Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

Wegelerstraße 6 • 53115 Bonn • Germany
phone +49 228 73-3427 • fax +49 228 73-7527
www.ins.uni-bonn.de

M. Griebel, A. Hullmann

An efficient sparse grid Galerkin approach for the numerical valuation of basket options under Kou's jump-diffusion model

INS Preprint No. 1202

February 2012

An efficient sparse grid Galerkin approach for the numerical valuation of basket options under Kou's jump-diffusion model

M. Griebel and A. Hullmann

Abstract We use a sparse grid approach to discretize a multi-dimensional partial integro-differential equation (PIDE) for the deterministic valuation of European put options on Kou's jump-diffusion processes. We employ a generalized generating system to discretize the respective PIDE by the Galerkin approach and iteratively solve the resulting linear system. Here, we exploit a newly developed recurrence formula, which, together with an implementation of the unidirectional principle for non-local operators, allows us to evaluate the operator application in linear time. Furthermore, we exploit that the condition of the linear system is bounded independently of the number of unknowns. This is due to the use of the Galerkin generating system and the computation of L_2-orthogonal complements. Altogether, we thus obtain a method that is only linear in the number of unknowns of the respective generalized sparse grid discretization. We report on numerical experiments for option pricing with the Kou model in one, two and three dimensions, which demonstrate the optimal complexity of our approach.

1 Introduction

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 certain given assumptions. Here, for some simple options, e.g. European vanillas, analytical formulas exist, but for more complex financial instruments like American put options or arithmetic average Asian options this is no longer the case and numerical approximations must be made. This gave rise to the discipline of computational finance, in which most valuation problems

Institute for Numerical Simulation, University of Bonn
{griebel, hullmann}@ins.uni-bonn.de

are solved by means of Monte Carlo simulations or the numerical solution of partial differential equations.

In recent decades, it has become clear that the assumptions made in the original paper do not hold in practice. This can be seen e.g. from effects like the 'volatility smile', which means that the implied volatility of an option is not independent of the strike price. Thus, observable option prices are not conform with the model assumption that logarithmic returns follow a Brownian motion with constant volatility. Obviously, the normal distribution underestimates the probability of extreme events like the Black Monday in 1987, the crash of Long-Term Capital Management in 1998 or the collapse of Lehman brothers in 2008, just to mention a few. In May 2010, the Dow Jones plunged without obvious reasons by almost 1000 points, commonly referred to as "Flash Crash". 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.

By adding jumps to the model of the geometric Brownian motion we can take such effects into account and fix issues like the volatility smile. Even though analytical formulas exist for certain option types on jump processes [Kou02, Mer76], numerical valuation is nevertheless needed in most practical cases. Then, additional computational difficulties stem from an integral term, which makes the usual Black-Scholes PDE a partial integro-differential equation (PIDE). Due to the non-locality of the integro-operator we obtain linear systems with densely populated matrices, which, treated naively, would result in a substantial additional computational complexity of $\mathcal{O}(N^3)$, with N being the degrees of freedom. Thus, the convolution integral is evaluated by the fast Fourier transform in [AA00], which reduces the complexity of the system matrix application to $\mathcal{O}(N \log N)$. However, Kou's jump-diffusion model admits an even faster operator application with $\mathcal{O}(N)$ complexity for the finite difference case [Toi08]. In this paper, we introduce a comparable approach for the Galerkin method and exploit it in our numerical solver.

An additional numerical challenge is the pricing of basket options, i.e. options on multiple underlyings. This usually leads to the so-called 'curse of dimensionality', which means that the cost complexity for the approximation to the solution of a problem grows exponentially with the dimension. For example, a d-dimensional mesh with a resolution of h in each direction results in a storage and cost complexity of at least $\mathcal{O}(h^{-d})$. Sparse grid discretizations [BG04] can circumvent this problem to some extent. They result in a complexity of only $\mathcal{O}(h^{-1}\,(\log h^{-1})^{d-1})$, which allows for huge savings for higher values of d, whereas – depending on the smoothness assumptions on the function – the convergence rate of the error is unchanged or only affected by a small logarithmic term.

In this paper, we show how generalized sparse grid generating systems can be used in the described PIDE setting, i.e. for the pricing of basket options with the Kou model. We use the unidirectional principle [BZ96, Bun92] – a

fast matrix-vector multiplication for sparse grids, 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 an optimal preconditioning based on a H^1-norm equivalence with L_2-orthogonal subspaces, we obtain a method which altogether scales only linearly with respect to the degrees of freedom of our sparse grid discretization.

The remainder of this paper is organized as follows: In Sect. 2 we present Kou's model, its multi-dimensional generalization and we discuss how the pricing of European options on jump-diffusion processes leads to a PIDE problem. Section 3 deals with relevant aspects of the numerical treatment of our PIDE, i.a. optimal preconditioning, the unidirectional principle for non-local operators and the recurrence formula for Kou's model in the Galerkin approach. In Sect. 4 we test our approach on basket put options in one, two and three dimensions. Final remarks in Sect. 5 conclude the paper.

2 Option pricing with Kou's model

In this section we describe a one-dimensional model for jump-diffusion processes as presented in [Kou02]. Then, we discuss its generalization to the multi-dimensional setting. Finally, we sketch how a PIDE arises from the option pricing problem on such a jump-diffusion process.

2.1 One-dimensional model

Kou's jump-diffusion model [Kou02] assumes that the price of a security S fulfills the stochastic differential equation (SDE)

$$\frac{\mathrm{d}S(t)}{S(t-)} = \mu \mathrm{d}t + \sigma \mathrm{d}W(t) + \mathrm{d}\left(\sum_{j=1}^{N(t)} (V_j - 1)\right), \tag{1}$$

where t denotes time, $W(t)$ 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 $\{V_j\}_{j\in\mathbb{N}}$ denotes a sequence of jumps. These jumps are assumed to be independent identically distributed with density

$$h_{p,\eta,\mu}(v) = \begin{cases} (1-p)\eta v^{\eta-1} & \text{for} \quad v < 1\,, \\ p\mu v^{-\mu-1} & \text{for} \quad v \geq 1\,, \end{cases}$$

where p and $1-p$ denote the probabilities of jumping upwards and downwards, respectively, while $\eta > 0$ and $\mu > 1$ are parameters that control the jump sizes. The density of $Y_j := \log(V_j), j \in \mathbb{N}$, is then given by

$$\kappa_{p,\eta,\mu}(z) = \begin{cases} (1-p)\eta e^{\eta z} & \text{for} \quad z < 0 \,, \\ p\mu e^{-\mu z} & \text{for} \quad z \geq 0 \,, \end{cases} \tag{2}$$

which is an asymmetric double exponential distribution. Furthermore, in this model all random variables $W(t), N(t), Y_j$ are assumed to be independent. The dynamics of S in the SDE (1) can then be given by

$$S(t) = S(0) \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W(t)\right) \cdot \prod_{j=1}^{N(t)} V_j$$

and we have

$$\mathbb{E}(V_j) = \mathbb{E}(\exp(Y_j)) = (1-p)\frac{\eta}{\eta+1} + p\frac{\mu}{\mu-1} \,.$$

2.2 Multi-dimensional case and dependence modelling

We now consider a d-dimensional price process $\mathbf{S} = (S_1, \ldots, S_d)$ with state space $\mathbb{R}^d$. The components $S_i, i = 1, \ldots, d$, of $\mathbf{S}$ follow the dynamics

$$S_i(t) = S_i(0) \exp\left(\left(\mu_i - \frac{1}{2}\sigma_i^2\right) t + \sigma_i W_i(t)\right) \cdot \prod_{j=1}^{N_i(t)} V_{i,j} \cdot \prod_{j=1}^{\tilde{N}(t)} \tilde{V}_{i,j} \,, \tag{3}$$

where μ_i and σ_i denote drift and volatility constants, N_i and $\tilde{N}$ are Poisson processes with rates λ_i and $\tilde{\lambda}$, respectively, and $\{\log V_{i,j}\}_{j\in\mathbb{N}}$ and $\{\log \tilde{V}_{i,j}\}_{j\in\mathbb{N}}$ are sequences of jumps of the component i.

Obviously, the Brownian part of the process is decorrelated across dimensions. A priori, this assumption may look unrealistic, it however can easily be achieved by a standard principal component transformation of the covariance matrix, see [Rei04] for example.

In our model (3), every price process S_i has two sources of jumps. The first one leaves other processes $S_{i'}, i' \neq i$ unaffected. It is realized by the Poisson process $N_i(t)$ and i.i.d. double exponential random variables $\{\log V_{i,j}\}_{j\in\mathbb{N}}$ with parameters p_i, η_i, μ_i, see (2). The second one takes care of the correlations of jumps over the dimensions. It consists of a jump process $\tilde{N}(t)$ and the associated random variables $\{\log \tilde{V}_{i,j}\}_{j\in\mathbb{N}}$, which are again assumed to be i.i.d. double exponential random variables with parameters $\tilde{p}_i, \tilde{\eta}_i, \tilde{\mu}_i$. It is noteworthy that even though the jump sizes are independent, dependence of the d price processes $(S_1, \ldots, S_d)$ is created nevertheless, as jumps hap-

pen at the same time in *all* components of $\mathbf{S}$. This joint jump term can be understood as a risk factor affecting the whole market and not just a single company or industrial sector.

2.3 Representation of the multi-dimensional process as Lévy process

We now identify our d-dimensional generalization (3) of the Kou model as a Lévy process. To this end, recall the following definition, compare also [CT04].

Definition 1 (Lévy process). A càdlàg stochastic process $(\mathbf{X}(t))_{0\leq t<\infty}$ on the filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\leq t<\infty}, \mathbb{P})$ with values in $\mathbb{R}^d$ and $\mathbf{X}(0) = \mathbf{0}$ a.s. 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 $\mathbf{X}(t-s)$, $0 \leq s < t < \infty$.
3. Stochastic continuity: $\lim_{t\to s} \mathbf{X}(t) = \mathbf{X}(s)$, where the limit is taken in probability.

