C. Reisinger[1,2] *and R. Wissmann*[1]

[1] *Mathematical Institute, University of Oxford, OX2 6GG, United Kingdom*
[2] *Oxford-Man Institute of Quantitative Finance, University of Oxford, OX2 6ED, United Kingdom*

Finite difference methods for medium- and high-dimensional derivative pricing PDEs

Contents

0.1 Introduction

Many models in financial mathematics and financial engineering, particularly in derivative pricing, can be formulated as partial differential equations (PDEs). Specifically, for the most commonly used continuous-time models of asset prices the value function of a derivative security, that is the option value as a function of the underlying asset price, is given by a PDE. This opens up the possibility to use accurate approximation schemes for PDEs for the numerical computation of derivative prices.

As the computational domain is normally a box, or can be restricted to one by truncation, the construction of tensor product meshes and spatial finite difference stencils is straightforward (see, e.g., [21]). Accurate and stable

splitting methods have become standard for efficient time integration (see, e.g., [9]).

Notwithstanding this, the more common approach in the financial industry appear to be Monte Carlo methods. This is partly a result of the perception that PDE schemes, although highly efficient for simple contracts, are less flexible and harder to adapt to more exotic features. In particular, the widespread belief is that PDE schemes become too slow for practical use if the number of underlying variables exceeds three.

Indeed, the increase in computational time and memory requirements of standard mesh based methods with the dimension is exponential, and has become known as the "curse of dimensionality". Various methods, such as sparse grids ([18], [12]), radial basis functions ([14]) and tensor approaches ([11] for an application to finance and [4] for a literature survey) have been proposed to break this curse. These methods can perform remarkably well for special cases, but have not been demonstrated to give accurate enough solutions for truly high dimensions in applications (larger than, say, five).

In conversations about numerical methods for high-dimensional PDEs inevitably the question comes up: "How high can you go?". This is a meaningful question if one considers a specific type of PDE with closely defined characteristics. But even within the fairly narrow class of linear second-order parabolic PDEs which are most common in finance, the difficulty of solving them varies vastly and depends on a number of factors: the input data such as volatilities and correlations, the boundary data (payoff), and the quantity of interest (usually the solution of the PDE at a single point).

It is inherent to the methods presented in this paper that it is not the nominal dimension of a PDE which matters. A PDE which appears inaccessible to numerical methods in its raw form, may be very easily approximated if a more adapted coordinate system is chosen. This can be either because the solution is already adequately described by a low number of principal components (it has low "truncation dimension"), or because it can be accurately represented as the sum of functions of a low number of variables (it has low "superposition dimension").

To exploit such features, we borrow ideas from data analysis to represent the solutions by sums of functions which can be approximated by PDEs with low effective dimension. More specifically, the method is a 'dynamic' version of the anchored-ANOVA decompositions which were applied to integration problems in finance in [5]. A version which is equivalent in special cases has been independently derived via PDE expansions by [18]; a detailed error analysis is found in [16] and also in [7]; an efficient parallelisation strategy is proposed in [20]; and the method is extended to complex derivatives in [17] and to CVA computations in [2]. The link of these methods to anchored-ANOVA is already observed in [15] and [19]. We present here a systematic approach which extends [16] from Black-Scholes to more general models, and analyse the accuracy of the approximations by way of carefully chosen numerical tests.

In the remainder of this section, we describe the mathematical framework.

Then, in Section 0.2, we describe the standard approximation schemes. In Section 0.3, we define and explain in detail a dimension-wise decomposition. Section 0.4 summarises known theoretical results for the constant coefficient case, and offers a heuristic argument for the accuracy of a variable coefficient extension. Section 0.5 gives numerical results for test cases. We draw conclusions in Section 0.6.

Throughout this chapter, we study asset price processes of the form

$$\begin{aligned} dS_t^i &= \mu_i(S_t,t)\,dt + \sigma_i(S_t,t)\,dW_t^i, \qquad i=1,\dots,N, \quad t>0, \qquad (1)\\ S_0^i &= s_i, \qquad\qquad\qquad\qquad\qquad\quad i=1,\dots,N, \qquad (2) \end{aligned}$$

where W is an N-dimensional standard Brownian motion, $s \in \mathbb{R}^N$ a given initial state, the drift μ_i and local volatility σ_i are functions $\mathbb{R}^N \times [0,T] \to \mathbb{R}$, and we will allow the correlation between the Brownian drivers also to be 'local', i.e. given S_t at time t the instantaneous correlation matrix is $(\rho_{ij}(S_t,t))_{1\le i,j\le N}$. We consider European-style financial derivatives on S_T with maturity $T>0$ and payoff function $h\colon \mathbb{R}^N \to \mathbb{R}$, whose value function $V\colon \mathbb{R}^N \times [0,T] \to \mathbb{R}$ can be written as

$$V(s,t) = \mathbb{E}[\exp(-\textstyle\int_t^T \alpha(S_u,u)\,du) h(S_T) | S_t = s],$$

where α is a discount factor, possibly stochastic through its dependence on S, and V satisfies the Kolmogorov backward PDE (see, e.g. [13])

$$\begin{aligned} \frac{\partial V}{\partial t} + \sum_{i=1}^N \mu_i \frac{\partial V}{\partial s_i} + \frac{1}{2}\sum_{i,j=1}^N \sigma_i\sigma_j\rho_{ij}\frac{\partial^2 V}{\partial s_i \partial s_j} - \alpha V &= 0,\\ V(s,T) &= h(s). \end{aligned}$$

For simplicity we consider functions defined on the whole of $\mathbb{R}^N$, but it will become clear how to deal with bounded domains.

Let $p(y,t;s,0)$ be the transition density function of S_t at y given state s at $t=0$. Then if α does not depend on S, we can write

$$V(s,0) = \exp(-\textstyle\int_0^T \alpha(u)\,du) \displaystyle\int_{\mathbb{R}^N} p(y,T;s,0) h(y)\,dy.$$

Here, p satisfies the Kolmogorov forward equation

$$\begin{aligned} -\frac{\partial p}{\partial t} - \sum_{i=1}^N \frac{\partial}{\partial y_i}(\mu_i p) + \frac{1}{2}\sum_{i,j=1}^N \frac{\partial^2}{\partial y_i \partial y_j}(\sigma_i\sigma_j\rho_{ij}p) &= 0,\\ p(y,0;s,0) &= \delta(y-s), \end{aligned}$$

where δ is the Dirac distribution centred at 0.

Most commonly, one is interested in approximating the value of $V(s_0,0)$ for a given, fixed $s_0 \in \mathbb{R}^N$, and derivatives of V with respect to s_0.

