AMFR-W NUMERICAL METHODS FOR SOLVING HIGH DIMENSIONAL SABR/LIBOR PDE MODELS*

J.G. LÓPEZ-SALAS†, S. PÉREZ-RODRÍGUEZ‡, AND C. VÁZQUEZ§

Abstract. In this work we mainly develop a new numerical methodology to solve a PDE model recently proposed in the literature for pricing interest rate derivatives. More precisely, we use high order in time AMFR-W methods, which belong to a class of W-methods based on Approximate Matrix Factorization (AMF) and are specially suitable in the presence of mixed spatial derivatives. High order convergence in time allows larger time steps which combined with the splitting of the involved operators, highly reduces the computational time for a given accuracy. Moreover, the consideration of a large number of underlying forward rates makes the PDE problem high dimensional in space, so the use of AMFR-W methods with a sparse grids combination technique represents another innovative aspect, making AMFR-W more efficient than with full grids and opening the possibility of parallelization. Also the consideration of new homogeneous Neumann boundary conditions provides another original feature to avoid the difficulties associated to the presence of boundary layers when using Dirichlet ones, specially in advection dominated regimes. These Neumann boundary conditions motivate the introduction of a modified combination technique to overcome a decrease in the accuracy of the standard combination technique.

Key words. SABR-LIBOR market models, high dimensional PDEs, AMFR-W methods, finite differences, sparse grids combination technique

AMS subject classifications. 65M06, 65M20, 65M50, 65W10, 91G30, 91G80

1. Introduction.

High dimensional parabolic Partial Differential Equations (PDEs) arise in many fields of science and engineering problems, as for example in computational biology for stochastic gene networks [2] or in computational finance for pricing financial derivatives [33], when a large number of underlying stochastic factors are involved in their equivalent stochastic formulations. In fact, each stochastic factor gives rise to one spatial-like variable in the corresponding PDE. In this high dimensional setting, when using finite differences for the spatial discretization, the complexity of standard grid based approaches grows exponentially with the dimensions of the problem as well as the computational times, thus giving rise to the so called *curse of dimensionality*. Thus, alternative techniques to the standard full grid are required. Also, the use of high order time integration schemes turns out to be very convenient to allow larger time steps and therefore reduce the computational time to get a prescribed accuracy.

In the present work we mainly propose a new numerical technique for solving the high dimensional PDE problem governing the Stochastic Alpha Beta Rho-LIBOR Market Model (SABR-LMM) PDE model introduced in [33]. The more classical LMM has been introduced for pricing interest rate derivatives which depend on the evolution

*Submitted to the editors DATE.

Funding: This work was funded by the Spanish Grants MTM2016-76497-R, PID2019-108584-RB-I00 and MTM2016-77735-C3-3-P, as well as Xunta de Galicia grant ED431C2018/033, all including FEDER funding. First and third authors acknowledge the support received from the Centro de Investigacin de Galicia "CITIC", funded by Xunta de Galicia and the European Union (European Regional Development Fund- Galicia 2014-2020 Program), by grant ED431G 2019/01.

†Department of Economics and CITIC, Universidade da Coruña, Campus de Elviña s/n, 15071 - A Coruña, Spain (jose.lsalas@udc.es).

‡Departamento de Análisis Matemático, Universidad de La Laguna, 38208, La Laguna, Tenerife, Canary Islands, Spain (sperezr@ull.edu.es).

§Department of Mathematics and CITIC, Universidade da Coruña, Campus de Elviña s/n, 15071 - A Coruña, Spain (carlosv@udc.es).

of a certain number of forward LIBOR rates, when their volatility was assumed to be constant (see [5], for example). More recently, the consideration of stochastic volatility has been incorporated for a better fit to market data by combining the classical LMM and the SABR model for stochastic volatility in [21, 36, 38], with different modelling approaches. In these works, the number of stochastic factors depends on the number of forward rates and volatilities that are considered. Their formulations are posed in terms of expectations to be solved by means of Monte Carlo techniques. Although we are aware of the evolution of LMM as a consequence of the financial crisis in 2007 to incorporate the practical presence of a multicurve setting (see [35], for example) and the recent ongoing studies related to LIBOR transitions not consolidated in market practice yet [34], we have chosen to start from the classical version of LMM as described in [5].

More recently, in [33] a PDE formulation is obtained for the Mercurio and Morini model presented in [36] and a set of numerical methods are proposed to solve it. More precisely, the combination of standard finite differences in space and a $\theta-$method in time are proposed on uniform full grids. Moreover, by arguing that these standard finite difference methods based on traditional full grids are not able to price interest rate derivatives with more than three or four stochastic forward rates, a sparse grid combination technique is applied. A rigorous analysis of finite differences schemes in the sparse grid combination technique in arbitrary dimensions is carried out in [39]. In order to overcome the curse of dimensionality one can try to use high order discretizations in time and space, mainly to reduce the required discretization points to achieve certain accuracy. Although for sparse grids in space there are lots of works (see [26] and the references therein), for time discretization only schemes up to order two have been applied. Following this objective, unlike in [33], in the present article we propose higher order discretization techniques in time based on a special class of W-methods [41], the AMFR-W methods introduced in [12].

These AMFR-W-methods are specially suitable for parabolic problems involving mixed spatial derivatives as it is the case in the SABR-LMM model proposed in [33]. In [12] it is shown that they are unconditionally stable regardless the spatial dimension on linear constant coefficients PDEs with mixed derivative terms with both periodic boundary conditions and homogeneous Dirichlet boundary conditions. Moreover, as the ADI methods [26, 28], the AMFR-W scheme takes advantage of the structure of the linear system obtained from the spatial discretization, so it can be decomposed into tridiagonal systems which can be solved in linear run-time. Thus, the computational effort is significantly reduced. However, while classical ADI schemes are of order two in time, the proposed AMFR-W scheme exhibits order three in time when standard full grids are used. An additional innovative aspect of the present work is the application of the AMFR-W methods in the context of sparse grids combination technique, which turns out to be an efficient tool for solving the SABR-LMM model in the required high dimensional setting. Furthermore, the introduction of more appropriate Neumann boundary conditions motivates the consideration of a modified combination technique to improve the convergence.

LMMs are usually simulated by means of Monte Carlo method, in contrast with the here proposed PDE methodology. We aim to avoid the slow Monte Carlo rate of convergence, $O(1/\sqrt{M})$ for all dimensions, M being the number of simulations.

The plan of the article is the following. In Section 2 we present the PDE model and justify the introduction of new homogeneous Neumann boundary conditions at the so called outflow boundaries. In Section 3, we introduce the space discretization of the PDE problem with finite differences to obtain an ODE system in suitable form for

the application of the AMFR-W method. In Section 4 we describe the application of AMFR-W method to the ODE system to get the fully discretized problem. Section 5 is devoted to the methodology of sparse grids, including the standard and the modified combination techniques. In Section 6 we present and discuss the numerical results obtained for full and sparse grids. Finally, Section 7 contains some conclusions.

2. PDE formulation of the SABR-LMM model. As indicated in the previous section, we mainly address a new and more efficient numerical solution of the PDE formulation introduced in [33] for the SABR-LMM proposed by Mercurio and Morini in [36] to price a financial derivative which depends on a certain number of forward LIBOR rates, i.e a swaption. In this section we introduce the PDE model and we incorporate some new boundary conditions. Although we refer the reader to [33] for the statement of the model and further details, we need to introduce some financial concepts and their notations related to interest rates derivatives to be used along this article. In this respect, we also refer the reader to the textbook [5].

