Reduced models for sparse grid discretizations of the multi-asset Black-Scholes equation

Advances in Computational Mathematics manuscript No.
(will be inserted by the editor)

Reduced Models for Sparse Grid Discretizations of the Multi-Asset Black-Scholes Equation

Benjamin Peherstorfer · Pablo Gómez · Hans-Joachim Bungartz

Received: date / Accepted: date

Abstract This work presents reduced models for pricing basket options with the Black-Scholes and the Heston model. Basket options lead to multi-dimensional partial differential equations (PDEs) that quickly become computationally infeasible to discretize on full tensor grids. We therefore rely on sparse grid discretizations of the PDEs, which allow us to cope with the curse of dimensionality to some extent. We then derive reduced models with proper orthogonal decomposition. Our numerical results with the Black-Scholes model show that sufficiently accurate results are achieved while gaining speedups between 80 and 160 compared to the high-fidelity sparse grid model for 2-, 3-, and 4-asset options. For the Heston model, results are presented for a single-asset option that leads to a two-dimensional pricing problem, where we achieve significant speedups with our model reduction approach based on high-fidelity sparse grid models.

Keywords option pricing · proper orthogonal decomposition · sparse grids · Black-Scholes equation

Benjamin Peherstorfer
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139, USA
E-mail: pehersto@mit.edu

Pablo Gómez and Hans-Joachim Bungartz
Department of Informatics
Technische Universität München
Boltzmannstr. 3
85748 Garching, Germany

1 Introduction

We consider pricing European call options that give the right, but not the obligation, to buy an underlying asset (e.g., stock) at the expiration time $0 < T \in \mathbb{R}^+ = \{x \in \mathbb{R} \,|\, x \geq 0\}$ for a strike price $K \in \mathbb{R}^+$ [51]. The value of the option at time $t = T$ (future) is known: If the value $x \in \mathbb{R}^+$ of the underlying asset is below or equal K, then the option is not exercised and its value is zero. If the value x of the asset is higher than K, then the option is worth the difference between x and K. Thus, the price of the option at time $t = T$ is $\max\{x - K, 0\}$; however, to sell the option at time $t = 0$ (now), we need the price of the option at $t = 0$ and not only at $t = T$. Several models exist to price such call options at time $t = 0$. We consider here the Black-Scholes [2] and the Heston [25] model. The pricing models used in the following are simple models that are appropriate to showcase our numerical methods but that ignore many real-world details, see, e.g., [51,15] for more sophisticated models.

We consider multi-asset or basket options where the option gives the right to buy multiple assets. This leads to multi-dimensional partial differential equations (PDEs) with a spatial domain in $\mathbb{R}^d$ where $d > 1$. In general, a closed form solution does not exist anymore when pricing basket options, and thus one has to resort to numerical methods; however, this is computationally expensive due to the multi-dimensional setting. Additionally, the price of an option has to be updated repeatedly. It depends on several parameters describing the underlying assets, e.g., the drifts, volatilities, and correlations. These parameters are subject to change as new data becomes available.

Monte Carlo methods are often used to deal with multi-dimensional option pricing problems. Monte Carlo approaches transform the option pricing problem into a numerical quadrature problem and compute the scalar option price value for a given asset price. Whereas standard Monte Carlo is flexible and easy to use, its convergence rate is limited by $1/\sqrt{M}$, where $M \in \mathbb{N}$ is the number of samples. Advanced techniques, such as quasi, adaptive, or multi-level Monte Carlo, as well as Smolyak quadrature, are necessary to achieve a sufficiently fast convergence, see, e.g., [15,14,26,13,16]. Monte Carlo methods usually compute the scalar option price value for a given asset price only; they do not construct a function that can be evaluated at an arbitrary asset price. Therefore, it can become costly to, e.g., compute Greeks (derivatives) for hedging strategies [33,22], even though sophisticated methods exist [15]. Besides that, if the parameters change, one has to start all over again.

Model reduction approaches derive low-cost surrogates of the computationally expensive option pricing problems that provide approximate but computationally cheap solutions. The PDE corresponding to the option pricing model at hand is therefore discretized on a grid with, e.g., finite differences and finite element methods, and the full solution is obtained by, e.g., Galerkin projection, see [33,32]. In [46], a model reduction method for option pricing based on the proper orthogonal decomposition (POD) is presented. The obtained reduced model is employed for model calibration. In [48], POD is used to

speedup hedging models for Asian options with one and two underlyings. Another POD-based reduced model is derived in [10]. The authors of [9] discuss how to construct especially well-suited basis functions for option pricing and show that only a small number of them is required to achieve sufficiently accurate results. In [41] it is shown that [9] is in essence POD. This approach is then extended in [40] to price options with two underlying assets, but with the strong limitation that the volatilities of both assets have to be equal. The reduced basis method is used in [39] for model calibration, and in [35] to price options with the Heston model. In [21,8], American options, which require additional inequality constraints, are priced with reduced models.

All of the listed work consider model reduction for single-asset options or for limited cases of 2-asset options, even though often more sophisticated pricing models than the Black-Scholes and the Heston model are used. One reason for this limitation to single-asset options could be that a direct discretization of the multi-dimensional PDEs corresponding to basket options on Cartesian grids (so-called full grids) already becomes computationally infeasible for a small number of underlying assets (curse of dimensionality). Recently, discretizations based on sparse grids have been introduced, which are computationally feasible for moderately high-dimensional option pricing problems [5,53]. In [42,44,30,31], the multi-dimensional Black-Scholes equation is solved with the sparse grid combination technique [18], which splits the high-dimensional problem into many computationally cheap problems. A direct and adaptive sparse grid approach for up to 6-asset basket options is introduced in [6,7,22]. A sparse grid discretization for the multi-dimensional PDE corresponding to the Heston model is developed in [47,34]. Note that financial products with significantly more underlying assets (e.g., 128) can be priced with tensor-based approaches, see, e.g., [29].

We construct POD-Galerkin reduced models for basket options. The reduced models are developed in the context of the multi-asset Black-Scholes equation. We therefore discretize the multi-asset Black-Scholes equation with the direct sparse grid approach introduced in [6], construct a POD basis from pre-computed snapshot data, and derive the reduced model.

In the following Section 2, we briefly discuss the Black-Scholes and Heston model. We then focus on the multi-asset Black-Scholes model and derive the discrete operators stemming from the sparse grid discretization of the multi-asset Black-Scholes equation. In Section 4, we derive the reduced operators, discuss how to construct the POD basis vectors, and introduce the reduced models. Detailed numerical experiments are presented in Sections 5 and 6 for the Black-Scholes model, and in Section 7 for the Heston model.

2 Option Pricing with the Black-Scholes and the Heston Model

We state the Black-Scholes equation and give details on its parameters in Section 2.1, and apply the principal axis transformation to avoid the variable coefficients in Section 2.2. The Heston model is briefly discussed in Section 2.3.

2.1 Option Pricing with the Black-Scholes Model

Consider the value $\boldsymbol{x} = [x_1, \ldots, x_d]^T \in [0, \infty)^d$ of a basket of $d \in \mathbb{N}$ assets. We assume the value $\boldsymbol{x}$ of the assets follows a geometric Brownian motion with given drift and standard deviation [51]. Consider now the corresponding European call basket option with d underlying assets and the strike price $K \in \mathbb{R}^+$. We define $r \in \mathbb{R}^+$ as the risk-free interest rate, $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_d]^T \in \mathbb{R}^d$ as the drift, $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_d]^T \in \mathbb{R}^d$ as the volatility, and

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \ldots & \rho_{1d} \\ \rho_{12} & 1 & \ldots & \rho_{2d} \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{1d} & \rho_{2d} & \ldots & 1 \end{bmatrix} \in \mathbb{R}^{d \times d}$$

as the correlation matrix. For a compact notation, we collect all these parameters in the vector $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_p]^T \in \mathcal{D} \subset \mathbb{R}^p$, with $p = d + d + 0.5 \cdot (d^2 - d)$. We further select a maximum asset value $\boldsymbol{x}^{\max} = [x_1^{\max}, \ldots, x_d^{\max}] \in (0, \infty)^d$, following the procedure in [22,28], and define the spatial domain $\Omega = [0, x_1^{\max}] \times \ldots [0, x_d^{\max}] \subset [0, \infty)^d$. With the forward time $t \in [0, T]$, where T is the maturity, we consider the function $u : \Omega \times [0, T] \times \mathcal{D} \to \mathbb{R}^+$ in a suitable function space[1] as the solution of the multi-variate Black-Scholes equation

$$\frac{\partial u}{\partial t} + \frac{1}{2} \sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d} \eta_i x_i \frac{\partial u}{\partial x_i} - ru = 0\,, \quad \boldsymbol{x} \in \Omega. \tag{1}$$

Note that the choice of the maximum asset value $\boldsymbol{x}^{\max}$ is discussed in detail in [22,28]. We only note here that it can be chosen such that the effect on the solution is negligible except near the boundary. Since we will only consider European call options, the terminal condition of (1) is given by the payoff function

$$u(\boldsymbol{x}, T; \boldsymbol{\mu}) = \max \left\{ \frac{1}{d} \sum_{i=1}^{d} x_i - K, 0 \right\}.$$

We refer to, e.g., [15,51] for details. We impose Dirichlet boundary conditions

$$u(\boldsymbol{x}, t; \boldsymbol{\mu}) = u(\boldsymbol{x}, T; \boldsymbol{\mu}) \cdot \mathrm{e}^{-r(T-t)}\,, \quad \boldsymbol{x} \in \partial\Omega, t \in [0, T]\,, \tag{2}$$

where $\partial\Omega$ is the boundary of the spatial domain Ω.

