Sparse grid combination technique for Hagan SABR/LIBOR market model

J. G. López-Salas and C. Vázquez

Abstract SABR models have been used to incorporate stochastic volatility to LIBOR market models (LMM) in order to describe interest rate dynamics and price interest rate derivatives. From the numerical point of view, the pricing of derivatives with SABR/LIBOR market models (SABR/LMMs) is mainly carried out with Monte Carlo simulation. However, this approach could involve excessively long computational times. In the present chapter we propose an alternative pricing based on partial differential equations (PDEs). Thus, we pose the PDE formulation associated to the SABR/LMM proposed by Hagan [17]. As this PDE is high dimensional in space, traditional full grid methods (like standard finite differences or finite elements) are not able to price derivatives over more than one or two underlying interest rates and their corresponding stochastic volatilities. In order to overcome this curse of dimensionality, a sparse grid combination technique is proposed. So as to assess on the performance of the method a comparison with Monte Carlo is presented.

1 Introduction

The LMM [5, 19, 23] has become the most popular interest rate model. The main reason is the agreement between this model and Black's formulas, which are the standard formulas employed in the market [6]. The standard LIBOR market model considers constant volatilities for the forward rates, no volatility smile modeling is taken into account.

Among the different stochastic volatility models offered in the literature, the SABR model proposed by Hagan, Kumar, Lesniewski and Woodward [16] in the year 2002 stands out for becoming the market standard to reproduce the price of European options. The SABR model can not be used to price derivatives whose payoff depends on several forward rates. In fact, SABR model works in the termi-

Department of Mathematics, Faculty of Informatics, Campus Elviña s/n, 15071-A Coruña (Spain). e-mail: jose.lsalas@udc.es (J. G. López-Salas) and carlosv@udc.es (C. Vázquez).

nal measure, under which both the forward rate and its volatility are martingales. This can always be done if we work with one forward rate in isolation at a time. Under this same measure, however, the process for another forward rate and for its volatility would not be driftless.

In order to allow LMM to fit market volatility smiles, different extensions of the LMM that incorporate the volatility smile by means of the SABR model were proposed. These models are known as SABR/LIBOR market models (SABR/LMMs). In this chapter we will deal with the model proposed by Hagan et al. in [17].

While Monte Carlo [12] simulation remains the industry's tool of choice for pricing interest rate derivatives within SABR/LMM setting, several difficulties motivate researchers to address alternative approaches based on PDE formulations. The first issue is that the convergence of Monte Carlo methods, although it depends only very weakly on the dimension of the problem, is very slow. The second drawback of Monte Carlo methods is the valuation of options with early-exercise, like in the case of the American options, due to the so-called "Monte Carlo on Monte Carlo" effect. However, the modification of the PDE to a linear complementarity problem is usually straightforward. Finally, the weakest point of Monte Carlo methods appears to be the computation of the sensitivities of the solution with respect to the underlyings, the so-called "Greeks", which are very used by traders, and are directly given by the partial derivatives of the PDE solution.

In view of previous arguments, in the present chapter we pose the equivalent PDE formulation for the SABR/LMM proposed by Hagan. From the numerical point of view, one main difficulty in this PDE formulation lies in its high dimensionality in space-like variables. In order to cope with this so-called *curse of dimensionality* several methods are available in the literature, see [3, 11] for example, which can be put into three categories. The first group uses the Karhunen-Loeve transformation to reduce the stochastic differential equation to a lower dimensional equation, therefore this results in a lower dimensional PDE associated to the previously reduced SDE. The second category gathers those methods which try to reduce the dimension of the PDE itself, like for example dimension-wise decomposition algorithms. Finally, the third category groups the methods which reduce the complexity of the problem in the discretization layer, like for example the method of sparse grids, which we use in the present chapter.

The sparse grid method was originally developed by Smolyak [28], who used it for numerical integration. It is mainly based on a hierarchical basis [29, 30], a representation of a discrete function space which is equivalent to the conventional nodal basis, and a sparse tensor product construction. Zenger [32] and Bungartz and Griebel [7] extended this idea and applied sparse grids to solve PDEs with finite elements, finite volumes and finite differences methods. Besides working directly in the hierarchical basis, the sparse grid can also be computed using the combination technique [14] by linearly combining solutions on traditional Cartesian grids with different mesh widths. This is the approach we follow in this chapter. Recently, this technique has been used for a financial application related to the pricing of basket options in [18]. Also in our previous work [21] we have posed the analogous PDE formulation for the SABR/LMM proposed by Mercurio and Morini in [22].

Moreover, we have used the same numerical methodology based on the sparse grids combination technique to solve the resulting high dimensional PDE problem.

The chapter is organized as follows. In Section 2 we pose the PDE formulation for the Hagan SABR/LMM. In Section 3 we describe the use of a full grid finite differences scheme for the Hagan model. Numerical results show the limitations of the full grid method when the number of forward rates increases. Therefore, in Section 4 we describe the sparse grid combination technique applied to the Hagan SABR/LMM and show numerical results that illustrate the behaviour of the method when the number of forward rates increases. For this purpose, a comparison with Monte Carlo simulation results is used.

2 The Hagan SABR/LMM PDE

We first consider a set of $N-1$ LIBOR forward rates F_i, $1 \leq i \leq N-1$, $\mathbf{F} = (F_1, \ldots, F_{N-1})$ on the tenor structure $[T_0, T_1, \ldots, T_{N-1}, T_N]$, the accruals being $\tau_i = T_{i+1} - T_i$. Hagan SABR/LMM is defined by the following system of stochastic differential equations [17]:

$$\begin{aligned} dF_i(t) &= \mu^{F_i}(t) F_i(t)^{\beta_i} dt + \alpha_i V_i(t) F_i(t)^{\beta_i} dW_i^{\mathscr{Q}}(t), \quad F_i(0) \text{ given}, \\ dV_i(t) &= \mu^{V_i}(t) V_i(t) dt + \sigma_i V_i(t) dZ_i^{\mathscr{Q}}(t), \quad V_i(0) = 1, \end{aligned} \tag{1}$$

which are posed on a probability space $\{\Omega, \mathscr{F}, \mathscr{Q}\}$ with filtration $\{\mathscr{F}_t\}$, $t \in [T_0, T_N]$. On one hand, μ^{F_i} is the drift of the i-th forward rate, $\beta_i \in [0,1]$ is the variance elasticity coefficient, $W_i^{\mathscr{Q}}$ is a standard Brownian motion under the risk neutral measure $\mathscr{Q}$, and $\boldsymbol{\rho}$ is the correlation matrix between the forward rates, i.e.

$$< dW_i^{\mathscr{Q}}(t), dW_j^{\mathscr{Q}}(t) >= \rho_{ij} dt, \quad \forall i, j \in \{1, \ldots, N-1\}.$$

On the other hand, V_i is the stochastic volatility of the forward rate F_i, μ^{V_i} is the drift of the i-th stochastic volatility, α_i is a deterministic (constant) instantaneous volatility coefficient used to embed in the model any initial value of the volatility process V_i, $Z_i^{\mathscr{Q}}$ is a standard Brownian motion, and $\boldsymbol{\theta}$ is the correlation matrix between the stochastic volatilities, i.e.

$$< dZ_i^{\mathscr{Q}}(t), dZ_j^{\mathscr{Q}}(t) >= \theta_{ij} dt, \quad \forall i, j \in \{1, \ldots, N-1\}.$$

Besides, the Brownian motions driving the forward rates are correlated with those ones driving the stochastic volatilities, $\boldsymbol{\phi}$ will denote the correlation matrix between the forward rates and their stochastic volatilities, i.e.

$$< dW_i^{\mathscr{Q}}(t), dZ_j^{\mathscr{Q}}(t) >= \phi_{ij} dt, \quad \forall i, j \in \{1, \ldots, N-1\}.$$

Thus, the correlation structure is given by the block-matrix

$$
\boldsymbol{P} = \begin{bmatrix} \boldsymbol{\rho} & \boldsymbol{\phi} \\ \boldsymbol{\phi}^\top & \boldsymbol{\theta} \end{bmatrix},
$$

which is assumed to be positive definite.

The drifts of the forward rates and their stochastic volatilities are determined by the chosen numeraire. Under the terminal probability measure associated with choosing the bond $P(t,T_N)$ as numeraire, the drifts of the forwards rates are given by

$$
\mu^{F_i}(t) = \begin{cases} -\alpha_i V_i(t) \displaystyle\sum_{j=i+1}^{N-1} \frac{\tau_j F_j(t)^{\beta_j}}{1+\tau_j F_j(t)} \rho_{ij} \alpha_j V_j(t) & \text{if } j < N-1, \\ 0 & \text{if } j = N-1, \end{cases}
$$

while the drifts of the stochastic volatilities are given by

$$
\mu^{V_i}(t) = \begin{cases} -\sigma_i \displaystyle\sum_{j=i+1}^{N-1} \frac{\tau_j F_j(t)^{\beta_j}}{1+\tau_j F_j(t)} \phi_{ij} \alpha_j V_j(t) & \text{if } j < N-1, \\ 0 & \text{if } j = N-1. \end{cases}
$$

Our model for the correlation structure is taken from Rebonato [25], who suggests the following functional parameterization:

$$
\rho_{ij} = \exp[-\lambda_1 |T_i - T_j|], \tag{2}
$$

$$
\theta_{ij} = \exp[-\lambda_2 |T_i - T_j|], \tag{3}
$$

$$
\phi_{ij} = \operatorname{sign}(\phi_{ii}) \sqrt{|\phi_{ii}\phi_{jj}|} \exp[-\lambda_3 (T_i - T_j)^+ - \lambda_3 (T_j - T_i)^+]. \tag{4}
$$