A zero coupon *bond* with maturity at time T pays its holder one unit of currency at time T. The zero coupon value at time $t < T$ will be denoted by $P(t,T)$, and is also referred as the discount factor from time T to time t. A *tenor structure* is defined as a set of ordered payment dates $T_0 < T_1 < \ldots < T_{N-1} < T_N$. The time gap between two consecutive tenor dates is denoted by $\tau_i = T_{i+1} - T_i$. In view of previous definitions, a payment of x units at time T_i is worth $xP(t,T_i)$ at time $t < T_i$.

Next, we introduce the forward rates that enter in the LMM. We consider the *forward* interest rate $F_i(t)$ as an interest rate we can contract to borrow or lend money during the future time period $[T_i, T_{i+1}]$, which is fixed at time T_i. Moreover, the value of $F_i(t)$ can be expressed in terms of discount factors in the form:

$$F_i(t) = F(t;T_i,T_{i+1}) = \frac{1}{\tau_i}\left(\frac{P(t,T_i)}{P(t,T_{i+1})} - 1\right) \text{ where } t \le T_i.$$

Conversely, the price of a zero coupon bond at time T_i that matures at T_j, $P(T_i,T_j)$, can be expressed in terms of forward LIBOR rates as

$$P(T_i,T_j) = \prod_{k=i}^{j-1} \frac{1}{1+\tau_k F_k(T_i)}.$$

Among all interest rate derivatives, the simplest one is the caplet. A *caplet* is a European call option on a forward rate. Thus, if the maturity of a caplet is T_{i+1}, at that time the holder of the caplet receives the payoff $\tau_i(F_i(T_i)-K)^+$, so its discounted payoff at time $t < T_{i+1}$ is given by $P(t,T_{i+1})\tau_i(F_i(T_i)-K)^+$, where $(\cdot)^+$ denotes the function $\max(\cdot,0)$ and K is the strike (a fixed interest rate) of the caplet. If constant volatilities are assumed as it is the case in the classical LMM, the caplet price can be analytically computed with a Black's formula (see [5], for details).

An interest rate *swap* (IRS) is a contract to exchange interest payments at future fixed dates. At every time instant in the prescribed set of dates $T_{a+1},\ldots,T_b$ the contract holder pays a fixed interest rate K and receives a floating forward LIBOR rate $F_i(T_i)$, which is fixed at time T_i. At time T_a the value of the IRS is given by

$$\text{IRS}(T_a;T_a,\ldots,T_b) = \sum_{i=a}^{b-1} P(T_a,T_{i+1})\tau_i(F_i(T_a)-K). \tag{2.1}$$

A European $T_a \times (T_b - T_a)$ *swaption* is an option that gives the right to enter a swap at the future time T_a (swaption maturity). The underlying swap length $T_b - T_a$

is referred as the tenor of the swaption. Therefore, the discounted swaption payoff to time t is equal to $P(t,T_a)\big(\mathrm{IRS}(T_a;T_a,\ldots,T_b)\big)^+$.

In the forthcoming section devoted to numerical results, several examples address the pricing of caplets and swaptions. Note that the payoff of a caplet just involves one forward rate, therefore its price at any time before maturity only depends on this particular forward rate. In the case of IRS or swaptions, their payoffs at expiration date depend on a certain number of forward rates, so their price at any time before expiration will also depend on them.

In the Mercurio and Morini model it is assumed that a generic European interest rate derivative depends on the evolution of $N-1$ forward rates, F_1, $F_2,\ldots,F_{N-1}$ associated to bonds with maturity related to the tenor structure, and a common stochastic volatility, V. Let $u=u(t,F_1,F_2,\ldots,F_{N-1},V)$ denote the value at time $t\in[0,T]$ of this European interest rate derivative, with forward rates $F_i\in[0,F_i^{max}]$, $i=1,2,\ldots,N-1$, and volatility $V\in[0,V^{max}]$. Note that the previously described caplets, IRS and swaptions are particular cases. As stated in [33], the function u satisfies the following PDE

$$\begin{aligned}\frac{\partial u}{\partial t}+\frac{1}{2}\sigma^2V^2\frac{\partial^2u}{\partial V^2}+\frac{1}{2}V^2\sum_{i,j=1}^{N-1}\alpha_i\alpha_j\rho_{ij}F_i^\beta F_j^\beta\frac{\partial^2u}{\partial F_i\partial F_j}\\ +\sigma V^2\sum_{i=1}^{N-1}\alpha_i\phi_iF_i^\beta\frac{\partial^2u}{\partial F_i\partial V}+\sum_{i=1}^{N-1}\mu_iF_i^\beta\frac{\partial u}{\partial F_i}\quad=0,\end{aligned}\tag{2.2}$$

where σ is the volatility of the stochastic volatility V, α_i is a deterministic (constant) instantaneous volatility coefficient of the forward rate F_i, ρ_{ij} is the correlation between the forward rates F_i and F_j, ϕ_i is the correlation between F_i and the stochastic volatility V and μ_i is the drift of the i-th forward rate. Moreover, when the bond $P(t,T_1)$ is chosen as the numeraire, the drifts μ_i depend on the forward rates as $\mu_1=0$, $\mu_i=\alpha_iV^2\sum_{j=2}^{i}\frac{\tau_jF_j^\beta}{1+\tau_jF_j}\rho_{ij}\alpha_j$, $i\geq2$. The parameter $\beta\in[0,1]$ is the elasticity of variance, which usually is 0, 0.5 or 1, that corresponds to stochastic volatilities with normal, CIR or log-normal dynamics. For the correlation structure, as in [33] we consider the expression $\rho_{ij}=\mathrm{e}^{-\lambda|T_i-T_j|}$, which depends on the constant parameter λ.

In view of the form of the differential operator governing the PDE, (2.2) must be completed with a final condition $u(T,F_1,F_2,\ldots,F_{N-1},V)=g(T,F_1,F_2,\ldots,F_{N-1})$, where g represents the derivative payoff, the expression of which depends on the interest rate derivative we are dealing with.

In order to apply the method of lines (MoL) to discretize the previous model it is more convenient to write (2.2) in terms of the time to maturity $T-t$ instead of the physical time t, so that the final condition turns into an initial condition. In an abuse of notation we keep the notation t for the new formulation after this change in the time variable. More precisely, we rewrite the model by denoting $u=u(t,F_1,F_2,\ldots,F_{N-1},F_N)$ the value of the interest rate derivative at time $T-t$, $t\in[0,T]$, with forward rates $F_i\in[0,F_i^{max}]$, $i=1,2,\ldots,N-1$, and volatility $V\in[0,V^{max}]$. After some easy calculus, we obtain the equation

$$\frac{\partial u}{\partial t}=\sum_{i=1}^{N}d_i\frac{\partial^2u}{\partial F_i^2}+\sum_{i=1}^{N-1}\sum_{k=i+1}^{N}m_{ik}\frac{\partial^2u}{\partial F_i\partial F_k}+\sum_{i=2}^{N-1}a_i\frac{\partial u}{\partial F_i},\tag{2.3}$$

with the *initial condition*

$$u(0,F_1,F_2,\ldots,F_{N-1},F_N)=g(T,F_1,F_2,\ldots,F_{N-1}),\tag{2.4}$$

where

$$d_i = d_i(F_i, F_N) = \begin{cases} \frac{1}{2}\alpha_i^2 \rho_{ii} F_i^{2\beta} F_N^2, & \text{if } 1 \le i \le N-1, \\ \frac{1}{2}\sigma^2 F_N^2, & \text{if } i = N. \end{cases} \tag{2.5}$$