2.2 Principal Axis Transformation of the Black-Scholes Equation

The original Black-Scholes equation (1) contains variable coefficients that depend on the asset price $\boldsymbol{x}$, see [15]. It is therefore common to not directly

[1] Suitable function spaces for a transformed Black-Scholes equation will follow in Section 3.2.

discretize and solve (1) but to first transform it such that these variable coefficients are eliminated. After discretization, this leads to a numerically better conditioned system of linear equations. Following [22,43], we apply the principal axis transformation, which we briefly summarize here:

1. We first switch time by considering $\tau = T - t$ instead of t. Therefore, our equation is solved forward in time, i.e., we have an initial condition instead of a terminal condition as in the original Black-Scholes formulation.
2. The variable coefficients are eliminated by the log transformation $y_i = \log x_i$

$$\frac{\partial u}{\partial \tau} - \frac{1}{2}\sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} \frac{\partial^2 u}{\partial y_i \partial y_j} - \sum_{i=1}^{d} \left(\eta_i - \frac{1}{2}\sigma_i^2\right) \frac{\partial u}{\partial y_i} + ru = 0\,.$$

3. An eigendecomposition $\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T = \boldsymbol{\Lambda}$ of the $d \times d$ covariance matrix $\boldsymbol{\Lambda}$ with $\Lambda_{ij} = \sigma_i \sigma_j \rho_{ij}$ for all $1 \leq i, j \leq d$ is computed. The eigenvectors are the columns of $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_d] \in \mathbb{R}^{d \times d}$ and the eigenvalues form the diagonal of the matrix $\boldsymbol{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$. We then transform $\boldsymbol{z} = \boldsymbol{Q}^T \boldsymbol{y}$ and obtain the equation

$$\frac{\partial u}{\partial \tau} - \frac{1}{2}\sum_{i=1}^{d} \lambda_i(\boldsymbol{\mu}) \frac{\partial^2 u}{\partial z_i^2} - \sum_{i=1}^{d} \hat{\eta}_i \frac{\partial u}{\partial z_i} + ru = 0, \qquad \hat{\eta}_i = \sum_{j=1}^{d} \left(\eta_j - \frac{1}{2}\sigma_j^2\right) Q_{ji}\,,$$

   where we denote the i-th eigenvalue by $\lambda_i(\boldsymbol{\mu})$ to indicate the dependence on the parameter $\boldsymbol{\mu}$.
4. The drift term is eliminated by the translation $\hat{z}_i = z_i + \tau\hat{\eta}_i$, and the reactive term is removed by replacing u by $\hat{u} = u/\mathrm{e}^{-r\tau}$.

After the transformation, we obtain the initial value problem

$$\frac{\partial \hat{u}}{\partial \tau} - \frac{1}{2}\sum_{s=1}^{d} \lambda_s(\boldsymbol{\mu}) \frac{\partial^2 \hat{u}}{\partial \hat{z}_s^2} = 0, \qquad \hat{\boldsymbol{z}} \in \Omega\,, \tag{3}$$

with the initial condition

$$\hat{u}(\hat{\boldsymbol{z}}, 0; \boldsymbol{\mu}) = \max\left\{ \frac{1}{d}\sum_{i=1}^{d} \exp\left(\sum_{j=1}^{d} Q_{ij}\hat{z}_j\right) - K, 0 \right\},$$

and boundary condition

$$\hat{u}(\hat{\boldsymbol{z}}, \tau; \boldsymbol{\mu}) = \hat{u}(\hat{\boldsymbol{z}}, 0; \boldsymbol{\mu}), \qquad \hat{\boldsymbol{z}} \in \partial\Omega, \tau \in [0, T]\,. \tag{4}$$

The solution function u of the original Black-Scholes equation can be restored with

$$u(\boldsymbol{x}, t; \boldsymbol{\mu}) = \hat{u}(\hat{\boldsymbol{z}}, \tau; \boldsymbol{\mu}) \cdot \mathrm{e}^{-r\tau}\,. \tag{5}$$

We once again refer to [22] for details.

2.3 Option Pricing with the Heston Model

The Heston model [25] extends the Black-Scholes model by allowing the volatilities of the underlying assets to vary. This doubles the number of dimensions of the corresponding option pricing problem compared to the Black-Scholes model. An option with d underlying assets therefore leads to a $2d$-dimensional pricing problem in case of the Heston model [47]. We restrict the following discussion of the Heston model to a single asset option and therefore to a two-dimensional pricing problem.

The governing equation of the Heston model [25] is of the form

$$\frac{\partial u}{\partial t} = -\frac{1}{2} v x^2 \frac{\partial^2 u}{\partial x^2} - \rho \xi v x \frac{\partial^2 u}{\partial x \partial v} - \frac{1}{2} \xi^2 v \frac{\partial^2 u}{\partial v^2} - \kappa(\theta - v)\frac{\partial u}{\partial v} - r x \frac{\partial u}{\partial x} + r u \,. \quad (6)$$

where now u depends on the asset price x and on the variance v, with correlation ρ. We further have the long-run variance $\theta \in \mathbb{R}^+$, the mean reversion rate $\kappa \in \mathbb{R}^+$, and the volatility of the volatility $\xi \in \mathbb{R}^+$. These parameters are collected into the parameter $\boldsymbol{\mu} = [\theta, \kappa, \xi]$. Similarly to the Black-Scholes equation, we have the terminal condition

$$u(x, v, T; \boldsymbol{\mu}) = \max\{x - K, 0\}$$

for European call options. We refer to [25] for a discussion on boundary and limit conditions. A log-transformation of the asset price in (6) eliminates coefficients with a dependence on x, see, e.g., [25, 47].

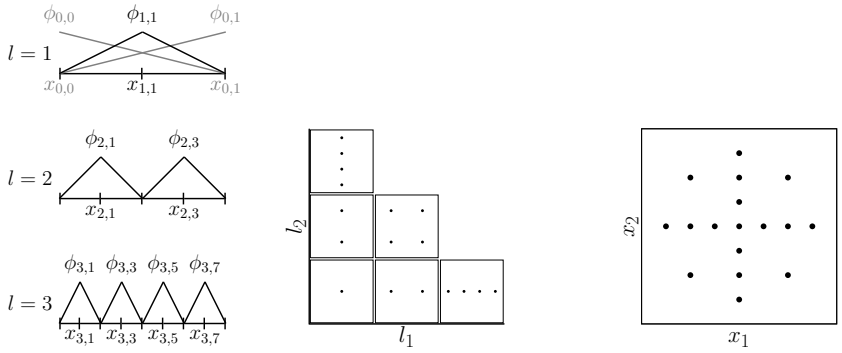
3 Sparse Grid Discretization

In this section, we discretize the transformed Black-Scholes equation (3) in the spatial domain Ω with finite elements and in the temporal domain with the implicit Euler method. In the spatial domain, we rely on sparse grids that allow us to cope with the curse of dimensionality to some extent [5, 37]. We first give a brief overview of sparse grids and then derive the discrete operators corresponding to the Black-Scholes equation. Note that a similar derivation can be applied to the Heston model, see [34] for a detailed discussion.

3.1 Sparse Grids and Sparse Grid Spaces

Sparse grids have become a versatile tool to deal with multi-dimensional problems and have a wide range of applications [5, 12, 37]. The original idea was introduced for numerical quadrature, but sparse grids were first used for the discretization of PDEs in [53]. We only give an overview of the basic sparse grid framework, which is required in later sections, and refer to the survey [5] for details.

Let $\mathcal{V}$ be a function space with domain $\Omega \subset \mathbb{R}^d$, e.g., we can think of $\mathcal{V} = H_0^1(\Omega)$ as a Sobolev space. We approximate a function $f \in \mathcal{V}$ by its



(a) hier. basis (b) hierarchical increments (c) sparse grid

Fig. 1: The one-dimensional hierarchical basis up to level three is shown in (a). The grid points of the hierarchical increments with level up to three and of the two-dimensional sparse grid of level three are plotted in (b) and (c), respectively.

interpolant $f^{(\infty)} \in \mathcal{V}_\ell^{(\infty)} \subset \mathcal{V}$ in the finite-dimensional space $\mathcal{V}_\ell^{(\infty)}$ of piecewise d-linear functions with mesh width $h_\ell = 2^{-\ell}, \ell \in \mathbb{N}$ in each direction. The superscript $^{(\infty)}$ will become clear with the hierarchical decomposition in (8).

We construct a multi-level basis, the so-called hierarchical basis, for the space $\mathcal{V}_\ell^{(\infty)}$. Let $\phi(x) = \max\{1-|x|, 0\}$ be the standard hat function, and define by dilation and translation the hierarchical basis function $\phi_{l,i}(x) = \phi(2^l x - i)$ with the level $l \in \mathbb{N}$ and index $i \in \mathbb{N}$, see Figure 1. We derive the multi-dimensional hierarchical basis functions by the tensor product

$$\phi_{\boldsymbol{l},\boldsymbol{i}} = \prod_{j=1}^{d} \phi_{l_j,i_j} \,,$$

where now the level $\boldsymbol{l} \in \mathbb{N}^d$ and index $\boldsymbol{i} \in \mathbb{N}^d$ are vectors with d components. With these hierarchical basis functions we can span the so-called hierarchical increments $\mathcal{W}_{\boldsymbol{l}}$ as

$$\mathcal{W}_{\boldsymbol{l}} = \operatorname{span}\left\{\phi_{\boldsymbol{l},\boldsymbol{i}} \,|\, 1 \le i_j < 2^{l_j}, i_j \notin 2\mathbb{N}, 1 \le j \le d\right\} . \tag{7}$$

The hierarchical increment $\mathcal{W}_{\boldsymbol{l}}$ is spanned by those hierarchical basis functions of level $\boldsymbol{l}$ for which all components of the index $\boldsymbol{i}$ are odd. With (7) we define the space of piecewise d-linear functions corresponding to mesh width $h_\ell = 2^{-\ell}$ as direct sum

$$\mathcal{V}_\ell^{(\infty)} = \bigoplus_{|\boldsymbol{l}|_\infty \le \ell} \mathcal{W}_{\boldsymbol{l}} \,, \tag{8}$$

where $|l|_\infty = \max_j |l_j|$. The limit $|\boldsymbol{l}|_\infty \le \ell$ also explains the superscript $^{(\infty)}$ in the notation.

The hierarchical basis leads to better conditioned systems when applied to the discretization of PDEs with the finite element method [52], but, more

importantly, the hierarchical decomposition (8) allows us to define many different approximation spaces. In particular, we can ask for the approximation space with maximum level ℓ in each dimension that has the best cost-benefit ratio. The benefit is equal to the interpolation error in the L^2 norm, and the (computational) costs are measured with the number of basis functions. If we set $\mathcal{V}$ to the space of functions with bounded mixed derivatives up to order two [5], the result of this optimization problem is the sparse grid space of dimension d and level ℓ

$$\mathcal{V}_\ell^{(1)} = \bigoplus_{|\boldsymbol{l}|_1 \leq \ell+d-1} \mathcal{W}_{\boldsymbol{l}}\,, \tag{9}$$

where $|\boldsymbol{l}|_1 = \sum_j l_j$, see [5] for details and proofs. The grid points of the sparse grid of dimension two and level three as well as of the corresponding hierarchical increments are shown in Figure 1. We also note that different benefit and error measures lead to different types of sparse grids [5,4].

