Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 222 (2008) 193–209
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

On coordinate transformation and grid stretching for sparse grid pricing of basket options

C.C.W. Leentvaar, C.W. Oosterlee*

Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, The Netherlands

Received 2 October 2006; received in revised form 5 April 2007

Abstract

We evaluate two coordinate transformation techniques in combination with grid stretching for pricing basket options in a sparse grid setting. The sparse grid technique is a basic technique for solving a high-dimensional partial differential equation. By creating a small hypercube sub-grid in the 'composite' sparse grid we can also determine hedge parameters accurately. We evaluate these techniques for multi-asset examples with up to five underlying assets in the basket.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Option pricing; Multi-asset options; Sparse grids; Coordinate transformation

1. Introduction

The topic of this article is the accurate evaluation of basket option prices and the corresponding hedge parameters with partial differential equations (PDEs). Basket options are exotic options, whose payoff functions are based on more than one underlying asset. As the number of the underlying assets increases, the number of the dimensions increases as well in the multi-dimensional pricing partial differential equation and the size of the discrete problem grows exponentially. It is therefore necessary to use numerical techniques that are based on a relatively small number of grid points but that also maintain a satisfactory accuracy.

The sparse grid method is employed here. It is based on a combination of solutions of smaller-sized problems in order to approximate the full grid solution. This method is one of the key techniques for the numerical solution of high-dimensional partial differential equations. The payoff function of a basket call option is, however, non-differentiable or even discontinuous on any hyper-plane that is typically not parallel to a corresponding low-dimensional grid hyper-plane. Therefore, a straightforward application of the sparse grid method may not work satisfactorily, as the mixed derivatives are not bounded for this type of function which is a necessary requirement for the convergence of sparse grid solutions. We show – especially through numerical experiments – that the combination of coordinate transformation, grid stretching and the use of non-equidistant grids may result in satisfactory accuracy for basket call prices with the sparse grid method. Parts of this work are inspired by the Ph.D. thesis work of C. Reisinger [14] (which

* Corresponding author.
E-mail addresses: c.c.w.leentvaar@tudelft.nl (C.C.W. Leentvaar), c.w.oosterlee@tudelft.nl (C.W. Oosterlee).

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.10.015

is in German), in particular the combination of coordinate transformation, grid stretching and sparse grid technique. We however compare two different coordinate transformations in this paper. A nonlinear transformation from [14] is extended to allow for different continuous dividend yields. We further employ different numbers of grid points per dimension, which is not standard in the sparse grid setting. The computation of the Greeks from each sub-grid in the sparse grid method is not straightforward and therefore we present a technique for doing this accurately. Three types of contracts are computed: a European basket call, a European digital call and a Bermudan basket put.

In Section 2, the basket option is discussed, the governing multi-dimensional Black–Scholes equation is presented with its payoff function, i.e., its final condition, plus boundary conditions. The coordinate transformation and grid stretching are presented in Section 3. Numerical implementation by the use of Kronecker products is described in Section 4, the sparse grid technique in Section 5 and a method to extract the hedge parameters from the sparse grid solution is in Section 6. Finally the numerical experiments and conclusions are in Sections 7 and 8, respectively.

2. Basket call options

A European plain vanilla call option is a contract which gives the holder the right to buy an underlying asset S for a fixed price K at maturity time T, see, for example [2,10]. A basket call option contract gives the holder the right to buy an underlying *basket of assets* for a fixed exercise price K. This type of option belongs to the so-called exotic options. The payoff function of a European basket call is typically based on the weighted sum of the assets $S_1, \ldots S_d$ in the basket, and it reads

$$u(\mathbf{S}, T) = \max\left\{\sum_{k=1}^{d} w_k S_k - K, 0\right\}, \tag{1}$$

where w_k are the percentages or the weights of the assets in the basket and $\mathbf{S} = (S_1, S_2, \ldots, S_d)$ is a vector of d asset prices. To price a basket call option with d underlying assets, the multi-dimensional Black–Scholes partial differential equation is used, as derived in [12,21]

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sum_{k=1}^{d}\sum_{\ell=1}^{d} \rho_{k\ell}\sigma_k\sigma_\ell S_k S_\ell \frac{\partial^2 u}{\partial S_k \partial S_\ell} + \sum_{k=1}^{d} (r - \delta_k)\, S_k \frac{\partial u}{\partial S_k} - ru = 0. \tag{2}$$

In this equation, σ_k is the volatility of asset k, $\rho_{k\ell}$ is the correlation between the assets k and ℓ, r is the risk-free interest rate, δ_k is the continuous dividend yield, t is the time ($0 \leqslant t \leqslant T$) and u is the option price. In this work the underlying asset price dynamics is assumed to be the multi-dimensional geometric Brownian motion.

The PDE (2) is a second-order partial differential equation in d dimensions and the 2 times d boundary conditions are mandatory. As the asset price domain is truncated $S_k \in [0, S_k^{\max}]$, we first of all need a boundary condition at $S_k = 0$. When using the reduced form of Eq. (2), where each coefficient belonging to a derivative with respect to S_k vanishes at $S_k = 0$, a $(d-1)$-dimensional partial differential equation remains at the boundary. This is called the *natural boundary condition* in [14]. In particular, the boundary condition at $S_1 = 0$ or $S_2 = 0$ for a two-asset option is represented by the well-known one-dimensional Black–Scholes equation for a vanilla option.

Also for $S_k = S_k^{\max}$ a boundary condition must be prescribed. If $S_k^{\max}$ is large enough, i.e. $w_k S_k^{\max} \gg K$, a linearity condition can be applied, which means that the option price can be assumed to show a linear growth in that coordinate direction. In this case we set the second derivative with respect to S_k equal to zero at that boundary, as in [18,19]. All other derivatives remain present (including the mixed derivatives). An appropriate size of the truncated domain is important for this boundary condition not to have a negative effect on the option prices at the spot price and/or at the exercise price K.

3. Coordinate transformation

Coordinate transformations – typically – are employed to transform a given PDE into another one whose solution is easier to achieve. In basket option pricing an important reason for using a coordinate transformation is to simplify the payoff function (1). This function is non-differentiable along a hyper-plane $\sum_{k=1}^{d} w_k S_k = K$ in the d-dimensional domain. This plane crosses the Cartesian S_i-grid, which may hamper satisfactory accuracy of the so-called sparse

grid combination technique, described in Section 5. A coordinate transformation from S_k to x_i can be written in the form

$$x_i = f_i\left(S_1, S_2, \ldots, S_d\right), \tag{3}$$

$$S_k = f_k^{-1}\left(x_1, x_2, \ldots, x_d\right). \tag{4}$$

We write $x_i = x_i(\mathbf{S})$ and $S_k = S_k(\mathbf{x})$ where $\mathbf{S}$ and $\mathbf{x}$ are d-dimensional vectors. If the transformations (3) and (4) are applied to the partial differential equation (2), it changes into

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{d}\sum_{j=1}^{d}\alpha_{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d}\beta_i\frac{\partial u}{\partial x_i} - ru = 0, \tag{5}$$

where

$$\alpha_{ij} = \sum_{k=1}^{d}\sum_{\ell=1}^{d} a_{k\ell}\frac{\partial x_i}{\partial S_k}\frac{\partial x_j}{\partial S_\ell}, \tag{6}$$

$$\beta_i = \sum_{k=1}^{d}\sum_{\ell=1}^{d} a_{k\ell}\frac{\partial^2 x_i}{\partial S_k \partial S_\ell} + \sum_{k=1}^{d} b_k\frac{\partial x_i}{\partial S_k}, \tag{7}$$

with $a_{k\ell} = \frac{1}{2}\rho_{k\ell}\sigma_k\sigma_\ell S_k(\mathbf{x})S_\ell(\mathbf{x})$ and $b_k = (r - \delta_k)S_k(\mathbf{x})$.

The new coordinate x_1 is now chosen equal to the basket value

$$x_1 = \sum_{k=1}^{d} w_k S_k. \tag{8}$$

With this coordinate the new payoff function reads

$$u(\mathbf{x}, T) = \max\{x_1 - K, 0\}. \tag{9}$$

This transformed payoff function is only dependent on x_1 and thus non-differentiable in only one coordinate direction. It now makes sense, for example, to use a truncation of this coordinate as in the one-dimensional case presented in [11]

$$x_1^{\max} = K\exp\left(\sqrt{2\sigma^2 T\log 100}\right). \tag{10}$$

We can, however, also safely use $x_1^{\max} = 3K$. With this important first coordinate after transformation, it may be possible to reduce the number of points in the other coordinates, as stated in [14,18].