Moreover, for $i = 1, \ldots, N-1$, $k = i+1, \ldots, N$, we define

$$m_{ik} = m_{ik}(F_i, F_k, F_N) = \begin{cases} \alpha_i \alpha_k \rho_{ik} F_i^{\beta} F_k^{\beta} F_N^2, & \text{if } i+1 \le k \le N-1, \\ \alpha_i \sigma \phi_i F_i^{\beta} F_N^2, & \text{if } k = N. \end{cases} \tag{2.6}$$

Note that the last term in (2.3) is only defined for $N \ge 3$. In this case, for each $i = 2, \ldots, N-1$ we define:

$$a_i = a_i(F_2, \ldots, F_i, F_N) = \left(\sum_{j=2}^{i} \alpha_i \alpha_j \rho_{ij} \Phi_\beta(F_j, \tau_j) \right) F_i^{\beta} F_N^2, \tag{2.7}$$

where Φ_β is the scalar function $\Phi_\beta(x, \tau) := \dfrac{\tau x^\beta}{1+\tau x}$, $\tau > 0$, $x \ge 0$.

In next paragraphs we will discuss about the appropriate boundary conditions to add to (2.3)-(2.4) to define the initial-boundary value problem. First, note that in [33] the following time-independent boundary conditions were considered to complete the formulation (2.3)-(2.4):

$$\begin{aligned} &\text{if } F_j = 0 \text{ or } F_j = F_j^{max},\ 1 \le j \le N-1, \\ &\qquad u(t, F_1, \ldots, F_{N-1}, F_N) = g(T, F_1, \ldots, F_{N-1}), \\ &u(t, F_1, \ldots, F_{N-1}, 0) = u(0, F_1, \ldots, F_{N-1}, 0) = g(T, F_1, \ldots, F_{N-1}), \\ &\frac{\partial u}{\partial V}(t, F_1, \ldots, F_{N-1}, V^{max}) = 0. \end{aligned} \tag{2.8}$$

These boundary conditions are appropriate when $\beta = 0$ since the PDE coefficients are independent of the forward rates (F_j, $j \le N-1$) and the advection terms are moderate. When $\beta > 0$, they also can be appropriate when we have a small number of forward rates and a derivative with short maturity. Nevertheless, as soon as this number increases then the dimension of the PDE grows, and the advection-dominance of the PDE (2.3) becomes more relevant. It is well-known that imposing a Dirichlet boundary condition at the outflow boundaries $F_i = F_i^{max}$, $i = 1, \ldots, N-1$, in an advection-dominated setting could give rise to boundary layers (see, for instance, [27, Sect. I.5]), and some previous numerical results with this model corroborate that.

When $\beta \in (0, 1]$, in order to show that the larger the dimension N the more advection-dominant the PDE (2.3) becomes, it is better to write the PDE (2.3) in the following conservative form

$$u_t + \nabla \cdot (\underline{c} u) = \nabla \cdot (D \nabla u) + s(\underline{x}, t, u), \quad \underline{x} \in \Omega \subset \mathbb{R}^N,\ t > 0\,, \tag{2.9}$$

where $\underline{x} = (F_1, \ldots, F_N)$, $\underline{c} = (c_1, \ldots, c_N)^T$, $D = (D_{ik})_{i,k=1}^N$, $c_i = c_i(\underline{x}, t)$, $D_{ik} = D_{ik}(\underline{x}, t)$.

Note that in multi-dimensional advection-diffusion-reaction PDEs with variable coefficients, some additional advection terms arise in the conservative form coming from the partial derivatives of the coefficients of the second order diffusion terms.

After some manipulations it can be proved that PDE (2.3) admits an expression of type (2.9) with $D_{ii} = d_i$, $D_{ik} = m_{ik}/2$, $\forall i \neq k$, and

$$
\begin{aligned}
&c_i = \left(P_i^+ - P_i^-\right)\alpha_i F_i^\beta F_N^2 + \alpha_i \phi_i F_i^\beta \sigma F_N, \quad 1 \leq i \leq N-1,\\
&c_N = P_N^+ \sigma F_N^2 + \sigma^2 F_N,
\end{aligned} \tag{2.10}
$$

where

$$
\begin{aligned}
P_i^+ = &\quad \alpha_i \frac{\beta}{F_i^{1-\beta}} + \sum_{\substack{j=1\\ j\neq i}}^{N-1} \frac{1}{2}\alpha_j \rho_{ij} \frac{\beta}{F_j^{1-\beta}}, \quad 1 \leq i \leq N-1,\\
P_N^+ = &\quad \sum_{j=1}^{N-1} \frac{1}{2}\alpha_j \phi_j \frac{\beta}{F_j^{1-\beta}}, \quad P_1^- = 0, \quad P_2^- = \alpha_2\, \Phi_\beta(F_2, \tau_2),\\
P_i^- = &\quad \alpha_i \Phi_\beta(F_i, \tau_i) + \sum_{j=2}^{i-1} \alpha_j \rho_{ij} \Phi_\beta(F_j, \tau_j), \quad 3 \leq i \leq N-1.
\end{aligned}
$$

Moreover, the reaction term takes the form $s(\underline{x}, t, u) = \delta u$, $\delta = P\, F_N^2 + Q\, \sigma F_N + \sigma^2$, where P and Q depend on the forward rates $F_1, \ldots, F_{N-1}$.

Obviously, when $0 < \beta < 1$ the PDE (2.3) is equivalent to (2.9) whenever $F_j > 0$, $\forall j = 1, \ldots, N-1$, due to the lack of differentiability of $\underline{c}$ when $F_j = 0$. Since we intend to use this expression (2.9) to study the behaviour of the PDE at the boundaries $F_j = F_j^{max}$, we will assume for now that $F_j > 0$, $1 \leq j \leq N-1$. On the other hand, when $\beta = 1$ this additional assumption is not needed.

In most of practical cases we have that

$$
\tau_i F_i^{max} \leq \frac{\beta}{2-\beta}, \quad 2 \leq i \leq N-2, \qquad \tau_{N-1} F_{N-1}^{max} \leq \frac{\beta}{1-\beta}, \tag{2.11}
$$

what implies that, for $2 \leq i \leq N-1$[1],

$$
\begin{aligned}
P_i^+ - P_i^- = &\quad \alpha_i \left(\frac{\beta}{F_i^{1-\beta}} - \Phi_\beta(F_i, \tau_i) \right) + \sum_{j=2}^{i-1} \frac{1}{2}\alpha_j \rho_{ij} \left(\frac{\beta}{F_j^{1-\beta}} - 2\Phi_\beta(F_i, \tau_i) \right)\\
&+ \frac{1}{2}\alpha_1 \rho_{i1} \frac{\beta}{F_1^{1-\beta}} + \sum_{j=i+1}^{N-1} \frac{1}{2}\alpha_j \rho_{ij} \frac{\beta}{F_j^{1-\beta}} \geq 0.
\end{aligned} \tag{2.12}
$$

We must observe that in the case $\beta = 1$,

$$
\begin{aligned}
&\frac{\beta}{F_i^{1-\beta}} - \Phi_\beta(F_i, \tau_i) = 1 - \frac{\tau_i F_i}{1+\tau_i F_i} = \frac{1}{1+\tau_i F_i} \geq 0,\\
&\frac{\beta}{F_j^{1-\beta}} - 2\Phi_\beta(F_j, \tau_j) = 1 - 2\frac{\tau_i F_i}{1+\tau_i F_i} = \frac{1-\tau F_j}{1+\tau_i F_j} \geq 0 \quad \text{if } \tau_j F_j \leq 1,
\end{aligned}
$$

and the second requirement in (2.11) is superfluous.

