BGRZ94] H. Bungartz, M. Griebel, D. Röschke, and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math.*, 2:21–45, 1994.

[BHPS11] H. Bungartz, A. Heinecke, D. Pflüger, and S. Schraufstetter. Option pricing with a direct adaptive sparse grid approach. *Journal of Computational and Applied Mathematics*, 236(15):3741–3750, 2011.

[BL11] A. Brandt and O. Livne. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2011.

[BM02] G. Beylkin and M. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proc. Natl. Acad. Sci. USA*, 99:10246–10251, 2002.

[BNT10] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, 2010.

[BP94] A. Berman and R. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Society for Industrial and Applied Mathematics, 1994.

[BPX90] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55(191):1–22, 1990.

[Bra07a] D. Braess. *Finite Elemente: Theorie, Schnelle Löser und Anwendungen in der Elastizitatstheorie*. Springer London, Limited, 2007.

[Bra07b] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2007.

[BS73] F. Black and M. Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81(3):637–654, 1973.

[Bun92a] H. Bungartz. An adaptive Poisson solver using hierarchical bases and sparse grids. In P. de Groen and R. Beauwens, editors, *Proceedings of the IMACS International Symposium, Brussels, 2.-4. 4. 1991*, pages 293–310, Amsterdam, 1992. Elsevier.

[Bun92b] H. Bungartz. *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*. Dissertation, Fakultät für Informatik, Technische Universität München, 1992.

[BZ96] R. Balder and C. Zenger. The solution of multidimensional real Helmholtz equations on sparse grids. *SIAM J. Sci. Comput.*, 17:631–646, 1996.

[CK04] P. Casazza and G. Kutyniok. *Frames of subspaces*, volume 345 of *Contemporary Mathematics*, pages 87–113. Amer. Math. Soc., 2004.

[CMO97] R. Caflisch, W. Morokoff, and A. Owen. Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *J. Computational Finance*, 1:27–46, 1997.

[CT04] R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC Financial Mathematics Series, 2004.

[CV05] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. *SIAM J. Numer. Anal.*, 43:1596–1626, 2005.

[Dah96] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl*, 2:341–361, 1996.

[DM93] W. Dahmen and C. Micchelli. Using the refinement equation for evaluating integrals of wavelets. *SIAM Journal on Numerical Analysis*, 30(2):507–537, 1993.

[DSS09] T. Dijkema, C. Schwab, and R. Stevenson. An adaptive wavelet method for solving high-dimensional elliptic PDEs. *Constructive Approximation*, 30(3):423–455, 2009.

[EG04] A. Ern and J. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer New York, 2004.

[Feu05] C. Feuersänger. Dünngitterverfahren für hochdimensionale elliptische partielle Differentialgleichungen. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2005.

[Feu10] C. Feuersänger. *Sparse Grid Methods for Higher Dimensional Approximation*. Dissertation, Institut für Numerische Simulation, Universität Bonn, September 2010.

[FHMM13] A. Falcó, L. Hilario, N. Montés, and M. Mora. Numerical strategies for the Galerkin-proper generalized decomposition method. *Mathematical and Computer Modelling*, 57(7-8):1694–1702, 2013.

[Gar06] J. Garcke. Regression with the optimised combination technique. In W. Cohen and A. Moore, editors, *Proceedings of the 23rd ICML '06*, pages 321–328, New York, NY, USA, 2006. ACM Press.

[GG03] T. Gerstner and M. Griebel. Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87, 2003.

[GGT01] J. Garcke, M. Griebel, and M. Thess. Data mining with sparse grids. *Computing*, 67(3):225–253, 2001.

[GH09] J. Garcke and M. Hegland. Fitting multidimensional data using gradient penalties and the sparse grid combination technique. *Computing*, 84(1-2):1–25, 2009.

[GH10a] M. Griebel and J. Hamaekers. Tensor product multiscale many-particle spaces with finite-order weights for the electronic Schrödinger equation. *Zeitschrift für Physikalische Chemie*, 224:527–543, 2010.

[GH10b] M. Griebel and M. Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *J. Complexity*, 26:455–489, 2010.