For the definition of the remaining coordinates, two basic choices are available: via a linear transformation or via a nonlinear, normalized, transformation.

3.1. Linear coordinate transformation

A linear coordinate transformation can be written in the form

$$\mathbf{x} = \mathbf{G}\mathbf{S}, \tag{11}$$

with $\mathbf{G}$ the transformation matrix. The first row of matrix $\mathbf{G}$ is defined by the weights of the basket option. The other coefficients are chosen as follows (see for example [18], Chapter 5)

$$g_{ij} = \begin{cases} -w_j & j = i-1, i \neq 1, \\ w_j & j \neq i-1, i \neq 1 \vee i = 1. \end{cases} \tag{12}$$

Applying (12) to (6) and (7) gives

$$\alpha_{ij} = \sum_{k=1}^{d}\sum_{\ell=1}^{d} a_{k\ell}g_{ik}g_{j\ell}, \qquad \beta_i = \sum_{k=1}^{d} b_k g_{ik}. \tag{13}$$

Note that $\partial^2 x_i/\partial S_k \partial S_\ell = 0$ with this transformation, because x_i is linear in S_k. It is easy to see that coordinate transformation (12) is nonsingular. The boundary conditions transform accordingly. Coordinate x_1 is defined on $[0, x^{\max}]$. At $x_1 = 0$ all asset prices are zero and therefore the option price itself is also set to zero. This is a Dirichlet condition. The linearity condition for x_1 towards infinity remains valid; $x_1^{\max}$ corresponds with $\sum w_k S_k^{\max}$. For the other coordinates $x_i,\ i \neq 1$ we set linearity conditions on both the boundaries, as these transformed coordinates do not have their left-hand boundaries at $x_k = 0$. Therefore, it is not true in general that the coefficients of the particular derivatives vanish, which implies that the use of the linearity conditions both at $x_i = x^{\min}$ as on $x_i = x^{\max}$ with $i > 1$ makes good sense.

3.2. Nonlinear transformation

It is also reasonable to employ a nonlinear transformation, with normalized coordinates $x_j,\ \ j > 1$. By normalization one can guarantee that the transformed coordinate directions remain in a $(d-1)$-dimensional *unit* hyper-cube. This transformation was developed for equally distributed basket put options ($\forall i, j\ w_i = w_j$) in [14,15]. With basket weights w_k included, it reads

$$x_i = \begin{cases} \sum_{k=1}^{d} w_k S_k & i = 1, \\ \dfrac{w_{i-1} S_{i-1}}{\sum_{k=i-1}^{d} w_k S_k} & i > 1. \end{cases} \tag{14}$$

Correspondingly, we find the inverse transformation

$$S_k = \begin{cases} \dfrac{1}{w_1} x_1 x_2 & k = 1, \\ \dfrac{1}{w_k} x_1 x_{k+1} \prod_{j=1}^{k} (1 - x_j) & 1 < k < d, \\ \dfrac{1}{w_d} x_1 \prod_{j=1}^{d} (1 - x_j) & k = d. \end{cases} \tag{15}$$

Again the sum of the weighted assets in the basket is used for the first coordinate. Before the new coefficients (6) and (7) are derived, we define the following function

$$\hat{f}_{ik} := \begin{cases} x_{k+1} \prod_{j=i+1}^{k} (1 - x_j) & i < k < d, \\ \prod_{j=i+1}^{k} (1 - x_j) & i < k = d, \\ x_{k+1} & i = k < d, \\ 1 & i = k = d, \\ 0 & i > k. \end{cases} \tag{16}$$

Using (16), the coefficients (6) are transformed to

$$\alpha_{11} = x_1^2 \sum_{k=1}^{d} \sum_{\ell=1}^{d} \hat{\rho}_{k\ell} \hat{f}_{1k} \hat{f}_{1\ell},$$

$$\alpha_{1j} = x_1 x_j (1 - x_j) \sum_{k=1}^{d} \sum_{\ell=1}^{d} \left(\hat{\rho}_{k,j-1} - \hat{\rho}_{k\ell}\right) \hat{f}_{1k} \hat{f}_{j\ell}, \quad \forall 1 < j \leq d,$$

$$\alpha_{ij} = x_i (1 - x_i) x_j (1 - x_j) \sum_{k=1}^{d} \sum_{\ell=1}^{d} \left(\hat{\rho}_{k\ell} - \hat{\rho}_{i-1,\ell} - \hat{\rho}_{k,j-1} + \hat{\rho}_{i-1,j-1}\right) \hat{f}_{ik} \hat{f}_{j\ell}, \quad \forall 1 < i, j \leq d,$$

with $\hat{\rho}_{k\ell} = \hat{\rho}_{\ell k} = \frac{1}{2}\rho_{k\ell}\sigma_k\sigma_\ell$, and $\alpha_{ij} = \alpha_{ji}$. The coefficients (7) now become

$$\beta_1 = \sum_{k=1}^{d}(r-\delta_k)\hat{f}_{1k},$$

$$\begin{aligned}\beta_i &= x_i(1-x_i)\left(r-\delta_1-\sum_{k=1}^{d}\left((r-\delta_k)\,\hat{f}_{ik}\right)\right)\\ &\quad + x_i(1-x_i)\left(\sum_{\ell=1}^{d}\left(-2\hat{\rho}_{i-1,i-1}x_i+(2x_i-1)(\hat{\rho}_{k,i-1}+\hat{\rho}_{\ell,j-1})+2(1-x_i)\hat{\rho}_{k\ell}\right)\hat{f}_{ik}\hat{f}_{i\ell}\right).\end{aligned}$$

The boundary conditions for x_1 are the same as in the case of the linear transformation. Furthermore, it can be shown that $\alpha_{ij} = 0$ and $\beta_j = 0$ for $x_j = 0$ and $x_j = 1$ with $i \geqslant 1$ and $j > 1$. This means that on these boundaries, the coefficients of the derivatives with respect to x_j vanish and the natural boundary conditions can again be applied.

We will compare the accuracy of basket option prices and hedge parameters after employing one of these grid transformations. In addition we will evaluate the use of grid stretching, as described below.

3.3. Coordinate stretching

After applying one of the two transformation techniques, a non-differentiable payoff (or in the case of a digital, discontinuous) function remains only along the x_1-direction. Analytic grid stretching in this coordinate direction represents a technique, which may cluster grid points in the region of interest and which can improve the accuracy of the solution in the case of a payoff function that is discontinuous [5,18].

The coordinate x_1 can be written as a function of the new coordinate y via the stretching function ψ. We need the derivative of the grid function, ψ', and the second derivative ψ''. Eq. (5) changes to

$$\begin{aligned}&\frac{\partial u}{\partial t}+\sum_{i=2}^{d}\sum_{j=2}^{d}\hat{\alpha}_{ij}\frac{\partial^2 u}{\partial x_i\partial x_j}+\hat{\alpha}_1(\psi'(y))^{-2}\left(\frac{\partial^2 u}{\partial y^2}-\frac{\psi''(y)}{\psi'(y)}\frac{\partial u}{\partial y}\right)\\ &\quad+2\sum_{i=2}^{d}\hat{\alpha}_{i1}\psi'(y)^{-1}\frac{\partial^2 u}{\partial x_i\partial y}+\sum_{i=2}^{d}\hat{\beta}_i\frac{\partial u}{\partial x_i}+\hat{\beta}_1\psi'(y)^{-1}\frac{\partial u}{\partial y}-ru=0.\end{aligned} \tag{17}$$

Note that $\hat{\alpha}_i$ and $\hat{\beta}_i$ are now functions of the vector $(\psi(y), x_2, x_3, \ldots, x_d)^t$. The stretching function used in this paper, from [18] reads

$$\psi(y) = K\left(1+\frac{1}{15}\sinh\left(c_2 y + c_1(1-y)\right)\right), \tag{18}$$

$$c_1 = \sinh^{-1}\left(15\left(x^{\min}-K\right)/K\right), \tag{19}$$

$$c_2 = \sinh^{-1}\left(15\left(x^{\max}-K\right)/K\right), \tag{20}$$

with $15/K$ a reference parameter.