The interpolation error in the L^2 norm of the interpolant $f^{(1)} \in \mathcal{V}_\ell^{(1)}$ with superscript $^{(1)}$ of $f \in \mathcal{V}$ is in $\mathcal{O}(2^{-2\ell}\ell^{d-1})$, and thus slightly worse than of the interpolant $f^{(\infty)} \in \mathcal{V}_\ell^{(\infty)}$ of the full grid space $\mathcal{V}_\ell^{(\infty)}$ with error in $\mathcal{O}(2^{-2\ell})$; however, the number of grid points, and thus the computational costs, is only in $\mathcal{O}(2^\ell \ell^{d-1})$ for the sparse grid space, compared to $\mathcal{O}(2^{\ell d})$ for $\mathcal{V}_\ell^{(\infty)}$. Thus, it is distinctly cheaper to approximate a multi-dimensional function f in a sparse than in a full grid space, provided f has bounded mixed derivatives up to order two. Note that we have considered homogeneous boundary values so far. We refer to [5,11,37] and references therein for details on sparse grids in general, and on sparse grids with grid points at the boundary in particular.

3.2 Spatial Discretization with Sparse Grids

We discretize the Black-Scholes equation with the finite element method on sparse grids. We first derive the weak formulation of (3) and then project with Ritz-Galerkin onto a finite-dimensional sparse grid space.

Let $\partial\Omega$ be the boundary of the spatial domain Ω and let the Sobolev space $H_0^1(\Omega)$ be our ansatz and test space such that all basis and test functions in $H_0^1(\Omega)$ vanish at the boundary $\partial\Omega$, see Section 3.1 and [5]. We then obtain with integration by parts the variational formulation

$$\langle \dot{\hat{u}}, \phi \rangle_{L^2} - \frac{1}{2} \sum_{s=1}^{d} \lambda_s(\boldsymbol{\mu}) \left\langle \frac{\partial \hat{u}}{\partial \hat{z}_s}, \frac{\partial \phi}{\partial \hat{z}_s} \right\rangle_{L^2} = 0\,, \qquad \phi \in H_0^1(\Omega)\,, \tag{10}$$

where $\langle \cdot, \cdot \rangle_{L^2}$ denotes the L^2 dot product. The boundary integrals vanish because of the homogeneous Dirichlet boundary conditions. Existence and uniqueness of the solution is guaranteed in these spaces, see [5] and references therein. The weak form (10) is derived in detail in [22]. Note that for the Heston model the weak form is derived in [34]. We use Ritz-Galerkin for the projection onto the finite-dimensional sparse grid space $\mathcal{V}_\ell^{(1)}$ with boundary

points, spanned by the basis functions in $\{\phi_1, \dots, \phi_{N_B}\}$. Therefore, we introduce the ordering $\phi_1, \dots, \phi_N, \phi_{N+1}, \dots, \phi_{N_B}$ of the hierarchical basis functions of $\mathcal{V}_\ell^{(1)}$ where the first N functions correspond to the inner grid points and the rest to the boundary points. For each inner grid point we have one degree of freedom, the hierarchical coefficient, and thus N is also the number of degrees of freedom. The sparse grid function $\hat{u}$ can be represented as the linear combination

$$\hat{u}(\hat{\boldsymbol{z}}, \tau; \boldsymbol{\mu}) = \sum_{j=1}^{N} \hat{u}_j(\tau; \boldsymbol{\mu}) \phi_j(\hat{\boldsymbol{z}}) + \sum_{j=N+1}^{N_B} \hat{u}_j(\tau; \boldsymbol{\mu}) \phi_j(\hat{\boldsymbol{z}}) \tag{11}$$

for a fixed $\tau \in [0, T]$. Note that the coefficients $\hat{u}_1(\tau; \boldsymbol{\mu},), \dots, \hat{u}_{N_B}(\tau; \boldsymbol{\mu}) \in \mathbb{R}$ still depend on the time τ. We emphasize that our notation does not distinguish between the solution of (3) and its discretized counterpart (11) because the context will always make it clear which of them we mean, and we want to keep the notational burden to a minimum. The coefficients of the linear combination (11) corresponding to the inner and boundary points are denoted by

$$\hat{\boldsymbol{u}}^I(\tau; \boldsymbol{\mu}) = [\hat{u}_1(\tau; \boldsymbol{\mu}), \dots, \hat{u}_N(\tau; \boldsymbol{\mu})]^T, \hat{\boldsymbol{u}}^B(\tau; \boldsymbol{\mu}) = [\hat{u}_{N+1}(\tau; \boldsymbol{\mu}), \dots, \hat{u}_{N_B}(\tau; \boldsymbol{\mu})]^T,$$

where the Dirichlet boundary conditions (4) fixate $\hat{\boldsymbol{u}}^B(\tau; \boldsymbol{\mu})$ at all time steps and thus we skip the time dependence in the notation and write $\hat{\boldsymbol{u}}^B(\boldsymbol{\mu})$ instead.

With the representation (11), the system of ordinary differential equations (ODE) follows as

$$\underbrace{\begin{bmatrix} \boldsymbol{B}^I & \boldsymbol{B}^B \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}} \begin{bmatrix} \dot{\hat{\boldsymbol{u}}}^I(\tau; \boldsymbol{\mu}) \\ \dot{\hat{\boldsymbol{u}}}^B(\boldsymbol{\mu}) \end{bmatrix} - \underbrace{\frac{1}{2} \sum_{s=1}^{d} \lambda_s(\boldsymbol{\mu}) \begin{bmatrix} \boldsymbol{L}_s^I & \boldsymbol{L}_s^B \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{L}(\boldsymbol{\mu})} \begin{bmatrix} \hat{\boldsymbol{u}}^I(\tau; \boldsymbol{\mu}) \\ \hat{\boldsymbol{u}}^B(\boldsymbol{\mu}) \end{bmatrix} = 0 \tag{12}$$

where $\boldsymbol{0}$ and $\boldsymbol{I}$ are the zero and identity matrices, respectively. Note that $\dot{\hat{\boldsymbol{u}}}^B(\boldsymbol{\mu}) = \boldsymbol{0}$ because the boundary conditions do not change over time. The inner operators $\boldsymbol{B}^I \in \mathbb{R}^{N \times N}$ and $\boldsymbol{L}^I \in \mathbb{R}^{N \times N}$ take only the inner coefficients into account and are defined as

$$\boldsymbol{B}^I = \left(\langle \phi_i, \phi_j \rangle_{L^2}\right)_{1 \le i,j \le N}, \quad \boldsymbol{L}_s^I = \left(\left\langle \frac{\partial \phi_i}{\partial \hat{z}_s}, \frac{\partial \phi_j}{\partial \hat{z}_s} \right\rangle_{L^2}\right)_{1 \le i,j \le N}. \tag{13}$$

Similarly, we obtain

$$\boldsymbol{B}^B = \left(\langle \phi_i, \phi_j \rangle_{L^2}\right)_{\substack{1 \le i \le N, \\ N+1 \le j \le N_B}}, \quad \boldsymbol{L}_s^B = \left(\left\langle \frac{\partial \phi_i}{\partial \hat{z}_s}, \frac{\partial \phi_j}{\partial \hat{z}_s} \right\rangle_{L^2}\right)_{\substack{1 \le i \le N, \\ N+1 \le j \le N_B}} \tag{14}$$

for the operators that also consider boundary points. System (12) depends on the parameter $\boldsymbol{\mu}$ indirectly through the eigenvalues $\lambda_1(\boldsymbol{\mu}), \dots, \lambda_d(\boldsymbol{\mu})$, but the operators $\boldsymbol{B}^I, \boldsymbol{B}^B$, $\boldsymbol{L}_s^I$, and $\boldsymbol{L}_s^B, 1 \le s \le d$ are independent of $\boldsymbol{\mu}$.

The efficient assembly of the discrete operators (13) and (14) based on the hierarchical basis and sparse grids requires specialized algorithms. These algorithms are necessary because the hierarchical basis functions have non-local support and thus lead to full matrices. We do not discuss the details of these algorithms here but refer to [22,5].

3.3 Temporal Discretization with Implicit Euler

The system of ODEs (12) is discretized in the time domain with the implicit Euler method [22]. At each time step we have the (hierarchical) coefficient vector

$$\hat{\boldsymbol{u}}_\tau(\boldsymbol{\mu}) = [u_1^\tau(\boldsymbol{\mu}), \ldots, u_{N_B}^\tau(\boldsymbol{\mu})]^T \in \mathbb{R}^{N_B}$$

split into inner and boundary points

$$\hat{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu}) = [u_1^\tau(\boldsymbol{\mu}), \ldots, u_N^\tau(\boldsymbol{\mu})]^T, \hat{\boldsymbol{u}}^B(\boldsymbol{\mu}) = \hat{\boldsymbol{u}}_\tau^B(\boldsymbol{\mu}) = [u_{N+1}^\tau(\boldsymbol{\mu}), \ldots, u_{N_B}^\tau(\boldsymbol{\mu})]^T .$$

Let $\delta\tau$ be the time step size, then we solve the system of linear equations

$$(\boldsymbol{B} - \delta\tau \boldsymbol{L}(\boldsymbol{\mu}))\hat{\boldsymbol{u}}_{\tau+\delta\tau}(\boldsymbol{\mu}) = \boldsymbol{B}\hat{\boldsymbol{u}}_\tau(\boldsymbol{\mu}) \tag{15}$$

in every time step. System (15) involves matrices with size $N_B \times N_B$ but has only N degrees of freedom. By representing it as

$$\left(\begin{bmatrix} \boldsymbol{B}^I & \boldsymbol{B}^B \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} - \delta\tau \begin{bmatrix} \boldsymbol{L}^I(\boldsymbol{\mu}) & \boldsymbol{L}^B(\boldsymbol{\mu}) \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\right) \begin{bmatrix} \hat{\boldsymbol{u}}_{\tau+\delta\tau}^I(\boldsymbol{\mu}) \\ \hat{\boldsymbol{u}}^B(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^I & \boldsymbol{B}^B \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu}) \\ \hat{\boldsymbol{u}}^B(\boldsymbol{\mu}) \end{bmatrix} \tag{16}$$

we can bring the boundary operators to the right-hand side and obtain the final $N \times N$ system

$$\underbrace{\left(\boldsymbol{B}^I - \delta\tau \boldsymbol{L}^I(\boldsymbol{\mu})\right)}_{\boldsymbol{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}} \hat{\boldsymbol{u}}_{\tau+\delta\tau}^I(\boldsymbol{\mu}) = \underbrace{\boldsymbol{B}^I \hat{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu}) + \delta\tau \boldsymbol{L}^B(\boldsymbol{\mu})\hat{\boldsymbol{u}}^B(\boldsymbol{\mu})}_{\boldsymbol{b}_\tau(\boldsymbol{\mu}) \in \mathbb{R}^N} . \tag{17}$$

System (17) requires a matrix-vector product with the $\boldsymbol{L}^B(\boldsymbol{\mu})$ operator of size $N \times N_B$ to impose the Dirichlet boundary conditions in every time step.

