DELFT UNIVERSITY OF TECHNOLOGY

REPORT 06-13

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

C.C.W. Leentvaar and C.W. Oosterlee

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2006

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

C.C.W. Leentvaar and C.W. Oosterlee
Delft University of Technology,
Delft Institute of Applied Mathematics,
Mekelweg 4, 2628 CD Delft, The Netherlands.
{c.c.w.leentvaar,c.w.oosterlee}@ewi.tudelft.nl

October 2, 2006

Abstract

We evaluate two coordinate transformation techniques in combination with a coordinate 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.

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 is based on more than one underlying asset. As the number of the underlying assets increases, the number of the dimensions also increases 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 which 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 the key technique for the numerical solution of high dimensional partial differential equations. We would like to point out that the pay-off function of a basket call option is non-differentiable 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. The numerical experiments show the influence of analytic grid stretching with coordinate transformations and non-equidistant grids (i.e., with different number of fine grid points in each coordinate direction) on the accuracy of basket option prices and hedge parameters. We show -especially through numerical experiments- that the combination of coordinate transformation 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 PhD. thesis work of C. Reisinger [12].

In Section 2, the basket option is discussed, the governing multi-dimensional Black-Scholes equation is presented with its pay-off, 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 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, 9]. A basket call option contract, on the other hand, 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 pay-off of a European basket call is typically based on the weighted sum of the assets $S_1, \dots S_d$ in the basket, and it reads

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

where w_k are the percentages or the weights of the assets in the basket and $\mathbf{S} = (S_1, S_2, \dots, S_d)$ 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 [11, 19]

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

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.

Equation (2) is an anti-diffusion-type equation. However, it comes with equation (1), which is a final condition and therefore the problem is well-posed. The PDE (2) is a second order partial differential equation in d dimensions and $2d$ 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 equation (2), where every 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 [12]. In particular, the boundary condition at $S_1 = 0$ or $S_2 = 0$ for a two-asset option is represented by the well-known 1D 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. $w_k S_k^{\max} >> K$, 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 [16, 17]. 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 pay-off 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 sparse grid 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

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

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

Without loss of generality 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 pay-off condition reads

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

This transformed pay-off 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 1D case presented by Kangro [10]

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

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 [12, 16].

For the definition of the remaining coordinates, two basic choices are available: via a linear transformation or via a non-linear, 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 [16], 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

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

Note that $\partial^2 x_i / \partial S_k \partial S_\ell = 0$ with this transformation, because x_i is linear in S_k. The coordinate transformation must be non-singular. It is easy to see that (12) is non-singular, as it can be transformed to

$$\begin{pmatrix} w_1 & w_2 & \dots & w_{d-1} & w_d \\ -w_1 & w_2 & \dots & w_{d-1} & w_d \\ w_1 & -w_2 & \dots & w_{d-1} & w_d \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_1 & w_2 & \dots & -w_{d-1} & w_d \end{pmatrix} \rightarrow \begin{pmatrix} w_1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 2w_d \\ 0 & -2w_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -2w_{d-1} & 0 \end{pmatrix},$$

which is non-singular.

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 the both 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 Non-linear transformation

It is also reasonable to employ a non-linear 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 [12, 13]. 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

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

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

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

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

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

$$\beta_i = x_i (1 - x_i) \left(r - \delta_1 - \sum_{k=1}^{d} \left((r - \delta_k) \hat{f}_{ik} \right) \right) +$$

$$+ x_i (1 - x_i) \left(\sum_{\ell=1}^{d} (-2 \hat{\rho}_{i-1,i-1} x_i + (2x_i - 1)(\hat{\rho}_{k,i-1} + \hat{\rho}_{\ell,j-1}) + 2(1 - x_i) \hat{\rho}_{k\ell}) \hat{f}_{ik} \hat{f}_{i\ell} \right).$$

It follows that if $\delta_i = 0$ (i = 1,...,d), $\beta_1 = r$ and the first term for each β_i, $i > 1$ vanishes, because $\sum_{k=1}^{d} \hat{f}_{ik} = 1$.

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.