4. Set-up of a matrix

A general form for Eq. (2), (5) or (17) is

$$\frac{\partial u}{\partial t}+\sum_{i=1}^{d}\sum_{j=1}^{d}\alpha_{ij}\frac{\partial^2 u}{\partial x_i\partial x_j}+\sum_{i=1}^{d}\beta_i\frac{\partial u}{\partial x_i}-ru=0, \tag{21}$$

where we focus on the x-grid, for simplicity. For discretization of (21) *Kronecker products*, based on the one-dimensional discrete operators are used to set up the multi-dimensional discrete equation.

4.1. Difference stencils, Kronecker products

Eq. (21) contains three types of derivatives: first, second and the mixed derivatives. The latter will be constructed by the use of a Kronecker product of the difference stencils of two first derivatives. For the other two derivatives, the standard second-order central differences are used. We define a grid with N_i points per coordinate and with $h_i = N_i^{-1}$ as the mesh-size.

To reduce the overall number of grid points, high-order discretization stencils ($\mathbf{O}(h^4)$ for example) would be a choice, but as the final condition is non-differentiable, it is well-known [6,16] that these high-order central differences without substantial enhancements do not result in the desired accuracy, but in at most second-order accuracy. Therefore, we focus here on second-order accuracy for the tensor-product grid discretization (sparse grid accuracy is typically somewhat lower, see Section 5).

For the boundary conditions only the first derivative needs to be discretized, because the second derivative is either zero (linearity condition) or the coefficient in front is zero (natural condition) and thus this derivative vanishes. At the boundary, where we can use a linearity condition as the boundary condition [18,19], we choose a backward difference scheme for the first derivative

$$\left.\frac{\mathrm{d}u}{\mathrm{d}x_i}\right|_1 = \frac{-3u_0 + 4u_1 - u_2}{2h_i} + \mathbf{O}(h_i^2), \tag{22}$$

$$\left.\frac{\mathrm{d}u}{\mathrm{d}x_i}\right|_N = \frac{3u_N - 4u_{N-1} + u_{N-2}}{2h_i} + \mathbf{O}(h_i^2). \tag{23}$$

The Kronecker product [17], defined as in Definition 1, is the basis for the set-up of a matrix arising from a d-dimensional PDE problem. The Kronecker products will be employed based on one-dimensional discretization stencils.

Definition 1. Given matrix $\mathbf{A}$ of size $k \times \ell$ and matrix $\mathbf{B}$ of size $m \times n$, then the *Kronecker product* or tensor product of $\mathbf{A}$ and $\mathbf{B}$ is a matrix $\mathbf{C}$ of size $k \cdot m \times \ell \cdot n$, which has the pattern:

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n_1}B \\ a_{21}B & a_{22}B & \cdots & a_{2n_1}B \\ \vdots & \vdots & & \vdots \\ a_{m_11}B & a_{m_12}B & \cdots & a_{m_1n_1}B \end{pmatrix}.$$

Furthermore, we define the repeated Kronecker product:

$$\bigotimes_{m=1}^{d} A_m := A_1 \otimes A_2 \otimes \cdots \otimes A_d.$$

Kronecker products are associative and non-commutative operations. The order is determined by the subscripts and the associative hierarchy does not matter.

The grid ordering is important when using the Kronecker product. We use the standard lexicographical ordering of the grid points. Consider a two-dimensional grid with 5 points for coordinate 1 and 4 points for coordinate 2. The grid point vectors read $[x_1^{(0)}, x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_1^{(4)}]$ and $[x_2^{(0)}, x_2^{(1)}, x_2^{(2)}, x_2^{(3)}]$, respectively. If we use the Kronecker product (with $\mathbf{e}$ the all-one vector), we obtain

$$\left(\mathbf{e}_{x_2} \otimes \mathbf{x_1}, \mathbf{x_2} \otimes \mathbf{e}_{x_1}\right) = \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} x_1^{(0)} \\ x_1^{(1)} \\ x_1^{(2)} \\ x_1^{(3)} \\ x_1^{(4)} \end{bmatrix}, \begin{bmatrix} x_2^{(0)} \\ x_2^{(1)} \\ x_2^{(2)} \\ x_2^{(3)} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) = \begin{pmatrix} (x_1^{(0)}, x_2^{(0)}) \\ \vdots \\ (x_1^{(4)}, x_2^{(0)}) \\ (x_1^{(0)}, x_2^{(1)}) \\ \vdots \\ (x_1^{(4)}, x_2^{(1)}) \\ \vdots \end{pmatrix}.$$

We see that with the Kronecker products the lexicographical grid ordering is obtained and it is therefore possible to set up a dimension-independent grid generating routine.

For the generation of the difference stencils, the same automatic scheme with these Kronecker products can be used. For example the mixed derivative term in two dimensions can be written as:

$$\frac{\partial^2}{\partial x_1 \partial x_2} = \frac{\partial}{\partial x_2}\left(\frac{\partial}{\partial x_1}\right). \tag{24}$$

In stencil notation, an example of the discretization of the mixed derivative reads

$$\frac{\partial^2}{\partial x_1 \partial x_2}|_{h_1,h_2} \hat{=} \frac{1}{4h_1h_2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} + \mathbf{O}(h_1^2 + h_2^2). \tag{25}$$

We can write, according to (24)

$$\frac{\partial}{\partial x_2} \otimes \frac{\partial}{\partial x_1} = \begin{bmatrix} -1/2h_2 \\ 0 \\ 1/2h_2 \end{bmatrix} \otimes \begin{bmatrix} -1/2h_1 & 0 & 1/2h_1 \end{bmatrix} = \frac{1}{4h_1h_2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \tag{26}$$

which is the same as (25).

If there are N_k points per coordinate, then the multi-dimensional grid function X_i for coordinate i reads

$$X_i = \bigotimes_{m=i}^{d-1} \mathbf{e}_{x_{d+i-m}} \otimes \mathbf{x}_i \otimes \bigotimes_{m=1}^{i-1} \mathbf{e}_{x_{i-m}}, \tag{27}$$

where $\mathbf{e}_{x_i}$ is the all one vector of length $N_i + 1$. The grid points are presented in multi-dimensional representation, $\mathbf{X} = [X_1, X_2, \ldots, X_d]$ of size $\prod_{i=1}^{d}(N_i+1)\times d$, as defined in (27). In Eq. (21), the coefficients α_{ij} and β_i are functions of all grid points. We can now simply evaluate the coefficients α_{ij} as a function of $\mathbf{X}$, and obtain the multi-dimensional representation of the coefficients.

If we want to express a stencil of a derivative with respect to coordinate i in a d-dimensional way, we use

$$\left[\frac{\partial}{\partial x_i}\right]^d = \bigotimes_{m=i}^{d-1} \mathbf{I}_{d+i-m} \otimes \left[\frac{\partial}{\partial x_i}\right]^1 \otimes \bigotimes_{m=1}^{i-1} \mathbf{I}_{i-m}, \tag{28}$$

where $\mathbf{I}_m$ is the identity matrix of size $(N_m + 1) \times (N_m + 1)$ and $[\frac{\partial}{\partial x_i}]^d$ is the finite difference stencil of the first derivative term in Eq. (21) in a d-dimensional representation. $[\frac{\partial}{\partial x_i}]^1$ represents the discretized first derivative for coordinate x_i.