4 Reduced Models for the Black-Scholes Equation

In this section, we introduce reduced models that allow us to price basket options. A reduced basis is constructed with POD, which is then used to derive the reduced system. We emphasize that we do not reduce the dimension d (i.e., the number of underlying assets) but the number of degrees of freedom N, see [44, 23].

4.1 POD-Galerkin Reduced Models

POD [49] is a procedure to compute orthonormal basis vectors to represent a given set of data points (e.g., solutions of a PDE) such that the first basis vector (dimension) has the highest variance, and all the following vectors as much variance as possible under the constraint that they have to be orthonormal. Let

$$\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_1), \ldots, \hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_M)] \in \mathbb{R}^{N \times M} \tag{18}$$

be our data or snapshot matrix with zero mean and with $M \in \mathbb{N}$ solutions (snapshots) $\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_1), \dots, \hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_M)$ of (17). The parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M \in \mathcal{D}$ are chosen such that the snapshots capture the important system behavior of (17). The selection of the snapshots significantly influences the quality of the reduced model. Many selection algorithms are based on greedy sampling approaches, see, e.g., [50,45]. There are also sampling procedures that explicitly target problems with a large number of parameters, i.e., where the dimension p of the parameter vector $\boldsymbol{\mu}$ is large [3,24,38]. Goal-oriented sampling procedures take into account the output of interest, e.g., the option price in the setting of option pricing, see [50,3]. The POD-Greedy [20] method is widely used for time-dependent problems. We use here a simple random sampling from a uniform distribution in the parameter domain $\mathcal{D}$. Our numerical results will confirm that the random sampling is sufficient here; however, we note that our reduced modeling approach based on high-fidelity sparse grid models is applicable to the more sophisticated sampling strategies as well.

The POD basis vectors or POD modes for the snapshots in $\hat{\boldsymbol{U}}$ correspond to the left singular vectors of the singular value decomposition (SVD) of $\hat{\boldsymbol{U}}$. Thus, let $\hat{\boldsymbol{U}} = \boldsymbol{W}\boldsymbol{S}\boldsymbol{V}^T$ be the SVD of $\hat{\boldsymbol{U}}$ with the singular values $\zeta_1 > \zeta_2 > \dots > 0$ and the left singular vectors as columns of $\boldsymbol{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_N] \in \mathbb{R}^{N \times N}$. We define $\boldsymbol{W}_k = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_k] \in \mathbb{R}^{N \times k}$ to contain the first k POD basis vectors that are the solution to the minimization problem

$$\underset{\{\boldsymbol{w}_i\}_{i=1}^k}{\arg\min} \sum_{i=1}^{M} \|\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_i) - \sum_{j=1}^{k} (\boldsymbol{w}_j^T \hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}_i)) \boldsymbol{w}_j\|_2^2 \tag{19}$$

under the constraint $\boldsymbol{w}_i^T \boldsymbol{w}_j = \delta_{ij}$ for all $1 \le i, j \le k$ with the Kronecker delta δ_{ij}. The POD basis vectors depend on the choice of the inner product in (19). A careful selection of the inner product can improve the approximation results [20,27]. Our numerical results will confirm that this computationally simple approach is already sufficient for us, see Sections 5 to 7.

Given the POD basis vectors in $\boldsymbol{W}_k$, we construct the reduced operators for the system (17) as

$$\tilde{\boldsymbol{B}}^I = \boldsymbol{W}_k^T \boldsymbol{B}^I \boldsymbol{W}_k, \quad \tilde{\boldsymbol{L}}_s^I = \boldsymbol{W}_k^T \boldsymbol{L}_s^I \boldsymbol{W}_k, \quad \tilde{\boldsymbol{L}}_s^B = \boldsymbol{W}_k^T \boldsymbol{L}_s^B, \tag{20}$$

for all $1 \le s \le d$. With $\tilde{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu}) = \boldsymbol{W}_k^T \hat{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu})$ we obtain the reduced system

$$\underbrace{\left(\tilde{\boldsymbol{B}}^I - \delta\tau \sum_{s=1}^{d} \lambda_s(\boldsymbol{\mu}) \tilde{\boldsymbol{L}}_s^I\right)}_{\tilde{\boldsymbol{A}}(\boldsymbol{\mu}) \in \mathbb{R}^{k \times k}} \tilde{\boldsymbol{u}}_{\tau+\delta\tau}^I(\boldsymbol{\mu}) = \underbrace{\tilde{\boldsymbol{B}}^I \tilde{\boldsymbol{u}}_\tau^I(\boldsymbol{\mu}) + \sum_{s=1}^{d} \lambda_s(\boldsymbol{\mu}) \tilde{\boldsymbol{L}}_s^B \boldsymbol{u}^B(\boldsymbol{\mu})}_{\tilde{\boldsymbol{b}}_\tau(\boldsymbol{\mu}) \in \mathbb{R}^k} \tag{21}$$

with $k \times k$ degrees of freedoms instead of $N \times N$ as the original system (17).

Let us make a few remarks on the computational procedure. First, the SG^{++} solver used in the following, does not give direct access to the boundary

operators $\boldsymbol{L}_s^B$ for $1 \leq s \leq d$ due to restrictions of the UpDown algorithm [22]. Therefore, we introduce the matrices $\boldsymbol{R}^I \in \mathbb{R}^{N \times N_B}$ and $\boldsymbol{R}^B \in \mathbb{R}^{N_B \times N}$ with

$$\boldsymbol{R}^I \begin{bmatrix} \boldsymbol{u}^I \\ \boldsymbol{u}^B \end{bmatrix} = \boldsymbol{u}^I \qquad \boldsymbol{R}^B \boldsymbol{u}^B = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}^B \end{bmatrix}$$

and represent $\boldsymbol{L}_s^B = \boldsymbol{R}^I \boldsymbol{L}_s^B \boldsymbol{R}^B$ for $1 \leq s \leq d$. Second, the snapshots to construct the matrix $\hat{\boldsymbol{U}}$ are pre-computed in a computationally expensive offline phase with costs depending on N_B. The same holds for the reduced operators (20) that are independent of the parameter $\boldsymbol{\mu} \in \mathcal{D}$ and thus can also be pre-computed in the offline phase. Finally, in the online phase, the parameter-dependent eigenvalues $\lambda_1(\boldsymbol{\mu}), \ldots, \lambda_d(\boldsymbol{\mu})$ are computed, which requires solving a $d \times d$ eigenvalue problem once, then the left-hand side operator $\tilde{\boldsymbol{A}}$ and the right-hand side boundary operator are assembled. Note that the Dirichlet boundary conditions change with the parameter $\boldsymbol{\mu}$. Note further that matrix-vector products with costs in $\mathcal{O}(kN_B)$ are required to assemble the right-hand side. The costs of the computational procedure of the implicit Euler steps only depend on k and d, but not on N or N_B. The evaluation of the solution function at an arbitrary asset value $\boldsymbol{x} \in \Omega$ requires the construction of the sparse grid function with N_B grid points from the reduced solution $\tilde{\boldsymbol{u}}_T^I(\boldsymbol{\mu})$.

4.2 Proper Orthogonal Decomposition of Sparse Grid Data

Our snapshot vectors, which are the columns of the matrix $\hat{\boldsymbol{U}}$, contain the hierarchical coefficients corresponding to the sparse grid function (11) representing the solution at the final time step and parameter $\boldsymbol{\mu}$. Since the hierarchical basis is a multi-level basis, it leads to multi-scale coefficients with very different absolute values for coefficients corresponding to basis functions of different hierarchical increments, see Section 3.1 and [5, Lemma 3.3]. Such multi-scale data is often computationally problematic in the context of POD and requires special treatment such as scalar-valued POD [54, Section 2.2] or problem-dependent inner products [27]; however, in case of the hierarchical basis, the coefficients of the sparse grid function decay in a controlled manner and thus this gives us insight into the POD reconstruction error of the snapshots. We emphasize that the following results are only valid for the snapshots included in $\hat{\boldsymbol{U}}$ and are not applicable to general solutions $\hat{\boldsymbol{u}}_\tau(\boldsymbol{\mu}) \in \mathbb{R}^{N_B}$.

The error of representing the snapshots in the POD basis is given by

$$\sum_{i=1}^{M} \|\hat{\boldsymbol{u}}_T(\boldsymbol{\mu}_i) - \sum_{j=1}^{k} (\boldsymbol{w}_j^T \hat{\boldsymbol{u}}_T(\boldsymbol{\mu}_i)) \boldsymbol{w}_j\|_2^2 = \sum_{j=k+1}^{N} \zeta_j^2 \tag{22}$$

where $\zeta_1 > \zeta_2 > \cdots > 0$ are the singular values. The following corollary provides a bound for the sum of the squares of the singular values for sparse grid snapshot matrices.