Therefore, $c_i \geq 0$, $i = 1, \ldots, N$, so it is clear that all the boundaries $F_i = F_i^{max}$ $1 \leq i \leq N$ are outflow boundaries since the outward normal vector on each one of these borders is the $N-$dimensional canonical vector $\mathbf{e}_i = (e_{ik})_{k=1}^N$, $e_{ii} = 1$, $e_{ik} \neq 0$, $\forall k \neq i$.

[1] In the sequel, $\sum_{i=j}^{k}(\cdot) = 0$ when $j > k$.

On the other hand, we need to take into account that these advection coefficients depend on N, $c_i = c_i^{(N)}$, and $F_N = V$ so if we increase the number of forward rates from $\{F_1, \ldots, F_{N-2}\}$ to $\{F_1, \ldots, F_{N-2}, F_{N-1}\}$, we have that

$$c_i^{(N)} = c_i^{(N-1)} + \frac{1}{2}\alpha_i \alpha_{N-1} \rho_{i,N-1} \frac{\beta}{F_{N-1}^{1-\beta}} F_i^{\beta} V^2, \quad i = 1, \ldots, N-2. \tag{2.13}$$

Since usually $F_i^{max} = F^{max}$, $i = 1, \ldots, N-1$ and $V^{max} >> F^{max}$, we can see that the advection increases with N whereas the diffusion coefficients do not.

A way to avoid the boundary layers that this advection-dominance can produce with conditions (2.8) when $0 < \beta \leq 1$, is to consider homogeneous Neumann boundary conditions instead [27], i.e.,

$$\begin{aligned} &u(t, F_1, \ldots, F_{N-1}, F_N) = g(T, F_1, \ldots, F_{N-1}), \quad \text{if } F_j = 0,\ 1 \leq j \leq N, \\ &\frac{\partial u}{\partial F_j}(t, F_1, \ldots, F_{N-1}, F_N) = 0, \quad \text{if } F_j = F_j^{max},\ 1 \leq j \leq N-1, \text{ or } F_N = V^{max}. \end{aligned} \tag{2.14}$$

On the one hand, these homogeneous Neumann boundary conditions are appropriate from the financial point of view. Actually, if we analyse the behaviour of the payoff of the swaption $T_1 \times (T_N - T_1)$, which is given by

$$g(T, F_1, F_2, \ldots, F_{N-1}) = \max\left\{ \sum_{i=1}^{N-1} \frac{\tau_i (F_i - K)}{(1+\tau_1 F_1) \cdots (1+\tau_i F_i)}, 0 \right\},$$

when some forward rate $F_j \to \infty$, we can consider the approximation

$$g(T, F_1, \ldots, F_{N-1}, V) \approx f(F_1, \ldots, F_{N-1}) := \sum_{i=1}^{N-1} \frac{\tau_i (F_i - K)}{P_i},$$

where $P_i := \prod_{l=1}^{i} (1 + \tau_l F_l)$, $1 \leq i \leq N-1$, and prove that

$$\frac{\partial f}{\partial F_j}(T, F_1, \ldots, F_{N-1}, V) \longrightarrow 0 \quad \text{when } F_j \to \infty, \quad j = 1, \ldots, N-1. \tag{2.15}$$

In order to prove this, it is enough to write the partial derivatives of f as

$$\frac{\partial f}{\partial F_j}(F_1, \ldots, F_{N-1}) = \frac{\tau_j}{(1+\tau_1 F_1) \cdots (1+\tau_{j-1} F_{j-1})(1+\tau_j F_j)^2} \left((1+\tau_j K) - \sum_{i=j+1}^{N-1} \frac{\tau_i (F_i - K)}{\tilde{P}_{ij}} \right),$$

where $\tilde{P}_{ij} = P_i / P_j = (1+\tau_{j+1} F_{j+1}) \cdots (1+\tau_i F_i)$, $i \geq j+1$. As in [33], an homogeneous Neumann boundary condition is imposed at $F_N = V^{max}$ because the price of the derivative becomes independent of V when V approaches to infinity.

On the other hand, the choice (2.14) becomes better when some discretization of the "spatial" variables F_j is used to approximate the solution of PDE (2.9) on a uniform spatial grid, as will be illustrated later on.

3. Space discretization with finite differences. Following the ideas given in [33], firstly a space discretization is performed on a uniform spatial grid on $\Omega = [0, F_1^{max}] \times [0, F_2^{max}] \times \cdots \times [0, F_{N-1}^{max}] \times [0, V^{max}]$. However, we propose here a different time discretization by using the AMFR-W-methods introduced in [12].

For the space discretization we consider N integers $(M_1, \ldots, M_{N-1}, M_N)$ to define on Ω the spatial grid with $M_i + 1$ equally spaced points (denoting $F_N^{max} = V^{max}$) at the F_i-direction $F_{i,j_i} = j_i h_i$, $0 \leq j_i \leq M_i$, $h_i = \dfrac{F_i^{max}}{M_i}$, $1 \leq i \leq N$, and discretize the derivatives in (2.3) with second order central finite differences at each spatial node $(F_{1,j_1}, \ldots, F_{N-1,j_{N-1}}, F_{N,j_N})$. More precisely, the MoL approximates the solution at each spatial point $u(t, F_{1,j_1}, \ldots, F_{N-1,j_{N-1}}, F_{N,j_N}) \approx U_{j_1,\ldots,j_{N-1},j_N}(t)$, where the values $U_{j_1,\ldots,j_{N-1},j_N}(t)$ need to satisfy the semi-discretized ODE system

$$
\begin{aligned}
\frac{d}{dt} U_{j_1,\ldots,j_{N-1},j_N}(t) = & \sum_{i=1}^{N} d_i(F_{i,j_i}, F_{N,j_N})\, \Delta^{(i)}_{j_1,\ldots,j_{N-1},j_N} \\
& + \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} m_{ik}(F_{i,j_i}, F_{k,j_k}, F_{N,j_N})\, \Delta^{(ik)}_{j_1,\ldots,j_{N-1},j_N} \\
& + \sum_{i=2}^{N-1} a_i(F_{2,j_2}, \ldots, F_{i,j_i}, F_{N,j_N})\, \nabla^{(i)}_{j_1,\ldots,j_{N-1},j_N},
\end{aligned} \tag{3.1}
$$

with $\Delta^{(i)}$, $\Delta^{(ik)}, \nabla^{(i)}$ representing the approximations with central differences of the derivatives $(\partial^2 u/\partial F_i^2)$, $(\partial^2 u/\partial F_i \partial F_k)$, $(\partial u/\partial F_i)$, respectively.