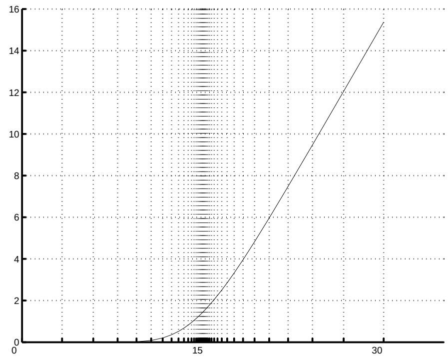


Figure 1: Solution of a vanilla call on a stretched grid ($K = 15$)

3.3 Coordinate stretching

After applying one of the two transformation techniques, a non-differentiable payoff condition remains only along the x_1-direction. Analytic grid stretching in a coordinate direction represents a technique, which may cluster grid points in the region of interest. In this way, stretching can improve the accuracy of the solution in the case of a payoff that is not differentiable [5,16]. The technique employed here relies on a 1D grid stretching along coordinate x_1 in the form of a bijective function ψ. With this function, coordinates on the *computational grid*, called y here, are equally spaced. They are related to non-equally distributed points on the x_k-grid. The center of grid stretching is placed in the region where $\sum_{k=1}^{d} w_k S_k = K$. The number of points decreases towards the domain boundaries.

Grid x can be written as a function of the new coordinate y via ψ. We need the derivative of the grid function, ψ', and the second derivative, ψ''. with stretching only in x_1, equation (5) changes to

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

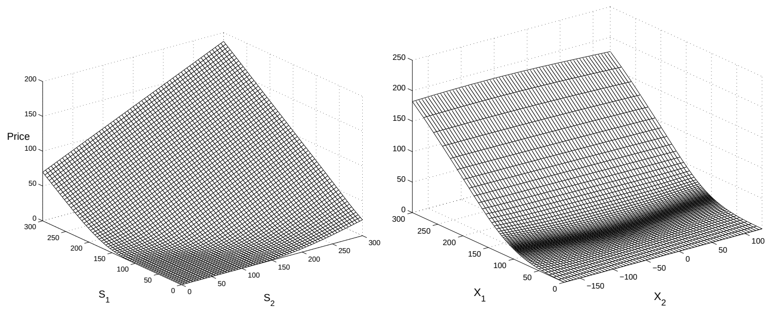
Note that $\hat{\alpha}_i$ and $\hat{\beta}_i$ are now functions of the vector $(\psi(y), x_2, x_3, \ldots, x_d)^t$. The stretching function used in this paper, from [16] reads

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

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

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

with $15/K$ a reference parameter. For a one-dimensional call option pricing problem, the influence of this stretching function on a grid is depicted in Figure 1. For a two-dimensional call the influence of the grid stretching on a transformed grid is depicted in Figure 2.



(a) No transformation, no stretching (b) Transformed and stretched grid

Figure 2: Effect of transformation of a solution of a two-asset option

4 Set-Up of a Matrix

A general form for equations (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

Equation (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 [6,14] 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 [16,17], we choose a backward difference scheme for the first derivative

$$\frac{du}{dx_i}|_1 = \frac{-3u_0 + 4u_1 - u_2}{2h_i} + \mathbf{O}(h_i^2), \tag{22}$$

$$\frac{du}{dx_i}|_N = \frac{3u_N - 4u_{N-1} + u_{N-2}}{2h_i} + \mathbf{O}(h_i^2). \tag{23}$$

The Kronecker product [15], 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 1D discretization stencils.

Definition 1

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

In Definition 2 we define a sequence of Kronecker products similarly to a summation or a product.

Definition 2 *The repeated Kronecker product is defined by:*

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

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 2D 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

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

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.

4.1.1 Example

Now we concentrate on the discretization matrix. Consider the two-dimensional Laplacian (Δ). In stencil notation, it reads

$$\frac{\partial^2}{\partial x_1^2}|_{h_1} + \frac{\partial^2}{\partial x_2^2}|_{h_2} \triangleq \begin{bmatrix} & 1/h_2^2 & \\ 1/h_1^2 & -2/h_1^2 - 2/h_2^2 & 1/h_1^2 \\ & 1/h_2^2 & \end{bmatrix} + \mathbf{O}(h_1^2 + h_2^2). \tag{24}$$

If we consider each derivative separately, then with the lexicographical grid ordering, the corresponding stencils read

$$\frac{\partial^2}{\partial x_1^2}|_{h_1} \triangleq \begin{bmatrix} 1/h_1^2 & -2/h_1^2 & 1/h_1^2 \end{bmatrix} + \mathbf{O}(h_1^2), \quad \frac{\partial^2}{\partial x_2^2}|_{h_2} \triangleq \begin{bmatrix} 1/h_2^2 \\ -2/h_2^2 \\ 1/h_2^2 \end{bmatrix} + \mathbf{O}(h_2^2). \tag{25}$$