Corollary 1 *Let $\hat{\boldsymbol{U}} \in \mathbb{R}^{N \times M}$ be the snapshot matrix as defined in* (18) *where the option pricing problem is discretized on a d-dimensional sparse grid with level ℓ and with homogeneous boundary values. The sum of the squares of the singular values $\zeta_1 > \zeta_2 > \cdots > 0$ of $\hat{\boldsymbol{U}}$ is bounded by*

$$\sum_{i=1}^{N} \zeta_i^2 \leq \sum_{l=0}^{\ell-1} 2^{-3l} \binom{d-1+l}{d-1} 4^{-3d} M \|\partial^{\mathbf{2}} \hat{u}\|_\infty^2$$

where $\|\partial^{\mathbf{2}} \hat{u}\|_\infty$ is the L^∞ norm of the derivatives up to order two in each dimension of the solution $\hat{u}$ of the continuous problem (3) *and N the number of degrees of freedom of the discretized problem.*

Proof The singular values of $\hat{\boldsymbol{U}}$ are related to the matrix entries through

$$\sum_{j=1}^{N} \zeta_j^2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \hat{U}_{ij}^2 \,. \tag{23}$$

It is shown in [5], if the solution function $\hat{u}$ of the continuous problem (3) has bounded mixed derivatives up to order two, the hierarchical coefficients are bounded by

$$|\hat{U}_{ij}|^2 = |\hat{u}_i^T(\boldsymbol{\mu}_j)|^2 \leq 4^{-d} 2^{-4|\boldsymbol{l}|_1} \|\partial^{\mathbf{2}} \hat{u}\|_\infty^2 \,, \tag{24}$$

where $\hat{u}_i^T(\boldsymbol{\mu}_j)$ is the hierarchical coefficient corresponding to the i-th basis function in the j-th snapshot, d is the dimension (i.e., number of assets), $\boldsymbol{l}$ the level of the basis function corresponding to node i, $\hat{u}$ the solution function of the continuous problem (3), and $\partial^{\mathbf{2}}$ the multi-variate derivative operator up to order two in each dimension, cf. also Section 3.1. The bound (24) shows that the coefficients decay rapidly with the level $\boldsymbol{l}$ of the corresponding basis function. Furthermore, a sparse grid space of level ℓ and dimension d is spanned by

$$2^i \cdot \binom{d-1+i}{d-1} \tag{25}$$

basis functions with $|\boldsymbol{l}|_1 = i + d$, $0 \leq i \leq \ell - 1$, see, again, [5, Lemma 3.6]. We therefore obtain for all $j = 1, \ldots, M$ that

$$\sum_{i=1}^{N} \hat{U}_{ij}^2 \leq \sum_{l=0}^{\ell-1} 2^l \binom{d-1+l}{d-1} 4^{-d} 2^{-4(l+d)} \|\partial^{\mathbf{2}} \hat{u}\|_\infty^2$$

holds. With $2^l 2^{-4(l+d)} 4^{-d} = 4^{-3d} 2^{-3l}$ and by combining (23) with (24) and (25), we obtain the (possibly very pessimistic) bound stated in Corollary 1. □

5 Numerical Results: Single-Asset Options with Black-Scholes

Even though our reduced models allow us to price basket options with multiple underlying assets, we first report results for single asset options, as those are also priced in most of the related, previous work on model reduction for option pricing [46,10,9,41,39,21].

In this and in the following section, we use the SG^{++}[2] sparse grid solver to compute solutions of the high-fidelity sparse grid model. It implements the sparse grid discretization approach summarized in Section 3. The SG^{++} solver does not assemble the matrices of system (17) but only provides routines for the matrix-vector product with the matrices. These methods are then used in conjunction with the conjugate gradient method to solve the system. Even though this matrix-free approach is well-suited if the Black-Scholes equation is discretized on a large number of grid points, leading to matrices that cannot be assembled anymore, it is usually slower in case of small matrices. Since a direct solver is the best choice for the small systems arising in our reduced models, and to ensure a fair runtime comparison between the reduced and the high-fidelity model, we exchanged the matrix-free solver routine of SG^{++} with a direct solver. Thus, assembled matrices are either formed or loaded and then the system (17) is solved by employing a QR decomposition. We use the implementation provided in the Eigen library [19]. Our SG^{++} solver with the direct solver is about 10 to 30 times faster than the matrix-free solver for the problems considered in the following; however, we emphasize that the scope of our direct solver is severely limited by the huge memory requirements. All of the following runtime measurements were performed on the cluster of the Munich Centre of Advanced Computing[3] on the nodes with a quad socket Intel Westmere-EX Xeon E7-4830 and 512 GB RAM, on a single core.

We consider the following setup for pricing a one-dimensional European call option. The option has only one underlying asset in the spatial domain $\Omega = [0.5, 4] \subset \mathbb{R}$ [28] and thus the option pricing problem depends only on two parameters $\boldsymbol{\mu} = [\eta, \sigma]^T \in \mathcal{D} = [0.03, 0.07] \times [0.2, 0.4] \subset \mathbb{R}^2$ that correspond to the drift η and volatility σ of the asset. We set the risk-free interest rate to $r = 0.05$, the time step size to $\delta\tau = 0.01$, the number of time steps to 100, and the strike price to $K = 1$. We note that the risk-free rate can be changed after the transformed equation (3) has been solved because it only changes the back transformation (5) into Cartesian coordinates. We compute $M = 600$ snapshots with the SG^{++} solver where we discretize on a sparse grid of level nine, i.e., the mesh width is $h = 2^{-9}$. Thus, our discretized system has $N = 511$ degrees of freedom. Note that a one-dimensional sparse grid coincides with an ordinary Cartesian full grid. The parameters $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M$ of the snapshots were selected randomly from a uniform distribution in the domain $\mathcal{D}$. Reduced models constructed from a different number M of snapshots are presented in the following section.

[2] http://www5.in.tum.de/SGpp

[3] http://www.mac.tum.de/

We report runtime and accuracy results of our reduced models for the option with $\boldsymbol{\mu} = [\eta, \sigma]^T = [0.05, 0.3]^T$. Table 1 shows the runtime and speedup of our reduced model as well as the absolute $\|\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}) - \boldsymbol{W}_k \tilde{\boldsymbol{u}}_T^I(\boldsymbol{\mu})\|_2$ and relative

$$\frac{\|\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu}) - \boldsymbol{W}_k \tilde{\boldsymbol{u}}_T^I(\boldsymbol{\mu})\|_2}{\|\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu})\|_2}$$

L^2 error with respect to the high-fidelity model, where $\hat{\boldsymbol{u}}_T^I(\boldsymbol{\mu})$ and $\tilde{\boldsymbol{u}}_T^I(\boldsymbol{\mu})$ are the high-fidelity and reduced solutions, respectively.

The first row of Table 1 corresponds to the high-fidelity model (i.e., the SG^{++} solver) and shows that it requires about 0.14 seconds to construct and assemble the operators and another 0.5 seconds to solve the system of linear equations. In the operator setup phase of the high-fidelity model, the operators are formed and then the system matrix is assembled. In case of the reduced model, the operators are pre-computed in the offline phase and then loaded from disk. Furthermore, the operator setup phase includes the transformation of the initial condition into the POD representation. After the operators are assembled or loaded, respectively, they can be reused for different parameter configurations. The reported solve runtime includes the QR decomposition of the already assembled system matrix, the implicit Euler time steps, and, in case of the reduced model, the transformation from the POD basis back into the hierarchical basis.

The following rows correspond to our reduced model for different numbers of POD modes k. Already with only $k = 10$ POD basis vectors we achieve a relative L^2 error below 10^{-3} with respect to the high-fidelity model, and we gain a speedup of about 400 when solving the system. We do not achieve any speedup in the operator phase, because in case of the 1-asset problem, assembling the full-order operators is faster than loading the reduced operators from disk. The price of the option is evaluated at $x = 1$, which is near the bend of the solution function and thus where the sparse grid discretization error is usually large. It is correct up to three digits if $k = 20$ POD modes are used.

6 Numerical Results: Basket Options with Black-Scholes

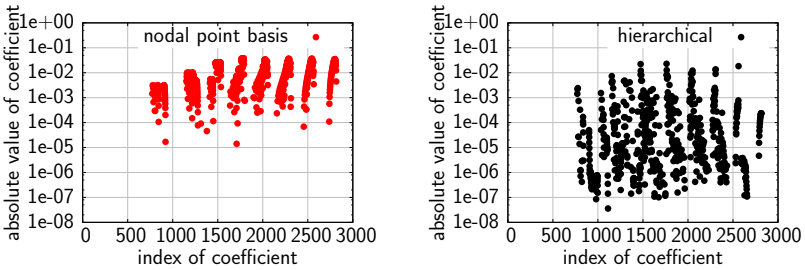
In this section, we report on the performance of our reduced model for pricing basket options with multiple underlying assets. We first discuss how the hierarchical basis effects the POD representation, and then show accuracy and runtime results for 2-, 3-, and 4-asset basket options where we gain speedups between 80 and 160 for solving the system of linear equations and still achieve an accuracy of about 10^{-2} with respect to the high-fidelity model.

6.1 POD Representation of Sparse Grid Data

We show details on how the hierarchical basis effects the POD representation with the help of a 2-asset basket option. We compute 150 snapshots with

Table 1: Black-Scholes: Runtime and accuracy results are reported for a 1-asset option priced by the high-fidelity model (first row) and our reduced models with 10 to 100 POD basis vectors. Our reduced model with 10 POD modes achieves an error below 10^{-3} with respect to the high-fidelity solution. Furthermore, the corresponding system is solved about 400 times faster than the one of the high-fidelity model. For this 1-asset problem, no speedup is achieved in the operator phase because loading the pre-computed reduced matrices from disk takes more time than assembling the full-order matrices.

#modes	operator setup		solve		price	abs. L^2	rel. L^2
	time[s]	speedup	time[s]	speedup			
-	0.14	1.0	0.598	1.0	0.142191	-	-
10	0.20	0.7	0.0014	427.1	0.143376	0.001453	0.000456
20	0.22	0.6	0.0028	213.5	0.142752	0.000559	0.000175
30	0.22	0.6	0.0042	142.3	0.142754	0.000543	0.000170
40	0.22	0.6	0.0061	98.0	0.142618	0.000423	0.000132
60	0.23	0.6	0.0106	56.4	0.142479	0.000278	0.000087
80	0.25	0.5	0.0162	36.9	0.142427	0.000241	0.000076
100	0.26	0.5	0.0220	27.1	0.142404	0.000219	0.000069



(a) nodal point basis (b) hierarchical basis

Fig. 2: The plot shows the coefficients of the nodal point and the hierarchical basis representation of the snapshots. The nodal point basis coefficients are in range $[10^{-5}, 10^{-1}]$, but the (multi-scale) hierarchical coefficients are distributed between 10^{-8} and 10^{-1}. All coefficients with an absolute value below 10^{-15} are not plotted.

parameters $\boldsymbol{\mu} = [\eta_1, \eta_2, \sigma_1, \sigma_2, \rho_{12}]^T$ chosen from a uniform distribution in the range $[0.06, 0.10] \times [0.07, 0.11] \times [0.2, 0.4] \times [0.3, 0.5] \times [-0.6, -0.4] \in \mathbb{R}^5$. These are then represented in the hierarchical and nodal point basis in the matrices $\hat{\boldsymbol{U}}^{\text{hier}} \in \mathbb{R}^{N_B \times 150}$ and $\hat{\boldsymbol{U}}^{\text{nodal}} \in \mathbb{R}^{N_B \times 150}$, respectively. Note that in this subsection we compute the POD basis vectors for the snapshots including the boundary points for a better visualization in the following figures.