[GH13a] M. Griebel and H. Harbrecht. A note on the construction of L-fold sparse tensor product spaces. *Constructive Approximation*, 38(2):235–251, 2013.

[GH13b] M. Griebel and H. Harbrecht. On the construction of sparse tensor product spaces. *Mathematics of Computations*, 82(282):975–994, 2013.

[GH13c] M. Griebel and A. Hullmann. An efficient sparse grid Galerkin approach for the numerical valuation of basket options under Kou's jump-diffusion model. In *Sparse grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 121–150. Springer Berlin Heidelberg, 2013.

[GH14a] M. Griebel and J. Hamaekers. Fast discrete Fourier transform on generalized sparse grids. In *Sparse grids and Applications*, volume 97 of *Lecture Notes in Computational Science and Engineering*, pages 75–108. Springer, 2014.

[GH14b] M. Griebel and A. Hullmann. On a multilevel preconditioner and its condition numbers for the discretized Laplacian on full and sparse grids in higher dimensions. In *Singular Phenomena and Scaling in Mathematical Models*, pages 263–296. Springer International Publishing, 2014.

[GHO15] M. Griebel, A. Hullmann, and P. Oswald. Optimal scaling parameters for sparse grid discretizations. *Numerical Linear Algebra with Applications*, 22(1):76–100, 2015.

[GK09] M. Griebel and S. Knapek. Optimized general sparse grid approximation spaces for operator equations. *Mathematics of Computations*, 78(268):2223–2257, 2009.

[GKS13] M. Griebel, F. Kuo, and I. Sloan. The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition. *Math. Comp.*, 82:383–400, 2013.

[GKT13] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques, 2013. Available at http://arxiv.org/abs/1302.7121 .

[GL96] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[GO94] M. Griebel and P. Oswald. On additive Schwarz preconditioners for sparse grid discretizations. *Numer. Math.*, 66:449–464, 1994.

[GO95a] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numer. Math.*, 70:163–180, 1995.

[GO95b] M. Griebel and P. Oswald. Tensor product type subspace splitting and multilevel iterative methods for anisotropic problems. *Adv. Comput. Math.*, 4:171–206, 1995.

[GO12] M. Griebel and P. Oswald. Greedy and randomized versions of the multiplicative Schwarz method. *Linear Algebra and its Applications*, 437(7):1596–1610, 2012.

[GOV05] M. Griebel, D. Oeltz, and P. Vassilevski. Space-time approximation with sparse grids. *SIAM J. Sci. Comput.*, 28(2):701–727, 2005.

[Gri91] M. Griebel. A parallelizable and vectorizable multi-level algorithm on sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for partial differential equations*, volume 31, pages 94–100, Braunschweig, 1991. Vieweg.

[Gri94a] M. Griebel. Multilevel algorithms considered as iterative methods on semidefinite systems. *SIAM Int. J. Sci. Stat. Comput.*, 15(3):547–565, 1994.

[Gri94b] M. Griebel. *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen.* Teubner Skripten zur Numerik. Teubner, Stuttgart, 1994.

[Gri98] M. Griebel. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing*, 61(2):151–179, 1998.

[Gri06] M. Griebel. Sparse grids and related approximation schemes for higher dimensional problems. In L. Pardo, A. Pinkus, E. Suli, and M.J. Todd, editors, *Foundations of Computational Mathematics (FoCM05), Santander*, pages 106–161. Cambridge University Press, 2006.

[GSZ92] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In P. de Groen and R. Beauwens, editors, *Iterative Methods in Linear Algebra*, pages 263–281. IMACS, Elsevier, North Holland, 1992.

[Hac85] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer Series in Computational Mathematics. Springer, 1985.

[Ham10] J. Hamaekers. *Sparse Grids for the Electronic Schrödinger Equation: Construction and Application of Sparse Tensor Product Multiscale Many-Particle Spaces with Finite-Order Weights for Schrödinger's Equation.* Südwestdeutscher Verlag für Hochschulschriften, Saarbrücken, 2010.

[Heg03] M. Hegland. Additive sparse grid fitting. In *Curve and Surface Fitting (Saint-Malo, 2002)*, Mod. Methods Math., pages 209–218. Nashboro Press, Brentwood, TN, 2003.