As of the grid ordering, the expression for the mixed derivative $(\partial^2 u/\partial x_i \partial x_j)$ stencil with respect to i for $j > i$ reads

$$\left[\frac{\partial^2}{\partial x_i \partial x_j}\right]^d = \bigotimes_{m=j}^{d-1} \mathbf{I}_{d+j-m} \otimes \left[\frac{\partial}{\partial x_j}\right]^1 \otimes \bigotimes_{m=i+1}^{j-1} \mathbf{I}_{i+j-m} \otimes \left[\frac{\partial}{\partial x_i}\right]^1 \otimes \bigotimes_{m=1}^{i-1} \mathbf{I}_m. \tag{29}$$

By the use of the point-wise row-product $\diamond$, the matrix $\mathbf{A}_h$ in semi-discretized equation (31) reads

$$\mathbf{A}_h = \sum_{i=1}^{d}\sum_{j=1}^{d} \alpha_{ij}(\mathbf{X}) \diamond \left[\frac{\partial^2}{\partial x_i \partial x_j}\right]^d + \sum_{i=1}^{d} \beta_i(\mathbf{X}) \diamond \left[\frac{\partial}{\partial x_i}\right]^d - r \bigotimes_{m=1}^{d} \mathbf{I}_{d+1-m}. \tag{30}$$

After discretizing the derivatives with respect to all x_i we obtain a system

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{u_h}}{\mathrm{d}t} + \mathbf{A_h}\mathbf{u_h} + \mathbf{b_h}(t) = 0, \\ \mathbf{u_h}(\mathbf{x}, T) = \mathbf{u}_h^0. \end{cases} \tag{31}$$

where $\mathbf{u}_h$ is the discrete solution and $\mathbf{A}_h$ the matrix. Vector $\mathbf{b}_h$ is the vector representing the values at the boundary (if the boundary is a Dirichlet condition). For a basket put option, $\mathbf{b_h}(t)$ is dependent on time t. We use the second-order Crank–Nicolson method to integrate (31) in time. In principle one could also employ a stretching in the time direction [7], for improved efficiency but this has not been pursued here. We need to solve a linear system $\mathbf{Mu} = \mathbf{v}$ for every time-step. The matrix $\mathbf{M}$ is a sparse matrix with 2 times d off-diagonals.

5. Sparse grids

Solving Eq. (21) on a tensor-product grid of size $\prod_{i=1}^{\mathrm{d}} N_i$ is an extensive work. Working on this tensor product, the so-called *full*, grid consumes too much memory, when d increases. This is called the curse of dimensionality [1]. For example, the numerical solution of a five-asset option with 32 points per coordinate gives rise to more than 32 million points.

The sparse grid approach, developed in [3,20] is a technique that splits the full grid problem of N^d points up into layers of sub-grids. Each sub-grid represents a coarsening in one or more coordinates up to a minimal required number of points. In the so-called *sparse grid combination technique*, the partial solutions that are computed on these grids, are combined a-posteriori by interpolation to a certain point or region. The sparse grid solution corresponding to a full grid solution on an equidistant grid of size N^d, meaning that the mesh-size in each direction is $h = N^{-1}$, is a combination of d layers, where combination coefficients are determined by Newton's binomial expression.

Consider a d-dimensional problem with mesh-sizes $h_i = N_i^{-1}$ with N_i the number of grid points for coordinate i, $1 \leqslant i \leqslant d$.

Definition 2. A multi-index $\mathcal{I}_d$ belonging to a d-dimensional grid is a collection of numbers $n_i,\ i = 1, \ldots, d$, which represent a d-dimensional grid with N_i grid points in coordinate i, with $N_i = c_i 2^{n_i}$ (c_i some positive constant).

With the aid of the constants c_i, it is possible to construct a non-equidistant grid.

According to Definition 2 the multi-index $\mathcal{I}_d$ of an equidistant full grid with N_l points per coordinate reads $\mathcal{I}_d = \{l, l, \ldots, l\}$, with l the layer number ($c_i = 1$, for example). If $c_1 \neq c_j$ $(=c_A)$, $j > 1$, then a non-equidistant grid of size $(c_1 2^l \times c_A 2^l \times c_A 2^l \times \cdots)$ can be constructed with $\mathcal{I}_d = \{l, l, \ldots, l\}$ and we have to give the vector elements c_i explicitly.

Definition 3. The sum of a multi-index $|\mathcal{I}_d|$ is defined by

$$|\mathcal{I}_d| := \sum_{i=1}^{d} n_i. \tag{32}$$

The full grid solution will be denoted by u_l^f, indicating N_l points per coordinate direction; the sparse grid solution after the combination will be denoted by u_l^c and the exact solution by u_E. Now, we can define [8]

Definition 4. The combined sparse grid solution u_l^c corresponding to a full grid solution u_l^f reads

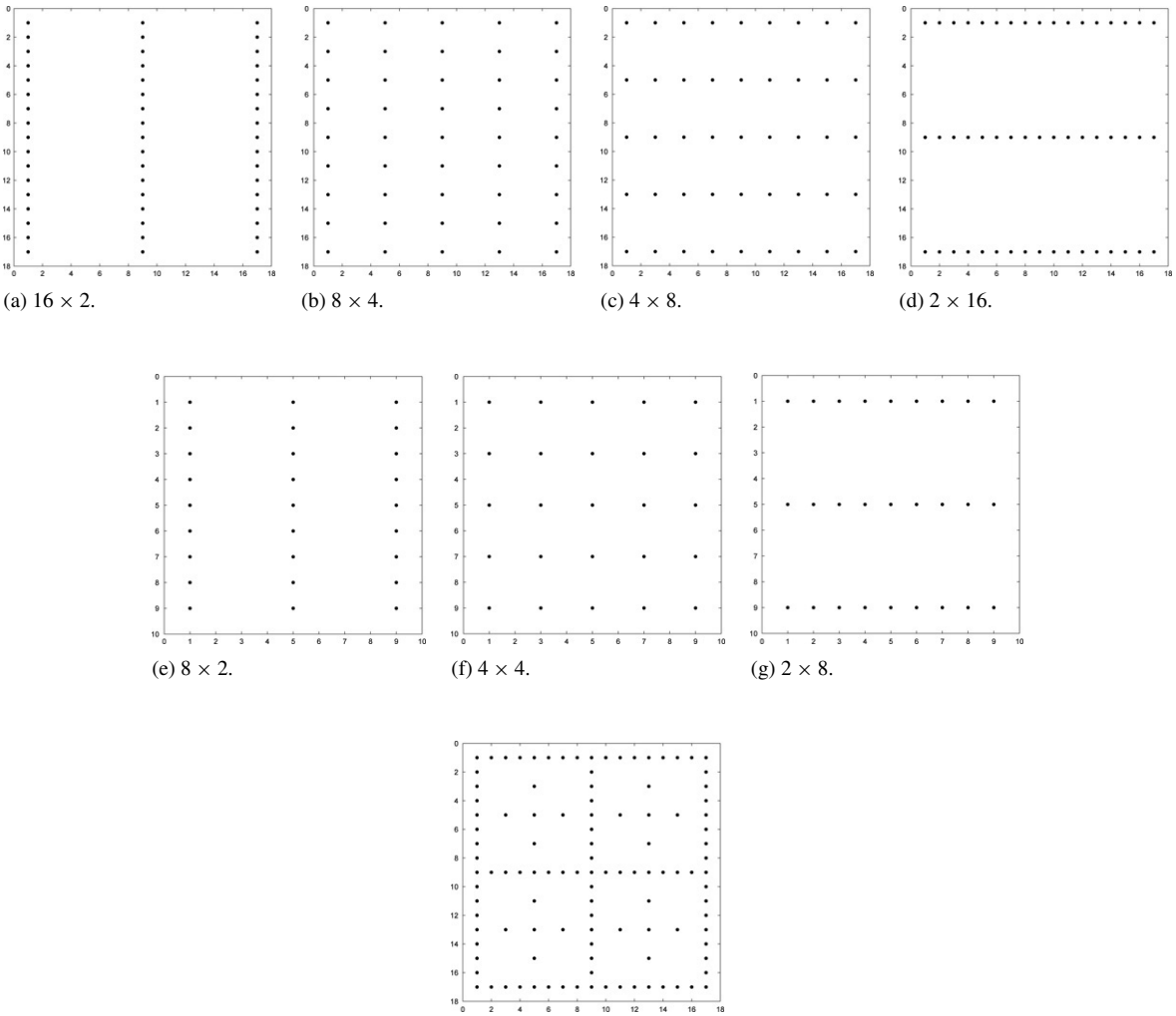
$$u_l^c = \sum_{m=l}^{l+d-1} (-1)^{m+1} \binom{d-1}{m-l} \sum_{|\mathcal{I}_d|=m} u_{\mathcal{I}_d}^f, \tag{33}$$

with $u_{\mathcal{I}_d}^f$ being the solution of the problem on a grid with multi-index $\mathcal{I}_d$ such that $|I_d|$ equals m. This means that the sparse grid solution u_l^c mimics the full grid solution u_l^f.