As a first step, we change the time direction to *time-to-maturity*, $t \to T-t$, to obtain

$$\frac{\partial V}{\partial t} = \sum_{i=1}^{N} \mu_i \frac{\partial V}{\partial s_i} + \frac{1}{2} \sum_{i,j=1}^{N} \sigma_i \sigma_j \rho_{ij} \frac{\partial^2 V}{\partial s_i \partial s_j} - \alpha V, \tag{3}$$

$$V(s,0) = h(s), \tag{4}$$

where we keep the symbols t and V for simplicity. We now transform the PDE into a standard form by using a rotation and subsequent translation of the spatial coordinates. For a given orthogonal matrix $Q \in \mathbb{R}^{N\times N}$, define $\beta\colon \mathbb{R}^N \times [0,T] \to \mathbb{R}^N$ componentwise by

$$\beta_i(x,t) \equiv \sum_{j=1}^{N} Q_{ji} \int_0^t \mu_j(x, T-u)\, du \tag{5}$$

for $1 \le i \le N$. We then introduce new spatial coordinates x via

$$x(s,t) = Q^{\mathrm{T}} s + \beta(s_0, t) \tag{6}$$

and set

$$a = Q^{\mathrm{T}} s_0 + \beta(s_0, T). \tag{7}$$

We write $s(x,t) = Q(x - \beta(s_0,t))$ for the inverse transform.

A simple calculation shows that the PDE (3–4) transforms into

$$\frac{\partial V}{\partial t} = \mathcal{L}V := \sum_{k,l=1}^{N} \lambda_{kl} \frac{\partial^2 V}{\partial x_k \partial x_l} + \sum_{k=1}^{N} \kappa_k \frac{\partial V}{\partial x_k} - \alpha V, \tag{8}$$

$$V(x,0) = g(x) := h(s(x,0)), \tag{9}$$

for a function $V\colon \mathbb{R}^N \times [0,T] \to \mathbb{R}$, $T > 0$, where we still call the transformed function V by slight abuse of notation, and

$$\begin{aligned} \lambda_{kl}(x,t) &= \frac{1}{2} \sum_{i,j=1}^{N} Q_{ik} Q_{jl} \sigma_i \sigma_j \rho_{ij}, \\ \kappa_k(x,t) &= \sum_{i=1}^{N} Q_{ik} \left[\mu_i - \mu_i(s_0, T-t)\right], \end{aligned} \tag{10}$$

where σ_i and ρ_{ij} are functions of $(s(x,t), T-t)$.

For a constant (i.e., independent of time and the spatial coordinates), positive semidefinite coefficient matrix $\Sigma = (\Sigma_{ij})_{1\le i,j\le N} = (\sigma_i \sigma_j \rho_{ij})_{1\le i,j\le N}$, we can choose Q to be the matrix of eigenvectors of Σ sorted by eigenvalue size[1], i.e.,

$$Q = (q_1, \ldots, q_N), \quad \frac{1}{2}\Sigma q_i = \lambda_i q_i, \quad \lambda_1 \ge \ldots \ge \lambda_N \ge 0, \tag{11}$$

[1] If Σ has eigenvectors with multiplicity larger than 1, then this decomposition is not unique. In that case, we can simply choose any such matrix Q.

and get $(\lambda_{kl})_{1\leq k,l,\leq N} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ a constant diagonal matrix.

If μ does not depend on the spatial coordinates x but only on t, then the difference under the sum in equation (10) vanishes identically and thus $\kappa(x,t) \equiv 0$.

Moreover, if α is also only a function of t, the zero order term can be eliminated from (8) by considering $\exp\left(\int_0^t \alpha(T-u)\,du\right) V$.

If all this is satisfied, then $\mathcal{L}$ simplifies to the N-dimensional heat operator in (12). Keeping the symbol V for the transformed value function and $\mathcal{L}$ for the operator for simplicity, we obtain

$$\frac{\partial V}{\partial t} = \mathcal{L}V = \sum_{k=1}^{N} \lambda_k \frac{\partial^2 V}{\partial x_k^2}, \tag{12}$$

$$V(x,0) = g(x), \tag{13}$$

for $x \in \mathbb{R}^N$, $t \in (0,T)$, $\lambda = (\lambda_1, \ldots, \lambda_N) \in \mathbb{R}^N_+$.

In all other cases, i.e. if Σ is not constant and μ depends on s, a transformation to a diagonal diffusion without drift is generally not possible. By translation to $s = s_0 + \int_0^T \mu(s_0, u)\,du$ and choosing Q as the eigenvectors of $\Sigma(s,T)$, one obtains $\lambda_{kl}(a,0) = 0$ for $k \neq l$ and $\kappa_k(a,0) = 0$, but these coefficients are non-zero for other (x,t).

0.2 Finite difference schemes

In this section, we describe the finite difference schemes used for the one- and two-dimensional versions of (12) and (8) which we will need to construct the dimension-wise splitting introduced in Section 0.3. We choose the Crank-Nicolson scheme for the one-dimensional equations, Brian's scheme [1] for multi-dimensional PDEs without cross-derivatives, and the Hundsdorfer-Verwer (HV) scheme [8] for PDEs with cross-derivative terms. These are established techniques from the literature which are routinely used in financial institutions for derivative pricing, and can be replaced by a method of choice. As such, this section can be skipped without loss of continuity.

We follow standard procedure (see, e.g., [21]) to define a finite difference approximation V_h to V, where $h = (\Delta t, \Delta x_1, \ldots, \Delta x_d)$ contains both the time step size $\Delta t > 0$ and the spatial mesh sizes $\Delta x_i > 0$, $i = 1, \ldots, d$, where d is

the dimension of the PDE. We first define basic finite difference operators

$$\begin{aligned}
\delta_t V_h(\cdot,t) &= \frac{V_h(\cdot,t+\Delta t)-V_h(\cdot,t)}{\Delta t},\\
\delta_x^i V_h(\cdot,t) &= \frac{V_h(\cdot+\Delta x_i,t)-V_h(\cdot-\Delta x_i,t)}{2\Delta x_i},\\
\delta_x^{i,i} V_h(\cdot,t) &= \frac{V_h(\cdot+\Delta x_i,t)-2V_h(\cdot,t)+V_h(\cdot-\Delta x_i,t)}{\Delta x_i^2},\\
\delta_x^{i,j} V_h &= \delta_x^i \delta_x^j V, \qquad i\neq j,
\end{aligned}$$

and then an approximation to $\mathcal{L}$ by

$$L(t) = \sum_{i=1}^{d} \kappa_i(\cdot,t)\,\delta_x^i + \sum_{i,j=1}^{d} \lambda_{ij}(\cdot,t)\,\delta_x^{i,j} - \alpha(\cdot,t),$$

where the operator $\kappa_i(\cdot,t_n)\delta_x^i$, applied to V_h, at a point $x=(x_{j_1},\ldots,x_{j_d})$ is

$$((\kappa_i(\cdot,t_n)\delta_x^i)V_h)_{j_1,\ldots,j_d} = \kappa_i(x,t_n)\frac{V_h(x+\Delta x_i e_i,t_n)-V_h(x-\Delta x_i e_i,t_n)}{2\Delta x_i},$$

e_i the ith unit vector, and similar for the σ and α terms.