The characteristic exponent $\psi : \mathbb{R}^d \to \mathbb{C}$ of $\mathbf{X}(t)$, which satisfies

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

allows for the unique Lévy-Khinchin representation

$$\psi(\xi) = -\frac{1}{2}\langle \xi, \mathbf{Q}\xi\rangle + i\langle \gamma, \xi\rangle + \int_{\mathbb{R}^d} \left(e^{i\langle \xi, \mathbf{z}\rangle} - 1 - i\langle \xi, \mathbf{z}\rangle \mathbb{1}_{\{|\mathbf{z}|\leq 1\}}\right)\nu(\mathrm{d}\mathbf{z}),$$

where $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is the symmetric covariance matrix of the continuous part of $\mathbf{X}$, $\gamma \in \mathbb{R}^d$ is the drift of $\mathbf{X}$ and ν is the Lévy measure, which satisfies

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

This condition ensures that the activity for large jumps is finite and possibly infinite for very small jumps only.

For our multi-dimensional Kou model presented in Subsect. 2.2, the logarithmic returns follow a Lévy process. We thus can rewrite (3) as

$$S_i(t) = S_i(0)\exp\left(X_i(t)\right) \quad \text{for} \quad i = 1, \ldots, d\,,$$

with $(\mathbf{X}(t))_{0\leq t<\infty}$ being a $\mathbb{R}^d$-valued Lévy process with characteristic triplet $(\mathbf{Q}, \nu, \gamma)$. Here, the elements of covariance matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$ satisfy

$$q_{ij} = \delta_{ij}\sigma_i^2$$

for all $i, j = 1, \ldots, d$, the drift is

$$\gamma_i = \mu_i - \frac{\sigma_i^2}{2} + \int_{\mathbb{R}^d} z_i \mathbb{1}_{\{|z|\leq 1\}} \nu(\mathrm{d}z)$$

for $i = 1, \ldots, d$, and the Lévy measure ν has finite activity and can be expressed by

$$\nu(\mathrm{d}z) = \sum_{i=1}^{d} \lambda_i \kappa_{p_i,\eta_i,\mu_i}(z_i)\mathrm{d}z_i \otimes \bigotimes_{\substack{i'=1 \\ i'\neq i}}^{d} \delta(\mathrm{d}z_{i'}) + \tilde{\lambda} \bigotimes_{i=1}^{d} \kappa_{\tilde{p}_i,\tilde{\eta}_i,\tilde{\mu}_i}(z_i)\mathrm{d}z_i \,. \tag{4}$$

In (4), the i-th summand represents the jumps in the i-th component of $\mathbf{X}$ generated by the Poisson process $N_i(t)$ and the associated jump sizes $(\log V_{i,j})_{j\in\mathbb{N}}$ are distributed with density $\kappa_{p_i,\eta_i,\mu_i}$, see (2). The remaining components are unaffected by these jumps, which is expressed by the delta distribution $\bigotimes_{i'\neq i} \delta(\mathrm{d}z_{i'})$. The last term in (4) is due to the Poisson process $\tilde{N}(t)$ with jumps occurring in all components at the same time. The jump sizes $(\log \tilde{V}_{i,j})_{j\in\mathbb{N}}$ in the i-th component are distributed with the probability density $\kappa_{\tilde{p}_i,\tilde{\eta}_i,\tilde{\mu}_i}$.

Remark 1. Even though the jump sizes are all independent, the stochastic processes X_i, and thus S_i, are not. Otherwise, the right-hand side of

$$\mathbb{E}\left(e^{i\langle \xi, \mathbf{X}(t)\rangle}\right) = e^{-t\psi(\xi)}$$

could be written as a product in $\xi_1, \ldots, \xi_d$, which is not possible since ψ does in general *not* decompose into a sum of one-dimensional functions over the dimensions.

Note here that the product $\bigotimes_{i=1}^{d} \kappa_{\tilde{p}_i,\tilde{\eta}_i,\tilde{\mu}_i}(z_i)$ in (4) can be seen as a rank-one approximation to a general finite Lévy measure. Then, this concept can easily be carried over to more complex dependencies by adding further Poisson processes in combination with jump sizes that are again independent of other dimensions. This can be done in the spirit of a low rank approximation, see [BM02], i.e. as the approximation of a non-separable density function by a small sum of separable functions. Such a generalization would be treatable by the numerical approach proposed in this paper as well.

2.4 Option pricing

We now want to price a European option with the payoff $g : \mathbb{R}^d \to \mathbb{R}$ depending on the state of the process $\mathbf{S}$ at the time of maturity T. In this paper, we

predominantly price put options with strike price K and unit weights, i.e.

$$g(\mathbf{S}(T)) = \max(0, K - \sum_{i=1}^{d} S_i(T)) \,. \tag{5}$$

Note, however, that the same approach can also be used to valuate call options. For the conventional Black–Scholes model, arbitrage considerations show that the fair price of a European option is given by the discounted expected value of the payoff function under the unique risk-neutral measure. 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]. We assume that the risk-neutral dynamics of our d price processes is given by

$$S_i(t) = S_i(0) \exp\left(rt + X_i(t)\right) \quad \text{for} \quad i = 1, \ldots, d \,, \tag{6}$$

where $\mathbf{X}$ is a Lévy process with a triplet $(\mathbf{Q}, \nu, \gamma)$, s.t.

$$e^{-tr} S_i(t)$$

are martingales for $i = 1, \ldots, d$. Then, the value of a European option at time t with payoff g can be given by

$$V(t, \mathbf{s}) = \mathbb{E}(e^{-r(T-t)} g(\mathbf{S}(T)) \mid \mathbf{S}(t) = \mathbf{s}) \,. \tag{7}$$

This function V is known to satisfy the PIDE (8) of the following theorem, see [CT04, RSW10] for further information.

Theorem 1 (Backward PIDE for European options). *Let* $\mathbf{S}$ *be an exponential Lévy model* (6) *with Lévy triplet* $(\mathbf{Q}, \nu, \gamma)$*, which has a non-vanishing diffusion matrix* $\mathbf{Q}$*, and let* ν *satisfy*

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

for $i = 1, \ldots, d$*. 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 (7) *is a solution of the backward PIDE for European options*

$$\begin{aligned} &\frac{\partial V}{\partial t}(t, \mathbf{s}) + \frac{1}{2} \sum_{i,j=1}^{d} s_i s_j q_{ij} \frac{\partial^2 V}{\partial s_i \partial s_j}(t, \mathbf{s}) + r \sum_{i=1}^{d} s_i \frac{\partial V}{\partial s_i}(t, \mathbf{s}) - r V(t, \mathbf{s}) \\ &+ \int_{\mathbb{R}^d} \left(V(t, \mathbf{s} e^{\mathbf{z}}) - V(t, \mathbf{s}) - \sum_{i=1}^{d} s_i \left(e^{z_i} - 1\right) \frac{\partial V}{\partial s_i}(t, \mathbf{s}) \right) \nu(\mathrm{d}\mathbf{z}) = 0 \end{aligned} \tag{8}$$

on $(0,T) \times \mathbb{R}^d_{>0}$*, where* $V(t, \mathbf{s}e^{\mathbf{z}}) := V(t, s_1 e^{z_1}, \ldots, s_d e^{z_d})$*, with the terminal condition given by*

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

Under the same constraints as in Theorem 1 and with the assumption of finite activity, i.e. $\lambda := \nu(\mathbb{R}^d) < \infty$, which holds in case of the Kou model (3) with $\lambda = \sum_{i=1}^d \lambda_i + \tilde{\lambda}$, we can rewrite the PIDE as

$$\frac{\partial}{\partial \tau} u(\tau, \mathbf{x}) - \frac{1}{2} \sum_{i,j=1}^{d} q_{ij} \frac{\partial^2}{\partial x_i x_j} u(\tau, \mathbf{x}) + \lambda u(\tau, \mathbf{x}) - \int_{\mathbb{R}^d} u(\tau, \mathbf{x} + \mathbf{z}) \nu(\mathrm{d}\mathbf{z}) = 0 \quad (9)$$

with $e^{-r\tau} u(\tau, \mathbf{x}) = V(t, \mathbf{s})$. To this end, we use the variable transforms $\tau = T - t$ and

$$x_i = \log s_i + \tau (r - \xi_i - \frac{q_{ii}}{2}) \quad \text{with} \quad \xi_i = \int_{\mathbb{R}^d} (e^{z_i} - 1) \, \nu_i(\mathrm{d}z_i) \qquad (10)$$

for $i = 1, \ldots, d$, with ν_i being the marginal distribution of ν in the i-th direction. Then, the terminal condition becomes an initial condition for $\tau = 0$, and reads

$$u(0, \mathbf{x}) = g(\exp(\mathbf{x}))$$

with the exp-function applied componentwise to the vector $\mathbf{x} \in \mathbb{R}^d$.

3 Numerical treatment

In this section, we discuss the numerical treatment of the transformed PIDE (9). We start with a simple θ-scheme for time discretization and a Galerkin method using generalized sparse grids for space discretization. The resulting set of linear equations is then solved by an optimally preconditioned iterative method, which we discuss in Subsect. 3.3. To this end, we present a fast matrix-vector multiplication for the sparse grid generating systems in combination with non-local operators and the Galerkin recurrence formula for the Kou model. All these ingredients are essential to obtain an overall solution method with optimal linear complexity in the degrees of freedom of our discretization.

3.1 Time discretization and weak formulation

First, we localize our space domain $\mathbb{R}^d$ to a rectangular domain

$$\mathcal{D} = (\alpha_1, \beta_1) \times \cdots \times (\alpha_d, \beta_d) \tag{11}$$

and assume zero boundary conditions. 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 price process has left $\mathcal{D}$ during the time to maturity. It is well-known that the pointwise error introduced by this approximation decreases exponentially with the domain size, see for example [CV05].