If the sub-grids are simply combined without any interpolation, which means that all the evaluated points in every sub-grid are added with the binomial coefficients, we obtain the combined solution as schematically depicted in Fig. 1(h) for the two-dimensional case.

The number of points of a d-dimensional problem in a full grid with $n_i = l$ reads

$$N_{\text{full}} = \left(\prod_{i=1}^{d} c_i\right) \left(2^l\right)^d. \tag{34}$$



(a) 16 × 2. (b) 8 × 4. (c) 4 × 8. (d) 2 × 16.

(e) 8 × 2. (f) 4 × 4. (g) 2 × 8.

(h) Combined.

Fig. 1. Construction of a two-dimensional sparse grid; (a)–(d): grids on layer 5, (e)–(g): grids on layer 4; (h) combined sparse grid solution.

From Eq. (33) it follows that the number of problems to be solved in the sparse grid technique reads

$$Z_{l,d} = \sum_{m=l}^{l+d-1} \binom{m-1}{d-1} = \frac{l}{d}\binom{l+d-1}{d-1} - \frac{l-d}{d}\binom{l-1}{d-1}. \tag{35}$$

Furthermore, the number of points employed for a grid with $|I_d| = m$ reads

$$N_{|I_d|=m} = \left(\prod_{i=1}^{d} c_i\right) 2^m. \tag{36}$$

Combining (35) and (36) results in the total number of points employed within the sparse grid technique

$$N_{l,\text{total}} = \sum_{m=l}^{l+d-1} N_{|I_d|=m} \binom{m-1}{d-1} = \left(\prod_{i=1}^{d} c_i\right) \sum_{m=l}^{l+d-1} \binom{m-1}{d-1} 2^m. \tag{37}$$

It is known that the error of the discrete solution from a second-order finite difference scheme of the Laplacian can be split [9] as

$$u_l^f - u_E = C_1(x_1, h_1)h_1^2 + C_1(x_2, h_2)h_2^2 + D(x_1, h_1, x_2, h_2)h_1^2 h_2^2. \tag{38}$$

With the combination technique as in Definition 4 and the splitting in (38), the absolute error, which is dimension-dependent, reads [4],

$$\epsilon_l = |u_l^c - u_E| = \mathbf{O}(h_l^2(\log_2 h_l^{-1})^{d-1}), \tag{39}$$

for the Laplacian. As we use, in the experiments to follow, a higher number of grid points on the coarsest grids by using the vector $\mathbf{c}$, the influence of the logarithmic term in (39) is less pronounced. The convergence ratio in the case where $h_l = c^{-1} \times 2^{-l}$ reads:

$$\frac{\epsilon_l}{\epsilon_{l+1}} = \frac{h_l^2 \left(\log_2 h_l^{-1}\right)^{d-1}}{h_{l+1}^2 \left(\log_2 h_{l+1}^{-1}\right)^{d-1}} = 4\left(\frac{l + \log_2 c}{l + 1 + \log_2 c}\right)^{d-1}. \tag{40}$$

For the non-equidistant case, the value for c taken would be the value for the first coordinate, because this coordinate is dominating. In our numerical experiments we will use a factor of $c = 4$ in the equidistant case and $c = 16$ in the non-equidistant case.

6. Hedge parameters

The Greeks are the derivatives of the option price. In this paper, we concentrate on Δ_k, the first derivative w.r.t. asset price k, $\Gamma_{k,k}$, the second derivative of the price and the correlation parameter $\Gamma_{k,\ell}$, $(k \neq \ell)$, based on the mixed derivative of the price. We use numerical differentiation of the solution originating from the sparse grid combination technique to obtain these Greeks. When using transformation and stretching the equations for Δ_k and $\Gamma_{k,\ell}$ read

$$\Delta_k = \frac{\partial u}{\partial S_k} = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i} \frac{\partial x_i}{\partial S_k}, \tag{41}$$

$$\Gamma_{k,\ell} = \frac{\partial^2 u}{\partial S_k \partial s_\ell} = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i} \frac{\partial^2 x_i}{\partial S_k \partial S_\ell} + \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2 u}{\partial x_i \partial x_j} \frac{\partial x_i}{\partial S_k} \frac{\partial x_j}{\partial S_\ell}. \tag{42}$$

In higher dimensions, we typically do not have the complete solution on the whole domain available, as we work with only a set of sparse grid solutions. The solution of the PDE on a region in the d-domain can, however, be obtained relatively easy by interpolation of the sparse grid solutions.

Consider a point $\mathbf{x} = \mathbf{x}_0$, where we wish to evaluate the option price u, Δ_k and $\Gamma_{k,\ell}$. Then from each sub-grid we interpolate the solution to a part of the finest full grid of size N_R^d. This means a successive interpolation to N_R^d points. The combination of all sub-grids is then straightforward, because we need to combine the interpolated solutions to the part of the finest grid. After this combination of solutions, we apply numerical differentiation for obtaining the Greeks on the relevant part of the finest grid. Schematically this is depicted in Fig. 2.

As we use a higher-order Lagrange interpolation for this purpose, we need $4 \times d$ points adjacent to point $\mathbf{x}_0$. The point $\mathbf{x}_0$ is placed in the middle of the region of interest. On each side of $\mathbf{x}_0$, we thus need two adjacent points for the first and second derivatives and four adjacent points for the mixed derivative.

7. Numerical experiments

The discrete multi-dimensional Black–Scholes equation is solved for European basket call options (Section 7.1), European digital basket call options (Section 7.2) and Bermudan basket put options (Section 7.3) defined on three, four and five assets. The aim here is to evaluate numerically the spatial accuracy achieved by the numerical techniques presented above. In the test experiments, the time-step is fixed at $\delta t = 10^{-3}$. The upper bound of the domain for the asset prices is $S_i^{\max} = 3K$. As in our test experiments we do not use more than 256 points per coordinate direction,

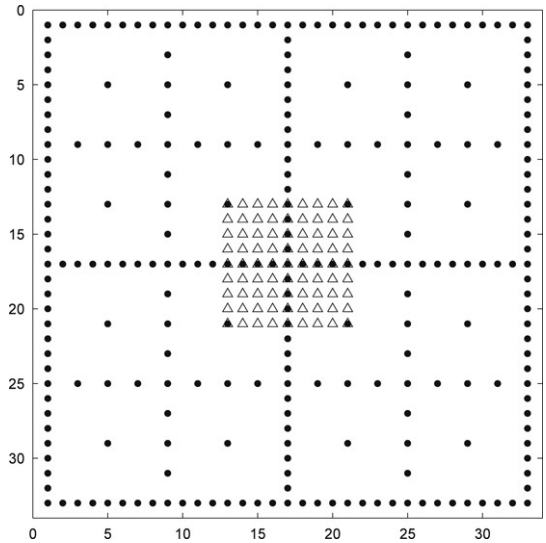


Fig. 2. Representation of the interpolated Ω_R from the sparse grid.

Table 1
Three-asset option with the three formulations on *an equidistant full grid* of $(2^{l+2} \times 2^{l+2} \times 2^{l+2})$, $c_1 = c_2 = c_3 = 4$

l	Eq. (2)		Eqs. (5) and (12)		Eqs. (5) and (14)		#unknowns
	Price	Error	Price	Error	Price	Error	
1	12.8618	3.83×10^{-1}	13.9367	6.92×10^{-1}	13.9042	6.59×10^{-1}	512
2	13.1501	9.48×10^{-2}	13.1957	4.92×10^{-2}	13.1731	7.18×10^{-2}	4 096
3	13.2214	2.35×10^{-2}	13.2355	9.35×10^{-3}	13.2319	1.30×10^{-2}	32 768
4	13.2390	5.85×10^{-3}	13.2416	3.28×10^{-3}	13.2408	4.08×10^{-3}	262 144
5	13.2434	1.46×10^{-3}	13.2441	7.68×10^{-4}	13.2439	9.59×10^{-4}	2 097 152

the error of the time integration, $\mathbf{O}(\delta t^2)$, is negligible compared to the spatial discretization error, $\mathbf{O}(h^2)$. So, in these model experiments we focus on the sparse grid spatial accuracy and neglect the effect of a discretization in time.