Figure 2 shows the absolute value of the coefficients corresponding to the two representations for one particular snapshot. It confirms that the hierarchical basis leads to coefficients at different scales in $[10^{-8}, 10^{-1}]$ whereas the coefficients corresponding to the nodal point basis are only in $[10^{-5}, 10^{-1}]$. All coefficients with an absolute value below 10^{-15} are not plotted.

We compute the POD basis vectors (left singular vectors) $\boldsymbol{W}^{\text{nodal}}$ and $\boldsymbol{W}^{\text{hier}}$ of the snapshot matrices $\hat{\boldsymbol{U}}^{\text{nodal}}$ and $\hat{\boldsymbol{U}}^{\text{hier}}$, respectively. The first four

POD basis vectors are visualized in Figure 3. The first column corresponds to the nodal point basis representation. In the second and third column, the POD vectors are plotted for the hierarchical basis representation, but whereas the second column directly plots the entries of $\boldsymbol{W}^{\text{hier}}$, the third column plots the sparse grid function f^j that is defined by the coefficients in column j of $\boldsymbol{W}^{\text{hier}}$, i.e.,

$$f^j = \sum_{i=1}^{N_B} W_{ij}^{\text{hier}} \phi_i \,,$$

where $\phi_1, \ldots, \phi_{N_B}$ are the hierarchical basis functions. The nodal point (column one) and the hierarchical (column three) basis representation lead to POD modes with a similar structure.

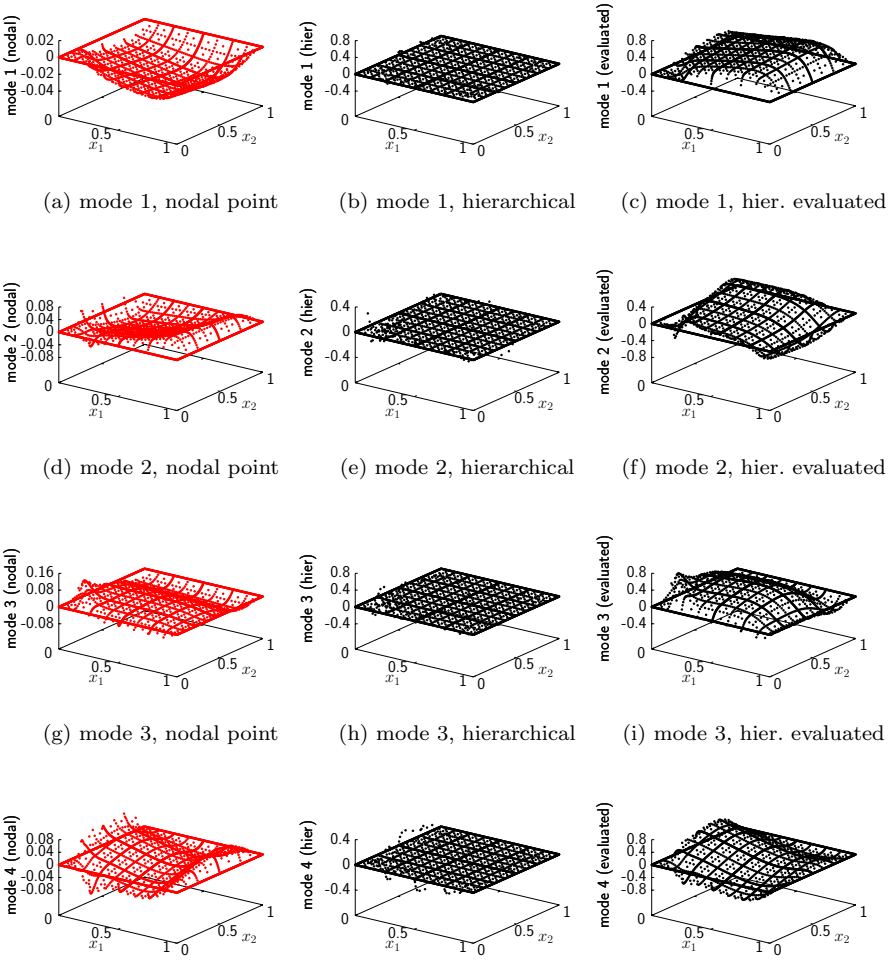
6.2 Pricing Basket Options

We consider 2-, 3-, and 4-asset options with a symmetric correlation matrix $\boldsymbol{\Sigma}$. A 2-asset option has 5 parameters, a 3-asset option has 9 parameters, and a 4-asset option has 14 parameters. In case of two assets, we have $\boldsymbol{\eta}^{(2)} = [0.08, 0.09]^T$, $\boldsymbol{\sigma}^{(2)} = [0.3, 0.4]^T$, and $\rho_{12}^{(2)} = \rho_{21}^{(2)} = -0.5$. For three and four assets, we set $\boldsymbol{\eta}^{(3)} = [0.1, 0.02, 0.04]^T$, $\boldsymbol{\sigma}^{(3)} = [0.2, 0.3, 0.4]^T$, $\boldsymbol{\eta}^{(4)} = [0.05, 0.05, 0.05, 0.05]^T$, $\boldsymbol{\sigma}^{(4)} = [0.4, 0.25, 0.3, 0.4]^T$, and

$$\boldsymbol{\Sigma}^{(3)} = \begin{bmatrix} 1.0 & -0.7 & -0.1 \\ -0.7 & 1.0 & 0.1 \\ -0.1 & 0.1 & 1.0 \end{bmatrix}, \quad \boldsymbol{\Sigma}^{(4)} = \begin{bmatrix} 1 & 0.1 & -0.4 & 0.2 \\ 0.1 & 1 & 0.3 & -0.1 \\ -0.4 & 0.3 & 1 & 0 \\ 0.2 & -0.1 & 0 & 1 \end{bmatrix}. \tag{26}$$

We compute 500, 600, and 700 snapshots for 2-, 3-, and 4-asset options, respectively, where the parameters are chosen randomly from a uniform distribution with the range ± 0.02 around the drifts, and ± 0.1 around the volatilities and correlations. We discretize the 2-asset option pricing problem on a sparse grid of level seven ($N = 769$ degrees of freedom, $N_B = 1281$ grid points), the 3-asset problem on a grid of level six ($N = 1023$ degrees of freedom, $N_B = 3713$ grid points), and the 4-asset problem also on a grid of level six ($N = 2561$ degrees of freedom, $N_B = 20,481$ grid points). Besides that, we consider the same setting as in Section 5.

The runtime and accuracy results of our reduced model for basket options are shown in Table 2. Again, the first row of each block corresponds to the high-fidelity model, and the following rows to reduced models with different numbers of POD modes. We report the price of the options at stock price $\boldsymbol{x} = \mathbf{1}$, which is near the bend of the solution and thus where the sparse grid discretization error is usually large. Still, the reduced model can predict the price well. We also achieve a relative L^2 error between 10^{-4} and 10^{-2}, which is similar to what is reported in [6,7,22]. The error increases with the number of underlying assets. One reason might be that the condition of the system (17) becomes worse with increasing dimension d, see [6,7,22] for a detailed study.



(a) mode 1, nodal point (b) mode 1, hierarchical (c) mode 1, hier. evaluated

(d) mode 2, nodal point (e) mode 2, hierarchical (f) mode 2, hier. evaluated

(g) mode 3, nodal point (h) mode 3, hierarchical (i) mode 3, hier. evaluated

(j) mode 4, nodal point (k) mode 4, hierarchical (l) mode 4, hier. evaluated

Fig. 3: The plots visualize the first four POD basis vectors for the 2-asset snapshot matrix. On the left, the POD vectors corresponding to the nodal point basis representation are shown. The POD vectors for the hierarchical representation are plotted in column two, and the corresponding sparse grid function is plotted in column three. The nodal point (first column) and the hierarchical basis representation (third column) lead to POD basis vectors with a similar structure.

Another reason could be that the number of parameters increases drastically with the number of underlyings, which allows for many different system behaviors and so a larger number of POD modes is required. Still, to obtain an accuracy of about 10^{-2} to $5 \cdot 10^{-2}$ with respect to the high-fidelity model, we achieve speedups in solving the system of linear equations of 160 (40 POD

modes), 80 (100 POD modes), and 130 (300 POD modes) for the 2-, 3-, and 4-asset option pricing problem, respectively. We also note that the speedup drops quickly if we increase the number of POD modes k, but that this becomes less significant as we increase the number of underlying assets. The reason is that the reported runtime also includes the transformation of the reduced solution from the POD basis back into the hierarchical basis that takes a larger share of the runtime when the number of assets, and consequently the number of sparse grid points, is high. Thus, an increase in the number of POD modes has less effect on the overall solver runtime in the case of four assets than in the case of two and three assets. In the operator phase, speedups of about 2, 6, and 20 are achieved for the 2-, 3-, and 4-asset option pricing problem, respectively. The results show that the speedup increases as the number of assets is increased. We note again that the operators have to be loaded only once and can then be reused for different parameter configurations. Let us finally consider the runtime of the offline phase for the 4-asset example. The runtime is dominated by the snapshot generation. We generate $M = 700$ snapshots. According to Table 2 the operator setup (once) needs 14595 seconds and solving the corresponding systems 700 times requires about 47 seconds each. This gives a total of about 13 hours for creating a reduced model for the 4-asset problem on a single core if operators are reused.

In Table 3 we report results for the 4-asset problem where we increased the parameter domain. Whereas we originally built the reduced model for the range ± 0.02 around the drifts and ± 0.1 around the volatilities and correlation coefficients, we now also consider the two times and four times larger ranges with ± 0.04 and ± 0.08 around the drifts and ± 0.2 and ± 0.4 around the volatilities and the correlation coefficients, respectively. The results reported in Table 3 show that the accuracy of the reduced model only slightly decreases as the parameter domain is increased. This is the expected behavior because a larger parameter domain leads to a richer high-fidelity model for which the reduced model requires more POD modes.