The stencil notation for the identity matrix is $\begin{bmatrix}0 & 1 & 0\end{bmatrix}$ for $\mathbf{I}_1$ and $\begin{bmatrix}0\\1\\0\end{bmatrix}$ for $\mathbf{I}_2$. We now find the following stencil with the Kronecker products

$$\mathbf{I}_2 \otimes \frac{\partial^2}{\partial x_1^2}|_{h_1} = \begin{bmatrix}0\\1\\0\end{bmatrix} \otimes \begin{bmatrix}1/h_1^2 & -2/h_1^2 & 1/h_1^2\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\ 1/h_1^2 & -2/h_1^2 & 1/h_1^2\\ 0 & 0 & 0\end{bmatrix}, \tag{26}$$

$$\frac{\partial^2}{\partial x_2^2}|_{h_2} \otimes \mathbf{I}_1 = \begin{bmatrix}1/h_1^2\\ -2/h_1^2\\ 1/h_1^2\end{bmatrix} \otimes \begin{bmatrix}0 & 1 & 0\end{bmatrix} = \begin{bmatrix}0 & 1/h_2^2 & 0\\ 0 & -2/h_2^2 & 0\\ 0 & 1/h_2^2 & 0\end{bmatrix}. \tag{27}$$

Adding (26) and (27) results in (24). The Kronecker product technique also works in higher dimensions, for other derivative stencils, with different boundary conditions etc. In particular, this method is useful for the mixed derivative. The mixed derivative can be written as

$$\frac{\partial^2}{\partial x_1 \partial x_2} = \frac{\partial}{\partial x_2}\left(\frac{\partial}{\partial x_1}\right). \tag{28}$$

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

$$\frac{\partial^2}{\partial x_1 \partial x_2}|_{h_1,h_2} \triangleq \frac{1}{4h_1h_2}\begin{bmatrix}1 & 0 & -1\\ 0 & 0 & 0\\ -1 & 0 & 1\end{bmatrix} + \mathbf{O}(h_1^2 + h_2^2). \tag{29}$$

We can write, according to (28)

$$\frac{\partial}{\partial x_2} \otimes \frac{\partial}{\partial x_1} = \begin{bmatrix}-1/2h_2\\ 0\\ 1/2h_2\end{bmatrix} \otimes \begin{bmatrix}-1/2h_1 & 0 & 1/2h_1\end{bmatrix} = \frac{1}{4h_1h_2}\begin{bmatrix}1 & 0 & -1\\ 0 & 0 & 0\\ -1 & 0 & 1\end{bmatrix}, \tag{30}$$

and again the Kronecker product leads to an elegant construction.

4.2 General dimensions

With the examples in Section 4.2.1 in mind and with Definitions 1 and 2 we can construct the multi-dimensional grid representation and difference molecules. If there are N_k points per coordinate, then the multi-dimensional grid function X_i for coordinate i reads

$$X_i = \bigotimes_{m=i}^{d-1} \mathbf{e}_{x_{d+i-m}} \otimes \mathbf{x}_i \otimes \bigotimes_{m=1}^{i-1} \mathbf{e}_{x_{i-m}}, \tag{31}$$

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 (31). In equation (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

$$\left[\frac{\partial}{\partial x_i}\right]^d = \bigotimes_{m=i}^{d-1} \mathbf{I}_{d+i-m} \otimes \left[\frac{\partial}{\partial x_i}\right]^1 \otimes \bigotimes_{m=1}^{i-1} \mathbf{I}_{i-m}, \tag{32}$$

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

$$\left[\frac{\partial^2}{\partial x_i \partial x_j}\right]^d = \bigotimes_{m=j}^{d-1} \mathbf{I}_{d+j-m} \otimes \left[\frac{\partial}{\partial x_j}\right]^1 \otimes \bigotimes_{m=i+1}^{j-1} \mathbf{I}_{i+j-m} \otimes \left[\frac{\partial}{\partial x_i}\right]^1 \otimes \bigotimes_{m=1}^{i-1} \mathbf{I}_m. \quad (33)$$