So far we have introduced Hagan SABR/LMM. Now suppose we need to price an interest rate product $u(t,\mathbf{F},\mathbf{V})$ whose payoff at expiry T_N is a function of forward rates from F_1 to F_{N-1}, and also of their stochastic volatilities $\mathbf{V} = (V_1,\ldots,V_{N-1})$. If G is the payoff of the option, then the arbitrage-free value of the option relative to a numeraire $\mathscr{N}$ is given by

$$
u(t,\mathbf{F}(t),\mathbf{V}(t)) = \mathbb{E}^{\mathscr{Q}} \left(\frac{G(T,\mathbf{F}(T),\mathbf{V}(T))}{\mathscr{N}(T)} \,\middle|\, \mathscr{F}_t \right). \tag{5}
$$

Thus, the value u of the option satisfies the PDE

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \frac{1}{2} \sum_{i,j=1}^{N-1} \theta_{ij} \sigma_i V_i \sigma_j V_j \frac{\partial^2 u}{\partial V_i \partial V_j} + \frac{1}{2} \sum_{i,j=1}^{N-1} \rho_{ij} \alpha_i V_i F_i^{\beta_i} \alpha_j V_j F_j^{\beta_j} \frac{\partial^2 u}{\partial F_i \partial F_j} + \\
\sum_{i,j=1}^{N-1} \phi_{ij} \alpha_i V_i F_i^{\beta_i} \sigma_j V_j \frac{\partial^2 u}{\partial F_i \partial V_j} + \sum_{i=1}^{N-1} \mu^{F_i}(t) F_i^{\beta_i} \frac{\partial u}{\partial F_i} + \sum_{i=1}^{N-1} \mu^{V_i}(t) V_i \frac{\partial u}{\partial V_i} = 0,
\end{aligned} \tag{6}
$$

with the terminal condition given by the derivative payoff,

$$u(T,\mathbf{F},\mathbf{V}) = g(T,\mathbf{F},\mathbf{V}),$$

on $[0,T]\times\mathbb{R}^{N-1}\times\mathbb{R}^{N-1}$. For simplicity of notation, we have used the relative payoff $g(\cdot) = \dfrac{G(\cdot)}{\mathscr{N}(T)}$. This PDE was be derived by applying multi-dimensional Itô's Lemma to u, see [27] for details.

Hereafter, for sake of brevity in the notation, let us consider the following operator:

$$\begin{aligned}\mathscr{L}[u] =& \frac{\partial u}{\partial t} + \frac{1}{2}\sum_{i,j=1}^{N-1}\theta_{ij}\sigma_i V_i \sigma_j V_j \frac{\partial^2 u}{\partial V_i \partial V_j} + \frac{1}{2}\sum_{i,j=1}^{N-1}\rho_{ij}\alpha_i V_i F_i^{\beta_i}\alpha_j V_j F_j^{\beta_j}\frac{\partial^2 u}{\partial F_i \partial F_j} + \\ & \sum_{i,j=1}^{N-1}\phi_{ij}\alpha_i V_i F_i^{\beta_i}\sigma_j V_j \frac{\partial^2 u}{\partial F_i \partial V_j} + \sum_{i=1}^{N-1}\mu^{F_i}(t)F_i^{\beta_i}\frac{\partial u}{\partial F_i} + \sum_{i=1}^{N-1}\mu^{V_i}(t)V_i\frac{\partial u}{\partial V_i},\end{aligned}$$

where u is a function defined on the domain $[0,T]\times\mathbb{R}^{N-1}\times\mathbb{R}^{N-1}$.

3 Finite Differences Method with full grids

In this section we introduce a full grid finite differences method to solve the problem (6). Domain truncation and boundary conditions are proposed. Notice that while the choice of the range of the time variable is totally unambiguous, $[0,T]$, an a priori choice must be made about which values of the space variables are too high or too low to be of interest, so far we will denote them by $[F_i^{min}, F_i^{max}]$ and $[V_i^{min}, V_i^{max}]$. Selecting boundary values such that the option of interest is too deeply in or out-of-the money is a common and reasonable choice.

We are going to define a $(2N-1)$-dimensional mesh with the time sampled from today (time 0) to the final expiry of the option (time T) at $M+1$ points uniformly spaced by the time step $\Delta t = \dfrac{T}{M}$.

The variables representing the forward rates $\mathbf{F} = (F_1,\ldots,F_{N-1})$ and their stochastic volatilities $\mathbf{V} = (V_1,\ldots,V_{N-1})$, often referred as the "space variables", will be sampled at R_i+1 and S_i+1 points, $i = 1,\ldots,N-1$, spaced by $h_i = \dfrac{F_i^{max}-F_i^{min}}{R_i}$ and $\hat{h}_i = \dfrac{V_i^{max}-V_i^{min}}{S_i}$, respectively.

For a given mesh, each point is uniquely determined by the time level m ($m = 0,\ldots,M$), the index vectors of the $N-1$ forward rates $\mathbf{f} = (f_1,\ldots,f_i,\ldots,f_{N-1})$ and stochastic volatilities $\mathbf{v} = (v_1,\ldots,v_i,\ldots,v_{N-1})$, where $f_i = 0,\ldots,R_i$ and $v_i = 0,\ldots,S_i$. We seek approximations of the solution at these mesh points, which will be denoted by

$$U^m_{\mathbf{f},\mathbf{v}} \approx u(m\Delta t, (f_i h_i)_{1\le i\le N-1}, (v_i\hat{h}_i)_{1\le i\le N-1}).$$

It is natural for this PDE to be solved backwards in time. We approximate the time derivative by the time-forward approximation

$$\left.\frac{\partial u}{\partial t}\right|_{t=m\Delta t,\mathbf{F}=(f_i h_i)_{1\le i\le N-1},\mathbf{V}=(v_i \hat{h}_i)_{1\le i\le N-1}} = \left.\frac{\partial u}{\partial t}\right|_{m,\mathbf{f},\mathbf{v}} \approx \frac{U^{m+1}_{\mathbf{f},\mathbf{v}} - U^{m}_{\mathbf{f},\mathbf{v}}}{\Delta t}.$$

For the space derivatives we have chosen second-order approximations. We will write $\mathbf{f}_{i\pm 1}$ to mean the forward rates index vector $(f_1,\ldots,f_i\pm 1,\ldots,f_{N-1})$ which corresponds to the forward rates point $(f_1 h_1,\ldots,(f_i\pm 1)h_i,\ldots,f_{N-1}h_{N-1})$. The same notation will be used in the case of the stochastic volatilities index vector.
The first derivatives are approximated by central differences:

- $\left.\dfrac{\partial u}{\partial F_i}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f}_{i+1},\mathbf{v}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}}}{2h_i}$,
- $\left.\dfrac{\partial u}{\partial V_i}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f},\mathbf{v}_{i+1}} - U^m_{\mathbf{f},\mathbf{v}_{i-1}}}{2\hat{h}_i}$.

The second derivatives are approximated by:

- $\left.\dfrac{\partial^2 u}{\partial F_i^2}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f}_{i+1},\mathbf{v}} - 2U^m_{\mathbf{f},\mathbf{v}} + U^m_{\mathbf{f}_{i-1},\mathbf{v}}}{h_i^2}$,
- $\left.\dfrac{\partial^2 u}{\partial V_i^2}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f},\mathbf{v}_{i+1}} - 2U^m_{\mathbf{f},\mathbf{v}} + U^m_{\mathbf{f},\mathbf{v}_{i-1}}}{\hat{h}_i^2}$.

The cross derivatives terms are approximated by:

- For $i\neq j$, $\left.\dfrac{\partial^2 u}{\partial F_i \partial F_j}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f}_{i+1,j+1},\mathbf{v}} + U^m_{\mathbf{f}_{i-1,j-1},\mathbf{v}} - U^m_{\mathbf{f}_{i+1,j-1},\mathbf{v}} - U^m_{\mathbf{f}_{i-1,j+1},\mathbf{v}}}{4h_i h_j}$,
- For $i\neq j$, $\left.\dfrac{\partial^2 u}{\partial V_i \partial V_j}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f},\mathbf{v}_{i+1,j+1}} + U^m_{\mathbf{f},\mathbf{v}_{i-1,j-1}} - U^m_{\mathbf{f},\mathbf{v}_{i+1,j-1}} - U^m_{\mathbf{f},\mathbf{v}_{i-1,j+1}}}{4\hat{h}_i \hat{h}_j}$,
- $\left.\dfrac{\partial^2 u}{\partial F_i \partial V_j}\right|_{m,\mathbf{f},\mathbf{v}} \approx \dfrac{U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j+1}} + U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j-1}} - U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j-1}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j+1}}}{4h_i \hat{h}_j}$.