The upper and lower bounds in (11) are chosen by

$$\alpha_i = x_i - \gamma\sqrt{T \cdot \mathrm{var}(X_i(T))}\,, \tag{12}$$

$$\beta_i = x_i + \gamma\sqrt{T \cdot \mathrm{var}(X_i(T))}\,, \tag{13}$$

where γ is a proportionality constant relating the domain size and the standard deviation of the stochastic process in that direction. Furthermore, $\mathbf{x} = (x_1 \ldots, x_d)$ denotes a point of interest, which might be, like e.g. in (10), a transformed initial state vector $\mathbf{s}$ for an option we want to price by the evaluation of $e^{-rT}u(T, \mathbf{x}) = V(0, \mathbf{s})$.

We then can write the PIDE (9) with the definition

$$Lu(\tau, \mathbf{x}) := \frac{1}{2}\sum_{i,j=1}^{d} q_{ij}\frac{\partial^2}{\partial x_i x_j}u(\tau, \mathbf{x}) - \lambda u(\tau, \mathbf{x}) + \int_{\mathbb{R}^d} u(\tau, \mathbf{x} + \mathbf{z})\nu(\mathrm{d}\mathbf{z})$$

as

$$\frac{\partial u(\tau, \mathbf{x})}{\partial \tau} - Lu(\tau, \mathbf{x}) = 0 \qquad \text{on} \qquad (0, T) \times \mathcal{D}\,, \tag{14}$$

with an extension of u by zero outside of $\mathcal{D}$, i.e. $u|_{(0,T)\times\mathbb{R}^d\setminus\mathcal{D}} = 0$, which is relevant to the integral term.

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

$$t_i = i\Delta t\,, \qquad i = 0, \ldots, M\,,$$

with $\Delta t = \frac{T}{M}$ and apply the well-known θ-scheme with $\theta = \frac{1}{2}$.[1] Then, (14) can be expressed for $i = 0, \ldots, M - 1$ as a sequence of stationary elliptic PIDEs

$$\frac{u^{(i+1)}(\mathbf{x}) - u^{(i)}(\mathbf{x})}{\Delta t} - L\left(\theta u^{(i+1)}(\mathbf{x}) + (1 - \theta)u^{(i)}(\mathbf{x})\right) = 0$$

on $\mathcal{D}$ with $u^{(i)}(\mathbf{x}) \approx u(t_i, \mathbf{x}), i = 0, \ldots, M$.

Next, a weak formulation in space will give us a sequence of $H_0^1(\mathcal{D})$-elliptic problems: Find $u^{(i+1)} \in H_0^1(\mathcal{D})$, s.t.

[1] More sophisticated techniques, e.g. space-time sparse grids [GO07], are available, but this is beyond the scope of this work.

$$\Delta t^{-1}(u^{(i+1)}, v)_{L_2} + \theta a(u^{(i+1)}, v) = r^{(i)}(v) \quad \forall v \in H_0^1(\mathcal{D}) \tag{15}$$

with

$$\begin{aligned} a(u,v) = \frac{1}{2} \int_{\mathcal{D}} \sum_{i,j=1}^{d} q_{ij} \frac{\partial u(\mathbf{x})}{\partial x_i} \frac{\partial v(\mathbf{x})}{\partial x_j} \mathrm{d}\mathbf{x} \\ + \lambda (u,v)_{L_2} - \int_{\mathcal{D}} \int_{\mathbb{R}^d} u(\tau, \mathbf{x}+\mathbf{z}) \nu(\mathrm{d}\mathbf{z}) v(\mathbf{x}) \mathrm{d}\mathbf{x} \end{aligned}$$

and

$$r^{(i)}(v) = \Delta t^{-1}(u^{(i)}, v)_{L_2} - a((1-\theta)u^{(i)}, v) \; .$$

Before equation (15) can be discretized in space, we need to transform the domain $\mathcal{D}$ to $\Omega := (0,1)^d$, see (11). This is done by a linear affine scaling $\mathcal{T} : \mathcal{D} \to \Omega$

$$\mathcal{T}(\mathbf{x}) = \left(\frac{x_1 - \alpha_1}{\beta_1 - \alpha_1}, \ldots, \frac{x_d - \alpha_d}{\beta_d - \alpha_d} \right) \; . \tag{16}$$

When we apply the domain transformation (16), and take the assumptions made about $(q_{ij})_{i,j=1}^d$ and ν in Subsect. 2.2 into account, the bilinear form $a(u,v)$ in the variational equation (15) can be given on Ω by

$$a(u,v) = \sum_{i=1}^{d} \frac{\sigma_i^2}{2(\beta_i - \alpha_i)^2} \int_{\Omega} \frac{\partial u(\mathbf{x})}{\partial x_i} \frac{\partial v(\mathbf{x})}{\partial x_i} \mathrm{d}\mathbf{x} + \lambda \int_{\Omega} u(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{17}$$

$$\begin{aligned} &- \sum_{i=1}^{d} (\beta_i - \alpha_i) \lambda_i \int_{\Omega} \int_{\mathbb{R}^d} u(\mathbf{x} + z_i \mathbf{e}_i) \kappa_{p_i, \eta_i, \mu_i}((\beta_i - \alpha_i) z_i) \mathrm{d}z_i v(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &- \tilde{\lambda} \int_{\Omega} \int_{\mathbb{R}^d} u(\mathbf{x} + \mathbf{z}) \prod_{i=1}^{d} (\beta_i - \alpha_i) \kappa_{\tilde{p}_i, \tilde{\eta}_i, \tilde{\mu}_i}((\beta_i - \alpha_i) z_i) \mathrm{d}z_i \; v(\mathbf{x}) \mathrm{d}\mathbf{x} \; , \end{aligned} \tag{18}$$

where $\mathbf{e}_i$ denotes the i-th unit vector and

$$\lambda := \sum_{i=1}^{d} \lambda_i (\beta_i - \alpha_i) + \tilde{\lambda} \prod_{i=1}^{d} (\beta_i - \alpha_i) \; .$$

3.2 Space discretization by a sparse grid generating system

What is finally missing is a discretization of (15) with bilinear form (17) in space. To this end, we introduce a sparse grid generating system based on linear splines. First, we consider a one-dimensional multilevel system on the

interval $(0,1)$. On level l, $n_l = 2^l - 1$ hat functions

$$\phi_{l,i}(x) = \max(1 - 2^l \left| x - x_{l,i} \right|, 0)$$

exist, which are centered at the points of an equidistant mesh

$$x_{l,i} = 2^{-l} i$$

for $i = 1, \ldots, n_l$. The left side of Fig. 1 shows all functions on the first four

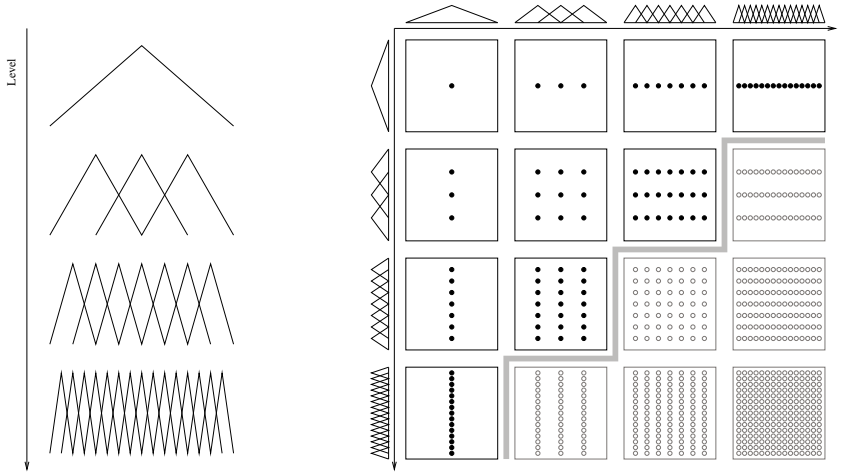

Fig. 1 The first four levels of a one-dimensional multilevel generating system (left). Two-dimensional tensorization and the sparse subspace (right)

levels. The spaces

$$V_l = \operatorname{span}\{\phi_{l,i} \mid 1 \le i \le n_l\}, l \in \mathbb{N}$$

possess the simple inclusion relation $V_l \subset V_{l+1}$. This makes the union of their basis functions $\bigcup_{l=1}^{k} \{\phi_{l,1}, \ldots, \phi_{l,n_l}\}$ a generating system and not a basis. The multi-dimensional case is based on the domain $\Omega = (0,1)^d$. By tensorization, we obtain spaces associated to a multi-index $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{N}^d$

$$V_{\mathbf{l}} = \bigotimes_{i=1}^{d} V_{l_i} \ .$$

For a given $\mathbf{l}$, $V_{\mathbf{l}}$ is the space of d-linear functions on a possibly anisotropic full grid space with

$$|V_{\mathbf{l}}| = \prod_{i=1}^{d} (2^{l_i} - 1) = \mathcal{O}(2^{|\mathbf{l}|_1})$$

degrees of freedom. It is spanned by the functions

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

with

$$\mathbf{i} \in \chi_\mathbf{l} := \{\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d : 1 \leq i_j \leq n_j, j = 1, \ldots, d\} \; .$$