Ignoring spatial boundaries for the time being, V_h is defined for all $(x,t)\in \mathbb{R}^d\times\{0,\Delta t,\ldots,T\}$ by the scheme

$$\begin{aligned}
\delta_t V_h &= \theta L(t+\Delta t)V_h(t+\Delta t) + (1-\theta)L(t)V_h(t), \qquad (14)\\
V_h(x,T) &= \phi(x),
\end{aligned}$$

where $\theta\in[0,1]$. Here, $\Delta t = T/N_t$, where N_t is the number of timesteps.

In practice, the scheme and solution need to be restricted to a bounded domain, and for simplicity we restrict ourselves here to a box where $x_{i,min}\le x_i\le x_{i,max}$. These may be given naturally, e.g., $x_{min}=0$ if x is a positive stock price, or by truncation of an infinite interval at suitably large values, e.g., a certain number of standard deviations away from the spot. Then with N_i the number of mesh intervals in coordinate direction x_i, $\Delta x_i = (x_{i,max}-x_{i,min})/N_i$, the mesh points are $x_{i,j}=x_{i,min}+j\Delta x_i$ for $j=0,\ldots,N_i$, $i=1,\ldots,d$. We denote the numerical solution on this mesh by U_n, this being the vector $(V_h((x_{i,j_i})_{i=1,\ldots,N},t_n))_{j_i=0,\ldots,N_i}$.

Let $L^n\equiv L(t_n)$ be the discretisation matrix at time step t_n, then this matrix is first decomposed into

$$L^n = L_0^n + L_1^n + \ldots + L_d^n,$$

where the individual $L_i^n, 1\le i\le d$, contain the contribution to L stemming from the first and second order derivatives in the ith dimension,

$$L_i^n = \kappa_i(\cdot,t_n)\delta_x^i + \lambda_{ii}(\cdot,t_n)\delta_x^{i,i} - \frac{1}{d}\alpha(\cdot,t_n),$$

and, following [9], we define one matrix F_0 which accounts for the mixed derivative terms,

$$L_0^n = \sum_{i \neq j} \lambda_{ij}(\cdot, t_n)\delta_x^{i,j}.$$

For $L_0^n = 0$, which contains the discretisation of (12) as a special case, a simple splitting scheme is given by the Douglas scheme [3],

$$\begin{aligned} Y_0 &= U_{n-1} + \Delta t L^{n-1} U_{n-1}, \\ (I - \theta \Delta t L_j^n) Y_j &= Y_{j-1} - \theta \Delta t L_j^{n-1} U_{n-1}, \qquad j = 1, \ldots, d, \\ U_n &= Y_d. \end{aligned} \tag{15}$$

The scheme is unconditionally stable for all $\theta \geq 1/2$ and of second order in time for $\theta = 1/2$ (otherwise of first order, see [10]).

A second order modification of the above scheme was proposed by Brian [1], where the first two steps are as above with $\theta = 1$ and step size $\Delta t/2$, and the last step (15) is replaced by a Crank-Nicholson-type step

$$\frac{U_n - U_{n-1}}{\Delta t} = \sum_{j=1}^{d} \frac{1}{2}(L_j^n + L_j^{n-1}) Y_j.$$

For $L_0^n \neq 0$, i.e. with cross-derivative terms present as in the general case of (14), second order gets lost and an iteration of the idea is needed. The Hundsdorfer-Verwer (HV) scheme [8],

$$\begin{aligned} Y_0 &= U_{n-1} + \Delta t L^{n-1} U_{n-1}, \\ (I - \theta \Delta t L_j^n) Y_j &= Y_{j-1} - \theta \Delta t L_j^{n-1} U_{n-1}, \qquad j = 1, 2, 3, \\ \widetilde{Y}_0 &= Y_0 + \frac{1}{2}\Delta t \left[L^n Y_3 - L^{n-1} U_{n-1}) \right], \\ (I - \theta \Delta t L_j^n) \widetilde{Y}_j &= Y_{j-1} - \theta \Delta t L_j^n Y_j, \qquad j = 1, 2, 3, \\ U_n &= \widetilde{Y}_3, \end{aligned}$$

defines a second order consistent ADI splitting for all θ, and can be shown to be *von Neumann* stable for $\theta \in \left[\frac{1}{2} + \frac{1}{6}\sqrt{3}, 1\right]$, see [6]. We use $\theta = \frac{1}{2} + \frac{1}{6}\sqrt{3} \approx 0.789$ in the computations.

A severe computational difficulty arises for d larger than approximately three, as the total number of operations is proportional to $N_t N_1 \ldots N_d$, i.e., grows exponentially in the dimension. In the numerical tests, we will use $N_1 = N_2 = 800$ and $N_t = 1000$ for the two-dimensional equations. These involve 6.4×10^8 unknowns. In [16], for a second-order extension, $N_1 = N_2 = N_3 = 500$ and $N_t = 50$ are used for the three-dimensional equations involved, i.e., 6.25×10^9 unknowns. It is clear that within this framework a further increase in the dimension will only be practically feasible by reducing the number of mesh points in each direction and consequently sacrificing accuracy.

0.3 Decomposition methods

In order to accurately approximate derivative prices with $N > 3$ factors, we define an approximate dimension-wise decomposition, in the spirit of anchored-ANOVA decompositions. Here, the starting point a of the transformed process, from (7), serves as an 'anchor'. We show the basic concept in a static setting in Section 0.3.1, and its application to constant and variable coefficient stochastic processes and PDEs in the subsequent sections.

We assume in this section that a suitable rotation and translation (see end of Section 0.1) has taken place, so that

$$\lambda_{ij}(a,0) = 0, \qquad i \neq j, \tag{16}$$
$$\kappa_i(a,0) = 0. \tag{17}$$

We then denote for simplicity

$$\lambda_i(x,t) \equiv \lambda_{ii}(x,t),$$

For brevity, we set $\alpha = 0$ in this section, but the extension to $\alpha \neq 0$ is straightforward.

0.3.1 Anchored-ANOVA decomposition

We follow here [5] to define the anchored-ANOVA decomposition of a function $g : \mathbb{R}^N \to \mathbb{R}$, with a given "anchor" $a \in \mathbb{R}^N$. For a given index set $u \subset \mathcal{N} = \{i : 1 \leq i \leq N\}$, denote by $a\backslash x_u$ the N-vector

$$(a\backslash x_u)_i = \begin{cases} x_i & i \in u, \\ a_i & i \notin u. \end{cases}$$

Then $g_u(a;\cdot)$ defined for all $x \in \mathbb{R}^N$ by $g_u(a;x) = g(a\backslash x_u)$ is a projection of g, where we make the dependence of g_u on the anchor a explicit in the notation.