7.1. European basket call options

Three-dimensional full grid computations for a three-asset option are used as a reference to evaluate the influence of the coordinate transformation, the grid stretching, the use of fewer points in certain grid directions and the use of sparse grids. For the sparse grid computation, the layer number l is used to compare the sparse and full grid solutions and hedge parameters. Four- and five-asset options are computed with the techniques preferred from the three-asset reference computations. The option parameters used in the experiments can be found in the Appendix. The spot price is chosen to be $S_1 = S_2 = S_3 = K$ (so $\sum_{k=1}^{3} w_k S_k = 1/3 \sum_{k=1}^{3} S_k = K$). The reference value u (spot) $= 13.2449$ is computed by using an accurate FFT-based pricing technique [13]. In Table 1, option prices obtained on a three-dimensional equidistant full grid are presented for the original formulation of the basket option pricing PDE (1), (2) as well as for the two types of transformations (linear and nonlinear). The total number of unknowns employed can be computed with (34) and is shown in the last column of the table. In the experiments with equidistant grids we have set $c_i = 4$, $1 \le i \le 3$. The errors in Table 1 are computed as the absolute error between the computed value and u(spot). We see in Table 1 that the use of only a grid transformation does not lead to improved accuracy on a full grid, as expected. Furthermore, the accuracy improves by a factor 4 with decreasing mesh sizes, which is also expected.

In Table 2, we can observe an interesting improvement in accuracy with the two coordinate transformations when non-equidistant grids are used with $c_1 = 16$ and $c_i = 4$, $i = 2, 3$. Four times fewer points have been used in these tests. Whereas the accuracy without transformation is worse compared to the results in Table 1, the effect of coordinate

Table 2
Three-asset option on a *non-equidistant full grid* of size $(2^{l+4} \times 2^{l+2} \times 2^{l+2})$, $c_1 = 16$, $c_2 = c_3 = 4$

l	Eq. (2)		Eqs. (5) and (12)		Eqs. (5) and (14)		#unknowns
	Price	Error	Price	Error	Price	Error	
1	13.0982	1.47×10^{-1}	13.2406	4.27×10^{-3}	13.2321	1.28×10^{-2}	2 048
2	13.2071	3.78×10^{-2}	13.2427	2.24×10^{-3}	13.2409	4.02×10^{-3}	16 384
3	13.2355	9.38×10^{-3}	13.2444	5.10×10^{-4}	13.2440	9.46×10^{-4}	131 072
4	13.2426	2.34×10^{-3}	13.2448	1.35×10^{-4}	13.2447	2.43×10^{-4}	1048 576

Table 3
Three-asset option with the three formulations on a regular sparse grid, representing a $(2^{l+2} \times 2^{l+2} \times 2^{l+2})$-grid $c_1 = c_2 = c_3 = 4$

l	Eq. (2)		Eqs. (5) and (12)		Eqs. (5) and (14)	
	Price	Error	Price	Error	Price	Error
1	12.8618	3.83×10^{-1}	13.9367	6.92×10^{-1}	13.9042	6.59×10^{-1}
2	13.4397	1.95×10^{-1}	13.1977	4.72×10^{-2}	13.1732	7.17×10^{-2}
3	13.1502	9.47×10^{-2}	13.2368	8.05×10^{-3}	13.2320	1.29×10^{-2}
4	13.3256	8.07×10^{-2}	13.2422	2.73×10^{-3}	13.2409	4.04×10^{-3}
5	13.2297	1.52×10^{-2}	13.2443	6.34×10^{-4}	13.2440	9.47×10^{-4}
6	13.2329	1.20×10^{-2}	13.2447	1.72×10^{-4}	13.2447	2.42×10^{-4}

Table 4
Three-asset option with the two coordinate transformation methods on a non-equidistant sparse grid, representing a $(2^{l+4} \times 2^{l+2} \times 2^{l+2})$-grid, $c_1 = 16$, $c_2 = c_3 = 4$

l	Eqs. (5) and (12)		Eqs. (5) and (14)	
	Price	Error	Price	Error
Without stretching				
1	13.2406	4.27×10^{-3}	13.2321	1.28×10^{-2}
2	13.2426	2.30×10^{-3}	13.2409	4.01×10^{-3}
3	13.2444	5.03×10^{-4}	13.2440	9.40×10^{-4}
4	13.2448	1.39×10^{-4}	13.2447	2.41×10^{-4}
With stretching				
1	13.2648	1.99×10^{-2}	13.2592	1.43×10^{-2}
2	13.2485	3.60×10^{-3}	13.2474	2.52×10^{-3}
3	13.2456	7.18×10^{-4}	13.2453	4.47×10^{-4}
4	13.2449	3.68×10^{-5}	13.2448	9.61×10^{-5}

transformation is positive in this respect. The size of the grid at layer number 3 is $128 \times 32 \times 32$. This solution is comparable to the solution on the $128 \times 128 \times 128$ grid from Table 1.

Results obtained with the sparse grid technique, corresponding to those in Tables 1 and 2, are presented in Table 3, for the equidistant case, and in Table 4, for the non-equidistant finest grids.

We observe the negative effect of the payoff function being not aligned to a grid line on the sparse grid accuracy in the second and third columns of Table 3, where the results with the original non-transformed grid are presented. The need to align the payoff function with a grid line can clearly be observed as the methods based on transformed coordinates show a very satisfactory accuracy.

We further notice that there is no significant difference between the linear and nonlinear coordinate transformations. In Table 4, grid stretching (18) is also included. A slightly better result is observed by using the stretching. It shows, however, that the reduction of grid points in the other (not the first) directions, by the choice for non-equidistant grids, is much more significant than the additional stretching of the first coordinate. The accuracy is dictated by the fewer grid points in the directions 2 and 3. Moreover, a fixed analytic grid stretching does often not place the clustered points at the desired position for accuracy in the Greeks. The Greeks need not have their largest gradients near the exercise price. The errors in the hedge parameters presented in Table 5 are computed as the difference between the

Table 5
Greeks of the three-asset option on a non-equidistant sparse grid

l	Δ_1 (41)	Error	$\Gamma_{1,1}$ (42)	Error	$\Gamma_{1,2}$ (42)	Error
Nonlinear transformation						
3	0.1960		1.5889×10^{-3}		1.4627×10^{-3}	
4	0.1968	8.28×10^{-4}	1.5882×10^{-3}	8.28×10^{-4}	1.4603×10^{-3}	2.41×10^{-6}
5	0.1970	1.71×10^{-4}	1.5880×10^{-3}	6.57×10^{-4}	1.4597×10^{-3}	5.85×10^{-7}
6	0.1970	4.50×10^{-5}	1.5879×10^{-3}	1.26×10^{-4}	1.4596×10^{-3}	1.27×10^{-7}
Nonlinear transformation and stretching						
1	0.1981		1.5817×10^{-3}		1.4542×10^{-3}	
2	0.1973	7.81×10^{-4}	1.5862×10^{-3}	7.88×10^{-4}	1.4580×10^{-3}	3.781×10^{-6}
3	0.1971	1.97×10^{-4}	1.5872×10^{-3}	5.92×10^{-4}	1.4588×10^{-3}	8.06×10^{-7}
4	0.1970	4.92×10^{-5}	1.5874×10^{-3}	1.47×10^{-4}	1.4590×10^{-3}	2.04×10^{-7}

values of two preceding layers, i.e.: $|v_{l+1} - v_l|$ where v_l corresponds to a hedge parameter on layer l. The difference in accuracy between a linear or a nonlinear transformation is negligible. The grid stretching slightly decreases the Greek's accuracies. We conclude that the use of grid stretching does not really pay off in these model examples.

An interesting notion is about the number of grids that we need to evaluate with sparse grids. Because of the choice of the c_i ($c_1 = 16, c_2 = c_3 = 4$), the layer number l can be chosen differently in the Tables 3 and 4 and therefore the number of grids employed is different. The number of grids for the equidistant case using Eq. (35) is 46, whereas for the non-equidistant case it is only 19. This is because in the latter case, the sparse grid evaluation is based on a $32 \times 8 \times 8$-grid rather than on an $8 \times 8 \times 8$-grid. The finest grids in both cases have 2^{14} points and therefore the non-equidistant has a lower complexity than the equidistant case.