The finite differences solution under the so-called θ-scheme satisfies

$$\frac{U^{m+1}_{\mathbf{f},\mathbf{v}} - U^{m}_{\mathbf{f},\mathbf{v}}}{\Delta t} + \theta W^m_{\mathbf{f},\mathbf{v}} + (1-\theta) W^{m+1}_{\mathbf{f},\mathbf{v}} = 0,$$

where $\theta \in [0,1]$ and $W^m_{\mathbf{f},\mathbf{v}}$ is the discretization given by

$$
\begin{aligned}
W^m_{\mathbf{f},\mathbf{v}} =& \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N-1} \theta_{ij}\sigma_i V_i \sigma_j V_j \frac{U^m_{\mathbf{f},\mathbf{v}_{i+1,j+1}} + U^m_{\mathbf{f},\mathbf{v}_{i-1,j-1}} - U^m_{\mathbf{f},\mathbf{v}_{i+1,j-1}} - U^m_{\mathbf{f},\mathbf{v}_{i-1,j+1}}}{4\hat{h}_i\hat{h}_j} + \\
& \frac{1}{2}\sum_{i=1}^{N-1} \sigma_i^2 V_i^2 \frac{U^m_{\mathbf{f},\mathbf{v}_{i+1}} - 2U^m_{\mathbf{f},\mathbf{v}} + U^m_{\mathbf{f},\mathbf{v}_{i-1}}}{\hat{h}_i^2} + \\
& \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N-1} \rho_{ij}\alpha_i V_i F_i^{\beta_i} \alpha_j V_j F_j^{\beta_j} \frac{U^m_{\mathbf{f}_{i+1,j+1},\mathbf{v}} + U^m_{\mathbf{f}_{i-1,j-1},\mathbf{v}} - U^m_{\mathbf{f}_{i+1,j-1},\mathbf{v}} - U^m_{\mathbf{f}_{i-1,j+1},\mathbf{v}}}{4h_i h_j} + \\
& \frac{1}{2}\sum_{i=1}^{N-1} \alpha_i^2 V_i^2 F_i^{2\beta_i} \frac{U^m_{\mathbf{f}_{i+1},\mathbf{v}} - 2U^m_{\mathbf{f},\mathbf{v}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}}}{h_i^2} + \\
& \sum_{i,j=1}^{N-1} \phi_{ij}\alpha_i V_i F_i^{\beta_i} \sigma_j V_j \frac{U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j+1}} + U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j-1}} - U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j-1}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j+1}}}{4h_i\hat{h}_j} + \\
& \sum_{i=1}^{N-1} \mu^{F_i}(m\Delta t) F_i^{\beta_i} \frac{U^m_{\mathbf{f}_{i+1},\mathbf{v}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}}}{2h_i} + \\
& \sum_{i=1}^{N-1} \mu^{V_i}(m\Delta t) V_i \frac{U^m_{\mathbf{f},\mathbf{v}_{i+1}} - U^m_{\mathbf{f},\mathbf{v}_{i-1}}}{2\hat{h}_i},
\end{aligned}
\tag{7}
$$

and with terminal condition $U^M_{\mathbf{f},\mathbf{v}} = g(T,\mathbf{F},\mathbf{V})$.

Three different θ values represent three canonical discretization schemes, $\theta = 0$ is the explicit scheme, $\theta = 1$ the fully implicit scheme and $\theta = 0.5$ the Crank-Nicolson scheme. The fully implicit discretization is the best method with respect to stability, whereas the Crank-Nicolson timestepping provides the best convergence rate. Although the explicit method is the simplest to implement, it has the disadvantage of being conditionally stable.

We shall first discriminate explicit and implicit parts as follows:

$$
\frac{U^m_{\mathbf{f},\mathbf{v}}}{\Delta t} - \theta W^m_{\mathbf{f},\mathbf{v}} = \frac{U^{m+1}_{\mathbf{f},\mathbf{v}}}{\Delta t} + (1-\theta) W^{m+1}_{\mathbf{f},\mathbf{v}}. \tag{8}
$$

As a result of such discretization we arrive to the linear system of equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is the band matrix of known coefficients, $\mathbf{x}$ is the vector of the unknown solutions $U^m_{\mathbf{f},\mathbf{v}}$ and $\mathbf{b}$ is the vector of known values corresponding to the right-hand side of (8).

Equation (8) can be rewritten as:

$$
\begin{aligned}
& \theta \sum_{i=1}^{N-1} (\hat{b}_i - \hat{r}_i) U^m_{\mathbf{f},\mathbf{v}_{i-1}} + \theta \sum_{i=1}^{N-1} (\hat{b}_i + \hat{r}_i) U^m_{\mathbf{f},\mathbf{v}_{i+1}} + \\
& \theta \sum_{i=1}^{N-1} (b_i - r_i) U^m_{\mathbf{f}_{i-1},\mathbf{v}} + \theta \sum_{i=1}^{N-1} (b_i + r_i) U^m_{\mathbf{f}_{i+1},\mathbf{v}} +
\end{aligned}
$$

$$
\begin{aligned}
&\theta \sum_{ij\in P} a_{ij}\left(U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j+1}} + U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j-1}} - U^m_{\mathbf{f}_{i-1},\mathbf{v}_{j+1}} - U^m_{\mathbf{f}_{i+1},\mathbf{v}_{j-1}}\right) + \\
&\theta \sum_{ij\in C} \hat{\psi}_{ij}\left(U^m_{\mathbf{f},\mathbf{v}_{i+1,j+1}} + U^m_{\mathbf{f},\mathbf{v}_{i-1,j-1}} - U^m_{\mathbf{f},\mathbf{v}_{i-1,j+1}} - U^m_{\mathbf{f},\mathbf{v}_{i+1,j-1}}\right) + \\
&\theta \sum_{ij\in C} \psi_{ij}\left(U^m_{\mathbf{f}_{i+1,j+1},\mathbf{v}} + U^m_{\mathbf{f}_{i-1,j-1},\mathbf{v}} - U^m_{\mathbf{f}_{i-1,j+1},\mathbf{v}} - U^m_{\mathbf{f}_{i+1,j-1},\mathbf{v}}\right) + \\
&\left(-1 - 2\theta \sum_{i=1}^{N-1} (\hat{b}_i + b_i)\right) U^m_{\mathbf{f},\mathbf{v}} = \\
&-\hat{\theta} \sum_{i=1}^{N-1} (\hat{b}_i - \hat{r}_i) U^{m+1}_{\mathbf{f},\mathbf{v}_{i-1}} - \hat{\theta} \sum_{i=1}^{N-1} (\hat{b}_i + \hat{r}_i) U^{m+1}_{\mathbf{f},\mathbf{v}_{i+1}} \\
&-\hat{\theta} \sum_{i=1}^{N-1} (b_i - r_i) U^{m+1}_{\mathbf{f}_{i-1},\mathbf{v}} - \hat{\theta} \sum_{i=1}^{N-1} (b_i + r_i) U^{m+1}_{\mathbf{f}_{i+1},\mathbf{v}} \\
&-\hat{\theta} \sum_{ij\in P} a_{ij}\left(U^{m+1}_{\mathbf{f}_{i+1},\mathbf{v}_{j+1}} + U^{m+1}_{\mathbf{f}_{i-1},\mathbf{v}_{j-1}} - U^{m+1}_{\mathbf{f}_{i-1},\mathbf{v}_{j+1}} - U^{m+1}_{\mathbf{f}_{i+1},\mathbf{v}_{j-1}}\right) \\
&-\hat{\theta} \sum_{ij\in C} \hat{\psi}_{ij}\left(U^{m+1}_{\mathbf{f},\mathbf{v}_{i+1,j+1}} + U^{m+1}_{\mathbf{f},\mathbf{v}_{i-1,j-1}} - U^{m+1}_{\mathbf{f},\mathbf{v}_{i-1,j+1}} - U^{m+1}_{\mathbf{f},\mathbf{v}_{i+1,j-1}}\right) \\
&-\hat{\theta} \sum_{ij\in C} \psi_{ij}\left(U^{m+1}_{\mathbf{f}_{i+1,j+1},\mathbf{v}} + U^{m+1}_{\mathbf{f}_{i-1,j-1},\mathbf{v}} - U^{m+1}_{\mathbf{f}_{i-1,j+1},\mathbf{v}} - U^{m+1}_{\mathbf{f}_{i+1,j-1},\mathbf{v}}\right) + \\
&\left(-1 + 2\hat{\theta} \sum_{i=1}^{N-1} (\hat{b}_i + b_i)\right) U^{m+1}_{\mathbf{f},v},
\end{aligned}
\tag{9}
$$

where $\hat{\theta} = (1-\theta)$, P is the set containing the permutations of the numbers $1,2,\ldots,N-1$ taken two at a time with repetition (the number of elements in P is $(N-1)^2$), C is the set containing the combinations of the numbers $1,2,\ldots,N-1$ taken two at a time without repetition (the number of elements in C is $\binom{N-1}{2} = 2^{-1}(N-1)(N-2)$) and the known coefficients $\hat{b}_i$, b_i, $\hat{r}_i$, r_i, $\hat{\psi}_{ij}$, ψ_{ij} and a_{ij} are defined as

$$
\begin{aligned}
&\hat{b}_i = \frac{\Delta t \sigma_i^2 V_i^2}{2\hat{h}_i^2}, \quad b_i = \frac{\Delta t \alpha_i^2 V_i^2 F_i^{2\beta_i}}{2h_i^2}, \\
&\hat{r}_i = \frac{\Delta t \mu^{V_i}(t) V_i}{2\hat{h}_i}, \quad r_i = \frac{\Delta t \mu^{F_i}(t) F_i^{\beta_i}}{2h_i}, \\
&\hat{\psi}_{ij} = \frac{\Delta t \theta_{ij} \sigma_i V_i \sigma_j V_j}{4\hat{h}_i \hat{h}_j}, \quad \psi_{ij} = \frac{\Delta t \rho_{ij} \alpha_i V_i F_i^{\beta_i} \alpha_j V_j F_j^{\beta_j}}{4h_i h_j}, \\
&a_{ij} = \frac{\Delta t \phi_{ij} \alpha_i V_i F_i^{\beta_i} \sigma_j V_j}{4h_i \hat{h}_j},
\end{aligned}
$$

where we have denoted $\mathbf{F} = (F_i = f_i h_i)_{1\le i\le N-1}$ and $\mathbf{V} = (V_i = v_i \hat{h}_i)_{1\le i\le N-1}$.