By the use of the point-wise row-product $\diamond$, the matrix $\mathbf{A}_h$ in semi-discretized equation (35) reads

$$\mathbf{A}_h = \sum_{i=1}^{d}\sum_{j=1}^{d} \alpha_{ij}(\mathbf{X}) \diamond \left[\frac{\partial^2}{\partial x_i \partial x_j}\right]^d + \sum_{i=1}^{d} \beta_i(\mathbf{X}) \diamond \left[\frac{\partial}{\partial x_i}\right]^d - r \bigotimes_{m=1}^{d} \mathbf{I}_{d+1-m}. \quad (34)$$

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

$$\begin{cases} \dfrac{d\mathbf{u_h}}{dt} + \mathbf{A_h}\mathbf{u_h} + \mathbf{b_h}(t) = 0, \\ \mathbf{u_h}(\mathbf{x}, T) = \mathbf{u}_h^0. \end{cases} \quad (35)$$

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 (35) in time. We need to solve a linear system $\mathbf{Mu} = \mathbf{v}$ for every time step. Matrix $\mathbf{M}$ is a sparse matrix with a tridiagonal-plus-$2(d-1)$-off-diagonals structure. Furthermore, the matrix is neither symmetric nor positive definite.

5 Sparse Grids

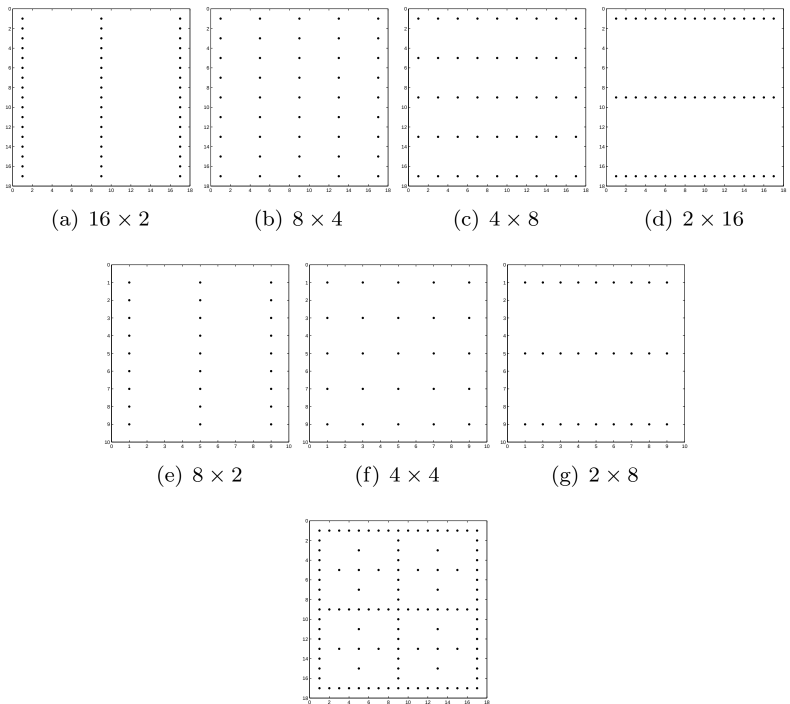
Solving equation (21) on a tensor-product grid of size $\prod_{i=1}^{d} N_i$ is an extensive work. Working on this tensor-product, so-called *full*, grid consumes too much memory, when d increases. This is called the curse of dimensionality [1]. For example, the numerical solution of a five asset option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

The sparse grid approach, developed by Zenger and co-workers [3, 18] 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 3 *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 3 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 is called 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 $(c_1 2^l \times c_A 2^l \times c_A 2^l \times \ldots)$ can be constructed with $\mathcal{I}_d = \{l, l, \ldots, l\}$ and we have to give the vector elements c_i explicitly.



(a) 16×2 (b) 8×4 (c) 4×8 (d) 2×16

(e) 8×2 (f) 4×4 (g) 2×8

(h) Combined

Figure 3: Construction of a 2D sparse grid; (a)–(d): grids on layer 5, (e)–(g): grids on layer 4; (h) combined sparse grid solution

Definition 4 *The sum of a multi-index $|\mathcal{I}_d|$ is defined by*

$$|\mathcal{I}_d| := \sum_{i=1}^{d} n_i. \tag{36}$$

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 [7]

Definition 5 *The combined sparse grid solution u_l^c corresponding to a full grid solution u_l^f reads*

$$u_l^c = \sum_{m=l}^{l+d-1} (-1)^{m+1} \binom{d-1}{m-l} \sum_{|\mathcal{I}_d|=m} u_{\mathcal{I}_d}^f, \tag{37}$$

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 sparse grid solutions as schematically depicted in Figure 3(h) for the 2D case.