We proceed to define a difference operator Δ recursively through $\Delta g_\emptyset = g_\emptyset$ and, for $u \neq \emptyset$,

$$\Delta g_u = g_u - \sum_{w \subset u} \Delta g_w = \sum_{w \subseteq u} (-1)^{|w|-|u|} g_w.$$

An exact decomposition of g is then given by the identity

$$g = \sum_{u \subseteq \mathcal{N}} \Delta g_u = \sum_{k=0}^{N} \sum_{|u|=k} \Delta g_u. \tag{18}$$

This enables the definition in [5] of successive dimension-wise approximations to the integral of g by truncation of the series.

0.3.2 Constant coefficient PDEs

We start by considering the N-dimensional heat equation

$$\frac{\partial V}{\partial t} = \mathcal{L}V = \sum_{k=1}^{N} \lambda_k \frac{\partial^2 V}{\partial x_k^2}, \tag{19}$$

$$V(\cdot, 0) = g, \tag{20}$$

with constant λ.

Given an initial-value problem of the form (19) and (20), and an index set $u \subseteq \mathcal{N}$, define a differential operator

$$\mathcal{L}_u = \sum_{k \in u} \lambda_k \frac{\partial^2}{\partial x_k^2},$$

and an approximation V_u of V as the solution to

$$\frac{\partial V_u}{\partial t} = \mathcal{L}_u V_u, \tag{21}$$

$$V_u(\cdot, 0) = g. \tag{22}$$

The definition in (21) is equivalent to saying

$$\frac{\partial V_u}{\partial t} = \mathcal{L}V_u,$$
$$V_u(x, 0) = g(a \backslash x_u),$$

i.e., projecting the initial condition, but it is not normally true that V_u from (21) is the projection of the solution V of (19) in the sense of Section 0.3.1.

From here on, we can proceed as in Section 0.3.1 to set

$$\Delta V_u = = \sum_{w \subseteq u} (-1)^{|w|-|u|} V_w.$$

To approximate V by lower-dimensional functions, we truncate the series in (18) and define

$$V_{0,s} = \sum_{k=0}^{s} \sum_{|u|=k} \Delta V_u = \sum_{k=0}^{s} c_k \sum_{|u|=k} V_u, \tag{23}$$

where c_k are integer constants which depend on the dimension N and s. The point to note is that V_u is essentially a $|u|$-dimensional function as it only depends on the fixed anchor and $|u|$ components of x.

In situations where one or several coordinates play a dominant role, it will be useful to consider a generalisation of (23) to

$$V_{r,s} = \sum_{k=0}^{s} c_k \sum_{|u|=k} V_{u \cup \{1,\ldots,r\}}, \quad r + s \leq N. \tag{24}$$

Here, all components $V_{u \cup \{1,\ldots,r\}}$ depend on all the $x_1, \ldots, x_r$.

0.3.3 Variable coefficients – full freezing

The simplest way to deal with variable coefficients is to"freeze" them at a constant value and then apply the methodology from Section 0.3.2. As we are interested in the PDE solution at the anchor point a, the obvious choice is to approximate κ_i and λ_{ij} by $\kappa_i(a,0)$ and $\lambda_{ij}(a,0)$.

For a given subset $u \subseteq \mathcal{N}$, we then define (note that in this case $\kappa_i(a,0) = 0$ and $\lambda_{ij} = 0$, $i \neq j$)

$$\begin{aligned} \frac{\partial V_u}{\partial t} &= \sum_{i \in u} \lambda_{ii}(a,0) \frac{\partial^2 V_u}{\partial x_i^2}, \\ V_u(x,0) &= g(x). \end{aligned}$$

0.3.4 Partial freezing

The full freezing approximation in Section 0.3.3 throws away more information than needed. In the following extension, we keep as much as possible of the original dynamics of the process in the low-dimensional cross-section the process is restricted to.

For given subset $u \subseteq \mathcal{N}$, we now define

$$\begin{aligned} \frac{\partial V_u}{\partial t} &= \sum_{i \in u} \kappa_i(a \backslash x_u, t) \frac{\partial V_u}{\partial x_i} + \sum_{i,j \in u} \lambda_{ij}(a \backslash x_u, t) \frac{\partial^2 V_u}{\partial x_i \partial x_j}, \\ V_u(x,0) &= g(x). \end{aligned}$$

Given the variability of the coefficients, there is generally no static coordinate transformation which reduces the PDE to the heat equation. The difference to the localized problem in the previous section is that since the PDE coefficients $\lambda(x,t)$ and $\kappa(x,t)$ change with spatial and time coordinates, the PDE will in general contain first order and non-diagonal second order terms.

0.3.5 Partial freezing and zero-correlation approximation

Here, motivated by $\lambda_{ij}(a,0) = 0$ for all $i \neq j$, we make the additional approximation that this holds for all x and t. So we define now

$$\begin{aligned} \frac{\partial V_u}{\partial t} &= \sum_{i \in u} \kappa_i(a \backslash x_u, t) \frac{\partial V_u}{\partial x_i} + \sum_{i \in u} \lambda_{ii}(a \backslash x_u, t) \frac{\partial^2 V_u}{\partial x_i^2}, \\ V_u(x,0) &= g(x). \end{aligned}$$

This extra approximation in addition to Section 0.3.4 does not give any

further dimension reduction, but simplifies the PDEs somewhat, i.e., no cross-derivative terms are present, which simplifies the construction of numerical schemes.

0.4 Theoretical results

In this section, we review the rigorous error analysis from [16] for the constant coefficient case in Section 0.3.2, and give a novel, more heuristic extension of this analysis to the variable coefficient setting of Section 0.3.4.

What is essential in the analysis is clearly the size of the diffusion and drift coefficients in the various directions, as well as the variability of the initial data jointly with respect to different sets of variables. The relevant measure of variability is defined in the following.

Definition 1 *Let*

$$\begin{aligned} C^{j,k,mix} &= \left\{ g \in C^b : \partial_{i_1}^j \dots \partial_{i_k}^j g \in C^b, \, \forall 1 \le i_1 < \dots < i_k \le N \right\}, \\ C^b &= \left\{ g : \mathbb{R}^N \to \mathbb{R} \text{ continuous} : |g(x)| \le c \text{ for all } x \text{ for some } c > 0 \right\}. \end{aligned}$$

The spaces of functions in Definition 1 allow us to measure whether a function is truly multi-dimensional by its cross-derivative with respect to sets of variables. The growth condition ensures well-posedness of the PDE.