The space

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

is equal to the underlying Sobolev space $H_0^1(\Omega)$ up to completion with the H^1-norm, see [BG04]. For our numerical computation, we have to resort to a finite-dimensional subset. To this end, we use an index set $\mathcal{I} \subset \mathbb{N}^d, |\mathcal{I}| < \infty$, which defines the subspaces included in the discretization as

$$V_\mathcal{I} = \sum_{\mathbf{l} \in \mathcal{I}} V_\mathbf{l} \; .$$

A proper choice of $\mathcal{I}$ now depends – besides the error we want to achieve – on the smoothness of the function class we want to approximate. For example, the full grid space with index set

$$\mathcal{I}_k^{(\infty)} = \{\mathbf{l} \in \mathbb{N}^d : |\mathbf{l}|_\infty \leq k\}$$

has the approximation property

$$\inf_{v \in V_{\mathcal{I}_k^{(\infty)}}} \|u - v\|^2_{H^s(\Omega)} \leq c 2^{-2(t-s)k} \|u\|^2_{H^t(\Omega)}$$

with rate[2] $t - s$ and $u \in H_0^t(\Omega)$. Its number of degrees of freedom grows by $\mathcal{O}(2^{kd})$. 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 $u \in H^t_{0,\text{mix}}(\Omega)$, the sparse grid index set

$$\mathcal{I}_k^{(1)} = \{\mathbf{l} \in \mathbb{N}^d : |\mathbf{l}|_1 \leq k + d - 1\} \tag{19}$$

circumvents this problem to some extent. The rate of the best approximation in dependence of k

$$\inf_{v \in V_{\mathcal{I}_k^{(1)}}} \|u - v\|^2_{H^s(\Omega)} \leq c 2^{-2(t-s)k} \|u\|^2_{H^t_{\text{mix}}(\Omega)}$$

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

is the same[3] as for the full grid space, i.e. $t - s$, but the number of degrees of freedom now only grows by $\mathcal{O}(2^k k^{d-1})$. This is a substantial improvement in comparison to the full grid case. For further details, see [GK09]. A related approach, adaptive sparse grids based on a linear spline basis, has been employed for option pricing without jumps in [BHPS11].

Next, to compensate for the transformation (16), we want our discretization to have a refinement level in each dimension $i = 1, \ldots, d$, which is logarithmically proportional to the size $\beta_i - \alpha_i$ of the respective dimension in $\mathcal{D}$. This then leads to anisotropic sparse grids. First, we need to determine the level shifts

$$\rho_i = \left\lfloor \log_2 \frac{\beta_1 - \alpha_1}{\beta_i - \alpha_i} \right\rfloor + 1$$

for $i = 1, \ldots, d$, that are necessary to compensate for the former anisotropies. Then, the index set

$$\mathcal{I}_k^{(*)} = \{\mathbf{l} \in \mathbb{N}^d : \frac{l_1 - 1}{\max(k - \rho_1, 1)} + \cdots + \frac{l_d - 1}{\max(k - \rho_d, 1)} \leq 1\} \tag{20}$$

gives us an anisotropic sparse grid on level k, with the classical sparse grid as a special case for $\rho_1 = \cdots = \rho_d = 1$. Anisotropies may reduce the sparse grid cost complexity even further, see [GH12] for a discussion in this direction.

Remark 2. It is furthermore 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]. For algorithmic reasons, we then need the additional condition

$$\mathbf{l} \in \mathcal{I}, \mathbf{k} \in \mathbb{N}^d, \mathbf{k} \leq \mathbf{l} \Rightarrow \mathbf{k} \in \mathcal{I} \,, \tag{21}$$

where $\mathbf{k} \leq \mathbf{l}$ is understood componentwise.

Now we present the final equations that are being solved numerically for the parameters and model assumptions specified in Subsect. 2.2, where also the domain transformation (16) is being considered. For any valid index set $\mathcal{I} \subset \mathbb{N}^d$, let us denote by $\mathbf{u} = (b_{\mathbf{l},\mathbf{i}})_{\mathbf{l}\in\mathcal{I},\mathbf{i}\in\chi_\mathbf{l}}$ a block vector of $N := \sum_{\mathbf{l}\in\mathcal{I}} |\chi_\mathbf{l}|$ coefficients used to represent functions

$$u(\mathbf{x}) = \sum_{\mathbf{l}\in\mathcal{I}} \sum_{\mathbf{i}\in\chi_\mathbf{l}} u_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) \in V_\mathcal{I} \,.$$

Then, our variational problem (15) can be discretized as follows: For $i = 0, \ldots, M - 1$ find $\mathbf{u}^{(i+1)}$ s.t.

$$\left(\Delta t^{-1}\mathbf{M} + \theta\mathbf{A}\right)\mathbf{u}^{(i+1)} = \left(\Delta t^{-1}\mathbf{M} + (\theta - 1)\mathbf{A}\right)\mathbf{u}^{(i)}, \tag{22}$$

with the block-structured mass matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$

[3] For $s = 0$ an additional logarithmic term appears in the error estimate.

$$(\mathbf{M})_{(\mathbf{l},\mathbf{i}),(\mathbf{k},\mathbf{j})} = \int_\Omega \phi_{\mathbf{l},\mathbf{i}}(\mathbf{x})\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x})d\mathbf{x}$$

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

$$(\mathbf{A})_{(\mathbf{l},\mathbf{i}),(\mathbf{k},\mathbf{j})} = a(\phi_{\mathbf{l},\mathbf{i}}, \phi_{\mathbf{k},\mathbf{j}})$$

with the bilinear form $a(\cdot,\cdot)$ from (17). The initial value $\mathbf{u}^{(0)}$ is set to the L_2-projection of the payoff function into the space $V_\mathcal{I}$.

Note here that the representation of functions in $V_\mathcal{I} = \sum_{\mathbf{l}\in\mathcal{I}} V_\mathbf{l}$ is not unique, since all subspaces $V_\mathbf{l}, \mathbf{l} \in \mathcal{I}$ appear in our discretization. This means that operator matrices $\mathbf{A}$ and $\mathbf{M}$ for this discretization have a non-trivial kernel and the system matrix cannot be inverted. However, Krylov subspace methods will converge as long as the right-hand side is within the range of the operator matrix, see [Kaa88, Gri94] for further details.

3.3 Preconditioning

We now want to find an optimal preconditioner for our linear system (22). To this end, we need to bound the quotient $\tilde{\lambda}_{\max}(\mathbf{A})/\tilde{\lambda}_{\min}(\mathbf{A})$ independently of $\dim V_\mathcal{I}$, where

$$\tilde{\lambda}_{\min}(\mathbf{A}) = \min_{\mathbf{u}\perp\ker(\mathbf{A})} \frac{\langle \mathbf{A}\mathbf{u}, \mathbf{u}\rangle}{\|\mathbf{u}\|_2^2}$$

and

$$\tilde{\lambda}_{\max}(\mathbf{A}) = \max_{\mathbf{u}\perp\ker(\mathbf{A})} \frac{\langle \mathbf{A}\mathbf{u}, \mathbf{u}\rangle}{\|\mathbf{u}\|_2^2}$$

denote the minimum and the maximum of the Rayleigh-quotient restricted to the subspace $\{\mathbf{u} : \mathbf{u} \perp \ker(\mathbf{A})\}$.

The preconditioning of generating systems on regular sparse grid spaces with index set (19) is discussed in [GO94, GO95] within the framework of multilevel subspace splittings. A simple diagonal scaling results in condition numbers that are bounded by $\mathcal{O}(k^{d-2})$, which is a substantial improvement, but does not give an optimal method yet.

To obtain optimal condition numbers of $\mathcal{O}(1)$ we proceed as follows: We rely on the stable subspace splitting

$$\{H^1; \|\cdot\|_{H^1}^2\} = \sum_{\mathbf{l}} \{W_\mathbf{l}; 2^{2|\mathbf{l}|_\infty}\|\cdot\|_{L_2}^2\}\,, \tag{23}$$

where $\mathbf{l} \in \mathbb{N}^d$ denotes a d-dimensional multi-index and $W_\mathbf{l}$ is the respective L_2-orthogonal complement space, see [GK09]. As the nodal basis on level $\mathbf{l}$ is a stable splitting for $V_\mathbf{l}$ with a condition number independent of $\mathbf{l}$ and $W_\mathbf{l} \subset V_\mathbf{l}$, we can derive an optimal preconditioner by splitting a function into

its orthogonal complements and then representing them in our generating system subspaces $V_{\mathbf{l}}$.

To explain this in more detail, let us first describe the case of a one-dimensional discretization up to level k, i.e. $\mathcal{I} = \{(l) : 1 \leq l \leq k\}$ and $d = 1$. Let $Q_l : V_k \to V_l$ be the L_2-orthogonal projection to V_l and $Q_0 = 0$. Because of the nestedness

$$V_l \subset V_{l'} \quad \text{for} \quad l \leq l',$$

the telescopic expansion

$$u_k = \sum_{l=1}^{k} (Q_l - Q_{l-1}) u_k \,, \qquad u_k \in V_k$$

is an orthogonal decomposition with $(Q_l - Q_{l-1})u_k \in W_l$.

The multi-dimensional case can be obtained by tensorization arguments. Then, for $u_{\mathcal{I}} \in V_{\mathcal{I}}$ with an index set $\mathcal{I} \subset \mathbb{N}^d$ satisfying (21), the expression

$$w_{\mathbf{l}} = \sum_{\mathbf{m} \in \{0,1\}^d} (-1)^{|\mathbf{m}|_1} Q_{l_1 - m_1, \ldots, l_d - m_d} u_{\mathcal{I}} \tag{24}$$

denotes the orthogonal complement of $u_{\mathcal{I}}$ in the space $W_{\mathbf{l}}, \mathbf{l} \in \mathcal{I}$.