With the results in Table 4 we show the accuracy of reduced models constructed from 700, 1500, and 2000 snapshots. The accuracy does not improve as we increase the number of snapshots. This indicates that our initial choice of 700 snapshots for the 4-asset problem was sufficient. Note that the errors reported in Table 4 might slightly change if the number of snapshots is increased. This is because the POD basis for the reduced model is generated with respect to the snapshots but is then tested on a solution that was not used as snapshot, see Section 4.1.

Finally, we show in Figure 4a a pointwise error plot for the 2-asset basket option. There is a highly oscillatory error near the kink at the strike price $K = 1$ that originates from the sparse grid discretization [47]. In Figure 4b we plot the relative L^2 error of the reduced approximation with 40 POD modes for each time step for the 3-asset problem. The error is larger in the first time steps because our snapshots consist of the solution at the final time step only. We therefore report in Table 5 how the accuracy of the reduced model changes if the snapshots also include the solutions at every, every 10-th, and every 25-

Table 2: Black-Scholes: Runtime and accuracy results are reported for the 2-, 3-, 4-asset basket options priced with the high-fidelity model (first row in each block) and with our reduced models for $k = 40$ to $k = 300$ POD basis vectors. We achieve speedups in solving the systems of linear equation between 80 and 160 and still obtain an accuracy of about 10^{-2} to $5{\cdot}10^{-2}$ with respect to the high-fidelity model. The speedup in the operator phase is between 2 and 20 and quickly improves with the number of assets. Note that the operators can be reused for different parameter configurations after they have been assembled or loaded.

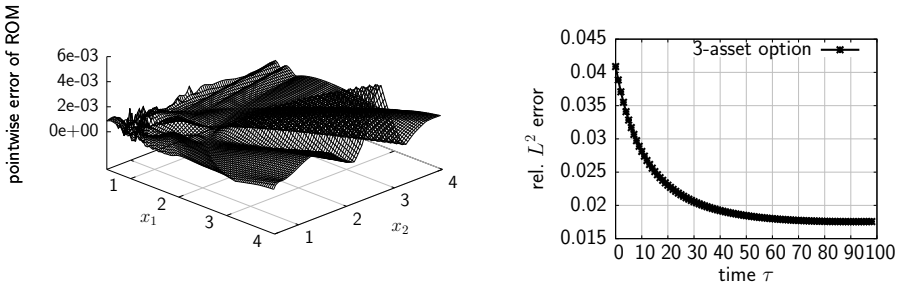
#modes	operator setup		solve		price	abs. L^2	rel. L^2
	time[s]	speedup	time[s]	speedup			
			2-asset basket				
-	4.53	1.0	1.29	1.0	0.119819	-	-
40	1.96	2.3	0.0081	159.2	0.119568	0.004968	0.003048
60	1.87	2.4	0.0129	100.0	0.119596	0.005513	0.003382
80	1.88	2.4	0.0186	69.3	0.120612	0.003724	0.002284
100	2.07	2.1	0.0257	50.1	0.120680	0.002822	0.001731
200	2.10	2.1	0.0804	16.1	0.119072	0.000835	0.000512
300	2.47	1.8	0.1697	7.6	0.120310	0.000602	0.000369
			3-asset basket				
-	139.83	1.0	2.998	1.0	0.086274	-	-
40	21.83	6.4	0.014	214.1	0.090982	0.065028	0.017566
60	22.63	6.1	0.020	149.9	0.086753	0.057331	0.015487
80	21.97	6.3	0.026	115.3	0.088119	0.055508	0.014994
100	22.34	6.2	0.034	88.1	0.086242	0.044121	0.011918
200	22.87	6.1	0.095	31.5	0.086697	0.031781	0.008585
300	23.40	5.9	0.186	16.1	0.085690	0.023374	0.006314
			4-asset basket				
-	14595	1.0	46.790	1.0	0.091571	-	-
40	766.16	19.0	0.120	389.9	0.080437	0.309468	0.064685
60	747.79	19.5	0.127	368.4	0.078553	0.288774	0.060359
80	751.89	19.4	0.144	324.9	0.083016	0.280076	0.058541
100	749.78	19.4	0.156	299.9	0.087010	0.278561	0.058225
200	755.66	19.3	0.236	198.2	0.090035	0.255502	0.053405
300	760.40	19.1	0.346	135.2	0.087448	0.253988	0.053088

Table 3: Black-Scholes: This table reports the price and accuracy obtained with reduced models of the 4-asset basket option pricing problem for different sizes of the parameter domain. The results confirm that the accuracy slightly decreases as the parameter domain is increased but that this can be compensated by increasing the dimension of the reduced model.

#modes	original param. domain		two times larger		four times larger	
	price	rel. L^2 err.	price	rel. L^2 err.	price	rel. L^2 err.
40	0.080437	0.064685	0.082293	0.072020	0.081089	0.076881
60	0.078553	0.060359	0.090056	0.069852	0.085374	0.073441
80	0.083016	0.058541	0.087592	0.068555	0.083794	0.071468
100	0.087010	0.058225	0.092695	0.068433	0.081353	0.069881
200	0.090035	0.053405	0.092996	0.067806	0.093087	0.068245
300	0.087448	0.053088	0.095273	0.064469	0.094025	0.067005

Table 4: Black-Scholes: We report the accuracy of reduced models constructed from 700, 1500, and 2000 snapshots, respectively, for the 4-asset basket option. The reduced models were constructed for the parameter domain with the range ± 0.08 around the drifts and ± 0.4 around the volatilities and correlations. The results confirm that the predicted prices as well as the L^2 errors change only slightly.

#modes	#snapshots = 700		#snapshots = 1500		#snapshots = 2000	
	price	rel. L^2 err.	price	rel. L^2 err.	price	rel. L^2 err.
40	0.081089	0.076881	0.080763	0.077686	0.074529	0.082620
60	0.085374	0.073441	0.084046	0.075550	0.082079	0.080490
80	0.083794	0.071468	0.078885	0.074863	0.080213	0.078906
100	0.081353	0.069881	0.081036	0.072778	0.083630	0.078918
200	0.093087	0.068245	0.084706	0.068320	0.085835	0.071286
300	0.094025	0.067005	0.088848	0.066846	0.085069	0.068860



(a) pointwise error of reduced solution (b) error at each time step

Fig. 4: Black-Scholes: In (a) the pointwise error with respect to the high-fidelity model is plotted for the reduced solution of the 2-asset option problem. The evolution of the relative L^2 error over time is plotted for the 3-asset problem in (b).

th time step. The results confirm that the solution at the final time step only slightly changes.

7 Numerical Results: Heston Model

A reduced model based on a high-fidelity sparse grid model can be derived for the PDE corresponding to the Heston model in a similar manner as for the Black-Scholes equation, see Sections 2.3 and 4. We consider here a single-asset option that corresponds to a two-dimensional pricing problem, i.e., the spatial domain of the PDE (6) is in $\mathbb{R}^2$. We generate 500 snapshots for parameters $[\kappa, \theta, \xi]$ drawn from a uniform distribution in

$$[1.5, 2] \times [0.2, 0.4] \times [0.1, 0.4] \subset \mathbb{R}^3 \,, \tag{27}$$

$$\tag{28}$$

and set $\rho = 0.9$. We follow a similar problem setup as in Section 6.2: The pricing problem is discretized on a sparse grid of level seven and we only keep

Table 5: Black-Scholes: We compare for the 3-asset problem the accuracy that we obtain if not only the solution at the final time step but also at every, every 10-th, and every 25-th time step are included into the snapshot matrix. The reported results confirm that the snapshots at the final time step are sufficient in our case because the solution changes only slightly during the time steps.

#modes	every 25-th time step		every 10-th time step		every time step	
	price	rel. L^2 err.	price	rel. L^2 err.	price	rel. L^2 err.
40	0.084808	0.018058	0.083605	0.019363	0.082674	0.018564
60	0.085601	0.015902	0.087300	0.015730	0.086174	0.016660
80	0.087268	0.015343	0.086921	0.015028	0.087827	0.015350
100	0.085617	0.015019	0.086350	0.014069	0.088316	0.014735
200	0.085623	0.010553	0.087271	0.011476	0.087914	0.011538
300	0.086333	0.008516	0.087201	0.009180	0.086871	0.009283

Table 6: Heston: Our model reduction approach introduced for the Black-Scholes model is also applicable to the Heston model. The reported results confirm that speedups up to 500 are achieved with the model reduction approach based on sparse grid discretizations and POD.

#modes	solve		avg abs L^2	min rel L^2	avg rel L^2	max rel L^2
	time[s]	speedup				
-	5.1777	1	-	-	-	-
40	0.0097	533.3	0.153285	0.112562	0.159706	0.195401
60	0.0160	322.6	0.094793	0.083360	0.099475	0.119408
80	0.0248	208.6	0.070438	0.044792	0.072663	0.107145
100	0.0361	143.2	0.070546	0.052427	0.073297	0.101834
200	0.1500	34.5	0.076412	0.047996	0.079195	0.099355
300	0.4539	11.4	0.026619	0.016613	0.027667	0.037096

the solutions at the final time step as snapshots. The solver is implemented in Matlab.

Table 6 reports runtime and accuracy results of our reduced model. We test the reduced model on ten option pricing problems with ten randomly (uniformly) chosen parameters in the domain (28). The minimum, average, and maximum relative L^2 error is reported. The spread between the minimum and maximum relative L^2 error is small. An average relative L^2 error of about 10^{-2} is achieved, as in the case of the Black-Scholes model; however, note, that significantly more POD modes were necessary than in the case of the Black-Scholes model. The reduced model is up to 500 times faster to evaluate than the high-fidelity sparse grid model here.

8 Conclusions and Outlook

Our model reduction approach combines a direct sparse grid discretization of the Black-Scholes equation with POD to derive reduced models that allow us to price basket options with up to four underlying assets. The solution process is about 80 to 160 times faster than the high-fidelity sparse grid model and thus

lets us price basket options with the simple Black-Scholes model in fractions of a second.