0.4.1 Constant coefficients

We follow here [16]. Let $\widehat{V}_{r,s} = V_{r,s} - V$ be the approximation error of $V_{r,s}$ from (24). Then the following holds.

Theorem 2 (Theorems 5 and 14 in [16]) *1. Assume $g \in C^{2,2,mixed}$ in (19–20). Then the expansion error $\widehat{V}_{r,1}$ satisfies*

$$\left\| \widehat{V}_{r,1}(\cdot,t) \right\|_\infty \le t^2 \sum_{r<i<j\le N} \lambda_k \lambda_i \left\| \frac{\partial^4 g}{\partial x_i^2 \partial x_j^2} \right\|_\infty . \tag{25}$$

2. Assume $g \in C^{2,3,mix}$ in (19–20). Then the expansion error $\widehat{V}_{r,2}$ satisfies

$$\left\| \widehat{V}_{r,2}(\cdot,t) \right\|_\infty \le t^3 \sum_{r<i<j<k\le N} \lambda_i \lambda_j \lambda_k \left\| \frac{\partial^6 g}{\partial x_i^2 \partial x_j^2 \partial x_k^2} \right\|_\infty . \tag{26}$$

The analysis in [16] derives PDEs for the error itself, and then makes use of standard maximum principle-type arguments to estimate the size of the error.

For instance, by using the PDEs satisfied by V and $V_{\{1,\ldots,r,i\}}$ for different i, it can be shown that

$$\begin{aligned}\frac{\partial}{\partial t}\widehat{V}_{r,1} &= \mathcal{L}_{\{1,\ldots,r\}}\widehat{V}_{r,1} \\ &\quad + \sum_{i=r+1}^{N}\left[\mathcal{L}_{\{1,\ldots,r,i\}} - \mathcal{L}_{\{1,\ldots,r\}}\right] V_{\{1,\ldots,r,i\}} + \left[\mathcal{L}_{\{1,\ldots,r\}} - \mathcal{L}\right] V \\ &= \sum_{k=1}^{r}\lambda_k \frac{\partial^2}{\partial x_k^2}\widehat{V}_{r,1} + \sum_{k=r+1}^{N}\lambda_k \frac{\partial^2}{\partial x_k^2}\left[V_{\{1,\ldots,r,k\}} - V\right]. \qquad (27)\end{aligned}$$

This is an inhomogeneous heat equation for $\widehat{V}_{r,1}$ with zero initial data and a right-hand side which can be shown to be small. As a consequence, the solution itself is small. Informally, the terms on the right-hand side $V^{\{1,\ldots,r,k\}} - V$ are of order $O(\lambda_{r+1} + \ldots + \lambda_N - \lambda_k)$, and hence the right-hand side is of order $O(\sum_{r<i<j\leq N}\lambda_i\lambda_j)$. A slightly more careful argument gives the precise bound (25), and a similar but lengthier argument for $V_{r,2}$ gives (26).

A number of comments are in order regarding the smoothness requirements dictated by the error bounds. First, most option payoffs are non-smooth, have kinks and discontinuities. This would appear to render (25) and its higher-order versions meaningless. A re-working of the derivation shows that g can actually be replaced by $V_{r,0}$, which is the solution to

$$\begin{aligned}\frac{\partial V_{r,0}}{\partial t} &= \sum_{k=1}^{r}\lambda_k \frac{\partial^2 V_{r,0}}{\partial x_k^2}, \\ V_{r,0}(x,0) &= g(x).\end{aligned}$$

So even if g itself is not smooth, $V_{r,0}$ will be smooth except in degenerate situations which are analysed in detail in [16]. Roughly speaking, as long as the location of kinks and discontinuities is not parallel to all of the first r coordinate axes, $V_{r,0}$ is smooth enough for the expansion error to be well-defined.

The second important point is that as (25) contains only mixed derivative terms, for any payoffs which depend only on, say, x_1 and x_k for some $k > 1$, the decomposition of the option price is exact. Moreover, the value of any derivative that can be statically replicated by options with such simple payoffs is found exactly. Again, a more detailed discussion is found in [16].

0.4.2 Variable coefficients

The transformation (6) with appropriate Q (see the discussion at the end of Section 0.1) ensures (16) and (17) but this is only true at $t = 0$ and $x = a$. However, using arguments similar to [16] and Section 0.4.1, we can still derive a PDE for the expansion error even for non-constant coefficients.

Straightforward calculus yields an expression similar to equation (27), namely

$$\begin{aligned}\frac{\partial}{\partial t}\widehat{V}_{r,1} &= \sum_{k,l=1}^{r}\lambda_{kl}(x,t)\frac{\partial^2}{\partial x_k x_l}\widehat{V}_{r,1}+\sum_{k=1}^{r}\kappa_k\frac{\partial}{\partial x_k}\widehat{V}_{r,1}\\ &\quad+\sum_{k=r+1}^{N}\left[\lambda_{kk}\frac{\partial^2}{\partial x_k^2}+2\sum_{l=1}^{r}\lambda_{kl}\frac{\partial^2}{\partial x_k x_l}\right]\left[V_{\{1,\ldots,r,k\}}-V\right] \quad (28)\\ &\quad-\sum_{k,l=r+1,k\neq l}^{N}\lambda_{kl}\frac{\partial^2}{\partial x_k x_l}V \quad (29)\\ &\quad+\sum_{k=r+1}^{N}\kappa_k\frac{\partial}{\partial x_k}\left[V_{\{1,\ldots,r,k\}}-V\right]. \quad (30)\end{aligned}$$

This equation contains three source terms, which determine the error size:

- The first term, (28), is similar to the source term appearing in the constant coefficient case. It is essentially a restricted differential operator applied to the difference between full and partial solution.
- The second term, (29), consists of the non-diagonal terms not captured at all in the expansion applied to the full solution. It contains the full solution rather than the difference between full and partial ones, but the λ_{kl} involved are zero for $t = 0$ and $x = a$.
- The third term, (30), where $\kappa_k(a, 0) = 0$, captures the changes in κ and again acts on the differences between partial and full solutions.

At $t = 0$ and $x = a$ all three source terms are zeros, because

$$V_{\{1,\ldots,r,k\}}(x,0)-V(x,0)=0 \quad \forall x\in\mathbb{R}^N \text{ and } \lambda_{kl}(a,0)=0, k\neq l.$$

Away from these initial coordinates the terms grow slowly and drive a non-zero error.

Instead of investigating this further theoretically, we give quantitative examples in the next section.

0.5 Numerical examples

In this section, we analyse the numerical accuracy of the decomposition from Section 0.3 for the approximation of European basket options, where the model for the underlying stock has variable coefficients. We list six "base" cases of how the PDE coefficients can be variable in Table 0.1.

TABLE 0.1: Different base cases with non-constant parameters.