The algorithmic implementation is straightforward. We first have to compute the projections by

$$(Q_{\mathbf{l}} u_{\mathcal{I}}, v_{\mathbf{l}})_{L_2} = (u_{\mathcal{I}}, v_{\mathbf{l}})_{L_2} \quad \text{for all} \quad v_{\mathbf{l}} \in V_{\mathbf{l}}, \quad \mathbf{l} \in \mathcal{I} \,. \tag{25}$$

To this end, all right-hand sides for $\mathbf{l} \in \mathcal{I}$ are extracted from the result of one single mass matrix multiplication applied to $u_{\mathcal{I}}$. The operator matrix on the left-hand side of (25) is just a d-fold tensor-product of one-dimensional mass matrices, which themselves are tridiagonal, and thus is easily invertible in $\mathcal{O}(2^{|\mathbf{l}|_1})$ operations for a fixed $\mathbf{l}$. Secondly, after having calculated all discrete functions $Q_{\mathbf{l}} u_{\mathcal{I}}$, we have to sum them up according to (24). This can also be done in $\mathcal{O}(2^{|\mathbf{l}|_1})$ operations, which is again linear in the degrees of freedom. After scaling with the $\mathbf{l}$-dependent weights in the norm equivalence (23) we then obtain an optimal preconditioner with a condition number bounded independently of the space $V_{\mathcal{I}}$. Note that this is achieved without explicitly discretizing the subspaces $W_{\mathbf{l}}$ and the overall system matrix with prewavelets or similar, more complicated basis functions, but merely by sticking to the more simple and natural generating system. A one-dimensional example[4] is given in Fig. 2. We observe that the oscillations on finer scales occur mainly at the position of the non-differentiable kink.

[4] The discretization in this example includes boundary functions otherwise not used for computation.

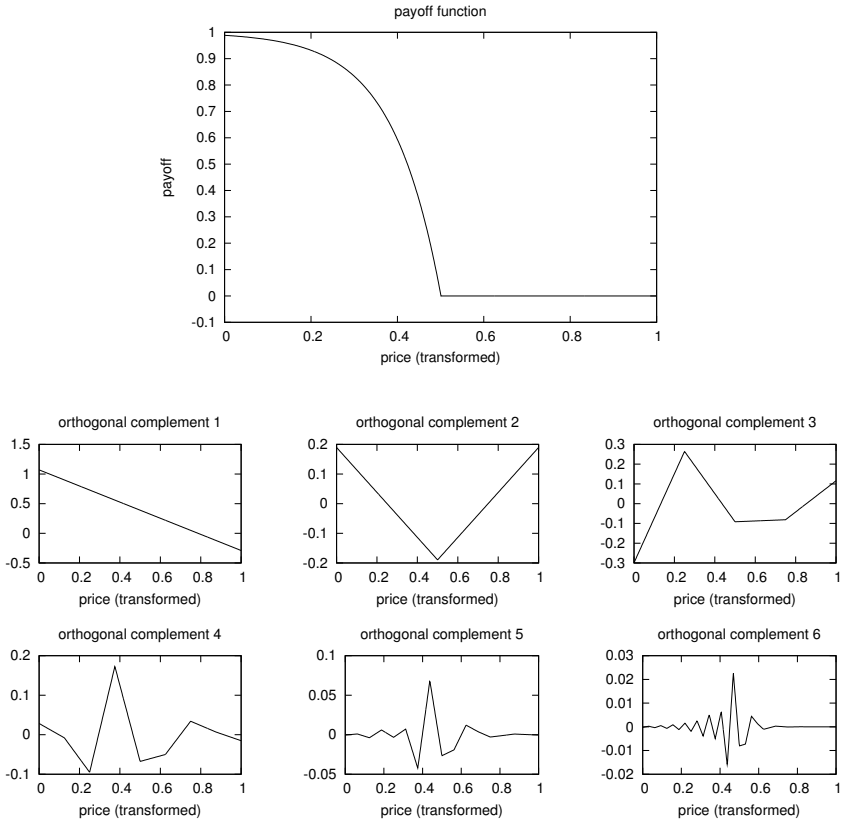

Fig. 2 The payoff function of a one-dimensional put option in the log-space (top). The first six orthogonal complements of this function (middle and bottom)

3.4 Operator application

Within such an iterative solver the matrix-vector multiplication must be frequently processed. Here, a complexity issue may occur due to the non-trivial interactions between the different subspaces involved in the generating system approach and, additionally, due to the integro-operator in the stiffness matrix. We now describe how a general linear tensor product operator[5]

$$A^{(1)} \otimes A^{(2)} \otimes \cdots \otimes A^{(d)} \tag{26}$$

can be applied to a sparse grid approximation $u_{\mathcal{I}} = [u_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}}$ of u using a number of operations, which is of optimal order, e.g. of $\mathcal{O}(2^k k^{d-1})$ for regular

[5] Note that the numbers $(1), \ldots, (d)$ in the exponents are no powers but indices, indicating that different operations may be carried out in different dimensions.

sparse grids (19). Consequently, all sums of tensor-product operators can then also be applied with optimal cost complexity.

If the index set $\mathcal{I}$ of our sparse grid would allow a representation $\mathcal{I} = I_1 \times \cdots \times I_d$ with $I_i \subset \mathbb{N}, i = 1, \ldots, d$, we could easily apply the tensor-product operator (26) dimension by dimension. But since the index $\mathcal{I}$ can in general not be represented as a simple cross product, we need a sophisticated algorithm known as the unidirectional principle to maintain optimal complexity. Our aim is to calculate

$$\mathbf{v}_\mathbf{l} = \sum_{\mathbf{l}' \in \mathcal{I}} \left(\mathbf{A}^{(1)}_{l_1,l'_1} \otimes \mathbf{A}^{(2)}_{l_2,l'_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,l'_d} \right) \mathbf{u}_{\mathbf{l}'}$$

for all $\mathbf{l} \in \mathcal{I}$, where $\mathbf{A}^{(j)}_{l_j,l'_j}, j = 1, \ldots, d$, denotes the operator matrix of $A^{(j)}$ in the weak formulation for discretization level l'_j and test function level l_j. First, let us investigate the one-dimensional case[6]. Up to the level k, we have

$$\begin{aligned} \mathbf{v}_l &= \sum_{l'=1}^{k} \mathbf{A}_{l,l'} \mathbf{u}_{l'} \\ &= \sum_{l'=1}^{l} \mathbf{A}_{l,l'} \mathbf{u}_{l'} + \sum_{l'=l+1}^{k} \mathbf{A}_{l,l'} \mathbf{u}_{l'} \ . \end{aligned} \tag{27}$$

Both sums in (27) need to be treated separately. To this end, we introduce the matrices $\mathbf{I}_{l,l'}$ which serve as restriction operators for $l < l'$ and as prolongation operators for $l > l'$. The matrices $\mathbf{A}_{l,l'}$ can then be written as

$$\mathbf{A}_{l,l'} = \begin{cases} \mathbf{A}_{l,l} \mathbf{I}_{l,l'} & \text{for} \quad l \geq l' \ , \\ \mathbf{I}_{l,l'} \mathbf{A}_{l',l'} & \text{for} \quad l < l' \ . \end{cases}$$

In the following two algorithms, the prolongations and restrictions are of central importance to efficiently transport intermediate results.

3.4.1 BottomUp algorithm

The BottomUp algorithm calculates all sums

$$\mathbf{v}_l = \sum_{l'=l+1}^{k} \mathbf{A}_{l,l'} \mathbf{u}_{l'} \tag{28}$$

[6] For notational convenience we drop the dimension index $^{(j)}$ in $\mathbf{A}^{(j)}_{l_j,l'_j}$ in the following calculations.

for $l = 1, \ldots, k$ in linear time. If we express these operations (28) by means of a matrix including all levels, we obtain an upper diagonal form

$$\begin{pmatrix} 0 & \mathbf{A}_{1,2} & \mathbf{A}_{1,3} & \ldots & \mathbf{A}_{k,k} \\ & 0 & \mathbf{A}_{2,3} & \ldots & \mathbf{A}_{2,k} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & \mathbf{A}_{k-1,k} \\ & & & & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{I}_{1,2}\mathbf{A}_{2,2} & \mathbf{I}_{1,3}\mathbf{A}_{3,3} & \ldots & \mathbf{I}_{1,k}\mathbf{A}_{k,k} \\ & 0 & \mathbf{I}_{2,3}\mathbf{A}_{3,3} & \ldots & \mathbf{I}_{2,k}\mathbf{A}_{k,k} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & \mathbf{I}_{k-1,k}\mathbf{A}_{k,k} \\ & & & & 0 \end{pmatrix} .$$

Obviously, the matrix can be expressed using the restrictions $\mathbf{I}_{l,l'}, l < l'$, and the isotropic matrices $\mathbf{A}_{l,l}$. This gives rise to a recursive formulation for $l = k-1, \ldots, 1$

$$\begin{aligned} \mathbf{v}_l &= \sum_{l'=l+2}^{k} \mathbf{A}_{l,l'} u_{l'} + \mathbf{A}_{l,l+1}\mathbf{u}_{l+1} \\ &= \mathbf{I}_{l,l+1} \left(\sum_{l'=(l+1)+1}^{k} \mathbf{A}_{l+1,l'}\mathbf{u}_{l'} + \mathbf{A}_{l+1,l+1}\mathbf{u}_{l+1} \right) \\ &= \mathbf{I}_{l,l+1} \left(\mathbf{v}_{l+1} + \mathbf{A}_{l+1,l+1}\mathbf{u}_{l+1} \right) . \end{aligned} \tag{29}$$

Clearly, all $\mathbf{v}_l$ for $1 \leq l \leq k$ 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.