3.1 Boundary conditions

In order to specify boundary conditions, a combination of mathematical, financial and heuristic reasoning allows us to find consistent and acceptable ones. There are several possibilities, see [8] for example.

We assume that forward rates and their stochastic volatilities are non negative and hence take values in the range zero to infinity. We first truncate the unbounded interval to a bounded one and then we must specify conditions at the new boundary. Thus we will consider the truncated domain $[F_i^{min}, F_i^{max}] \times [V_i^{min}, V_i^{max}]$, with $F_i^{min} = 0$ and $V_i^{min} = 0$.

For the forward rates we consider Dirichlet boundary conditions. Particularly, the terminal condition holds on the forward rates boundaries, i.e.

$$U^m_{\{\mathbf{f}|\exists f_i=0\},\mathbf{v}} = U^M_{\mathbf{f},\mathbf{v}}, \quad \forall m = 0,\dots,M-1,$$

$$U^m_{\{\mathbf{f}|\exists f_i=R_i\},\mathbf{v}} = U^M_{\mathbf{f},\mathbf{v}}, \quad \forall m = 0,\dots,M-1.$$

At the stochastic volatility boundaries we consider the following conditions:

$$\mathscr{L}[u] = 0, \quad V_k = 0, \tag{10}$$

$$\frac{\partial u}{\partial V_k} = 0, \quad V_k = V_{max}. \tag{11}$$

Thus, when $V_k = 0$ we require that the PDE itself must be satisfied on this boundary. When V_k approaches to infinity, the price of the derivative becomes independent of V_k. This is reflected by using Neumann conditions instead of the Dirichlet ones used for the forward rates boundaries.

For the boundary $V_k = V_{max}$ in order to maintain the second order accuracy in the discretization of the first derivative the ghost point method is considered. Let us consider the volatility index vector $\mathbf{s} = (v_1, v_2, \dots, S_k, \dots, v_{N-1})$. The ghost grid points $U_{\mathbf{f},\mathbf{s}_{k+1}}$ are added. Then, the finite differences scheme of equation (9) can also be applied at the boundary points $U_{\mathbf{f},\mathbf{s}}$. However, we now have more unknowns than equations. The additional equations come from the central finite differences discretization of the Neumann boundary condition (11):

$$\frac{U_{\mathbf{f},\mathbf{s}_{k+1}} - U_{\mathbf{f},\mathbf{s}_{k-1}}}{2\hat{h}_k} = 0,$$

which yields $U_{\mathbf{f},\mathbf{s}_{k+1}} = U_{\mathbf{f},\mathbf{s}_{k-1}}$. Inserting this into the finite differences equation at $V_k = V_{max}$ we achieve

$$\theta \sum_{\substack{i=1\\i\neq k}}^{N-1} (\hat{b}_i - \hat{r}_i) U^m_{\mathbf{f},\mathbf{s}_{i-1}} + \theta \sum_{\substack{i=1\\i\neq k}}^{N-1} (\hat{b}_i + \hat{r}_i) U^m_{\mathbf{f},\mathbf{s}_{i+1}} + 2\theta \hat{b}_k U^m_{\mathbf{f},\mathbf{s}_{k-1}} +$$

$$
\begin{aligned}
&\theta \sum_{i=1}^{N-1} (b_i - r_i) U^m_{\mathbf{f}_{i-1},\mathbf{s}} + \theta \sum_{i=1}^{N-1} (b_i + r_i) U^m_{\mathbf{f}_{i+1},\mathbf{s}} + \\
&\theta \sum_{\substack{ij \in P \\ j \neq k}} a_{ij} \big(U^m_{\mathbf{f}_{i+1},\mathbf{s}_{j+1}} + U^m_{\mathbf{f}_{i-1},\mathbf{s}_{j-1}} - U^m_{\mathbf{f}_{i-1},\mathbf{s}_{j+1}} - U^m_{\mathbf{f}_{i+1},\mathbf{s}_{j-1}} \big) + \\
&\theta \sum_{\substack{ij \in C \\ i \neq k, j \neq k}} \hat{\psi}_{ij} \big(U^m_{\mathbf{f},\mathbf{s}_{i+1,j+1}} + U^m_{\mathbf{f},\mathbf{s}_{i-1,j-1}} - U^m_{\mathbf{f},\mathbf{s}_{i-1,j+1}} - U^m_{\mathbf{f},\mathbf{s}_{i+1,j-1}} \big) + \\
&\theta \sum_{ij \in C} \psi_{ij} \big(U^m_{\mathbf{f}_{i+1,j+1},\mathbf{s}} + U^m_{\mathbf{f}_{i-1,j-1},\mathbf{s}} - U^m_{\mathbf{f}_{i-1,j+1},\mathbf{s}} - U^m_{\mathbf{f}_{i+1,j-1},\mathbf{s}} \big) + \\
&\left(-1 - 2\theta \sum_{i=1}^{N-1} (\hat{b}_i + b_i) \right) U^m_{\mathbf{f},\mathbf{s}} = \\
&- \hat{\theta} \sum_{\substack{i=1 \\ i \neq k}}^{N-1} (\hat{b}_i - \hat{r}_i) U^{m+1}_{\mathbf{f},\mathbf{s}_{i-1}} - \hat{\theta} \sum_{\substack{i=1 \\ i \neq k}}^{N-1} (\hat{b}_i + \hat{r}_i) U^{m+1}_{\mathbf{f},\mathbf{s}_{i+1}} - 2\hat{\theta} \hat{b}_k U^{m+1}_{\mathbf{f},\mathbf{s}_{k-1}} \\
&- \hat{\theta} \sum_{i=1}^{N-1} (b_i - r_i) U^{m+1}_{\mathbf{f}_{i-1},\mathbf{s}} - \hat{\theta} \sum_{i=1}^{N-1} (b_i + r_i) U^{m+1}_{\mathbf{f}_{i+1},\mathbf{s}} \\
&- \hat{\theta} \sum_{\substack{ij \in P \\ j \neq k}} a_{ij} \big(U^{m+1}_{\mathbf{f}_{i+1},\mathbf{s}_{j+1}} + U^{m+1}_{\mathbf{f}_{i-1},\mathbf{s}_{j-1}} - U^{m+1}_{\mathbf{f}_{i-1},\mathbf{s}_{j+1}} - U^{m+1}_{\mathbf{f}_{i+1},\mathbf{s}_{j-1}} \big) \\
&- \hat{\theta} \sum_{\substack{ij \in C \\ i \neq k, j \neq k}} \hat{\psi}_{ij} \big(U^{m+1}_{\mathbf{f},\mathbf{s}_{i+1,j+1}} + U^{m+1}_{\mathbf{f},\mathbf{s}_{i-1,j-1}} - U^{m+1}_{\mathbf{f},\mathbf{s}_{i-1,j+1}} - U^{m+1}_{\mathbf{f},\mathbf{s}_{i+1,j-1}} \big) \\
&- \hat{\theta} \sum_{ij \in C} \psi_{ij} \big(U^{m+1}_{\mathbf{f}_{i+1,j+1},\mathbf{s}} + U^{m+1}_{\mathbf{f}_{i-1,j-1},\mathbf{s}} - U^{m+1}_{\mathbf{f}_{i-1,j+1},\mathbf{s}} - U^{m+1}_{\mathbf{f}_{i+1,j-1},\mathbf{s}} \big) + \\
&\left(-1 + 2\theta \sum_{i=1}^{N-1} (\hat{b}_i + b_i) \right) U^{m+1}_{\mathbf{f},\mathbf{s}}.
\end{aligned}
\tag{12}
$$

3.2 Numerical results

It is not clear where to place F_i^{max} and V_i^{max}. On one hand, it is advantageous to place them far away of the initial forward rates. This reduces the error of the artificial boundary conditions. On the other hand a large computational domain requires a large discretization width. This increases the error of the approximation of the derivatives. In our experiments we will consider $F_i^{max} = 0.1$ and $V_i^{max} = 2.0$, which corresponds to interest rates of 10% and volatilities of 200%.

We are going to value $T_\alpha \times (T_\beta - T_\alpha)$ European swaptions, meaning that the swaption has maturity at time T_α and the length of the underlying swap is $(T_\beta - T_\alpha)$ (also known as the tenor of the swaption).

Some specifications of the financial product are given in Table 1 and the employed market data, taken from [4], are shown in Table 2. We will consider $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$ in the model for the correlation structure (2)-(4). Besides, the Crank-Nicolson scheme will be used in (8). For solving the system (9) the Gauss-Seidel iterative solver has been employed using a tolerance of 10^{-6}.

The numerical experiments have been performed with the following hardware and software configurations: two recent multicore Intel Xeon CPUs E5-2620 v2 clocked at 2.10 GHz (6 cores per socket) with 62 GBytes of RAM, CentOS Linux, GNU C++ compiler 4.8.2.

Table 1 Specification of the interest rate model.

Currency	EUR
Index	EURIBOR
Day Count	e30/360
Strike	5.5%

Table 2 Market data used in pricing. Data taken from 27th July 2004.