One important drawback of finite differences comes from the complexity of dealing with all these approximations when the dimension N is large. For the efficient manipulation of these differences, we propose to use $N-$dimensional multi-indices $\mathbf{j} = (j_1, \ldots, j_N)$, and the following Lemma 3.1 that is proved in the Appendix.

LEMMA 3.1. *Given N pairs of integers $m_i \leq M_i$ for $i = 1, \ldots, N$, let us define the set $\mathcal{I}_N = \{\mathbf{j} = (j_1, \ldots, j_N) \mid m_i \leq j_i \leq M_i,\ \forall i = 1, \ldots, N\}$. Moreover, for $M_T = \prod_{k=1}^{N} (M_k - m_k + 1)$, define the map $\vartheta : \mathcal{I}_N \longrightarrow \{m_1, m_1 + 1, \ldots, M_T + m_1 - 1\}$,*

$$
\vartheta(\mathbf{j}) = J = j_1 + \sum_{l=2}^{N} \left((j_l - m_l) \prod_{r=1}^{l-1} (M_r - m_r + 1) \right). \tag{3.2}
$$

Then, the map ϑ is bijective.

Besides the proof of Lemma 3.1, in the Appendix it is also included a practical way for computing the inverse map $\vartheta^{-1}(J) = \mathbf{j}$ (see (A.3) in Lemma A.1).

Note that due to the Dirichlet boundary conditions in (2.14), the values for $j_i = 0$ for $1 \leq i \leq N$ are given by the derivative payoff. As a consequence, the ODE system (3.1) is applied only when $j_i = 1, \ldots, M_i$, $1 \leq i \leq N$, so it has dimension $L = M_1 \cdots M_{N-1} M_N$. Then, we separate the multi-indices that correspond with finite differences nodes on the lower boundaries from the rest of them, so that two different bijections of type (3.2) are considered:

$$
\vartheta_0 : \mathcal{I}_N^{(0)} = \{\mathbf{j} = (j_1, \ldots, j_N) \mid 0 \leq j_i \leq M_i,\ \forall i = 1, \ldots, N\} \longrightarrow \{0, 1, \ldots, M - 1\},
$$

$$
\vartheta_1 : \mathcal{I}_N^{(1)} = \{\mathbf{k} = (k_1, \ldots, k_N) \mid 1 \leq k_i \leq M_i,\ \forall i = 1, \ldots, N\} \longrightarrow \{1, 2, \ldots, L\},
$$

where $M=\prod_{k=1}^{N}(M_k+1)$, $\mathcal{I}_N^{(1)}\subset\mathcal{I}_N^{(0)}$, and

$$\vartheta_0(\mathbf{j})=J=j_1+\sum_{l=2}^{N}\left(j_l\prod_{r=1}^{l-1}(M_r+1)\right),\quad \mathbf{j}\in\mathcal{I}_N^{(0)},$$
$$\vartheta_1(\mathbf{k})=K=k_1+\sum_{l=2}^{N}\left((k_l-1)\prod_{r=1}^{l-1}M_r\right),\quad \mathbf{k}\in\mathcal{I}_N^{(1)}.$$

The set of integers $\{0,1,\dots,M-1\}$ can be obtained as the union of the two disjoint sets $\mathsf{Inner}=\vartheta_0\left(\mathcal{I}_N^{(1)}\right)=\vartheta_0\left(\vartheta_1^{-1}\left(\{1,\dots,L\}\right)\right)$, $\mathsf{Outer}=\{0,1,\dots,M-1\}-\mathsf{Inner}$, and we consider the vector $Y(t)=(Y_J(t))_{J=0}^{M-1}$, where for each $J=0,\dots,M-1$, $(j_1,\dots,j_{N-1},j_N)=\vartheta_0^{-1}(J)$,

$$Y_J(t)=\begin{cases}U_{j_1,\dots,j_{N-1},j_N}(t), & \text{if } J\in\mathsf{Inner},\\ g(T,F_{1,j_1},\dots,F_{N-1,j_{N-1}}), & \text{if } J\in\mathsf{Outer}.\end{cases}\tag{3.3}$$

Note that $\vartheta_0(\mathbf{j}-\mathbf{e}_i)\in\{0,1,\dots,M-1\}$, for all $J\in\mathsf{Inner}$ with $\mathbf{j}=\vartheta_0^{-1}(J)\in\mathcal{I}_N^{(1)}$. Moreover, $\vartheta_0(\mathbf{j}-\mathbf{e}_i)=J-E_i$, for $E_1=1$, $E_i=\prod_{r=1}^{i-1}(M_r+1)$, $i\geq 2$. On the other hand, taking into account boundary conditions (2.14), for all $J\in\mathsf{Inner}$ we have that $\mathbf{j}=\vartheta_0^{-1}(J)\in\mathcal{I}_N^{(1)}$ and $\vartheta_0(\mathbf{j}+\mathbf{e}_i)\in\{0,1,\dots,M-1\}$ except when $j_i=M_i$, since $\mathbf{j}+\mathbf{e}_i=(\dots,M_i+1,\dots)\notin\mathcal{I}_N^{(0)}$. So, if $J\in\mathsf{Inner}$, then $J+E_i=\vartheta_0(\mathbf{j}+\mathbf{e}_i)$, for all $i=1,\dots,N$. However, for the case $j_i=M_i$ we take the virtual value $Y_{J+E_i}=Y_{J-E_i}$ due to the homogeneous Neumann conditions (2.14) at these boundaries. Therefore, taking into account (3.3), the ODE system (3.1) is shown in Sketch 1.

SKETCH 1. *ODE system* (3.1)

for $J\in$ *Outer* ***do***
  $Y_J'(t)=0$
end for
for $J\in$ *Inner* ***do***
  $\mathbf{j}=\vartheta_0^{-1}(J)=(j_1,\dots,j_N)$
  for $i=1,\dots,N$ ***do***
    $(d_i)_J=d_i(F_{i,j_i},F_{N,j_N});\quad (a_i)_J=a_i(F_{2,j_2},\dots,F_{i,j_i},F_{N,j_N})$

$$\Delta_J^{(i)}=\begin{cases}\dfrac{Y_{J+E_i}-2Y_J+Y_{J-E_i}}{h_i^2}, & \text{if } j_i\neq M_i\\ \dfrac{2Y_{J-E_i}-2Y_J}{h_i^2}, & \text{if } j_i=M_i\end{cases}$$

$$\nabla_J^{(i)}=\begin{cases}\dfrac{Y_{J+E_i}-Y_{J-E_i}}{2h_i}, & \text{if } j_i\neq M_i\\ 0, & \text{otherwise}\end{cases}$$

  end for
  for $1\leq i\leq N-1,\ i+1\leq k\leq N$ ***do***
    $(m_{ik})_J=m_{ik}(F_{i,j_i},F_{k,j_k},F_{N,j_N})$

$$\Delta_J^{(ik)}=\begin{cases}\dfrac{Y_{J+E_i+E_k}+Y_{J-E_i-E_k}-Y_{J+E_i-E_k}-Y_{J-E_i+E_k}}{4h_ih_k}, & \text{if } j_i\neq M_i\\ & \text{and } j_k\neq M_k\\ 0, & \text{otherwise}\end{cases}$$

  end for

$$Y'_J(t) = \sum_{i=1}^{N} (d_i)_J \ \Delta_J^{(i)} + \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} (m_{ik})_J \ \Delta_J^{(ik)} + \sum_{i=2}^{N-1} (a_i)_J \ \nabla_J^{(i)}$$

end for

Thus, we obtain the semi-discretized *autonomous initial value problem (IVP)*

$$Y' = \mathcal{F}(Y),\ Y(0) = Y_0,\ t \in [0,T], \qquad \mathcal{F}(Y) = \mathcal{F}_0(Y) + \sum_{i=1}^{N} \mathcal{F}_i(Y). \tag{3.4}$$