The number of underlying assets is limited by the high computational costs of the sparse grid solver. A tensor-based discretization [29] could allow for a larger number of underlying assets. Furthermore, a 4-asset option leads to 14, a 5-asset option to 20, and a 6-asset option already to 27 parameters. So far, the presented results suggest that this increasing number of parameters leads to a richer system behavior that has to be covered by the POD modes. A remedy could be to compute local reduced models for each hierarchical increment of the high-fidelity sparse grid function [1,36]. Besides that, dimensionality reduction could be applied to reduce the number of underlying assets, cf. [44,23].

We only discussed the Black-Scholes and the Heston model here, which both ignore important real-world phenomena. Future work therefore includes the extension to more sophisticated option pricing models, which require more advanced model reduction techniques. In [47], a versatile framework based on sparse grids for pricing problems is presented, including American and Asian options, as well as variable annuities. The sparse grid models introduced in [47] could therefore serve as high-fidelity models for model reduction. For example, consider American options, where additional inequality constraints have to be satisfied. A model reduction approach for American options is available, see, e.g., [21]. With the sparse grid model introduced in [47], the model reduction technique presented in [21] could be extended to basket options. Similarly, in [17], a sparse grid model is developed for Kuo's jump-diffusion model that takes into account effects such as the 'volatility smile'. This results in an additional integral term, which is efficiently treated by a sparse grid algorithm. It would be interesting to see if the evaluation of the integral term could be further accelerated by using a reduced modeling approach.

References

1. Amsallem, D., Zahr, M., Farhat, C.: Nonlinear model order reduction based on local reduced-order bases. International Journal for Numerical Methods in Engineering **92**(10), 891–916 (2012)
2. Black, F., Scholes, M.: The pricing of options and corporate liabilities. Journal of Political Economy **81**(3), 637–654 (1973)
3. Bui-Thanh, T., Willcox, K., Ghattas, O.: Model reduction for large-scale systems with high-dimensional parametric input space. SIAM Journal on Scientific Computing **30**(6), 3270–3288 (2008)
4. Bungartz, H.J., Griebel, M.: A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives. Journal of Complexity **15**(2), 167 – 199 (1999)
5. Bungartz, H.J., Griebel, M.: Sparse grids. Acta Numerica **13**, 1–123 (2004)
6. Bungartz, H.J., Heinecke, A., Pflüger, D., Schraufstetter, S.: Option pricing with a direct adaptive sparse grid approach. Journal of Computational and Applied Mathematics **236**(15), 3741–3750 (2011)
7. Bungartz, H.J., Heinecke, A., Pflüger, D., Schraufstetter, S.: Parallelizing a Black-Scholes solver based on finite elements and sparse grids. Concurrency and Computation: Practice and Experience pp. 1640–1653 (2012)
8. Burkovska, O., Haasdonk, B., Salomon, J., Wohlmuth, B.: Reduced basis methods for pricing options with the Black-Scholes and Heston model. arXiv:1408.1220 [math] (2014)

9. Cont, R., Lantos, N., Pironneau, O.: A reduced basis for option pricing. SIAM Journal on Financial Mathematics **2**(1), 287–316 (2011)
10. Falc, A., Chinesta, F., Gonzlez, M.: Model reduction methods in option pricing. Working Papers. Serie AD 2006-16, Instituto Valenciano de Investigaciones Econmicas, S.A. (Ivie) (2006). URL http://ideas.repec.org/p/ivi/wpasad/2006-16.html
11. Feuersänger, C.: Sparse grid methods for higher dimensional approximation. PhD thesis, Institut für Numerische Simulation, Universität Bonn (2010)
12. Garcke, J., Gerstner, T., Griebel, M.: Intraday foreign exchange rate forecasting using sparse grids. In: J. Garcke, M. Griebel (eds.) Sparse Grids and Applications, *Lecture Notes in Computational Science and Engineering*, vol. 88, pp. 81–105. Springer (2013)
13. Gerstner, T., Griebel, M.: Dimension-adaptive tensor-product quadrature. Computing **71**(1), 65–87 (2003)
14. Giles, M.: Multi-level Monte Carlo path simulation. Operations Research **56**(3), 607–617 (2008)
15. Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer (2004)
16. Griebel, M., Holtz, M.: Dimension-wise integration of high-dimensional functions with applications to finance. Journal of Complexity **26**, 455–489 (2010)
17. Griebel, M., Hullmann, A.: An efficient sparse grid Galerkin approach for the numerical valuation of basket options under Kou's jump-diffusion model. In: Sparse grids and Applications, Lecture Notes in Computational Science and Engineering, pp. 121–150. Springer (2013)
18. Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. In: P. de Groen, R. Beauwens (eds.) Iterative Methods in Linear Algebra, pp. 263–281. Elsevier (1992)
19. Guennebaud, G., Jacob, B., et al.: Eigen v3. http://eigen.tuxfamily.org (2010)
20. Haasdonk, B.: Convergence rates of the POD-Greedy method. M2AN Math. Model. Numer. Anal. **47**, 859–873 (2013)
21. Haasdonk, B., Salomon, J., Wohlmuth, B.: A reduced basis method for the simulation of American options. In: ENUMATH 2012 (2012)
22. Heinecke, A., Schraufstetter, S., Bungartz, H.J.: A highly-parallel Black-Scholes solver based on adaptive sparse grids. International Journal of Computer Mathematics **89**(9), 1212–1238 (2012)
23. Hepperger, P.: Option pricing in Hilbert space-valued jump-diffusion models using partial integro-differential equations. SIAM Journal on Financial Mathematics **1**(1), 454–489 (2010)
24. Hesthaven, J.S., Stamm, B., Zhang, S.: Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. ESAIM: Mathematical Modelling and Numerical Analysis **48**, 259–283 (2014)
25. Heston, S.L.: A closed-form solution for options with stochastic volatility with applications to bond and currency options. Review of Financial Studies **6**, 327–343 (1993)
26. Holtz, M.: Sparse grid quadrature in high dimensions with applications in finance and insurance. PhD thesis, Institut für Numerische Simulation, Universität Bonn (2008)
27. Jolliffe, I.: Principal Component Analysis. Springer (2002)
28. Kangro, R., Nicolaides, R.: Far field boundary conditions for Black-Scholes equations. SIAM Journal on Numerical Analysis **38**(4), 1357–1368 (2000)
29. Kiesel, R., Rupp, A., Urban, K.: Valuation of structured financial products by adaptive multiwavelet methods in high dimensions. Tech. rep., University of Ulm (2013)
30. Leentvaar, C., Oosterlee, C.: On coordinate transformation and grid stretching for sparse grid pricing of basket options. Journal of Computational and Applied Mathematics **222**(1), 193–209 (2008)
31. Leentvaar, C.C.W.: Pricing multi-asset options with sparse grids. PhD thesis, TU Delft (2008)
32. Linde, G., Persson, J., Von Sydow, L.: A highly accurate adaptive finite difference solver for the Black-Scholes equation. International Journal of Computer Mathematics **86**(12), 2104–2121 (2008)
33. Lötstedt, P., Persson, J., von Sydow, L., Tysk, J.: Space time adaptive finite difference method for European multi-asset options. Computers & Mathematics with Applications **53**(8), 1159–1180 (2007)

34. Maurus, S.: A multi-dimensional PDE solver for option pricing based on the Heston model and sparse grids. Thesis (S.M.), Institut für Informatik, Technische Universität München (2012)
35. Mayerhofer, A., Urban, K.: A reduced basis method for parabolic partial differential equations with parameter functions and application to option pricing (2014). Submitted
36. Peherstorfer, B., Butnaru, D., Willcox, K., Bungartz, H.: Localized discrete empirical interpolation method. SIAM Journal on Scientific Computing **36**(1), A168–A192 (2014)
37. Peherstorfer, B., Kowitz, C., Pflüger, D., Bungartz, H.J.: Selected recent applications of sparse grids. Numerical Mathematics: Theory, Methods and Applications **8**, 47–77 (2015)
38. Peherstorfer, B., Zimmer, S., Bungartz, H.J.: Model reduction with the reduced basis method and sparse grids. In: Sparse Grids and Applications. Springer (2013)
39. Pironneau, O.: Calibration of options on a reduced basis. Journal of Computational and Applied Mathematics **232**(1), 139–147 (2009)
40. Pironneau, O.: Reduced basis for vanilla and basket options. Risk and Decision Analysis **2**, 185–194 (2011)
41. Pironneau, O.: Proper orthogonal decomposition for pricing options. The Journal of Computational Finance **16**(1) (2012)
42. Reisinger, C.: Analysis of linear difference schemes in the sparse grid combination technique. IMA Journal of Numerical Analysis **33**(2), 544–581 (2013)
43. Reisinger, C., Wittum, G.: On multigrid for anisotropic equations and variational inequalities. Pricing multi-dimensional European and American options. Computing and Visualization in Science **7**(3-4), 189–197 (2004)
44. Reisinger, C., Wittum, G.: Efficient hierarchical approximation of high-dimensional option pricing problems. SIAM Journal on Scientific Computing **29**(1), 440–458 (2007)
45. Rozza, G., Huynh, D., Patera, A.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archives of Computational Methods in Engineering **15**(3), 1–47 (2007)
46. Sachs, E., Schu, M.: Reduced order models (POD) for calibration problems in finance. In: Numerical Mathematics and Advanced Applications, pp. 735–742. Springer (2008)
47. Schraufstetter, S.: A pricing framework for the efficient evaluation of financial derivatives based on Theta calculus and adaptive sparse grids. PhD thesis, Institut für Informatik, Technische Universität München (2012)
48. Schröter, T., Monoyios, M., Rometsch, M., Urban, K.: Model uncertainty and the robustness of hedging models (2012). Submitted
49. Sirovich, L.: Turbulence and the dynamics of coherent structures. Quarterly of Applied Mathematics pp. 561–571 (1987)
50. Veroy, K., Patera, A.: Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. International Journal for Numerical Methods in Fluids **47**(8-9), 773–788 (2005)
51. Wilmott, P., Dewynne, J., Howison, S.: Option Pricing: Mathematical Models and Computation. Oxford Financial Press (1994)
52. Yserentant, H.: Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. Applied Mathematics and Computation **19**(1-4), 347–358 (1986)
53. Zenger, C.: Sparse grids. In: W. Hackbusch (ed.) Parallel Algorithms for Partial Differential Equations, *Notes on Numerical Fluid Mechanics*, vol. 31, pp. 241–251. Vieweg (1991)
54. Zhou, Y.B.: Model reduction for nonlinear dynamical systems with parametric uncertainties. Thesis (S.M.), Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics (2012)