[Hes93] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[HGC07] M. Hegland, J. Garcke, and V. Challis. The combination technique and some generalisations. *Linear Algebra and its Applications*, 420(2–3):249–275, 2007.

[Hoe48] W. Hoeffding. A class of statistics with asymptotically normal distribution. *Ann. Math. Statistics*, 19:293–325, 1948.

[HSB12] A. Heinecke, S. Schraufstetter, and H. Bungartz. A highly parallel Black-Scholes solver based on adaptive sparse grids. *Int. J. Comput. Math.*, 89(9):1212–1238, 2012.

[IT09] J. Imai and K. Tan. Dimension reduction approach to simulating exotic options in a Meixner Lévy market. *IAENG International Journal of Applied Mathematics*, 39(4):265–275, 2009.

[JN99] M. Jung and S. Nepomnyaschikh. Variable additive preconditioning procedures. *Computing*, 62(2):109–128, 1999.

[JR08] J. Jakeman and S. Roberts. Stochastic Galerkin and collocation methods for quantifying uncertainty in differential equations: a review. *ANZIAM J.*, 50(C):C815–C830, 2008.

[Kaa88] E. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *Journal of Computational and Applied Mathematics*, 24(1-2):265 – 275, 1988.

[KK02] S. Knapek and F. Koster. Integral operators on sparse grids. *SIAM J. Num. Anal.*, 39(5):1794–1809, 2002.

[KL07] A. Knyazev and I. Lashuk. Steepest descent and conjugate gradient methods with variable preconditioning. *SIAM J. Matrix Analysis Applications*, 29(4):1267–1280, 2007.

[KOPT13] G. Kutyniok, K. Okoudjou, F. Philipp, and E. Tuley. Scalable frames. *Linear Algebra and its Applications*, 438(5):2225–2238, 2013.

[Kou02] S. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

[Kou07] S. Kou. Jump-diffusion models for asset pricing in financial engineering. In John R. Birge and Vadim Linetsky, editors, *Financial Engineering*, volume 15 of *Handbooks in Operations Research and Management Science*, pages 73 – 116. Elsevier, 2007.

[KSWW10] F. Kuo, I. Sloan, G. Wasilkowski, and H. Woźniakowski. On decompositions of multivariate functions. *Math. Comp.*, 79(270):953–966, 2010.

[KV13] S. Karimi and S. Vavasis. Detecting and correcting the loss of independence in nonlinear conjugate gradient, 2013. Available at http://arxiv.org/abs/1202.1479 .

[Kwo08] Y. Kwok. *Mathematical Models of Financial Derivatives.* Springer Finance. Springer London, 2008.

[LV07] J. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction.* Springer, 2007.

[MCC98] D. Madan, P. Carr, and E. Chang. The variance gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.

[Mer73] R. Merton. Theory of rational option pricing. *Bell Journal of Economics*, 4(1):141–183, 1973.

[Mer76] R. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2):125–144, 1976.

[Mit97] U. Mitzlaff. *Diffusionsapproximation von Warteschlangensystemen.* Doktorarbeit, TU Clausthal, 1997.

[NHW10] C. Schwab N. Hilber, S. Kehtari and C. Winter. Wavelet finite element method for option pricing in highdimensional diffusion market models. Research Report 2010-01, Seminar for Applied Mathematics, Swiss Federal Institute of Technology Zurich, 2010.

[NTW08] F. Nobile, R. Tempone, and C. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[Oet11] J. Oettershagen. Reduktion der effektiven Dimension und ihre Anwendung auf hochdimensionale Probleme. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2011.

[Osw92] P. Oswald. On discrete norm estimates related to multilevel preconditioners in the finite element method. In *Constructive Theory of Functions, Proc. Int. Conf. Varna 1991*, pages 203–214. Bulg. Acad. Sci., Sofia, 1992.

[Osw94] P. Oswald. *Multilevel Finite Element Approximation: Theory and Applications.* Teubner Skripten zur Numerik. Teubner, 1994.