In the splitting of the derivative $\mathcal{F}$, for each $i = 1, \ldots, N$, the term $\mathcal{F}_i(Y)$ contains the second order differences in the F_i-direction. That is, for all $J = 0, 1, \ldots, M-1$,

$$\left(\mathcal{F}_i(Y)\right)_J = \begin{cases} (d_i)_J \ \Delta_J^{(i)}, & \text{if } J \in \mathsf{Inner},\ (j_1, \ldots, j_N) = \vartheta_0^{-1}(J), \\ 0, & \text{if } J \in \mathsf{Outer}, \end{cases} \tag{3.5}$$

while the term $\mathcal{F}_0(Y)$ gathers the discretization corresponding to the remaining terms.

Clearly, $\forall i = 0, 1, \ldots, N$, $\mathcal{F}_i(Y) = \mathcal{A}_i Y$, where for all $J \in \mathsf{Outer}$, the J-th row of the constant matrix $\mathcal{A}_i$ is null. Besides, when $i \geq 1$ and $J \in \mathsf{Inner}$, the J-th row of $\mathcal{A}_i$ has at most three non-zero elements, located at the columns $J - E_i$, J and $J + E_i$. Therefore, the differential system $Y' = \mathcal{F}(Y)$ in (3.4) involves actually L unknowns since the corresponding equation for each $J \in \mathsf{Outer}$ is null. Once the semi-discretized linear IVP (3.4) has been posed, the AMFR-W-methods given in [12] can be applied for its time integration.

4. Time discretization.

4.1. AMFR-W-methods. AMFR-W-methods belong to the class of W-methods [41] for the time integration of IVPs of type (3.4). Thus, from an approximation Y_n of the solution $Y(t)$ at $t = t_n$ and a step size $\Delta t > 0$, an $s-$stage W-method gives the approximation Y_{n+1} at $t_{n+1} = t_n + \Delta t$ by

$$\begin{aligned} (I - \theta \Delta t W) K_r &= \Delta t \mathcal{F}\Big(Y_n + \sum_{j=1}^{r-1} a_{rj} K_j\Big) + \sum_{j=1}^{r-1} q_{rj} K_j, \quad r = 1, 2, \ldots, s, \\ Y_{n+1} &= Y_n + \sum_{r=1}^{s} b_r K_r. \end{aligned} \tag{4.1}$$

Each W-method is characterized by its coefficients (A, Q, b, θ), where $A = (a_{rj})_{j<r}$, $Q = (q_{rj})_{j<r}$ and $b = (b_r)$, and by the arbitrary matrix W. These kind of methods can also be understood as a generalization of Rosenbrock methods, which are obtained when $W = \mathcal{F}'(Y_n)$. In order to get W-methods of high order, W must be some rough approximation of $\mathcal{F}'(Y_n)$, and methods of order 3 and higher can be found in the literature under the assumption (see, for instance, [11, 37, 29, 15])

$$W - \mathcal{F}'(Y_n) = \mathcal{O}(\Delta t), \qquad \Delta t \to 0. \tag{4.2}$$

When $\mathcal{F}'(Y)$ admits a directional splitting of type (3.4), i.e. $\mathcal{F}'(Y_n) = \mathcal{F}'_0(Y_n) + \sum_{i=1}^{N} \mathcal{A}_i$, where the matrices $\mathcal{A}_i$ have simple structures, in [12] the authors propose

$$\Big(I - \theta \Delta t W\Big)^{-1} = \prod_{i=N}^{1} \Big(I - \nu \Delta t \mathcal{A}_i\Big)^{-1} \Big(2I - \big(I - \theta \Delta t \mathcal{F}'(Y_n)\big) \prod_{i=N}^{1} \Big(I - \nu \Delta t \mathcal{A}_i\Big)^{-1}\Big), \tag{4.3}$$

thus combining the Approximate Matrix Factorization (AMF) technique from [42, 27] for the matrix $I - \theta\Delta t W$ with a refinement to the solution of the linear systems [14, Section 3]. With this selection of the matrix W, the condition (4.2) is fulfilled and the introduction of a new parameter ν allows to improve the stability of the method.

Expanding these formulas on the semi-discretized IVP (3.4), each stage of the resulting AMFR-W-method (4.1)-(4.3) is computed as

$$\begin{aligned} K_r^{(0)} &= \Delta t\mathcal{F}(Y_n + \textstyle\sum_{j=1}^{r-1} a_{rj}K_j) + \sum_{j=1}^{r-1} q_{rj}K_j, \\ (I - \nu\Delta t\mathcal{A}_i)K_r^{(i)} &= K_r^{(i-1)}, \quad i = 1, \ldots, N, \\ \hat{K}_r^{(0)} &= 2K_r^{(0)} - (I - \theta\Delta t\mathcal{F}'(Y_n))K_r^{(N)}, \\ (I - \nu\Delta t\mathcal{A}_i)\hat{K}_r^{(i)} &= \hat{K}_r^{(i-1)}, \quad i = 1, \ldots, N, \\ K_r &= \hat{K}_r^{(N)}. \end{aligned} \tag{4.4}$$

In [12], different choices for the coefficients of these methods are tested. In this article we have used the proposed **AMFR-W2**, that is a 2-stage AMFR-W-method, with coefficients [27, p. 400]

$$a_{21} = 2/3, \quad q_{21} = -4/3, \quad b_1 = 5/4, \quad b_2 = 3/4. \tag{4.5}$$

Since (4.2) is fulfilled, this method is of order 3 for $\theta = (3+\sqrt{3})/6$ (in ODE sense). In [12], the authors also proved that the choice of the parameter ν depends on the number N of terms in the splitting in (3.4) to get unconditional stability. More precisely, they apply this method on the parabolic test problem given in [13] and guarantee that this method is unconditionally stable on linear constant coefficients PDEs with mixed derivatives of dimension N if $\nu \geq N\kappa_N\,\theta$ with the values of κ_N given in [12, Table 2], when both periodic and homogeneous Dirichlet boundary conditions. In [16], the authors solve the two-dimensional PDE for the well-known Heston model in options pricing. For this purpose, a hyperbolic change of variables is previously applied to the PDE, thus allowing the use of full non-uniform spatial meshes. However, we do not apply this change of variables since we use sparse grids to approximate efficiently the solution of (2.2) for higher spatial dimensions.

Obviously, if the solution of linear systems in (4.4) turns out too expensive from the computational point of view, the applicability of these schemes remains very limited. However, in the case of the PDE problem here addressed or similar multi-dimensional linear problems, due to the simple structure of the matrices $\mathcal{A}_i$, each linear system of type $(I - \nu\Delta t\mathcal{A}_i)K = G$ can be solved by using $\tilde{L}_i = \prod_{k\neq i}^{N} M_k$ tridiagonal linear systems of dimension M_i. In order to make easier the reading of this article, the details of this computation are included in Algorithm A.1 in the Appendix.