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

$$N_{full} = \left(\prod_{i=1}^{d} c_i\right) \left(2^l\right)^d. \tag{38}$$

From equation (37) it follows that the number of problems to be solved in the sparse grid technique reads

$$Z_{l,d} = \sum_{m=l}^{l+d-1} \binom{m-1}{d-1} = \frac{l}{d}\binom{l+d-1}{d-1} - \frac{l-d}{d}\binom{l-1}{d-1}. \tag{39}$$

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

$$N_{|I_d|=m} = \left(\prod_{i=1}^{d} c_i\right) 2^m. \tag{40}$$

Combining (39) and (40) results in the total number of points employed within the sparse grid technique

$$N_{l,total} = \sum_{m=l}^{l+d-1} N_{|I_d|=m} \binom{m-1}{d-1} = \left(\prod_{i=1}^{d} c_i\right) \sum_{m=l}^{l+d-1} \binom{m-1}{d-1} 2^m. \tag{41}$$

It is known that the error of the discrete solution from a second order finite difference scheme of the Laplacian can be split [8] as

$$u_l^f - u_E = C_1(x_1,h_1)h_1^2 + C_1(x_2,h_2)h_2^2 + D(x_1,h_1,x_2,h_2)h_1^2h_2^2. \tag{42}$$

With the combination technique as in Definition 5 and the splitting in (42), the absolute error, which is dimension-dependent, reads [4],

$$\epsilon_l = |u_l^c - u_E| = \mathbf{O}(h_l^2 \left(\log_2 h_l^{-1}\right)^{d-1}), \tag{43}$$

for the Laplacian.

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

$$\Delta_k = \frac{\partial u}{\partial S_k} = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}\frac{\partial x_i}{\partial S_k}, \tag{44}$$

$$\Gamma_{k,\ell} = \frac{\partial^2 u}{\partial S_k \partial s_\ell} = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}\frac{\partial^2 x_i}{\partial S_k \partial S_\ell} + \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2 u}{\partial x_i \partial x_j}\frac{\partial x_i}{\partial S_k}\frac{\partial x_j}{\partial S_\ell}. \tag{45}$$

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 easily 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

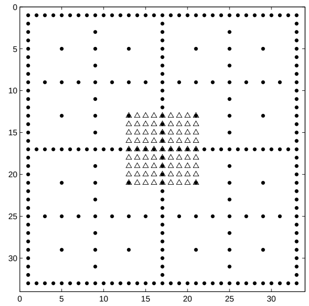


Figure 4: Representation of the interpolated Ω_R from the sparse grid

of solutions, we apply numerical differentiation for obtaining the Greeks on the relevant part of the finest grid. Schematically this is depicted in Figure 3.

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 options 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 $\delta t = 10^{-3}$. As in our test experiments we do not use more than 256 points per coordinate direction, 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.

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.

For the three-asset basket option, we use the following parameters:

- $K = 100$, $r = 0.04$, $T = 1$,
- $\delta_1 = \delta_2 = \delta_2 = 0$,
- $\sigma_1 = 0.3$, $\sigma_2 = 0.35$, $\sigma_3 = 0.4$, $\rho_{12} = \rho_{13} = \rho_{23} = 0.5$,
- $w_1 = w_2 = w_3 = 1/3$,
- u(spot)=13.2449.

The spot price is chosen to be $S_1 = S_2 = S_3 = K$ (so $\sum_{k=1}^{3} w_k S_k = K$). In Table 1, option prices obtained on a 3D 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 (38) and is shown in the last column of the table. In the experiments with equidistant grids we have set $c_i = 4$, $1 \leq i \leq 3$. We see in Table