For the four- and five-asset option examples discussed next, we evaluate the coordinate transformation with and without grid stretching. The non-equidistant grids are also employed. For the five-asset basket call we focus only on the nonlinear transformation. In the Tables 6 and 7, the results of these two option contracts are presented. The errors are computed as the difference between the option values in the point $S_i = K\ \forall i$ in two preceding layers. We observe that the non-equidistant grid also leads to very satisfactory accuracy here. The determination of the hedge parameters also works fine in higher dimensions. Grid stretching again does not seem to be necessary for obtaining small truncation errors. Note that the reason for a slight decrease in the grid convergence of the four-dimensional and the five-dimensional sparse grid solutions is due to the term $(\log(h_l^{-1}))^{d-1}$ in Eq. (38).

7.2. Digital options

The techniques presented can also be used for pricing options with discontinuous payoff functions. An example is the digital basket option. The payoff function of a digital basket call reads:

$$u(\mathbf{S}, T) = \begin{cases} 1 & \text{if } \sum_{i=1}^{d} w_i S_i > K \\ 0 & \text{elsewhere.} \end{cases} \tag{43}$$

The discontinuity is not aligned with a grid line without transformation. Here, other option parameters are chosen, in particular, a shorter time to maturity (leading to solutions with steep gradients), see the Appendix. In Table 8, the results for a digital basket call are presented. Again, the errors are computed as the difference between the option values in the point $S_i = K\ \forall i$ in two preceding layers. As expected, we observe a lower convergence of $\mathbf{O}(h)$ due to the discontinuity, but the accuracy is still satisfactory. The grid stretching gives the same error convergence, but a more accurate result. It is a helpful technique in the case of solutions with steep gradients.

7.3. Early exercise

We conclude the experiments with a – highly correlated – Bermudan put option. Also the weight parameters in this option are chosen differently (see the Appendix). A Bermudan put gives the holder the right to exercise at

Table 6
Four-asset option price, Δ_1 and $\Gamma_{1,1}$

l	Price	Error	Δ_1(41)	Error	$\Gamma_{1,1}$(42)	Error
Four-dimensional linear, no stretching						
1	13.6720		0.1450		8.6973×10^{-4}	
2	13.6618	1.0266×10^{-2}	0.1455	5.3596×10^{-4}	8.7153×10^{-4}	1.8062×10^{-6}
3	13.6597	2.0656×10^{-3}	0.1457	1.3006×10^{-4}	8.7310×10^{-4}	1.5607×10^{-6}
4	13.6590	6.6128×10^{-4}	0.1457	4.3688×10^{-5}	8.7349×10^{-4}	3.9636×10^{-7}
Four-dimensional linear and stretching						
1	13.6855		0.1464		8.6942×10^{-4}	
2	13.6642	2.1301×10^{-2}	0.1459	4.9735×10^{-4}	8.7176×10^{-4}	2.3415×10^{-6}
3	13.6597	4.4289×10^{-3}	0.1458	1.3399×10^{-4}	8.7288×10^{-4}	1.1237×10^{-6}
4	13.6586	1.1454×10^{-3}	0.1457	3.3631×10^{-5}	8.7313×10^{-4}	2.4849×10^{-7}
Four-dimensional nonlinear						
1	13.6471		0.1450		8.7344×10^{-4}	
2	13.6551	8.0568×10^{-3}	0.1456	5.6144×10^{-4}	8.7362×10^{-4}	1.8754×10^{-7}
3	13.6580	2.8561×10^{-3}	0.1457	1.1662×10^{-4}	8.7363×10^{-4}	7.2636×10^{-9}
4	13.6586	6.4266×10^{-4}	0.1457	3.0644×10^{-5}	8.7365×10^{-4}	1.6688×10^{-8}
Four-dimensional nonlinear and stretching						
1	13.6705		0.1465		8.7008×10^{-4}	
2	13.6605	9.9860×10^{-3}	0.1459	5.7513×10^{-4}	8.7256×10^{-4}	2.4738×10^{-6}
3	13.6588	1.6815×10^{-3}	0.1458	1.4322×10^{-4}	8.7310×10^{-4}	5.4651×10^{-7}
4	13.6584	4.4404×10^{-4}	0.1457	3.5827×10^{-5}	8.7324×10^{-4}	1.3813×10^{-7}

The sparse grid solution mimics a $(2^{l+4} \times 2^{l+2} \times 2^{l+2} \times 2^{l+2})$-grid, $c_1 = 16$, $c_2 = c_3 = c_4 = 4$.

Table 7
Five-asset option price, Δ_1 and $\Gamma_{1,1}$

l	Price	Error	Δ_1 (41)	Error	$\Gamma_{1,1}$ (42)	Error
Nonlinear transformation, no grid stretching						
1	12.6697		0.1176		6.0568×10^{-4}	
2	12.6788	9.1126×10^{-3}	0.1181	5.3724×10^{-4}	6.0556×10^{-4}	1.1569×10^{-7}
3	12.6821	3.2759×10^{-3}	0.1182	1.0939×10^{-4}	6.0549×10^{-4}	7.3138×10^{-8}
4	12.6829	7.4950×10^{-4}	0.1182	2.8927×10^{-5}	6.0548×10^{-4}	8.3213×10^{-9}
Nonlinear transformation and stretching						
1	12.6997		0.1189		6.0329×10^{-4}	
2	12.6863	1.3420×10^{-2}	0.1184	4.9556×10^{-4}	6.0494×10^{-4}	1.6557×10^{-6}
3	12.6838	2.4362×10^{-3}	0.1183	1.2326×10^{-4}	6.0527×10^{-4}	3.3178×10^{-7}
4	12.6832	6.3432×10^{-4}	0.1182	3.0835×10^{-5}	6.0536×10^{-4}	8.3943×10^{-8}

The sparse grid solution mimics a full grid of $(2^{l+4} \times 2^{l+2} \times 2^{l+2} \times 2^{l+2} \times 2^{l+2})$ points, $c_1 = 16$, $c_2 = c_3 = c_4 = c_5 = 4$.

discrete moments prior to the maturity date. In this experiment 10 exercise dates are allowed, which are equally spaced along the duration of the option contract. At each exercise date, the option value is the maximum of the computed value at the current date t_m and the payoff function: $u(\mathbf{S}, t_m) = \max\{u(\mathbf{S}, T), u(\mathbf{S}, t_m)\}$. In Table 9, we present the results of the computation with up to 5 underlying assets. We observe a satisfactory accuracy for all option contracts, although the convergence is irregular. This may be due to the correlation coefficients and early exercise. The nonlinear transformation works well for this case.

Although the results of these experiments give positive results for the use of sparse grids for basket options, it also gives rise to some serious thoughts on the applicability of the sparse grid method. Satisfactory sparse grid accuracy can be achieved for options whose payoff function coincides with a grid line after a coordinate transformation. This may, however, not be easily possible for complex payoff functions, as they are usually encountered in the financial industry. For those there is little hope for satisfactory sparse grid accuracy without any additional enhancements (making the method more complicated).

Table 8
Digital basket call option with 3, 4 and 5 underlying assets

l	Three-dimensional		Four-dimensional		Five-dimensional	
	Price	Error	Price	Error	Price	Error
Nonlinear transformation, no stretching						
1	0.5426		0.5591		0.5680	
2	0.4548	8.7831×10^{-2}	0.4511	1.0802×10^{-1}	0.4411	1.2498×10^{-1}
3	0.4960	4.1248×10^{-2}	0.5004	4.9322×10^{-2}	0.4971	5.6028×10^{-2}
4	0.4757	2.0286×10^{-2}	0.4763	2.4098×10^{-2}	0.4699	2.7174×10^{-2}
Nonlinear transformation, stretching						
1	0.4836		0.4858		0.4806	
2	0.4703	1.3379×10^{-2}	0.4699	1.5939×10^{-2}	0.4627	1.7953×10^{-2}
3	0.4770	6.6910×10^{-3}	0.4779	7.9018×10^{-3}	0.4715	8.8683×10^{-3}
4	0.4803	3.3539×10^{-3}	0.4818	3.9716×10^{-3}	0.4760	4.4551×10^{-3}

The sparse grid solution mimics a full grid of 2^{l+4} points in the first direction and 2^{l+2} in the other directions.