3.4.2 TopDown algorithm

The TopDown algorithm uses a similar recursive relation to calculate the left-hand sides of

$$\mathbf{v}_l = \sum_{l'=1}^{l} \mathbf{A}_{l,l'}\mathbf{u}_{l'} \tag{30}$$

for $1 \leq l \leq k$. The equations (30) can be rewritten as a matrix-vector multiplication with matrix

$$\begin{pmatrix} \mathbf{A}_{1,1} & & & & \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & & & \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{A}_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{A}_{k,1} & \mathbf{A}_{k,2} & \mathbf{A}_{k,3} & \ldots & \mathbf{A}_{k,k} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{1,1} & & & & \\ \mathbf{A}_{2,2}\mathbf{I}_{2,1} & \mathbf{A}_{2,2} & & & \\ \mathbf{A}_{3,3}\mathbf{I}_{3,1} & \mathbf{A}_{3,3}\mathbf{I}_{3,2} & \mathbf{A}_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{A}_{k,k}\mathbf{I}_{k,1} & \mathbf{A}_{k,k}\mathbf{I}_{k,2} & \mathbf{A}_{k,k}\mathbf{I}_{k,3} & \ldots & \mathbf{A}_{k,k} \end{pmatrix} ,$$

which can solely be expressed using the prolongations $\mathbf{I}_{l,l'}, l > l'$, and the isotropic operator matrices $\mathbf{A}_{l,l}$. We can precalculate vectors $\mathbf{w}_l$ for $l = 2, \ldots, k$ using the recursive relationship

$$\begin{aligned}\mathbf{w}_l &:= \sum_{l'=1}^{l} \mathbf{I}_{l,l'}\mathbf{u}_{l'} \\ &= \sum_{l'=1}^{l-1} \mathbf{I}_{l,l-1}\mathbf{I}_{l-1,l'}\mathbf{u}_{l'} + \mathbf{u}_l \\ &= \mathbf{I}_{l,l-1}\mathbf{w}_{l-1} + \mathbf{u}_l\end{aligned}$$

starting with $\mathbf{w}_1 := \mathbf{u}_1$. Now (30) can be expressed by

$$\mathbf{v}_l = \sum_{l'=1}^{l} \mathbf{A}_{l,l}\mathbf{I}_{l,l'}\mathbf{u}_{l'} = \mathbf{A}_{l,l}\sum_{l'=1}^{l} \mathbf{I}_{l,l'}\mathbf{u}_{l'} = \mathbf{A}_{l,l}\mathbf{w}_l \ . \tag{31}$$

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.

3.4.3 Multi-dimensional case

The multi-dimensional case can be reduced to the application of the one-dimensional algorithms by a splitting and rearrangement of the sum

$$v_{\mathbf{l}} = \sum_{\mathbf{l}' \in \mathcal{I}} \mathbf{A}^{(1)}_{l_1,l'_1} \otimes \mathbf{A}^{(2)}_{l_2,l'_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,l'_d} u_{\mathbf{l}'} \ ,$$

which then leads to the equation

$$\mathbf{v}_{\mathbf{l}} = \sum_{\substack{l'_1 \le l_1 \text{with} \\ (l'_1,l_2,\ldots,l_d)\in\mathcal{I}}} (\mathbf{A}^{(1)}_{l_1,l'_1} \otimes \mathbf{I} \otimes \cdots \otimes \mathbf{I}) \cdot \tag{32}$$

$$\sum_{\substack{(l'_2,\ldots,l'_d) \text{ with} \\ (l'_1,l'_2,\ldots,l'_d)\in\mathcal{I}}} \left(\mathbf{I} \otimes \mathbf{A}^{(2)}_{l_2,l'_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,l'_d}\right) \mathbf{u}_{(l'_1,l'_2,\ldots,l'_d)} \tag{33}$$

$$+ \sum_{\substack{(l'_2,\ldots,l'_d) \text{ with} \\ (l_1,l'_2,\ldots,l'_d)\in\mathcal{I}}} \left(\mathbf{I} \otimes \mathbf{A}^{(2)}_{l_2,l'_2} \otimes \cdots \otimes \mathbf{A}^{(d)}_{l_d,l'_d}\right) \cdot \tag{34}$$

$$\sum_{\substack{l'_1 > l_1 \text{ with} \\ (l'_1,l'_2,\ldots,l'_d)\in\mathcal{I}}} (\mathbf{A}^{(1)}_{l_1,l'_1} \otimes \mathbf{I} \otimes \cdots \otimes \mathbf{I})\mathbf{u}_{(l'_1,l'_2,\ldots,l'_d)} \ . \tag{35}$$

Now, (32) resembles the application of the one-dimensional TopDown algorithm, and (35) resembles the application of the one-dimensional BottomUp algorithm. The sums (33) and (34) are the result of a recursive application of the multi-dimensional algorithm with the first dimension left unchanged. The condition (21) ensures that the storage of intermediate results requires

space and time just in the same order as for $u_{\mathcal{I}}$ and $v_{\mathcal{I}}$. Specifically, we know that

$$(l_1, \ldots, l_d) \in \mathcal{I}, l_1' \leq l_1 \Rightarrow (l_1', l_2, \ldots, l_d) \in \mathcal{I}$$

in (32) and

$$(l_1', \ldots, l_d') \in \mathcal{I}, l_1' > l_1 \Rightarrow (l_1, l_2', \ldots, l_d') \in \mathcal{I}$$

in (34), which means that intermediate results can be represented as generalized sparse grid functions on the same index set $\mathcal{I}$.

The origins of this algorithm trace back to [Bun92, BZ96] with focus on partial differential operators. Now, with the generic concept presented in this paper, 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. 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 application is linear. But this is possible in the special case of the Kou model, which we will show in the next subsection. Note furthermore that, independently of this work, a similar abstraction of the unidirectional principle for multilevel discretizations has been presented in [Zei11].

3.5 Galerkin recurrence formula for the Kou model

In Subsect. 3.4, we have assumed that the matrices $\mathbf{A}^{(1)}_{l_1,l_1'}, \ldots, \mathbf{A}^{(d)}_{l_d,l_d'}$ can be applied in linear time. This is possible for differential operators, as their matrices are inherently sparse for nodal basis functions. However, integral operators do not have this nice property. In [Toi08], a recurrence formula is used to apply the integral operator of the Kou model for the finite difference case in linear time. We now derive a similar result for the Galerkin method.

In the following, let $\{\phi_i\}_{i=1}^{n_l}$ be a set of one-dimensional linear spline basis functions on level l with mesh-width $h = 2^{-l}$ and the relation

$$\phi_i(x) = \phi_{i+1}(x + h), \quad i = 1, \ldots, n_l - 1 \,. \tag{36}$$

In the Galerkin approach for the integro-operator of the Kou model, we need to calculate for all $i = 1, \ldots, n_l$ the expression[7]

$$v_i = \int_{[0,1]} \int_{\mathbb{R}} u(x + z) \kappa_{p,\eta,\mu}(z) \mathrm{d}z \phi_i(x) \mathrm{d}x \,.$$

Here, $\kappa_{p,\eta,\mu}$ is the Kou density function from (2), and $u : [0, 1] \to \mathbb{R}$ denotes a function that can be represented by

[7] Here and in the following, we have omitted time related indices and the dependence on other dimensions.

$$u(x) = \sum_{j=1}^{n_l} u_j \phi_j(x) \ .$$

This translates to the matrix-vector-product

$$\mathbf{v} = \mathbf{A}\mathbf{u}$$

with $\mathbf{v} = (v_1, \ldots, v_{n_l})$, $\mathbf{u} = (u_1, \ldots, u_{n_l})$ and

$$\begin{aligned}(\mathbf{A})_{i,j} &= \int_{[0,1]} \int_{\mathbb{R}} \phi_j(x+z) \kappa_{p,\eta,\mu}(z) \mathrm{d}z \phi_i(x) \mathrm{d}x \\ &= \int_{[0,1]} \int_{\mathbb{R}} \phi_j(z) \kappa_{p,\eta,\mu}(z-x) \phi_i(x) \mathrm{d}z \mathrm{d}x \ .\end{aligned} \tag{37}$$

The matrix $\mathbf{A}$ is dense but has Toeplitz structure: Applying (36) to both, ϕ_i and ϕ_j in (37) gives $(\mathbf{A})_{i,j} = (\mathbf{A})_{i+1,j+1}$. This property of the matrix would allow us to execute the matrix-vector multiplication in $\mathcal{O}(n_l \log n_l)$ instead of $\mathcal{O}(n_l^2)$.

A different approach that takes also the structure of $\kappa_{p,\eta,\mu}$ into account achieves even a linear runtime complexity. To this end, let us assume that $i, j \in \mathbb{N}, n_l \geq j \geq i+2$. Then we know that the interior of the supports of ϕ_i and ϕ_j is disjoint and that $z > x$ for $\phi_j(z) \neq 0$ and $\phi_i(x) \neq 0$. This allows us to obtain

$$\begin{aligned}(\mathbf{A})_{i,j} &= \int_{[0,1]} \int_{\mathbb{R}} \phi_j(z) k(z-x) \phi_i(x) \mathrm{d}z \ \mathrm{d}x \\ &= \int_{[0,1]} \int_{\mathbb{R}} \phi_j(z) p\mu e^{-\mu(z-x)} \phi_{i+1}(x+h) \mathrm{d}z \ \mathrm{d}x \\ &= \int_{[h,1+h]} \int_{\mathbb{R}} \phi_j(z) p\mu e^{-\mu(z-x+h)} \phi_{i+1}(x) \mathrm{d}z \ \mathrm{d}x \\ &= e^{-h\mu} (\mathbf{A})_{i+1,j}.\end{aligned}$$

A similar argument gives $(\mathbf{A})_{i,j} = e^{-h\eta} (\mathbf{A})_{i-1,j}$ 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

$$v_i^{\leftarrow} = \sum_{j=1}^{i-2} (\mathbf{A})_{i,j} u_j, \quad v_i^{\circ} = \sum_{j=i-1}^{i+1} (\mathbf{A})_{i,j} u_j, \quad v_i^{\rightarrow} = \sum_{j=i+2}^{n_l} (\mathbf{A})_{i,j} u_j \ .$$

With the recursive relationships

$$\begin{aligned} v_i^{\leftarrow} &= \sum_{j=1}^{i-3} (\mathbf{A})_{i,j} u_j + (\mathbf{A})_{i,i-2} u_{i-2} \\ &= e^{-h\eta} \sum_{j=1}^{(i-1)-2} (\mathbf{A})_{i-1,j} u_j + (\mathbf{A})_{i,i-2} u_{i-2} \\ &= e^{-h\eta} v_{i-1}^{\leftarrow} + (\mathbf{A})_{i,i-2} u_{i-2} \end{aligned}$$

for $i = 4, \ldots, n_l$ and

$$\begin{aligned} v_i^{\rightarrow} &= \sum_{j=i+3}^{n_l} (\mathbf{A})_{i,j} u_j + (\mathbf{A})_{i,i+2} u_{i+2} \\ &= e^{-h\mu} \sum_{j=(i+1)+2}^{n_l} (\mathbf{A})_{i+1,j} u_j + (\mathbf{A})_{i,i+2} u_{i+2} \\ &= e^{-h\mu} v_{i+1}^{\rightarrow} + (\mathbf{A})_{i,i+2} u_{i+2} \end{aligned}$$

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 $\mathbf{v} = \mathbf{A}\mathbf{u}$ in $\mathcal{O}(n_l)$ complexity even though the matrix $\mathbf{A}$ is dense.