	Eq (2)		Eq (5) and (12)		Eq (5) and (14)		
l	Price	Error	Price	Error	Price	Error	#unknowns
1	12.8618	$3.83 \cdot 10^{-1}$	13.9367	$6.92 \cdot 10^{-1}$	13.9042	$6.59 \cdot 10^{-1}$	512
2	13.1501	$9.48 \cdot 10^{-2}$	13.1957	$4.92 \cdot 10^{-2}$	13.1731	$7.18 \cdot 10^{-2}$	4096
3	13.2214	$2.35 \cdot 10^{-2}$	13.2355	$9.35 \cdot 10^{-3}$	13.2319	$1.30 \cdot 10^{-2}$	32768
4	13.2390	$5.85 \cdot 10^{-3}$	13.2416	$3.28 \cdot 10^{-3}$	13.2408	$4.08 \cdot 10^{-3}$	262144
5	13.2434	$1.46 \cdot 10^{-3}$	13.2441	$7.68 \cdot 10^{-4}$	13.2439	$9.59 \cdot 10^{-4}$	2097152

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

	Eq (2)		Eq (5) and (12)		Eq (5) and (14)		
l	Price	Error	Price	Error	Price	Error	#unknowns
1	13.0982	$1.47 \cdot 10^{-1}$	13.2406	$4.27 \cdot 10^{-3}$	13.2321	$1.28 \cdot 10^{-2}$	2048
2	13.2071	$3.78 \cdot 10^{-2}$	13.2427	$2.24 \cdot 10^{-3}$	13.2409	$4.02 \cdot 10^{-3}$	16384
3	13.2355	$9.38 \cdot 10^{-3}$	13.2444	$5.10 \cdot 10^{-4}$	13.2440	$9.46 \cdot 10^{-4}$	131072
4	13.2426	$2.34 \cdot 10^{-3}$	13.2448	$1.35 \cdot 10^{-4}$	13.2447	$2.43 \cdot 10^{-4}$	1048576

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

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 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 pay-off 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

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

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 $c_1 = c_2 = c_3 = 4$

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

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, $c_1 = 16, c_2 = c_3 = 4$

	Non-linear transformation					
l	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error	$\Gamma_{1,2}$ (45)	Error
3	0.1960		$1.5889 \cdot 10^{-3}$		$1.4627 \cdot 10^{-3}$	
4	0.1968	$8.28 \cdot 10^{-4}$	$1.5882 \cdot 10^{-3}$	$8.28 \cdot 10^{-4}$	$1.4603 \cdot 10^{-3}$	$2.41 \cdot 10^{-6}$
5	0.1970	$1.71 \cdot 10^{-4}$	$1.5880 \cdot 10^{-3}$	$6.57 \cdot 10^{-4}$	$1.4597 \cdot 10^{-3}$	$5.85 \cdot 10^{-7}$
6	0.1970	$4.50 \cdot 10^{-5}$	$1.5879 \cdot 10^{-3}$	$1.26 \cdot 10^{-4}$	$1.4596 \cdot 10^{-3}$	$1.27 \cdot 10^{-7}$
l	Non-linear transformation and stretching					
1	0.1981		$1.5817 \cdot 10^{-3}$		$1.4542 \cdot 10^{-3}$	
2	0.1973	$7.81 \cdot 10^{-4}$	$1.5862 \cdot 10^{-3}$	$7.88 \cdot 10^{-4}$	$1.4580 \cdot 10^{-3}$	$3.781 \cdot 10^{-6}$
3	0.1971	$1.97 \cdot 10^{-4}$	$1.5872 \cdot 10^{-3}$	$5.92 \cdot 10^{-4}$	$1.4588 \cdot 10^{-3}$	$8.06 \cdot 10^{-7}$
4	0.1970	$4.92 \cdot 10^{-5}$	$1.5874 \cdot 10^{-3}$	$1.47 \cdot 10^{-4}$	$1.4590 \cdot 10^{-3}$	$2.04 \cdot 10^{-7}$

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

pay-off with a grid line can clearly be observed as the methods based on transformed coordinates show a very satisfactory accuracy.

Although this is a positive result for the use of sparse grids for basket options (a result that can easily be generalized to pricing options with early exercise features, for example), 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 coincides with a grid line after a coordinate transformation. This may, however, not be easily possible for complex payoff structures, as they are usually encountered in the financial industry. For those there is little hope for satisfactory sparse grid accuracy without any enhancements (making the method more complicated).