Non-constant component	Parameter	Example
Time-dependent drift	$\mu = \mu(t)$	Exactly described by (5)–(6)
Time-dependent volatilities	$\sigma = \sigma(t)$	Sections 0.5.3 and 0.5.4
Time-dependent correlation	$\rho = \rho(t)$	Sections 0.5.1 and 0.5.2
Asset-dependent drift	$\mu = \mu(S)$	LIBOR market model in [17]
Asset-dependent volatilities	$\sigma = \sigma(S)$	Local vol – not considered
Asset-dependent correlation	$\rho = \rho(S)$	Section 0.5.5

Consider assets whose dynamics for the prices of $S_t^1, \ldots, S_t^N$ is given by

$$d(\log S_t^i) = -\frac{1}{2}\sigma_i^2(S_t, t)\, dt + \sigma_i(S_t, t)\, dW_t^i, \quad 1 \le i \le N,$$

under the risk-neutral measure with zero interest rates. By considering log prices as primitive variable in (1), in a Black-Scholes setting, i.e., if σ and ρ are constant, the PDE coefficients are constant. Generally, the Brownian motions W^i are correlated according to the correlation matrix

$$(\rho_{ij}(S, t))_{1 \le i,j \le N}.$$

We consider two possible correlation structures:

$$\rho_{simple}(\gamma) = \begin{pmatrix} 1 & \gamma & \gamma & \cdots & \gamma \\ \gamma & 1 & \gamma & \cdots & \gamma \\ \vdots & & \ddots & \vdots & \\ \gamma & \gamma & \gamma & \cdots & 1 \end{pmatrix}$$

for $\gamma \in (-1, 1)$ and

$$\rho_{exp,ij}(\gamma) = \exp(-\gamma |i - j|)$$

for $\gamma > 0$, where we replace γ by a function $\gamma : \mathbb{R}^N \times [0, T] \to \mathbb{R}$, possibly being asset- and time-dependent. The covariance matrix $\Sigma(S, t)$ is then fully characterised via $\Sigma_{ij}(S, t) = \sigma_i(S, t)\sigma_j(S, t)\rho_{ij}(S, t)$. Due to the asset- and time-dependency of correlations and volatilities, the asset distributions are no longer log-normal and hence a transformation of the pricing PDE to the standard heat equation is generally not possible.

As a test case, we choose a European arithmetic basket option with $N = 10$. The payout at maturity $T = 1$ is

$$h(S) = \max\left(\sum_{i=1}^{N} \omega_i S_i - K, 0\right),$$

with strike $K = 100$ and weights $\omega_i \in \mathbb{R}$, $i = 1, \ldots, N$. We will examine the

value at the point $S_{0,i} = 100$ for all i. As payout weight vectors ω we consider

$$\begin{aligned}\omega^1 &= (1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10, 1/10),\\ \omega^2 &= (4/30, 4/30, 4/30, 4/30, 4/30, 2/30, 2/30, 2/30, 2/30, 2/30),\\ \omega^3 &= (1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, -1/4, -1/4, -1/4).\end{aligned}$$

Using $V_{1,1}$ as approximation to V, we expect that the accuracy will be best for ω^1 and worst for ω^3, because ω^1 is parallel to the principal component of Σ and ω^3 closer to orthogonal.

The numerical parameters chosen were $N_1 = N_2 = 800$ and $N_t = 1000$, corresponding to a time step of size $\Delta t = 0.001$. For the reference Monte Carlo estimator V_{MC} we used 10^8 paths. This set-up reduces the discretization and simulation errors sufficiently for us to determine a good estimate of the expansion method's accuracy.

We implemented and tested two numerical algorithms for the solution of the PDE problems. One algorithm is the diagonal ADI method from Section 0.3.5 (with result denoted by $V_{PDE}^{diagADI}$), where we updated the diffusion coefficient values at every time step, and the PDE is solved numerically by Brian's scheme. The second method from Section 0.3.4 does incorporate the off-diagonal terms in the lower dimensional problems (denoted V_{PDE}^{HV}), where the numerical PDE solution is based on the Hundsdorfer-Verwer (HV) scheme.

We also compute the results for the fully frozen model from Section 0.3.3, i.e. with covariance matrix fixed at $\Sigma(s_0, T)$, both for the expansion (V_{PDE}^{loc}) and a full Monte Carlo estimator (V_{MC}^{loc}). This allows us to understand what contribution to the error comes from the variability of the coefficients, compared to the decomposition error already present for constant coefficients.

Our primary intention here is to give a proof of concept, rather than an in-depth study of the performance and convergence. We want to demonstrate that and how expansion methods can be used for variable coefficients.

0.5.1 Time-dependent simple correlation

For time-dependent simple correlation $\rho(t) = \rho_{simple}(t)$ the eigenvalues change over time. However, the lower $N - 1$ eigenvalues are identical and the subspace spanned by their eigenvectors does not change.

The following Table 0.2 shows results for $\sigma_i = 0.2$ and

$$\rho(t) = \rho_{simple}(0.8 - 0.8 \cdot (t/T - 0.5)^2) \in [\rho_{simple}(0.6), \rho_{simple}(0.8)].$$

PDE/ADI and PDE/HV results were almost identical and very close to the MC results. Only in the third case of ω^3 did they even differ in a statistically significant way, i.e., relative to the standard error σ_{MC}, from the MC computation. It is worth noting that the errors are even slightly larger in the fully frozen case, implying that the variable coefficients present no particular problem in this model.

TABLE 0.2: Time-dependent simple correlation.

		V_{MC}	$V_{PDE}^{diagADI}$	V_{PDE}^{HV}	V_{MC}^{loc}	V_{PDE}^{loc}
ω^1		6.9463	6.9451	6.9451	6.3784	6.3715
	σ_{MC}	0.0011			0.0010	
	Δ_{abs}		-0.0012	-0.0012		-0.0069
	Δ_{rel}		-0.02%	-0.02%		-0.11%
	Δ_{abs}/σ_{MC}		-1.06	-1.06		-6.73
ω^2		6.9602	6.9584	6.9584	6.3991	6.3932
	σ_{MC}	0.0011			0.0010	
	Δ_{abs}		-0.0018	-0.0018		-0.0059
	Δ_{rel}		-0.03%	-0.03%		-0.09%
	Δ_{abs}/σ_{MC}		-1.57	-1.57		-5.75
ω^3		7.5631	7.5816	7.5816	7.3585	7.4069
	σ_{MC}	0.0012			0.0012	
	Δ_{abs}		0.0185	0.0185		0.0484
	Δ_{rel}		0.24%	0.24%		-0.66%
	Δ_{abs}/σ_{MC}		14.96	14.96		-40.58

0.5.2 Time-dependent exponential correlation

For a time-dependent exponential correlation $\rho(t) = \rho_{exp}(t)$, the eigenvalues and eigenvectors change substantially over time, resulting in a significant contribution from non-zero off-diagonal elements in $\lambda(t)$.