Moreover, another interesting advantage of the AMFR-W-methods (4.4) when applied to the autonomous linear problem (3.4) with $\mathcal{F}(Y) = \mathcal{A}Y$, $\mathcal{A} = \mathcal{A}_0 + \sum_{i=1}^{N}\mathcal{A}_i$, comes from the fact that the matrix-vector product $\mathcal{F}'(Y_n)K_r^{(N)}$ is simply an extra evaluation of the derivative function $\mathcal{F}(K_r^{(N)})$ and the explicit computation of the matrix $\mathcal{A}_0$ is not actually necessary.

4.2. $\theta-$method + Gauss-Seidel as W-method. In [33], the authors applied a direct (backwards in time) time-space discretization with finite differences, that can be also interpreted as a W-method (4.1) with fixed time step-size, when a fixed

number of iterations of the Gauss-Seidel iterative scheme is used to solve the involved linear systems.

More precisely, for a fixed time step-size $\Delta t > 0$, the well-known $\theta-$method applied to (3.4), with $\theta \in [0,1]$, gives the approximations $W_n \approx Y(t_n)$, $t_n = n\Delta t$, $n = 0, 1, \ldots, M$, by using the formula

$$W_{n+1} = W_n + (1-\theta)\Delta t\mathcal{F}(W_n) + \theta\Delta t\mathcal{F}(W_{n+1}). \tag{4.6}$$

Therefore, when $\theta \neq 0$ and linear problems $\mathcal{F}(Y) = \mathcal{A}Y$ as (3.4) are considered, it is necessary to solve the linear system $(I - \theta\Delta t\mathcal{A})W_{n+1} = \beta_n$, at each time step, with $\beta_n = (I - \theta\Delta t\mathcal{A})W_n + \Delta t\mathcal{F}(W_n)$. In [33], the Gauss-Seidel iterative linear systems solver is performed until getting an error below a prescribed tolerance. Note that Gauss-Seidel method splits the coefficient matrix $\mathcal{A} = P+R$, where P is the triangular matrix whose entries are the lower-triangular part of $\mathcal{A}$ and its diagonal elements, while R stores its strictly upper-triangular part. By using this splitting, from a starting value $W_{n+1}^{(0)}$, this method computes iteratively approximations $W_{n+1}^{(r)} \approx W_{n+1}$ by solving only triangular systems

$$(I - \theta\Delta tP)W_{n+1}^{(r)} = \theta\Delta tRW_{n+1}^{(r-1)} + \beta_n, \qquad r = 1, 2, \ldots \tag{4.7}$$

After some algebraic manipulations and taking as a natural choice for the starting value $W_{n+1}^{(0)} = W_n$, these iterations can be written as $W_{n+1}^{(r)} = W_n + \sum_{j=1}^{r} \hat{K}_j$, $r = 1, 2, \ldots$, where the vectors $\hat{K}_j$ are sequentially computed by

$$(I - \theta\Delta tP)\hat{K}_r = \Delta t\mathcal{A}\left(W_n + \sum_{j=1}^{r-1}\theta\hat{K}_j\right) + \sum_{j=1}^{r-1}(-1)\hat{K}_j \qquad r = 1, 2, \ldots$$

If we compare this last formula with (4.1), clearly if this combination of θ-method + Gauss-Seidel iteration (4.6)-(4.7) is performed with a fixed number s of iterations, then the method can be included in the class of W-methods (4.1) with coefficients $W = P$, $a_{rj} = \theta$, $\forall j < r$, $q_{rj} = -1$, $\forall j < r$, $b_r = 1$, $r = 1, \ldots, s$. Therefore, if we include the discretization proposed in [33] in the W-methods framework, we can apply the order conditions given in [22, p.115] or [17, Sec.2.1] (in a similar notation as here), and we obtain that it achieves order 2 (for $s \geq 2$) in time only when $\theta = 1/2$ (Crank-Nicolson scheme), what is in agreement with the results obtained in the aforementioned article [33].

An advantage of expressing the scheme in [33] as a W-method is that it makes easier to compare its computational cost per time step with that of the here proposed **AMFR-W2** method (4.4)-(4.5). The scheme (4.6)-(4.7) with $\theta = 1/2$ and s Gauss-Seidel iterations needs to compute one evaluation of the derivative function $\mathcal{F}$ and s triangular linear systems of dimension $L = M_1 \cdots M_N$. On the other hand, **AMFR-W2** needs to evaluate four times the derivative function (as $\mathcal{F}'(Y_n)K_r^{(N)} = \mathcal{F}(K_r^{(N)})$) and to solve $(2\,\hat{L}_i)$ tridiagonal linear systems of dimension M_i, per each $i = 1, \ldots, N$. Since the triangular systems of dimension L need approximately $\mathcal{O}(L^2)$ operations, the tridiagonal ones of dimension M_i cost $\mathcal{O}(M_i)$ operations and each evaluation of derivative function involves $\mathcal{O}(L^2)$ operations, we have $\mathcal{O}((s+1)L^2)$ operations for (4.6)-(4.7) and $\mathcal{O}(4L^2 + 2NL)$ operations for **AMFR-W2** (4.4)-(4.5).

5. Sparse grids in space. Solving PDE problems as (2.2) on a full tensor product based grid with p^N grid points, with p being the number of grid points in each coordinate direction, can become a highly involved computational task, even prohibitive. As the number of underlying forward rates increases, clearly the dimension of the multi-dimensional pricing PDE (2.2) increases as well, so the computational cost of solving the fully discretized problem grows exponentially. Thus, the discretization using this so-called full grid also consumes too much memory. This drawback is referred as the curse of dimensionality. For example, pricing a swaption over five forward rates ruled by the same stochastic volatility, by means of a full grid with 128 points per coordinate gives rise to more than one four billion points. The storage of such a grid using double precision floating point format will need more than 32 thousand gigabytes of memory.

Because of the curse of dimensionality, traditional full grid methods, like finite differences, finite elements or finite volumes, are not able to price derivatives with high dimensional underlying processes, even in the most powerful supercomputers available nowadays. This limitation can be partially overcome by using a family of techniques known as sparse grid methods (see [6], for example). Sparse grids are useful numerical methods for solving high-dimensional PDEs because they are based on a relatively small number of grid points but also maintain a satisfactory accuracy. More precisely, let d denote the underlying problem's dimensionality and p the number of grid points in one coordinate direction at the boundary. On the one hand, regarding the considered number of degrees of freedom, full grid methods use $O(p^d)$ grid points, while sparse grid discretizations only employ $O\left(p(\log_2 p)^{d-1}\right)$ grid points. On the other hand, concerning accuracy, conventional methods converge at a rate of $O(p^{-2})$ when making use of second order schemes, whereas sparse grid methods converge at the only slightly deteriorated rate of $O(p^{-2}\,(\log_2 p)^{d-1})$. In [6], Bungartz and Griebel present an excellent survey of the fundamentals and the applications of sparse grids, with a focus on the solution of PDEs. Sparse grid were introduced in the early 1990s for the solution of PDEs by Zenger [44] and Griebel [18].