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 gradients near the exercise price. For the hedge parameters, presented in Table 5, the difference in accuracy between a linear or a non-linear 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 equation (39) 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. The following parameters are chosen (for the four- and, later, for the five-asset basket call):

- $K = 100$, $r = 0.04$, $T = 1$,
- $\sigma_1 = 0.3$, $\sigma_2 = 0.35$, $\sigma_3 = 0.4$ $\sigma_4 = 0.45$, $\sigma_5 = 0.25$,
- $\rho_{ij} = 0.5 \quad \forall 1 \leq i, j < 5,\ i \neq j$
- $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = 0$
- $w_1 = w_2 = w_3 = w_4 = w_5 = 1/d$

For the five-asset basket call we focus only on the non-linear transformation. In the Tables 6 and 7, the results of these two option contracts are presented. 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 4D and 5D sparse grid solutions is due to the term $(\log(h_l^{-1}))^{d-1}$ in equation (42).

	4D linear, no stretching					
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	13.6720		0.1450		$8.6973 \cdot 10^{-4}$	
2	13.6618	$1.0266 \cdot 10^{-2}$	0.1455	$5.3596 \cdot 10^{-4}$	$8.7153 \cdot 10^{-4}$	$1.8062 \cdot 10^{-6}$
3	13.6597	$2.0656 \cdot 10^{-3}$	0.1457	$1.3006 \cdot 10^{-4}$	$8.7310 \cdot 10^{-4}$	$1.5607 \cdot 10^{-6}$
4	13.6590	$6.6128 \cdot 10^{-4}$	0.1457	$4.3688 \cdot 10^{-5}$	$8.7349 \cdot 10^{-4}$	$3.9636 \cdot 10^{-7}$
	4D linear and stretching					
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	13.6855		0.1464		$8.6942 \cdot 10^{-4}$	
2	13.6642	$2.1301 \cdot 10^{-2}$	0.1459	$4.9735 \cdot 10^{-4}$	$8.7176 \cdot 10^{-4}$	$2.3415 \cdot 10^{-6}$
3	13.6597	$4.4289 \cdot 10^{-3}$	0.1458	$1.3399 \cdot 10^{-4}$	$8.7288 \cdot 10^{-4}$	$1.1237 \cdot 10^{-6}$
4	13.6586	$1.1454 \cdot 10^{-3}$	0.1457	$3.3631 \cdot 10^{-5}$	$8.7313 \cdot 10^{-4}$	$2.4849 \cdot 10^{-7}$
	4D non-linear					
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	13.6471		0.1450		$8.7344 \cdot 10^{-4}$	
2	13.6551	$8.0568 \cdot 10^{-3}$	0.1456	$5.6144 \cdot 10^{-4}$	$8.7362 \cdot 10^{-4}$	$1.8754 \cdot 10^{-7}$
3	13.6580	$2.8561 \cdot 10^{-3}$	0.1457	$1.1662 \cdot 10^{-4}$	$8.7363 \cdot 10^{-4}$	$7.2636 \cdot 10^{-9}$
4	13.6586	$6.4266 \cdot 10^{-4}$	0.1457	$3.0644 \cdot 10^{-5}$	$8.7365 \cdot 10^{-4}$	$1.6688 \cdot 10^{-8}$
	4D non-linear and stretching					
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	13.6705		0.1465		$8.7008 \cdot 10^{-4}$	
2	13.6605	$9.9860 \cdot 10^{-3}$	0.1459	$5.7513 \cdot 10^{-4}$	$8.7256 \cdot 10^{-4}$	$2.4738 \cdot 10^{-6}$
3	13.6588	$1.6815 \cdot 10^{-3}$	0.1458	$1.4322 \cdot 10^{-4}$	$8.7310 \cdot 10^{-4}$	$5.4651 \cdot 10^{-7}$
4	13.6584	$4.4404 \cdot 10^{-4}$	0.1457	$3.5827 \cdot 10^{-5}$	$8.7324 \cdot 10^{-4}$	$1.3813 \cdot 10^{-7}$

Table 6: Four-asset option price, Δ_1 and $\Gamma_{1,1}$. The sparse grid solution mimics a $(2^{l+4} \times 2^{l+2} \times 2^{l+2} \times 2^{l+2})$-grid, $c_1 = 16, c_2 = c_3 = c_4 = 4$