Table 0.3 shows results for $\sigma_i = 0.2$ and

$$\rho(t) = \rho_{exp}(0.25 - 0.6 \cdot (t/T - 0.5)^2)) \in [\rho_{exp}(0.1), \rho_{exp}(0.25)].$$

PDE/ADI results are again close to the MC results. The PDE/HV results differ somewhat more, against the expectation, but note that both solutions are significantly more accurate than the constant coefficient approximation. The third case, ω^3, is again the most challenging one for the dimension-wise method.

0.5.3 Time-dependent volatilities, simple correlation

For time-dependent $\sigma_i = \sigma(t)$, i.e., the case where all volatilities are time-dependent but equal, the eigenvalues $\lambda_1, \ldots, \lambda_N$ of Σ are simply scaled up or down over time and the matrix of eigenvectors stays constant. This means that all non-diagonal terms of λ vanish and the transformation to the heat equation is exact. This case is simple: it merely requires the solution of a heat equation with time-dependent diffusion coefficients.

For time-dependent $\sigma_i = \sigma_i(t)$, i.e., the case where the volatilities vary differently over time, the eigenvectors change with t. This in general leads to the appearance of non-zero off-diagonal terms. With no dependency on the asset values S, the initial PDE transformation means that those terms vanish at time $t = 0$ and then grow over time for $t > 0$.

TABLE 0.3: Time-dependent exponential correlation.

		V_{MC}	$V_{PDE}^{diagADI}$	V_{PDE}^{HV}	V_{MC}^{loc}	V_{PDE}^{loc}
ω^1		6.0662	6.0738	6.0590	6.8534	6.8477
	σ_{MC}	0.0010			0.0011	
	Δ_{abs}		0.0076	0.0885		-0.0057
	Δ_{rel}		0.13%	1.46%		-0.08%
	Δ_{abs}/σ_{MC}		7.82	90.88		-5.11
ω^2		6.1646	6.1695	6.1547	6.9109	6.9085
	σ_{MC}	0.0010			0.0011	
	Δ_{abs}		0.0049	-0.0099		-0.0024
	Δ_{rel}		0.08%	-0.16%		-0.03%
	Δ_{abs}/σ_{MC}		4.92	-10.00		-2.15
ω^3		9.6062	9.5346	9.7786	9.2907	9.3279
	σ_{MC}	0.0015			0.0015	
	Δ_{abs}		-0.0716	0.1724		0.0372
	Δ_{rel}		-0.75%	1.80%		-0.40%
	Δ_{abs}/σ_{MC}		-46.34	111.54		24.44

Table 0.4 shows results for $\rho = \rho_{simple}(0.7)$ and

$$\sigma_i(t) = 0.1(1 + t/T)\left(1 + \frac{i-1}{N-1}\right) \in [0.1, 0.2]\left(1 + \frac{i-1}{N-1}\right).$$

Both the PDE/diagonal ADI and PDE/HV results are fairly accurate for the first two test cases. They both struggle with the third one, producing errors of 2.42% and 2.66%. Given that a similar error is present in the fully localized case, i.e., for the model with constant coefficients, we conclude that this error is primarily due to the expansion method being applied to the challenging payout direction ω^3, rather than the non-constant nature of the coefficients.

0.5.4 Time-dependent volatilities, exponential correlation

Table 0.5 shows results for

$$\sigma_i(t) = 0.1(1 + t/T)\left(1 + \frac{i-1}{N-1}\right) \in [0.1, 0.2]\left(1 + \frac{i-1}{N-1}\right)$$

and

$$\rho(t) = \rho_{exp}(0.25 - 0.6 \cdot (t/T - 0.5)^2)) \in [\rho_{exp}(0.1), \rho_{exp}(0.25)].$$

By combining time-dependent volatilities with time-dependent correlation we have created a challenging scenario for our method. The PDE/diagonal ADI approach starts to be insufficient for the more complicated cases, differing by more than 4% for ω^3. The PDE/HV algorithm produces a relatively constant error of about 2% in all three test cases.

Contrasting with the fully frozen approximation it is evident that this is

TABLE 0.4: Time-dependent volatilities, simple correlation.

		V_{MC}	$V_{PDE}^{diagADI}$	V_{PDE}^{HV}	V_{MC}^{loc}	V_{PDE}^{loc}
ω^1		7.7987	7.7947	7.8234	5.1128	5.1123
	σ_{MC}	0.0013			0.0008	
	Δ_{abs}		-0.0040	0.0248		-0.0005
	Δ_{rel}		-0.05%	0.32%		-0.01%
	Δ_{abs}/σ_{MC}		-3.10	19.19		-0.57
ω^2		7.3183	7.3151	7.3416	4.7972	4.7961
	σ_{MC}	0.0012			0.0008	
	Δ_{abs}		-0.0032	0.0233		-0.0011
	Δ_{rel}		-0.04%	0.32%		-0.02%
	Δ_{abs}/σ_{MC}		-2.67	19.39		-1.41
ω^3		6.2074	6.3579	6.3723	4.0555	4.1658
	σ_{MC}	0.0010			0.0006	
	Δ_{abs}		0.1504	0.1649		0.1103
	Δ_{rel}		2.42%	2.66%		-2.72%
	Δ_{abs}/σ_{MC}		150.26	164.72		174.38

TABLE 0.5: Time-dependent volatilities, exp. correlation.

		V_{MC}	$V_{PDE}^{diagADI}$	V_{PDE}^{HV}	V_{MC}^{loc}	V_{PDE}^{loc}
ω^1		6.9951	7.0905	7.1454	5.1602	5.1595
	σ_{MC}	0.0012			0.0008	
	Δ_{abs}		0.0955	0.1503		-0.0007
	Δ_{rel}		1.36%	2.15%		-0.01%
	Δ_{abs}/σ_{MC}		83.12	130.87		-0.86
ω^2		6.5570	6.7953	6.7047	4.8380	4.8382
	σ_{MC}	0.0011			0.0008	
	Δ_{abs}		0.2383	0.1477		0.0002
	Δ_{rel}		3.63%	2.25%		0.00%
	Δ_{abs}/σ_{MC}		223.82	138.71		0.27
ω^3		9.6494	10.0537	9.8383	5.6868	5.7252
	σ_{MC}	0.0015			0.0009	
	Δ_{abs}		0.4042	0.1889		0.0384
	Δ_{rel}		4.19%	1.96%		0.67%
	Δ_{abs}/σ_{MC}		265.62	124.13		43.29

TABLE 0.6: Asset-dependent correlation.