	Start date	End date	LIBOR Rate (%)	Volatility (%)
T_0	29-07-04	29-07-05	2.423306	0
T_1	29-07-05	29-07-06	3.281384	24.73
T_2	29-07-06	29-07-07	3.931690	22.45
T_3	29-07-07	29-07-08	4.364818	19.36
T_4	29-07-08	29-07-09	4.680236	17.43
T_5	29-07-09	29-07-10	4.933085	16.15
T_6	29-07-10	29-07-11	5.135066	15.02
T_7	29-07-11	29-07-12	5.273314	14.24
T_8	29-07-12	29-07-13	5.376115	13.42

First of all, the results from pricing a 1×1 European swaption are discussed. The value ϑ of this swaption is the same as the price of the corresponding caplet, and so depends only on F_1. Hence, in one dimension a closed form expression for the price of a European swaption can be found by using Black's formula [6]:

$$\vartheta = P(T_0, T_2)\tau_1 \mathrm{Bl}(K, F(T_1, T_2; T_0), \nu_1),$$

where

$$\begin{aligned}
\mathrm{Bl}(K,F,\nu) &= F\Phi\big(d_1(K,F,\nu)\big) - K\Phi\big(d_2(K,F,\nu)\big),\\
d_1(K,F,\nu) &= \frac{\ln(F/K) + \nu^2/2}{\nu},\\
d_2(K,F,\nu) &= \frac{\ln(F/K) - \nu^2/2}{\nu},
\end{aligned}$$

$$v_i = \sigma_{Black}\sqrt{T_i},$$

where $P(T_0,T_2)$ is the price at time T_0 of a bond with maturity T_2 and σ_{Black} is the constant volatility of the forward rate. This value is equal to 0.659096 basis points (one basis point is one hundredth of one percent, $\frac{1\%}{100} = \frac{1}{10000}$). As Black-Scholes formula for caplets considers constant volatility σ_{Black}, in this first test the volatility of the volatility parameter of Hagan model is considered equal to zero, i.e., $\sigma_1 = 0$, therefore a standard LIBOR market model is used. The solution was found on several levels and Table 3 shows the convergence of the model. In all tables of this chapter, *Level* refers to the refinement level n, i.e., the mesh size is $h_i = 2^{-n} \cdot c_i$ in each coordinate direction, where c_i denotes the computational domain length in direction i, which is F_i^{max} in the case of the forward rates and V_i^{max} in the case of their stochastic volatilities. Besides, the solution and the error with respect to the exact solution are also shown in basis points. Additionally, the execution time is measured in seconds and the column labeled as *Grid points* shows the number of grid points employed in the full grid used by the finite differences method without taking into account the time coordinate.

When the volatilities of the volatilities σ_i, $1 \le i < N$, of the model are non zero or when the length of the underlying swap of the swaption being considered is greater than one, no closed form solutions are available. However, an estimate can be obtained from Monte Carlo simulations. On Table 4 Monte Carlo values for the 1×1 European swaption with $\sigma_1 = 0$ are shown for several numbers of paths (*#Paths*). More details about Monte Carlo simulation of SABR/LMMs can be found in the article [9].

Table 3 Convergence of the full grid finite differences solution in basis points in the pricing of a 1×1 swaption, $\sigma_1 = 0$, $V_1(0) = 1$, $\beta_1 = 1$, 128 time steps. Exact solution, 0.659096 basis points.

Level	Solution	Error	Time	Grid points
3	2.078086	1.418989	0.0024	81
4	1.108211	0.449114	0.0094	289
5	0.779033	0.119936	0.07	1089
6	0.672004	0.012907	0.53	4225
7	0.665176	0.006079	6.34	16641
8	0.661164	0.002067	84.12	66049
9	0.659380	0.000283	1122.86	263169
10	0.659032	0.000064	14288.34	1050625

Table 4 Convergence of the Monte Carlo solution in basis points in the pricing of a 1×1 swaption, $\sigma_1 = 0$, $V_1(0) = 1$, $\beta_1 = 1$, 128 time steps. Exact solution, 0.659096 basis points.

#Paths	Solution
10^5	0.616799
10^7	0.658598
10^9	0.659506

In Table 5 the pricing of the 1×1 European swaption with $\sigma_1 = 0.3$ for different resolution levels n are shown. In Table 6 the results for the 1×2 swaption are given. Note that with this numerical method it was not feasible to price the swaption past refinement level $n = 6$ due to the huge number of required grid points.

Table 5 Convergence of the full grid finite differences solution in basis points in the pricing of a 1×1 swaption, $\sigma_1 = 0.3$, $\phi_{11} = 0.4$, $V_1(0) = 1$, $\beta_1 = 1$, 128 time steps. Monte Carlo value using 10^7 paths, 1.657662 basis points.

Level	Solution	Time	Grid points
3	6.254822	0.0039	81
4	2.501988	0.0122	289
5	1.991646	0.07	1089
6	1.597470	0.62	4225
7	1.526047	7.48	16641
8	1.519841	98.45	66049
9	1.519742	1291.76	263169
10	1.519732	16238.98	1050625

Table 6 Convergence of the full grid finite differences solution in basis points in the pricing of a 1×2 swaption, $\sigma_i = 0.3$, $\phi_{ii} = 0.4$, $V_i(0) = 1$, $\beta_i = 1$, 128 time steps. Monte Carlo value using 10^7 paths, 4.564905 basis points.

Level	Solution	Time	Grid points
3	5.289644	1.03	6561
4	5.134938	33.84	83521
5	5.023293	1258.56	1185921
6	4.997679	60396.44	17850625

Theoretically, it is possible to solve the discrete system (9) for a general number of dimensions. However, in computational science, a major problem occurs when the number of dimensions increases. A natural way to reduce the discretization error is to decrease the mesh step in each coordinate direction. However, then the number of grid points in the resulting full grid grows exponentially with the dimension, i.e. the size of the discrete solution increases drastically. This is called the *curse of dimensionality* [2]. Therefore, this procedure of improving the accuracy by decreasing the mesh step is mainly bounded by two factors, the storage and the computational complexity. Due to these limitations, using a full grid discretization method which achieves sufficiently accurate approximations is only possible for problems with up to three or four dimensions, even on the most powerful machines presently available [7].

4 Sparse grids and the combination technique

Two approaches to try to overcome the curse of dimensionality are increasing the order of accuracy of the applied numerical approximation scheme or reducing the dimension of the problem by choosing suitable coordinates. Both approaches are not always possible for every option pricing problem. In this chapter we will take advantage of the sparse grid combination technique first introduced by Zenger and co-workers [14] in order to try to overcome the curse of dimensionality and allow to use the PDE formulation of SABR/LMM for the pricing problem we are dealing with. The combination technique replicates the structure of a so-called sparse grid by linearly combining solutions on coarser grids of the same dimensionality. This technique reduces the computational effort and the storage space involved with the mentioned traditional finite differences discretization methods. The number of sub-problems to solve increases, while the computational time per problem decreases drastically. This method can be implemented in parallel as each sub-grid is independent of the others. In the next two subsections we give a brief introduction to sparse grids and the combination technique. For a detailed discussion we refer to [7].

4.1 Sparse grids

First, we introduce some notations and definitions. Let $\mathbf{l} = (l_1, l_2, \ldots, l_d) \in \mathbb{N}_0^d$ denote a d-dimensional multi-index. Let $|\mathbf{l}|_1$ and $|\mathbf{l}|_\infty$ denote the discrete L_1-norm and $L_\infty-$norm of the multi-index $\mathbf{l}$, respectively, that are defined as

$$|\mathbf{l}|_1 = \sum_{k=1}^{d} l_k \quad \text{and} \quad |\mathbf{l}|_\infty = \max_{1 \le k \le d} l_k.$$

We define the anisotropic grid $\Omega_\mathbf{l}$ with mesh size $\mathbf{h} = (h_1, h_2, \ldots, h_d) = (2^{-l_1} c_1, 2^{-l_2} c_2, \ldots, 2^{-l_d} c_d)$ with multi-index $\mathbf{l}$ and grid length $\mathbf{c} = (c_1, c_2, \ldots, c_d)$.

Then, the full grid at refinement level $n \in \mathbb{N}$ and mesh size $h_i = 2^{-n} \cdot c_i$ for all i can be defined via the sequence of subgrids

$$\Omega^n = \Omega_{(n,\ldots,n)} = \bigcup_{|\mathbf{l}|_\infty \le n} \Omega_\mathbf{l}.$$

Figure 1 visualizes two dimensional full grids for levels $n = 0, \ldots, 4$.

The number of grid points in each coordinate direction of the full grid is $2^n + 1$ and therefore the number of grid nodes in the full grid increases with $O(2^{n \cdot d})$, i.e. grows exponentially with the dimensionality d of the problem.

The sparse grid Ω_s^n at refinement level n consists of all anisotropic Cartesian grids $\Omega_\mathbf{l}$, where the total sum of all refinement factors l_k in each coordinate direction equals the resolution n. Then, the sparse grid Ω_s^n is given by

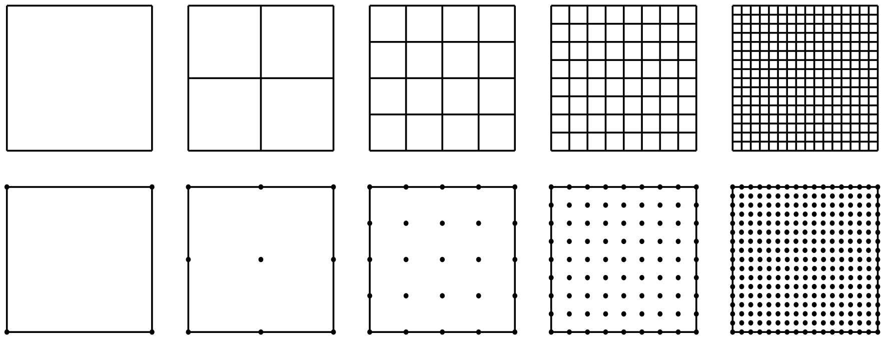


Fig. 1 Two-dimensional full grid hierarchy up to level $n = 4$.

$$\Omega_s^n = \bigcup_{|\mathbf{l}|_1 \le n} \Omega_{\mathbf{l}} = \bigcup_{|\mathbf{l}|_1 = n} \Omega_{\mathbf{l}}.$$

Figure 2 shows the two-dimensional grid hierarchy for levels $n = 0, \ldots, 4$.