Non-linear transformation, no grid stretching						
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	12.6697		0.1176		$6.0568 \cdot 10^{-4}$	
2	12.6788	$9.1126 \cdot 10^{-3}$	0.1181	$5.3724 \cdot 10^{-4}$	$6.0556 \cdot 10^{-4}$	$1.1569 \cdot 10^{-7}$
3	12.6821	$3.2759 \cdot 10^{-3}$	0.1182	$1.0939 \cdot 10^{-4}$	$6.0549 \cdot 10^{-4}$	$7.3138 \cdot 10^{-8}$
4	12.6829	$7.4950 \cdot 10^{-4}$	0.1182	$2.8927 \cdot 10^{-5}$	$6.0548 \cdot 10^{-4}$	$8.3213 \cdot 10^{-9}$
Non-linear transformation and stretching						
l	Price	Error	Δ_1 (44)	Error	$\Gamma_{1,1}$ (45)	Error
1	12.6997		0.1189		$6.0329 \cdot 10^{-4}$	
2	12.6863	$1.3420 \cdot 10^{-2}$	0.1184	$4.9556 \cdot 10^{-4}$	$6.0494 \cdot 10^{-4}$	$1.6557 \cdot 10^{-6}$
3	12.6838	$2.4362 \cdot 10^{-3}$	0.1183	$1.2326 \cdot 10^{-4}$	$6.0527 \cdot 10^{-4}$	$3.3178 \cdot 10^{-7}$
4	12.6832	$6.3432 \cdot 10^{-4}$	0.1182	$3.0835 \cdot 10^{-5}$	$6.0536 \cdot 10^{-4}$	$8.3943 \cdot 10^{-8}$

Table 7: Five-asset option price, Δ_1 and $\Gamma_{1,1}$. 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, $c_1 = 16, c_2 = c_3 = c_4 = c_5 = 4$

8 Conclusion

For pricing basket options with the multi-dimensional Black-Scholes equation a linear or a non-linear 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 not really significant on these non-equidistant grids. 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.

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 and M. Griebel. *Sparse Grids. Acta Numerica*, pages 147–269, May 2004.

[4] H.J. Bungartz, M. Griebel, D. Röschke, and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math.*, 2:21–45, 1994.

[5] N. Clarke and K. Parrot. Multigrid for American option pricing with stochastic volatility. *Appl. math. finance*, 6:177–179, 1999.

[6] M. Gerritsen and P. Olsson. Desiging an efficient solution strategy for fluid flows. 1. a stable higher order finite difference scheme and sharp shock resolution. *J. of Comp. Physics*, 129:245–262, 1996.

[7] M. Griebel, M. Schneider, and 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*, pages 263–281. Elsevier, Amsterdam, 1992.

[8] P. Hofmann. Asymptotic expansions of the discretization error of boundary value problems of the laplace equation in rectangular domains. *Numerische Mathematik*, 9:302–322, 1967.

[9] J.C. Hull. *Options, futures and other derivatives.* Prentice-Hall Int. Inc, London, 1989.

[10] R. Kangro and R. Nicolaides. Far field boundary conditions for Black-Scholes equations. *SIAM J. Numer. Anal.*, 38(4):1357–1368, 2000.

[11] Y. K. Kwok. *Mathematical models of financial derivatives*. Springer Finance. Springer-Verlag Singapore, second edition, 1998.

[12] C. Reisinger. *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben*. PhD thesis, Naturwissenschaftlich-Mathematischen Gesamtfakultät der Ruprecht-Karls-Universität Heidelberg, 2004.

[13] C. Reisinger and G. Wittum. On multigrid for anisotropic equations and variational inequalities: pricing multi-dimensional european and american options. *Computation and visualisation in science*, 7:189–197, 2004.

[14] A.S. Sabau and P.E. Raad. Comparison of compact and elastical finite difference solutions of stiff problems on nonuniform grids. *Computer & fluids*, 28:361–384, 1999.

[15] W Steeb. *Kronecker product of matrices and applications*. BI-Wissenschaftsverlag, Mannheim, 1991.

[16] D. Tavella and C. Randall. *Pricing financial instruments, the finite difference method*. Wiley, New York,, 2000.

[17] R. Windcliff, P.A. Forsyth, and R.A. Vetzal. Analysis of the stability of the linear boundary condition for the black-scholes equation. *Journal of Computational Finance*, 8:65–92, 2004.

[18] C. Zenger. Sparse grids. In *Proceedings of the 6th GAMM seminar, Notes on numerical fluid mechanics*, volume 31, 1990.

[19] Y. Zhu, X. Wu, and I Chern. *Derivative securities and difference methods*. Springer Verlag, New York, 2004.