So far, we have described the application of the integro-operator of the Kou model to a one-dimensional linear spline 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 (32)–(35), which exploits a given tensor product structure and requires only one-dimensional non-hierarchical applications of the Kou integral-operator in (29) and (31). This altogether allows the application of the operator matrices in the discretized equation (22) in just linear time.

4 Numerical experiments

In the following numerical experiments we use the described PIDE solver for the pricing of European basket put options. The focus of our studies is on space discretization, that means we have chosen the domain of computation large enough and the time steps small enough such that the main error now stems from the space discretization only. We are especially interested in the spatial convergence rates of our method. To this end, we look at the L_2-error of the the solution at $\tau = T$ and at the relative error of the option price at a predefined point of evaluation. Assuming a relationship

$$e \leq cN^{-\alpha}$$

with e being the error and N the total number of degrees of freedom, we can estimate the convergence rate α by computing

$$\alpha \approx -\frac{\log e_2 - \log e_1}{\log N_2 - \log N_1}$$

for two successive discretization spaces with N_1 and N_2 degrees of freedom and associated errors e_1 and e_2, respectively. As all computational steps can be carried out in linear complexity, this is a relevant measure for the efficiency of our method.

4.1 European put option

We now price a European put option on a single asset, i.e. $d = 1$, with the parameters

$$T = 0.2\ ,\ K = 1.0\ ,\ r = 0.00\ ,\ S = 1.0\ ,\ \sigma = 0.2\ ,$$

and a jump part

$$\lambda = 0.2\ ,\ \eta = 2.0\ ,\ \mu = 3.0 \text{ and } p = 0.5$$

using the equations (22). The domain is chosen as

$$\mathcal{D} = (-35.781, 35.773)$$

before being scaled down to Ω by (16), and the choice of $\Delta t = 2^{-11}T$ results in 2048 equidistant time steps. The reference price is given by 0.042647805, compare [Toi08]. Without the jump part, the correct Black–Scholes price of the option can be given analytically as 0.035670591. We will use both models to test our method. Figure 3 shows the convergence plots of the relative pricing error with respect to the degrees of freedom[8] with and without the jump part. Here and in Table 1 we see that we essentially get the same rate of 2 as for the simple Black–Scholes PDE. The last measured rate decreases slightly to about 1.8, which can be explained by the time discretization and domain truncation error starting to kick in at that level.

Figure 4 shows that the integro-term just adds a small constant factor to the runtime. Moreover, we see that the runtime is asymptotically proportional to the degrees of freedom as claimed in Subsect. 3.5.

[8] In one dimension, full and sparse grids are of course the same.

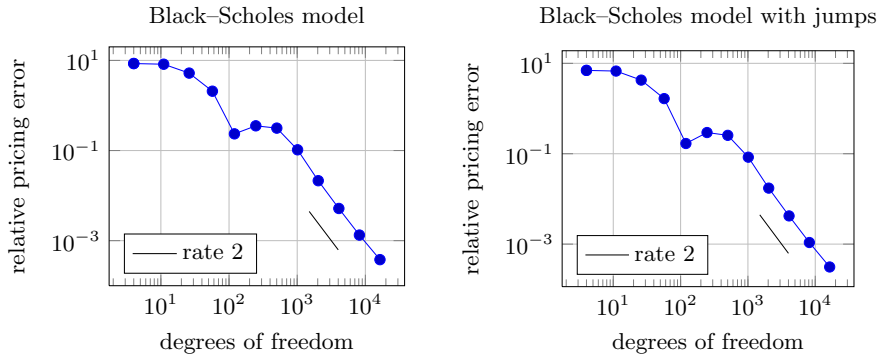

Fig. 3 The relative pricing error at $S_1 = 1.0$ versus the degrees of freedom of the discretization for the Black–Scholes model without jumps (left) and with Kou jumps (right)

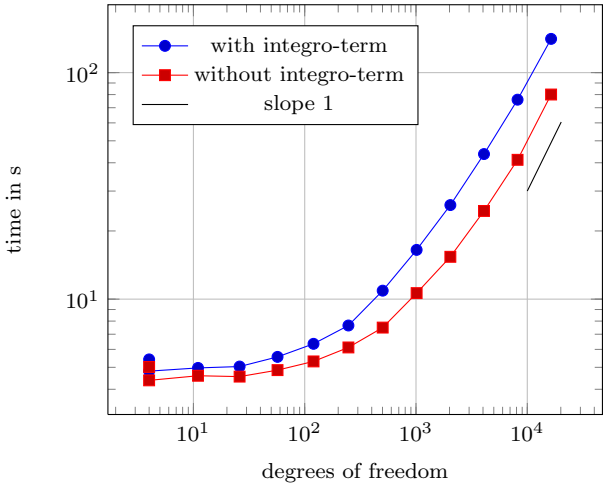

Fig. 4 The run time versus the degrees of freedom. The slope of 1 in this log-log plot relates to the formula $3 \cdot 10^{-3}x$, which indicates that the runtime grows linearly with the degrees of freedom

4.2 Two-dimensional example

We now define two underlyings by

$$r = 0.00 \quad \text{and} \quad \begin{cases} S_1 = 1.0\,, \ \sigma_1 = 0.2\,, \\ S_2 = 1.0\,, \ \sigma_2 = 0.2\,. \end{cases}$$

The first jump term

Table 1 Relative errors and rates α for the one-dimensional option pricing problem with and without jumps. The degrees of freedom (DOFs) refer to the size of the multilevel generating system

		without jumps		**with jumps**	
level	DOFs	pricing error	rate α	pricing error	rate α
2	4	$8.53 \cdot 10^{0}$	N/A	$6.98 \cdot 10^{0}$	N/A
3	11	$8.23 \cdot 10^{0}$	0.03	$6.75 \cdot 10^{0}$	0.03
4	26	$5.23 \cdot 10^{0}$	0.53	$4.26 \cdot 10^{0}$	0.54
5	57	$2.07 \cdot 10^{0}$	1.18	$1.66 \cdot 10^{0}$	1.20
6	120	$2.35 \cdot 10^{-1}$	2.93	$1.68 \cdot 10^{-1}$	3.07
7	247	$3.54 \cdot 10^{-1}$	−0.57	$2.94 \cdot 10^{-1}$	−0.77
8	502	$3.14 \cdot 10^{-1}$	0.17	$2.55 \cdot 10^{-1}$	0.20
9	1013	$1.04 \cdot 10^{-1}$	1.58	$8.38 \cdot 10^{-2}$	1.59
10	2036	$2.14 \cdot 10^{-2}$	2.26	$1.73 \cdot 10^{-2}$	2.26
11	4083	$5.18 \cdot 10^{-3}$	2.04	$4.21 \cdot 10^{-3}$	2.04
12	8178	$1.33 \cdot 10^{-3}$	1.96	$1.09 \cdot 10^{-3}$	1.95
13	16369	$3.79 \cdot 10^{-4}$	1.81	$3.12 \cdot 10^{-4}$	1.80

$$\lambda = 0.3 \quad \begin{cases} \eta_1 = 8.0\ ,\ \mu_1 = 8.0\ ,\ p_1 = 0.0\ , \\ \eta_2 = 8.0\ ,\ \mu_2 = 8.0\ ,\ p_2 = 0.0\ , \end{cases} \tag{38}$$

introduces negative jumps in both underlyings, while the second jump term

$$\lambda = 0.2\ ,\ \eta_1 = 10.0\ ,\ \mu = 10.0 \text{ and } p_1 = 0.5 \tag{39}$$

only affects the first dimension. First, we consider the geometric call option with payoff

$$g(\mathbf{x}) = \max(0, e^{\frac{x_1}{2}+\frac{x_2}{2}} - K) \tag{40}$$

on the domain $\mathcal{D} = (\frac{K}{2}, \frac{3K}{2})^2$ for $K = 1$ as suggested in [NHW10] for the Black-Scholes model. Note here that with this particular choice the non-differentiable kink is not included in the domain of computation, which is favorable for discretizations that rely on additional smoothness constraints like sparse grids. We include both jump terms and calculate the solution approximation $u_{\mathcal{I}^{(*)}}$ based on our PIDE solver approach with anisotropic sparse and full grids, compare (20). In this and in all following experiments, we choose $\Delta t = 2^{-8}T$, which results in 256 equidistant time steps. In Fig. 5 the L_2-error at $T = 0.2$ is measured against a full grid solution on level 10 after the linear transformation (16) to the unit square $\Omega = (0, 1)^2$. As we can see, we achieve roughly a rate of 2 with respect to the degrees of freedom used for the sparse grids, while the full grid rate deteriorates to $\frac{2}{d} = 1$ for $d = 2$.