The total number of nodes in the grid $\Omega_{\mathbf{l}}$ is $\prod_{k=1}^{d}(2^{l_k}+1) = O(2^{|\mathbf{l}|_1}) = O(2^n)$. In addition, there exist exactly $\binom{n+d-1}{d-1}$ grids $\Omega_{\mathbf{l}}$ with $|\mathbf{l}|_1 = n$,

$$\begin{aligned}
\binom{n+d-1}{d-1} &= \frac{(n+d-1)!}{(d-1)!n!} = \frac{(n+d-1)\cdot\ldots\cdot(n+1)n!}{(d-1)!n!} \\
&= \frac{n+(d-1)}{d-1}\cdot\frac{n+(d-2)}{d-2}\cdot\ldots\cdot\frac{n+(d-(d-1))}{d-(d-1)} \\
&= \left(1+\frac{n}{d-1}\right)\cdot\left(1+\frac{n}{d-2}\right)\cdot\ldots\cdot\left(1+\frac{n}{2}\right)\cdot\left(1+\frac{n}{1}\right) \\
&\le (1+n)^{d-1} = O(n^{d-1}).
\end{aligned}$$

Thus, the total number of grid points of the sparse grid Ω_s^n grows according to

$$\binom{n+d-1}{d-1}\cdot\prod_{k=1}^{d}(2^{l_k}+1) = O(n^{d-1})O(2^n) = O(n^{d-1}2^n), \tag{13}$$

which is far less the size of the corresponding full grid with $O(2^{nd})$ grid points. Let $h_n = 2^{-n}$, therefore the sparse grid employs $O(h_n^{-1}\cdot\log_2(h_n^{-1})^{d-1})$ grid points compared to $O(h_n^{-d})$ nodes in the full grid.

Bungartz and Griebel [7] show that the accuracy of the sparse grid using $O(h_n^{-1}\cdot\log_2(h_n^{-1})^{d-1})$ nodes is of order $O(h_n^2\log_2(h_n^{-1})^{d-1}))$ in the case of finite elements discretization and under certain smoothness conditions. Thus, the accuracy of the sparse grid is only slightly deteriorated from the accuracy $O(h_n^2)$ of conventional

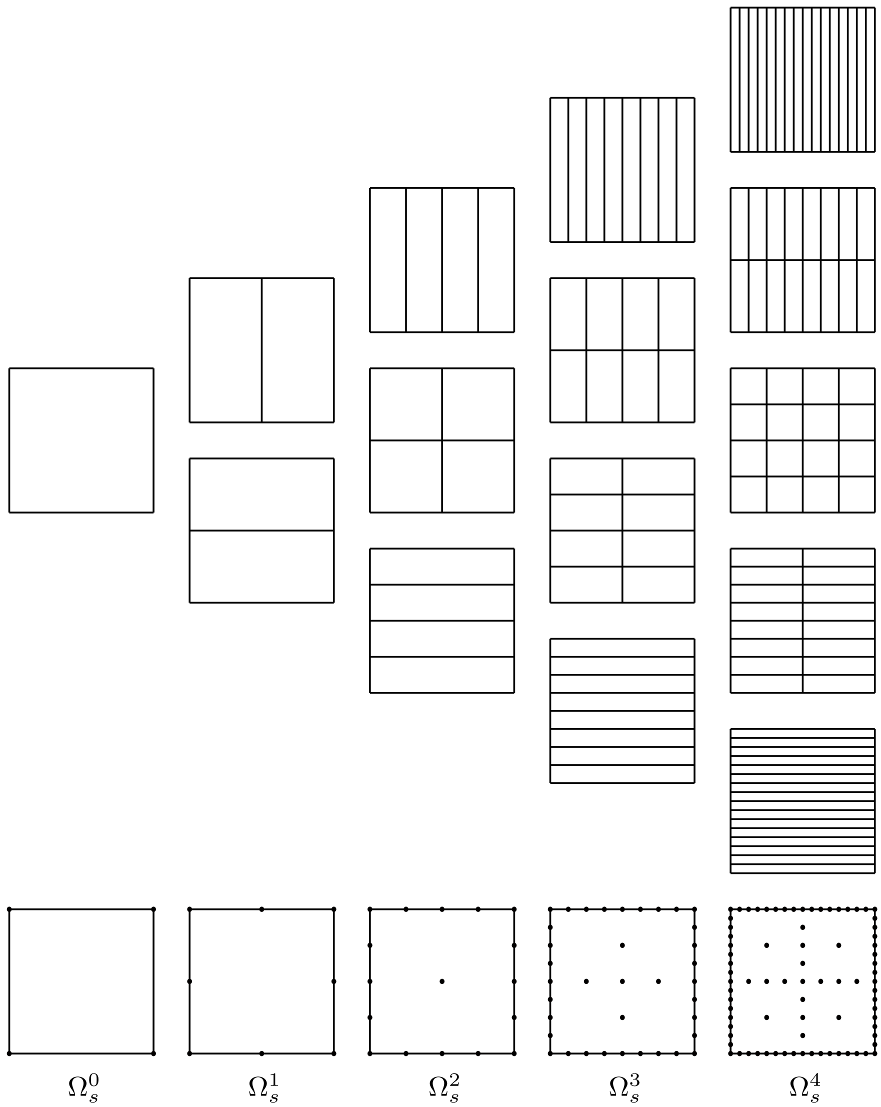
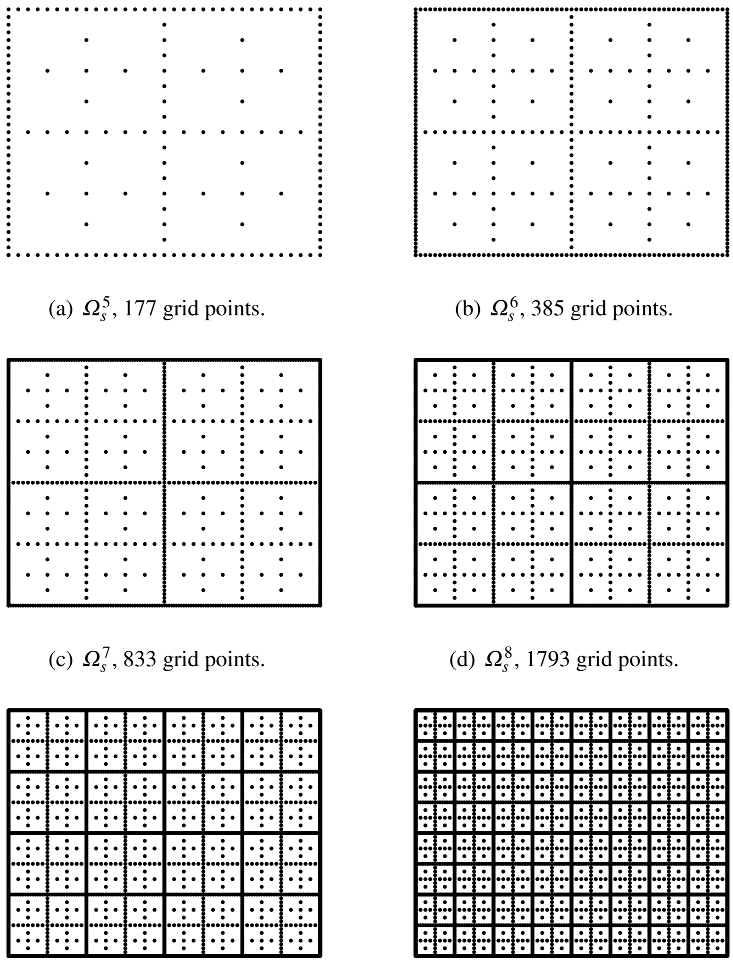


Fig. 2 Two-dimensional sparse grid hierarchy up to level $n = 4$.

full grid methods which need $O(h_n^{-d})$ grid points. Therefore, sparse grids need much less points than regular full grids to achieve a similar approximation quality.

However, the structure of a sparse grid is more complicated than the one of a full grid. Common PDE solvers usually manage only full grid solutions. Existing sparse grid methods working directly in the hierarchical basis involve a challenging implementation [1, 31]. This handicap can be circumvented with the help of the sparse grid combination technique which not only exploits the economical structure of the sparse grids but also allows for the use of traditional full grid PDE solvers.

Finally, two and three dimensional sparse grids for several resolution levels n are shown in Figures 3 and 4, respectively. Additionally, the growth of the grid points when increasing n can be observed.



(a) Ω_s^5, 177 grid points.

(b) Ω_s^6, 385 grid points.