		V_{MC}	$V_{PDE}^{diagADI}$	V_{PDE}^{HV}	V_{MC}^{loc}	V_{PDE}^{loc}
ω^1		6.7937	1.4032	6.7393	7.2138	7.2147
	σ_{MC}	0.0108			0.0010	
	Δ_{abs}		-5.3905	-0.0544		-0.0009
	Δ_{rel}		-79.35%	-0.80%		-0.01%
	Δ_{abs}/σ_{MC}		-497.65	-5.02		-0.75
ω^2		6.7910	6.7910	6.7534	7.2232	7.2239
	σ_{MC}	0.0109			0.0010	
	Δ_{abs}		-5.3660	-0.0376		-0.0008
	Δ_{rel}		-79.02%	-0.55%		-0.01%
	Δ_{abs}/σ_{MC}		-494.23	-3.47		-0.64
ω^3		7.4977	2.4238	7.3838	7.6708	7.6663
	σ_{MC}	0.0122			0.0010	
	Δ_{abs}		-5.0739	-0.1139		0.0045
	Δ_{rel}		-67.67%	-1.52%		0.06%
	Δ_{abs}/σ_{MC}		-416.90	-9.36		3.56

the first scenario in which the variability of the coefficients creates a major contribution to the overall error.

0.5.5 Asset-dependent correlation

Table 0.6 shows results for $\sigma_i = 0.2$ and

$$\rho(S) = \rho_{simple}\left(0.6 + 0.2\exp\left(-\frac{1}{N}\sum_{i}^{N}\frac{|S_i - 100|}{10}\right)\right) \in [\rho_{simple}(0.6), \rho_{simple}(0.8)].$$

Because of the added computational complexity of having to calculate the correlation for every vector of asset values encountered, these calculations were done with 10^6 Monte-Carlo paths, $J = 400$ grid points and $M = 400$ time steps.

Clearly, the PDE/diagonal ADI approach is insufficient and the non-diagonal PDE terms are necessary for the solution. The PDE/HV approach, which incorporates them, correspondingly gives fairly accurate results for ω^1 and ω^2, relative to the MC variance. As before, the accuracy decreases for the ω^3 case, which coincidentally depends only weakly on the chosen correlation dynamics.

0.6 Conclusions

This chapter describes a systematic approach to approximating medium- to high-dimensional PDEs in derivative pricing by a sequence of lower-dimensional PDEs, which are then accessible to state-of-the-art finite difference methods. The splitting is accurate especially in situations where the dynamics of the underlying stochastic processes can be described well by a lower number of components. In such situations, the decomposition can loosely be interpreted as a Taylor expansion with respect to small perturbations in the other directions.

To complement the theoretical analysis of the method in the constant parameter setting in earlier work, we describe here various extensions to variable parameters and analyse their accuracy through extensive numerical tests. Although the examples are necessarily specific, they are chosen to cover a spectrum of effects which occur in derivative pricing applications. As the approximation errors are determined locally by the variability of the solution and the parameters with respect to the different coordinates and time, the examples are to some extent representative of a wider class of situations.

Specifically, we designed test cases where different parameters varied with respect to spatial coordinates and time, and where the payoff varied most rapidly in different directions relative to the principle component of the covariance matrix. Across all cases, the ω^1 case, where the payout vector is parallel to the first eigenvector of Σ, showed the best accuracy, while the ω^3 case showed the worst. This was expected from the theoretical analysis and the results for constant coefficients, see Section 0.4.1.

Overall, our computations demonstrate that expansion methods can in principle be applied in this fashion to some variable coefficient asset models. Higher order methods or other extensions might be necessary to reduce the error sufficiently for real-world financial applications.

Bibliography

[1] P. L. T. Brian. A finite-difference method of higher-order accuracy for the solution of three-dimensional transient heat conduction problems. *AIChE J.*, 7:367–370, 1961.

[2] C. S. L. de Graaf, D. Kandhai, and C. Reisinger. Efficient exposure computation by risk factor decomposition. *arXiv preprint arXiv:1608.01197*, 2016.

[3] J. Douglas. Alternating direction methods for three space variables. *Numer. Math.*, 4(1):41–63, 1962.

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

[5] M. Griebel and M. Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *J. Complex.*, 26(5):455–489, 2010.

[6] T. Haentjens and K. J. in 't Hout. ADI finite difference schemes for the Heston-Hull-White PDE. *Journal of Computational Finance*, 16(1):83–110, 2012.

[7] N. Hilber, S. Kehtari, C. Schwab, and C. Winter. Wavelet finite element method for option pricing in highdimensional diffusion market models. Technical Report 2010–01, SAM, ETH Zürich, 2010.

[8] W. Hundsdorfer and J. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, volume 33. Springer, 2013.

[9] K. J. in 't Hout and S. Foulon. ADI finite difference schemes for option pricing. *Int. J. Numer. Anal. Mod.*, 7(2):303–320, 2010.

[10] K. J. in 't Hout and C. Mishra. Stability of ADI schemes for multidimensional diffusion equations with mixed derivative terms. *Appl. Numer. Math.*, 74:83–94, 2013.

[11] V. Kazeev, O. Reichmann, and C. Schwab. Low-rank tensor structure of linear diffusion operators in the TT and QTT formats. *Linear Algebra Appl.*, 438(11):4204–4221, 2013.

[12] C. C. W. Leentvaar and C. W. Oosterlee. On coordinate transformation and grid stretching for sparse grid pricing of basket options. *J. Comput. Appl. Math.*, 222:193–209, 2008.

[13] M. Musiela and M. Rutkowski. *Martingal Methods in Financial Modelling*. Springer, Berlin, second edition, 2005.

[14] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson. Improved radial basis function methods for multi-dimensional option pricing. *J. Comput. Appl. Math.*, 222(1):82–93, 2008.

[15] C. Reisinger. Asymptotic expansion around principal components and the complexity of dimension adaptive algorithms. In J. Garcke and M. Griebel, editors, *Sparse Grids and Applications*, number 88 in Springer Lectures Notes in Computational Science and Engineering, pages 263–276, 2012.

[16] C. Reisinger and R. Wissmann. Error analysis of truncated expansion solutions to high-dimensional parabolic PDEs. *arXiv preprint arXiv:1505.04639*, 2015.

[17] C. Reisinger and R. Wissmann. Numerical valuation of derivatives in high-dimensional settings via PDE expansions. *J. Comp. Fin.*, 18(4):95–127, 2015.

[18] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. *SIAM J. Sci. Comp.*, 29(1):440–458, 2007.

[19] P. Schröder, T. Gerstner, and G. Wittum. Taylor-like ANOVA-expansion for high dimensional problems in finance. Working paper, 2012.

[20] P. Schröder, P. Schober, and G. Wittum. Dimension-wise decompositions and their efficient parallelization. *Electronic version of an article published in Recent Developments in Computational Finance, Interdisciplinary Mathematical Sciences*, 14:445–472, 2013.

[21] D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method*. Wiley, New York, 2000.