Now we get back to our usual basket payoff

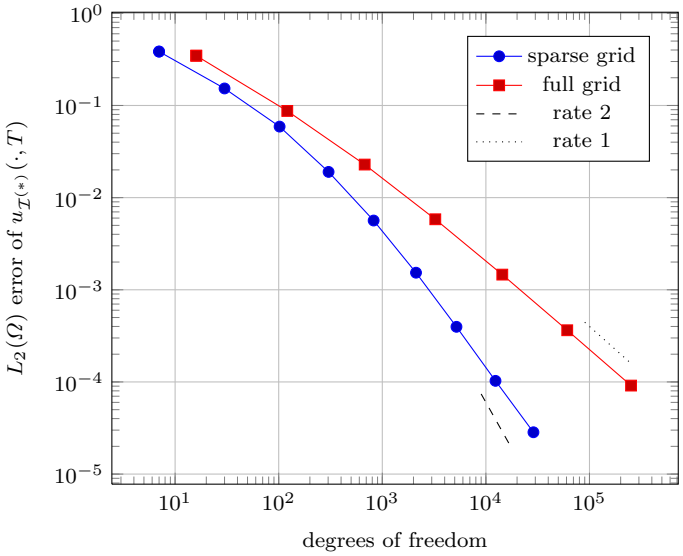

Fig. 5 $L_2(\Omega)$ error of the solution for the geometric call option (40) at $\tau = T$ with a full grid level 10 as reference solution.

$$g(\mathbf{x}) = \max(0, K - \sum_{i=1}^{d} e^{x_i}) \ , \tag{41}$$

which is equivalent to (5) after the logarithmic transformation of the coordinates (10). Strike and maturity are chosen as $K = 2$ and $T = 0.2$, respectively. We want to price the option on the underlyings with both jump terms and choose the domain

$$\mathcal{D} = (-0.984, 0.989) \times (-0.942, 0.948) \ ,$$

which now includes the non-differentiable kink of the payoff function. In terms of the L_2-error at $T = 0.2$, we achieve asymptotically the rate 2 for the sparse grid approach in contrast to the rate 1 for the full grid, see Fig. 6. This is a noteworthy result, as the payoff function lacks the mixed smoothness regularity, which is typically required by sparse grids. However, it is well-known [Tho97] that the finite element solutions to parabolic problems converge to full order due to the smoothing effect of the solution/propagation operator even when the initial data are nonsmooth.

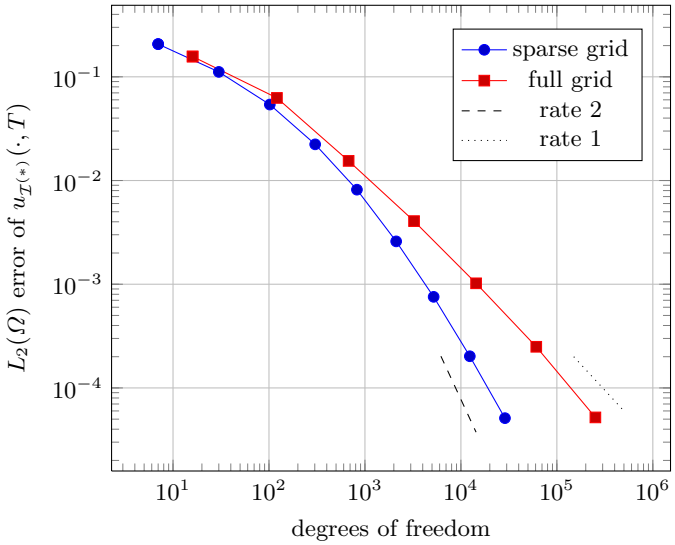

Fig. 6 $L_2(\Omega)$ error of the solution for the basket put option (41) at $\tau = T$ with a full grid level 10 as reference solution

4.3 Three-dimensional option

As an example for a three-dimensional option we set

$$T = 0.2\ ,\ K = 3.0\ ,\ r = 0.00 \quad \text{and} \quad \begin{cases} S_1 = 1.0\ ,\ \sigma_1 = 0.2\ , \\ S_2 = 1.0\ ,\ \sigma_2 = 0.1\ , \\ S_3 = 1.0\ ,\ \sigma_1 = 0.05\ , \end{cases}$$

and choose a jump term

$$\lambda = 0.2 \quad \begin{cases} \eta_1 = 4.0\ ,\ \mu_1 = 5.0\ ,\ p_1 = 0.5\ , \\ \eta_2 = 10.0\ ,\ \mu_2 = 11.0\ ,\ p_2 = 0.3\ , \\ \eta_3 = 13.0\ ,\ \mu_3 = 16.0,\ \ p_3 = 0.7\ . \end{cases}$$

The option's reference price of 0.04476 has been determined using a Monte Carlo simulation. For our PIDE approach, the parameter choice $\gamma = 30$ results in a domain

$$\mathcal{D} = (-3.302, 3.292) \times (-1.554, 1.555) \times (-0.873, 0.870)\ ,$$

which includes the non-differentiable kink. Figure 7 shows the relative pricing error with respect to the degrees of freedom for an anisotropic sparse grid and an anisotropic full grid discretization.

We observe that the convergence rate of the full grid approach is close to $\frac{2}{d}$ like in the former experiments. The sparse grid error convergence is somewhat erratic, but we clearly achieve a higher accuracy with less degrees of freedom.

Note here that the computation of a reference solution for the L_2-error measurement of the solution at maturity would be extremely demanding. We therefore omitted experiments for the L_2-norm and $d = 3$ here. Nevertheless, we expect analogous results as for the case $d = 2$, compare Figs. 5 and 6.

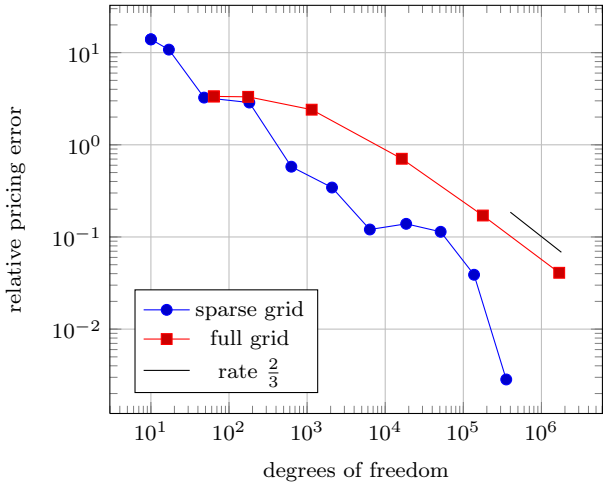

Fig. 7 Relative pricing error at S_1=S_2=S_3=1.0 with respect to the number of unknowns

5 Concluding remarks

We have presented a numerical method for the pricing of multi-dimensional basket options under Kou's jump-diffusion model. It involves a PIDE, i.e. a sum of tensor-product operators, and employs a general sparse grid discretization, which allows us to compute the solutions of moderate-dimensional problems. With the implementation of the unidirectional principle for non-local operators, an optimal preconditioner and a recurrence formula for the Kou model, we achieve linear runtime complexity with respect to the total number of degrees of freedom. The concept can easily be carried over to price more complex option types, e.g. early exercise options. It can also be generalized to a discretization by space-time sparse grids along the lines of [Oel06, GO07].

References

[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.

[Bel61] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[BG04] H. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.

[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*, 2011.

[BM02] G. Beylkin and M. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proc. Natl. Acad. Sci. USA*, 99:10246–10251, 2002.

[BS73] F. Black and M. Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81(3):637–654, 1973.

[Bun92] H. Bungartz. *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*. Dissertation, Fakultät für Informatik, Technische Universität München, November 1992.

[BZ96] R. Balder and C. Zenger. The solution of multidimensional real Helmholtz equations on sparse grids. *SIAM J. Sci. Comput.*, 17:631–646, May 1996.

[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.

[Feu10] C. Feuersänger. *Sparse Grid Methods for Higher Dimensional Approximation*. Dissertation, Institut für Numerische Simulation, Universität Bonn, September 2010.

[GG03] T. Gerstner and M. Griebel. Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87, 2003.

[GH12] M. Griebel and H. Harbrecht. On the construction of sparse tensor product spaces. *Mathematics of Computations*, 2012. to appear. Also available as INS Preprint No. 1104, University of Bonn.

[GK09] M. Griebel and S. Knapek. Optimized general sparse grid approximation spaces for operator equations. *Mathematics of Computations*, 78(268):2223–2257, 2009.

[GO94] M. Griebel and P. Oswald. On additive Schwarz preconditioners for sparse grid discretizations. *Numer. Math.*, 66:449–464, 1994.

[GO95] M. Griebel and P. Oswald. Tensor product type subspace splitting and multilevel iterative methods for anisotropic problems. *Adv. Comput. Math.*, 4:171–206, 1995.

[GO07] M. Griebel and D. Oeltz. A sparse grid space-time discretization scheme for parabolic problems. *Computing*, 81(1):1–34, 2007.

[Gri94] M. Griebel. *Multilevelmethoden als Iterationsverfahren über Erzeugendensysteme*. Teubner Skripten zur Numerik. Teubner, Stuttgart, 1994.

[Kaa88] E. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *Journal of Computational and Applied Mathematics*, 24(1-2):265 – 275, 1988.

[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.

[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*, 3:125–144, 1976.

[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.

[Oel06] D. Oeltz. *Ein Raum-Zeit Dünngitterverfahren zur Diskretisierung parabolischer Differentialgleichungen.* Dissertation, Institut für Numerische Simulation, Universität Bonn, 2006.

[Rei04] C. Reisinger. *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben.* Dissertation, Ruprecht-Karls-Universität Heidelberg, 2004.

[RSW10] N. Reich, C. Schwab, and C. Winter. On Kolmogorov equations for anisotropic multivariate Lévy processes. *Finance and Stochastics*, 14(4):527–567, 2010.

[Tho97] V. Thomée. *Galerkin finite element methods for parabolic problems.* Springer series in computational mathematics. Springer, 1997.

[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.

[Zei11] A. Zeiser. Fast matrix-vector multiplication in the sparse-grid Galerkin method. *J. Sci. Comput.*, 47(3):328–346, 2011.