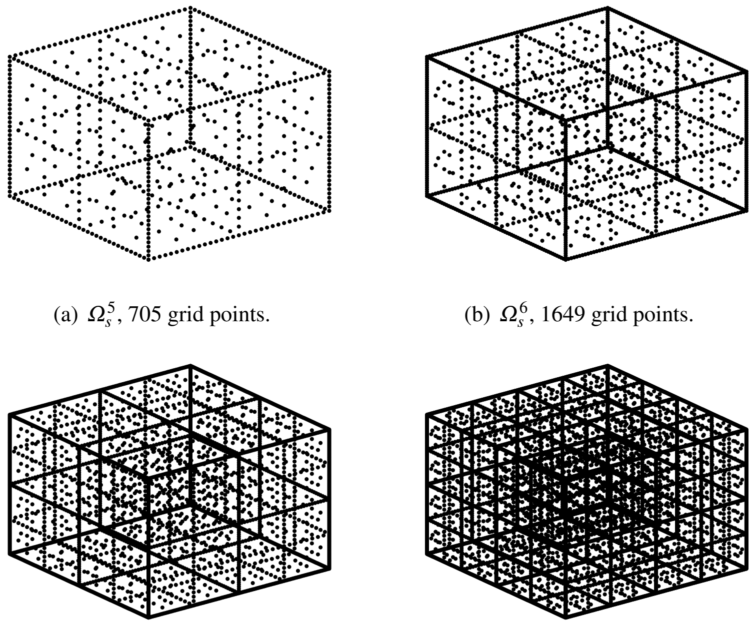
(c) Ω_s^7, 833 grid points.

(d) Ω_s^8, 1793 grid points.

(e) Ω_s^9, 3841 grid points.

(f) Ω_s^{10}, 8193 grid points.

Fig. 3 Two dimensional sparse grids for levels $n = 5, \dots, 10$.



(a) Ω_s^5, 705 grid points.

(b) Ω_s^6, 1649 grid points.

(c) Ω_s^7, 3809 grid points.

(d) Ω_s^8, 8705 grid points.

Fig. 4 Three dimensional sparse grids for levels $n = 5, 6, 7$ and 8.

4.2 Combination technique

Similar to the Richarson extrapolation [26], the so-called combination technique linearly combines the numerical solution on the sequence of anisotropic grids $\Omega_{\mathbf{l}}$ where

$$|\mathbf{l}|_1 = n - q, \quad q = 0, \ldots, d-1.$$

The combination technique reads

$$U_s^n = \sum_{q=0}^{d-1} (-1)^q \cdot \binom{d-1}{q} \cdot \sum_{|\mathbf{l}|_1 = n-q} U_{\mathbf{l}}, \quad l_k \geq 0, \quad \forall k = 1, \ldots, d, \tag{14}$$

where $U_{\mathbf{l}}$ denotes the numerical solution on the grid $\Omega_{\mathbf{l}}$ and U_s^n the combined solution on the sparse grid Ω_s^n.

The grids employed by the combination technique of level $n = 4$ in two dimensions are shown in Figure 5.

The idea of this technique is that the leading order errors from the dicretization on each grid cancel each other out in the combination solution.

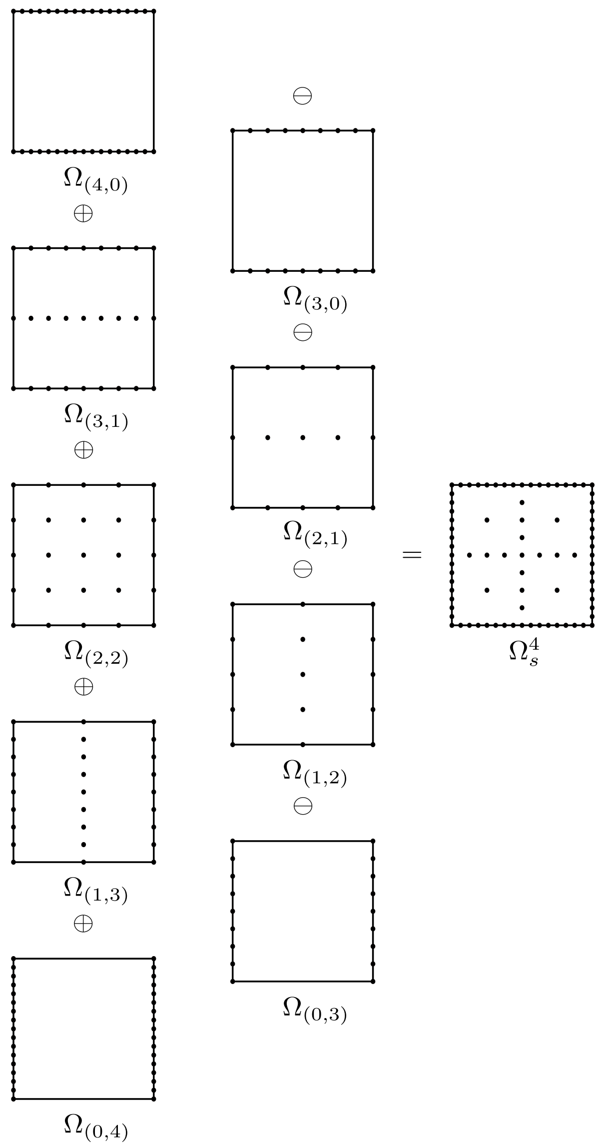


Fig. 5 Combination technique with level $n = 4$ in two dimensions.

The number of grid points involved in the approximation of U_s^n grows according to $O(n^{d-1} \cdot 2^n)$. In fact, from the formula (13) we have to solve $\binom{n+d-1}{d-1}$ problems with $O(2^n)$ unknowns, $\binom{n+d-2}{d-1}$ problems with $O(2^{n-1})$ unknowns, ... and $\binom{n}{d-1}$ problems with $O(2^{n-(d-1)})$ unknowns. This results in a total number of $O(n^{d-1} \cdot 2^n)$ grid points which is much less than the $O(2^{n \cdot d})$ grid nodes used by traditional full grid methods. Thus, the efficient use of sparse grids greatly reduces the computing time and the storage requirements which allows for the treatment of problems with ten variables and even more [7].

We have seen that the combination technique linearly combines the numerical solution on several traditional full grids. The solution can be calculated on each of these grids by using any existing PDE numerical method like finite differences, finite volume or finite elements. In addition, since all these sub-problems are independent the combination technique can be parallelized [13].

The combination technique approach presumes the existence of a so-called error splitting. It requires for an associated numerical approximation method on the full grid $\Omega_{\mathbf{l}}$ an error splitting of the form

$$u(\mathbf{x}) - U_{\mathbf{l}}(\mathbf{x}) = \sum_{k=1}^{d} \sum_{\substack{\{j_1,\ldots,j_k\} \\ \subseteq \{1,\ldots,d\}}} C_{j_1,\ldots,j_k}(\mathbf{x}, h_{j_1}, \ldots, h_{j_k}) \cdot h_{j_1}^p \cdot \ldots \cdot h_{j_k}^p, \tag{15}$$

at each grid point $\mathbf{x} \in \Omega_{\mathbf{l}}$. Here u denotes the exact solution of the partial differential equation under consideration, $U_{\mathbf{l}}$ the numerical solution on the grid $\Omega_{\mathbf{l}}$, $p > 0$ is the order of accuracy of the numerical approximation method with respect to each coordinate direction and the coefficient functions $C_{j_1,\ldots,j_k}$ of $\mathbf{x}$ and the mesh sizes h_{j_k}, $k = 1, \ldots, d$ are required to be bounded by a positive constant K such that

$$|C_{j_1,\ldots,j_k}(\mathbf{x}, h_{j_1}, \ldots, h_{j_k})| \leq K, \quad \forall k, 1 \leq k \leq d, \quad \forall \{j_1, \ldots, j_m\} \subseteq \{1, \ldots, d\}.$$

In [15] Griebel and Thurner showed that if the solution of the PDE is sufficiently smooth, the pointwise accuracy of the sparse grid combination technique is $O(n^{d-1} \cdot 2^{-n \cdot p}) = O([\log_2 h_n^{-1}]^{d-1} h_n^p)$, which is only slightly worse than $O(2^{-n \cdot p}) = O(h_n^p)$ obtained by the full grid solution.

The solution at points which do not belong to the sparse grid can be computed through interpolation. The applied interpolation method should provide at least the same order of accuracy of the numerical discretization scheme used to solve the PDE. Otherwise, the accuracy of the numerical scheme will be deteriorated.

4.3 Numerical results

Taking advantage of the previously described sparse grid combination technique, in this section we are pricing the same interest rate derivatives that have been valued in the former Section 3.2 where traditional full grid finite differences methods were considered. In addition to those products, we are going to price interest rate derivatives with up to four underlying LIBOR rates and their stochastic volatilities, showing that the sparse grid combination technique is able to cope with the curse of dimensionality up to a certain extent. As in the previous Section 3.2, we will use Crank-Nicolson scheme, we will consider the Gauss-Seidel iterative solver and the same boundary conditions as in Section 3.1. In the present case, we are interested in the evaluation of the solution at a single point which corresponds with the value of the forward rates at time zero (see Table 2) and $V_i(0) = 1$. The numerical solution on each grid handled by the combination technique is interpolated at this point using multilinear interpolation and then added up with the appropriate weights.

The sparse grid combination technique has been implemented to run on multi-core CPUs. The program was optimized and parallelized using OpenMP [33]. CPU times, measured in seconds, correspond to executions using 24 threads, so as to take advantage of Intel Hyperthreading. The speedups of the parallel version with respect to the pure sequential code are around 16. To the best of our knowledge, graphic processor units (GPUs) are not well-suited to parallelize the combination technique, due to the fact that the different grids employed by the combination technique involve memory accesses patterns totally different, therefore, it is not possible to access the device memory in a coalesced way [24], thus GPU global memory can not serve threads in parallel. In this scenario, the GPU code will be ill performing. In the work [10] the authors take advantage of GPUs to parallelize the solver of each full grid considered by the combination technique. However, they do not parallelize the combination technique itself.

In Table 7 a 1×1 European swaption is priced. The exact price of this derivative is 0.659096 basis points, as discussed in Section 3.2. These results are to be compared with those of Table 3, where it can be seen how the computational times and the grid points employed by the sparse grid combination technique have been substantially reduced.