Table 9
10-times exercisable Bermudan basket put option with 3, 4 and 5 underlying assets

Nonlinear transformation, no stretching						
l	Three-dimensional		Four-dimensional		Five-dimensional	
	Price	Error	Price	Error	Price	Error
3	10.3649		10.0422		10.3437	
4	10.3697	4.7423×10^{-3}	10.0463	4.0829×10^{-3}	10.3486	4.8573×10^{-3}
5	10.3702	5.2147×10^{-4}	10.0475	1.1571×10^{-3}	10.3491	4.7073×10^{-4}
6	10.3704	1.8426×10^{-4}	10.0475	2.0129×10^{-5}	10.3495	4.1991×10^{-4}

The sparse grid solution mimics a full grid of 2^{l+4} points in the first direction and 2^{l+2} in the other directions.

8. Conclusion

For pricing basket options with the multi-dimensional Black–Scholes equation a linear or a nonlinear coordinate transformation can be employed, in order to align the payoff function to a grid line. An additional stretching function concentrates points in the region around the exercise price. With the coordinate transformations it is possible to reduce the number of grid points in the $x_i, i > 1$ coordinates, which is highly advantageous. The effect of grid stretching is mainly significant on these non-equidistant grids if the maturity time is short (as then steep gradients in the solution occur). With the coordinate transformation the sparse grid combination technique can be efficiently employed to achieve very satisfactory grid accuracy in space. A significant reduction in the number of sparse grids that need to be processed can be achieved by a clever definition of the base grid. For the model problems evaluated, the difference in the accuracy between the linear or the nonlinear coordinate transformations is not significant. This includes the evaluation of the hedge parameters. Both the linear and the nonlinear transformation perform very well. The nonlinear transformation gives rise to a basket option problem with easier boundary conditions. A critical observation is about the generality of the sparse grid method in multi-asset option pricing. For highly complicated payoff functions that typically cannot be transformed to a low-dimensional hyper-plane the efficient use of sparse grids may be seen with some hesitation. The transformation also works for options with discontinuous payoff functions and options with early exercise.

Acknowledgement

The first author wants to thank the Dutch Technology Foundation STW for financial support.

Appendix. Option parameters

The chosen option parameters in the computation are presented in this appendix.

Standard basket call option

These parameters are used in Section 7.1 for the experiments presented in Tables 1–7.

$$K = 100 \qquad r = 4\% \qquad T = 1$$
$$\sigma = \begin{pmatrix} 0.3 & 0.35 & 0.4 & 0.45 & 0.25 \end{pmatrix}$$
$$\rho = \begin{pmatrix} 1.0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1.0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1.0 \end{pmatrix}$$
$$\delta_i = 0 \qquad w_i = 1/d.$$

Digital basket call option

These parameters are used in Section 7.2 for the experiments presented in Table 8.

$$K = 100 \qquad r = 5\% \qquad T = 0.25 \text{ (=3 months)}$$
$$\sigma = \begin{pmatrix} 0.30 & 0.35 & 0.40 & 0.33 & 0.27 \end{pmatrix}$$
$$\rho = \begin{pmatrix} 1.00 & 0.50 & 0.25 & 0.17 & 0.10 \\ 0.50 & 1.00 & -0.25 & -0.65 & -0.30 \\ 0.25 & -0.25 & 1.00 & 0.50 & 0.45 \\ 0.17 & -0.65 & 0.50 & 1.00 & 0.07 \\ 0.10 & -0.30 & 0.45 & 0.07 & 1.00 \end{pmatrix}$$
$$\delta = \begin{pmatrix} 0.02 & 0.03 & 0.06 & 0.04 & 0.07 \end{pmatrix} \qquad w_i = 1/d.$$

Bermudan basket put option

These parameters are used in Section 7.3 for the experiments presented in Table 9.

$$K = 50 \qquad r = 5\% \qquad T = 0.25$$
$$\sigma = \begin{pmatrix} 0.41 & 0.38 & 0.39 & 0.37 & 0.42 \end{pmatrix}$$
$$\rho = \begin{pmatrix} 1.00 & 0.95 & 0.90 & 0.86 & 0.81 \\ 0.95 & 1.00 & 0.95 & 0.90 & 0.86 \\ 0.90 & 0.95 & 1.00 & 0.95 & 0.90 \\ 0.86 & 0.90 & 0.95 & 1.00 & 0.95 \\ 0.81 & 0.86 & 0.90 & 0.95 & 1.00 \end{pmatrix}$$
$$\delta = \begin{pmatrix} 0.02 & 0.03 & 0.06 & 0.04 & 0.07 \end{pmatrix}$$
$$w_{3D} = \begin{pmatrix} 0.45 & 0.30 & 0.25 \end{pmatrix}$$
$$w_{4D} = \begin{pmatrix} 0.4 & 0.2 & 0.1 & 0.3 \end{pmatrix}$$
$$w_{5D} = \begin{pmatrix} 0.32 & 0.28 & 0.18 & 0.10 & 0.12 \end{pmatrix}.$$

References

[1] R. Bellman, Adaptive Control Processes; A Guided Tour, Princeton University Press, 1961.
[2] T. Björk, Arbitrage Theory in Continuous Time, Oxford University Press, 1998.
[3] H.J. Bungartz, M. Griebel, Sparse Grids, Acta Numerica (2004) 147–269.
[4] H.J. Bungartz, M. Griebel, D. Röschke, C. Zenger, Pointwise convergence of the combination technique for the Laplace equation, East-West Journal of Numerical Mathematics 2 (1994) 21–45.
[5] N. Clarke, K. Parrot, Multigrid for American option pricing with stochastic volatility, Applied Mathematical Finance 6 (1999) 177–179.

[6] M. Gerritsen, P. Olsson, Designing an efficient solution strategy for fluid flows. 1. A stable higher order finite difference scheme and sharp shock resolution, Journal of Computational Physics 129 (1996) 245–262.
[7] M.B. Giles, R. Carter, Convergence analysis of Crank–Nicolson and Rannacher time-marching, Report 05/16, Oxford University, 2005.
[8] M. Griebel, M. Schneider, C. Zenger, A combination technique for the solution of sparse grid problems, in: Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra, Elsevier, Amsterdam, 1992, pp. 263–281.
[9] P. Hofmann, Asymptotic expansions of the discretization error of boundary value problems of the laplace equation in rectangular domains, Numerische Mathematik 9 (1967) 302–322.
[10] J.C. Hull, Options, Futures and Other Derivatives, Prentice-Hall Int. Inc., London, 1989.
[11] R. Kangro, R. Nicolaides, Far field boundary conditions for Black–Scholes equations, SIAM Journal on Numerical Analysis 38 (4) (2000) 1357–1368.
[12] Y.K. Kwok, Mathematical Models of Financial Derivatives, second ed., in: Springer Finance, Springer-Verlag, Singapore, 1998.
[13] R. Lord, F. Fang, F. Bervoets, C.W. Oosterlee, A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes, SSRN. page http://ssrn.com/abstract=966046, 2007.
[14] C. Reisinger, Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben, Ph.D. Thesis, Naturwissenschaftlich-Mathematischen Gesamtfakultät der Ruprecht-Karls-Universität Heidelberg, 2004.
[15] C. Reisinger, G. Wittum, On multigrid for anisotropic equations and variational inequalities: Pricing multi-dimensional European and American options, Computation and Visualisation in Science 7 (2004) 189–197.
[16] A.S. Sabau, P.E. Raad, Comparison of compact and elastical finite difference solutions of stiff problems on nonuniform grids, Computers and Fluids 28 (1999) 361–384.
[17] W. Steeb, Kronecker Product of Matrices and Applications, BI-Wissenschaftsverlag, Mannheim, 1991.
[18] D. Tavella, C. Randall, Pricing Financial Instruments, the Finite Difference Method, Wiley, New York, 2000.
[19] R. Windcliff, P.A. Forsyth, R.A. Vetzal, Analysis of the stability of the linear boundary condition for the Black–Scholes equation, Journal of Computational Finance 8 (2004) 65–92.
[20] C. Zenger, Sparse grids, in: Proceedings of the 6th GAMM Seminar, in: Notes on Numerical Fluid Mechanics, vol. 31, 1990.
[21] Y. Zhu, X. Wu, I. Chern, Derivative Securities and Difference Methods, Springer Verlag, New York, 2004.