[Osw09] P. Oswald. Stable space splittings and fusion frames. *Wavelets XIII (V. Goyal, M. Papadakis, D. Van de Ville, eds.), Proc. SPIE San Diego*, 7446, 2009.

[Rei04] C. Reisinger. *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben.* Dissertation, Universität Heidelberg, 2004.

[Rei10] N. Reich. Wavelet compression of anisotropic integrodifferential operators on sparse tensor product spaces. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44:33–73, 1 2010.

[Rei12] C. Reisinger. Asymptotic expansion around principal components and the complexity of dimension adaptive algorithms. In *Sparse grids and Applications*, Lecture Notes in Computational Science and Engineering. Springer, 2012.

[RSW10] N. Reich, C. Schwab, and C. Winter. On Kolmogorov equations for anisotropic multivariate Lévy processes. *Finance and Stochastics*, 14(4):527–567, 2010.

[RW07] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. *SIAM Journal on Scientific Computing*, 29(1):440–458, 2007.

[RW13] C. Reisinger and R. Wissmann. Numerical valuation of derivatives in high-dimensional settings via PDE expansions, 2013. Available at http://arxiv.org/abs/1209.1909 .

[SCDD02] X. Shen, H. Chen, J. Dai, and W. Dai. The finite element method for computing the stationary distribution of an SRBM in a hypercube with applications to finite buffer queueing networks. *Queueing Syst. Theory Appl.*, 42(1):33–62, 2002.

[Sch12] S. Schraufstetter. *A Pricing Framework for the Efficient Evaluation of Financial Derivatives based on Theta Calculus and Adaptive Sparse Grids.* Dissertation, Institut für Informatik, Technische Universität München, München, 2012.

[SGW13] P. Schröder, T. Gerstner, and G. Wittum. Taylor-like ANOVA-expansion for high dimensional problems in finance, 2013. Working paper, available on demand.

[Sjö07] P. Sjöberg. Partial approximation of the master equation by the Fokker–Planck equation. In B. Kågström, E. Elmroth, J. Dongarra, and J. Waśniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, volume 4699 of *Lecture Notes in Computer Science*, pages 637–646. Springer Berlin Heidelberg, 2007.

[SLE09] P. Sjöberg, P. Lötstedt, and J. Elf. Fokker–Planck approximation of the master equation in molecular biology. *Computing and Visualization in Science*, 12:37–50, 2009.

[SS00] D. Schötzau and C. Schwab. Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method. *SIAM J. Numer. Anal.*, 38(3):837–875, 2000.

[ST03] C. Schwab and R. Todor. Sparse finite elements for elliptic problems with stochastic loading. *Numerische Mathematik*, 95(4):707–734, 2003.

[Tho06] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems.* Springer series in Computational Mathematics. Springer Berlin Heidelberg, 2006.

[Toi08] J. Toivanen. Numerical valuation of European and American options under Kou's jump-diffusion model. *SIAM J. Sci. Comput.*, 30(4):1949–1970, 2008.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[WHS08] G. Widmer, R. Hiptmair, and C. Schwab. Sparse adaptive finite elements for radiative transfer. *J. Comput. Phys.*, 227(12):6071–6105, 2008.

[Win09] C. Winter. *Wavelet Galerkin schemes for option pricing in multidimensional Lévy models.* PhD thesis, ETH Zürich, 2009.

[Xu92] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34:581–613, 1992.

[XZ02] J. Xu and L. Zikatanov. The method of alternating projections and the method of subspace corrections in Hilbert space. *Journal of the American Mathematical Society*, 15(3):573–598, 2002.

[Yse93] H. Yserentant. Old and new convergence proofs for multigrid methods. *Acta Numerica*, 2:285–326, 1993.

[Yse05] H. Yserentant. Sparse grid spaces for the numerical solution of the electronic Schrödinger equation. *Numerische Mathematik*, 101(2):381–389, 2005.

[Zei11] A. Zeiser. Fast matrix-vector multiplication in the sparse-grid Galerkin method. *J. Sci. Comput.*, 47(3):328–346, 2011.

[Zen91] C. Zenger. Sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31 of *Notes on Numerical Fluid Mechanics*, pages 241–251. Vieweg, 1991.