Table 7 Convergence of the sparse grid finite differences solution in basis points in the pricing of a 1×1 swaption, $\sigma_1 = 0$, $V_1(0) = 1$, $\beta_1 = 1$, 128 time steps. Exact solution, 0.659096 basis points.

Level	Solution	Error	Time	Grid points
3	6.715346	6.056250	0.04	37
4	2.182057	1.522961	0.05	81
5	1.097761	0.438665	0.05	177
6	0.782767	0.123671	0.05	385
7	0.663808	0.004712	0.06	833
8	0.657536	0.001560	0.11	1793
9	0.658183	0.000913	0.46	3841
10	0.659363	0.000267	2.32	8193

Next, in Table 8 a 1×1 European swaption is priced considering stochastic volatility. These results are to be compared with those of Table 5.

In the following Table 9, the pricing of a 1×2 European swaption taking into account stochastic volatilities is shown, as in the Table 6. For the higher resolution levels, the full grid method became very slow, while the sparse grid combination technique results much faster. Note that the combination technique is able to price successfully the 1×2 European swaption, this was not attainable in Table 6.

Finally, in Tables 10 and 11, 1×3 and 1×4 European swaptions are priced, respectively, taking into account stochastic volatilities. The pricing of these interest

Table 8 Convergence of the sparse grid finite differences solution in basis points in the pricing of a 1×1 swaption, $\sigma_1 = 0.3$, $\phi_{11} = 0.4$, $V_1(0) = 1$, $\beta_1 = 1$, 128 time steps. Monte Carlo value using 10^7 paths, 1.657662 basis points.

Level	Solution	Time
3	6.818116	0.05
4	2.694770	0.05
5	1.919198	0.05
6	1.596501	0.08
7	1.499332	0.12
8	1.505709	0.14
9	1.515855	0.64
10	1.521027	2.83

Table 9 Convergence of the sparse grid finite differences solution in basis points in the pricing of a 1×2 swaption, $\sigma_i = 0.3$, $\phi_{ii} = 0.4$, $V_i(0) = 1$, $\beta_i = 1$, 128 time steps. Monte Carlo value using 10^7 paths, 4.564905 basis points.

Level	Solution	Time
7	5.260049	0.21
8	4.951410	0.47
9	4.651916	1.45
10	4.424338	4.10
11	4.463664	17.04
12	4.515542	81.04
13	4.537787	472.07

rate derivatives was not viable with the full grid approach of Section 3. In order to be able to price derivatives over more than 4 LIBORs and their corresponding stochastic volatilities, the combination technique method should be parallelized to run on a cluster of processors. In the Chapter 13 of the book [11] Philipp Schrder et al. discuss the parallelization of the combination technique using MPI (*Message Passing Interface*) API. In [20] the authors parallelize the sparse grid combination technique taking advantage of a MapReduce framework, algorithms that are inherently fault tolerant.

Table 10 Convergence of the sparse grid finite differences solution in basis points in the pricing of a 1×3 swaption, $\sigma_i = 0.3$, $\phi_{ii} = 0.4$, $V_i(0) = 1$, $\beta_i = 1$, 128 time steps. Monte Carlo value using 10^7 paths, 7.648443 basis points.

Level	Solution	Time
11	9.177020	151.26
12	8.461583	431.29
13	7.455562	1219.71
14	7.442483	3849.56

Table 11 Convergence of the sparse grid finite differences solution in basis points in the pricing of a 1×4 swaption, $\sigma_i = 0.3$, $\phi_{ii} = 0.4$, $V_i(0) = 1$, $\beta_i = 1$, 8 time steps. Monte Carlo value using 10^7 paths, 11.674706 basis points.

Level	Solution	Time
15	11.316526	16595.66
16	11.564127	53184.37

Acknowledgements Partially financed by Spanish Grant MTM2013-47800-C2-1-P and by Xunta de Galicia (Grant CN2014/044). First author has also been founded by a FPU Spanish Grant.

References

1. Achatz, S.: Higher order sparse grid methods for elliptic partial differential equations with variable coefficients. Computing **71**(1), 1–15 (2003)
2. Bellmann, R.: Adaptive control processes: A guided tour. Princenton University Press (1961)
3. Beylkin, G., Mohlenkamp, M.J.: Algorithms for numerical analysis in high dimensions. SIAM Journal on Scientific Computing **26**(6), 2133–2159 (2005)
4. Blackham, J.: Sparse grid solutions to the libor market model. Master's thesis, Magdalen College, University of Oxford (2004)
5. Brace, A., Gatarek, D., Musiela, M.: The Market model of interest rate dynamics. Mathematical Finance **7**(2), 127–155 (1997)
6. Brigo, D., Mercurio, F.: Interest Rate Models - Theory and Practice. With Smile, Inflation and Credit, second edn. Springer (2007)
7. Bungartz, H.J., Griebel, M.: Sparse grids. Acta Numerica **13**, 147–269 (2004)
8. Duffy, D.J.: Finite Difference methods in financial engineering. A Partial Differential Equation Approach. Wiley Finance Series (2006)
9. Ferreiro, A.M., García, J.A., López-Salas, J.G., Vázquez, C.: SABR/LIBOR market models: Pricing and calibration for some interest rate derivatives. Applied Mathematics and Computation **242**, 65–89 (2014)
10. Gaikwad, A., Toke, I.M.: GPU Based Sparse Grid Technique for Solving Multidimensional Options Pricing PDEs. In: Proceedings of the 2nd Workshop on High Performance Computational Finance, WHPCF '09, pp. 6:1–6:9. ACM, New York, NY, USA (2009). DOI 10.1145/1645413.1645419. URL http://doi.acm.org/10.1145/1645413.1645419
11. Gerstner, T., Kloeden, P. (eds.): Recent Developments in Computational Finance. Foundations, Algorithms and Applications. World Scientific Publishers, Interdisciplinary Mathematical Science (2013)
12. Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer-Verlag, New York (2003)
13. Griebel, M.: The combination technique for sparse grids solution of PDEs on multiprocessor machines. Parallel Processing Letters **3**, 66–71 (1992)
14. Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. In: P. de Groen, R. Beauwens (eds.) Proceedings of the IMACS International Symposium on Iterative Methods in Linear Albegra, pp. 263–281. Elsevier, Amsterdam, 1992, Brussels (1991)
15. Griebel, M., Thurner, V.: The efficient solution of fluid dynamics problems by the combination technique. International Journal of Numerical Methods for Heat and Fluid Flow **5**(3), 251–269 (1995)
16. Hagan, P., Kumar, D., Lesniewski, A., Woodward, D.: Managing Smile Risk. Wilmott Magazine pp. 84–108 (2002)

17. Hagan, P., Lesniewski, A.: LIBOR market model with SABR style stochastic volatility. Working paper, available at http://lesniewski.us/papers/working/SABRLMM.pdf (2008)
18. Hendricks, C., Heuer, C., Ehrhardt, M., Gunter, M.: Higher order ADI schemes for parabolic equations in the combinations technique and application to finance. Journal of Computational and Applied Mathematics (to appear 2017)
19. Jamshidian, F.: LIBOR and swap market models and measures. Finance and Stochastic **1**, 293–330 (1997)
20. Larson, J.W., Hegland, M., Harding, B., Roberts, S., Stals, L., Rendell, A.P., Strazdins, P., Ali, M.M., Kowitz, C., Nobes, R., Southern, J., Wilson, N., Li, M., Oishi, Y.: 2013 International Conference on Computational Science: Fault-Tolerant Grid-Based Solvers: Combining Concepts from Sparse Grids and MapReduce. Procedia Computer Science **18**, 130–139 (2013)
21. López-Salas, J.G., Vázquez, C.: PDE formulation of some SABR/LIBOR market models and its numerical solution with a sparse grid combination technique. Preprint submitted for publication (2016)
22. Mercurio, F., Morini, M.: No-Arbitrage dynamics for a tractable SABR term structure Libor Model. Modeling Interest Rates: Advances in Derivatives Pricing, Risk Books (2009)
23. Miltersen, K., Sandmann, K., Sondermann, D.: Closed-form solutions for term structure derivatives with lognormal interest rates. Journal of Finance **52**(1), 409–430 (1997)
24. Nvidia Corporation: CUDA C Programming guide
25. Rebonato, R., Mckay, K., White, R.: The SABR/LIBOR Market Model. Pricing, Calibration and Hedging for Complex Interest-Rate Derivatives, first edn. John Wiley & Sons (2009)
26. Richardson, L.F.: The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. Philosophical Transactions of the Royal Society A **210**(459-470), 307–357 (1911)
27. Shreve, S.E.: Stochastic Calculus for Finance. Springer (2004)
28. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Dokl. Akad. Nauk SSR **148**, 1042–1045 (1963)
29. Yserentant, H.: On the multi-level splitting of finite element spaces. Numerische Mathematik **49**, 379–412 (1986)
30. Yserentant, H.: Hierarchical bases. In: Proceedings of the Second International Conference on Industrial and Applied Mathematics, ICIAM 91, pp. 256–276. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992). URL http://dl.acm.org/citation.cfm?id=166337.166365
31. Zeiser, A.: Fast Matrix-Vector Multiplication in the Sparse-Grid Galerkin Method. Journal of Scientific Computing **47**(3), 328–346 (2010)
32. Zenger, C.: Sparse grids. In: W. Hackbusch (ed.) Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, vol. 31, pp. 241–251. Vieweg-Verlag, 1991, Kiel, Germany (1990)
33. OpenMP web page: http://www.openmp.org