5.1. Standard sparse grid combination technique. Discretizations on sparse grids require hierarchical data structures. Therefore, specially designed PDE solvers are required, and their implementations become more and more complicated as the dimension of the problem increases [1, 43]. An efficient way to avoid intricate sparse grid implementations is given by the sparse grid combination technique, originally proposed by Griebel, Schneider and Zenger [20]. Basically, the combination technique solves the PDE on several independent and conventional Cartesian smaller-sized grids. Then, the solution in the sparse grid space is approximated by a suitable linear combination of these partial solutions on the coarser grid. This solution retains the advertised convergence rate of sparse grid methods if certain error expansions for the component approximations exist [7, 8, 3]. Note the rigorous analysis of finite differences schemes for the sparse grid combination technique in [39]. Further advantages of the technique are the possibility to solve the problem on each of the constituent grids using standard full grid solvers and the inherent parallelism of the method [19].

Let us introduce formally the sparse grid combination technique. We fix a multi-index $\boldsymbol{l} = (l_1, l_2, \ldots, l_d) \in \mathbb{N}_0^d$ and define its L_1-norm as $|\boldsymbol{l}|_1 = \sum_{i=1}^d l_i$. In the d-dimensional orthohedron $[0, c_1] \times [0, c_2] \times \ldots \times [0, c_d]$ $(c_i \in \mathbb{R}_{>0})$, we denote by $\Omega_{\boldsymbol{l}} = \Omega_{(l_1,\ldots,l_d)}$ an anisotropic[2] although full grid having uniform mesh spacing $h_i = 2^{-l_i} c_i$

[2]Mesh spacing differs in each coordinate direction.

in each coordinate direction $i \in \{1, \ldots, d\}$. Let $u_{\boldsymbol{l}}$ be the conventional finite difference solution to the PDE on grid $\Omega_{\boldsymbol{l}}$, extended to $[0, c_1]\times[0, c_2]\times\ldots\times[0, c_d]$ by interpolation. Then, the sparse grid combination solution u_n^s over the sparse grid Ω_n^s with refinement level n is given by the following linear combination

$$u_n^s = \sum_{q=0}^{d-1} (-1)^q \cdot \binom{d-1}{q} \cdot \sum_{|\boldsymbol{l}|_1 = n-q} u_{\boldsymbol{l}}. \tag{5.1}$$

Increasing the level n should give a more accurate solution to the problem. The grid solutions $u_{\mathbf{l}}$ involved in the inner sum of (5.1) all have $\mathbf{l} = (l_1, \ldots, l_d)$ such that $l_1 + \cdots + l_d = n - q$. The number of elements in each of these grids is $O(2^{n-q})$, regardless of the dimension, and the number of grid solutions in this inner sum is $\binom{n-q+d-1}{d-1}$ and grows like $O((n-q)^{d-1})$. Besides, $\left(\bigcup_{0\leq|\mathbf{l}|_1<n} \Omega_{\mathbf{l}}\right) \subset \Omega_n^s$. Therefore, the dimension of the sparse grid space on level n is $O(2^n n^{d-1}) = O(h^{-1}(\log_2 h)^{d-1})$ where $h = 2^{-n}$ is the finest grid size. This value can be compared with the full grid space dimension which is $O(2^{nd}) = O(h^{-d})$.

The combination technique works due to the cancellation mechanism of the error terms in the involved grids. This cancellation principle is well known in extrapolation techniques. Indeed, all lower order error terms cancel out in the the combination formula (5.1), see [25] for deep details in dimension two. Thus, the combination technique is able to produce accurate results in reasonable time. Several generalizations of the standard combination technique formula (5.1) have been developed [23].

The combination technique algorithm is embarrassingly parallel since all component grid solutions can be computed in parallel. In general, for refinement level n in d dimensions there are $\sum_{q=0}^{d-1}\binom{n-q+d-1}{d-1}$ component grids, which can be solved in parallel. In order to achieve optimal speed-ups one has to carefully deal with load imbalances, even in all those grids at the same refinement level, due to the anisotropic structure of the component grids.

The sparse grid combination technique was initially formulated for elliptic PDEs such as Laplace's and Poisson's equation. Later, it has also been applied to parabolic PDEs, specially for option pricing problems in finance [3, 30, 40, 31, 39, 9, 4, 32, 33, 26, 10, 24, 25]. Here we focus on the implementation of the sparse grid combination technique for parabolic equations. More precisely, we just consider the case where the solution is only needed at the final time, which is frequently the case in finance and particularly in the problem we address.

In this setting, the natural approach is the following. First, solve the parabolic equation on each of the full grids involved in the sparse grid combination technique formula (5.1) with a full grid method. Finally, combine these solutions only at the end. This method only requires interpolation from grid values at the final time, but not at intermediate time steps. If the numerical error due to the time discretization does not dominate the spatial error, we expect a pointwise rate of convergence proportional to $O(p^{-2}\,(\log_2 p)^{d-1})$ for our AMFR-W scheme applied to problems with smooth enough initial and boundary data. It is important to notice that interpolation techniques are required in order to approximate the solution at points not belonging to the sparse grid. The most straightforward approach is to interpolate at those points over all full grids handled by the combination technique, and then add up these results with the appropriate combination technique weights. Note also that the interpolation technique has to preserve the order of the used discretization scheme, so that the convergence

result remain valid for the entire domain. Otherwise, the convergence order only holds for grid nodes belonging to all sub-grids and therefore not affected by interpolation. A tensor based linear interpolation preserves the required order 2 of accuracy for second order finite difference discretizations.

5.2. Modified sparse grid combination technique. By means of the previously described standard sparse grid combination technique it is impossible to approximate accurately a Neumann boundary condition for degenerated Cartesian grids having very few points in the corresponding coordinate direction. The approximations of the solution in these grids becomes very poor, thus decreasing the accuracy of the combination technique approximation.

In order to overcome this drawback, a mild modification of the standard sparse grid combination technique (5.1) can be developed, just by forcing a minimum number of discretization steps in all grids involved in the combination procedure. More precisely, all levels in all dimensions start from a small but non zero value ψ, so that the modified combination technique formula reads

$$u_n^s = \sum_{q=0}^{d-1} (-1)^q \cdot \binom{d-1}{q} \cdot \sum_{|\boldsymbol{l}|_1 = n-q} u_{\psi \mathbf{1} + \boldsymbol{l}}, \tag{5.2}$$

where $\psi\mathbf{1} + \boldsymbol{l} = (\psi + l_1, \ldots, \psi + l_d)$. This modified sparse grid combination technique working over a modified sparse grid $\Omega_n^{s,\psi}$ produces more accurate approximations [4] at the cost of increasing the consumed time and memory. Although the number of subproblems to be solved is exactly the same as before, the number of degrees of freedom associated to each subproblem increases. In fact, the number of grid points in the combined sparse grid increases from $O(2^n n^{d-1})$ in the standard one to $O(2^{n+d\psi} n^{d-1})$ in the modified one. In this new setting ψ should be kept small (specifically $\psi = 1$ or 2 in the present work), otherwise the new modified combination technique will suffer soon the curse of dimensionality.

6. Numerical results. In this Section we present the obtained numerical results when the previously described methodologies are applied. More precisely, we show and discuss the results obtained by using the AMFR-W method with full grid, standard and modified sparse grids combination techniques to conveniently cope with the proposed homogeneous Neumann boundary conditions in the particular case $\beta = 1$.

For all products we will use the data presented in Table 1 where we consider the tenor structure $0 = T_0 < 0.5 < 1.0 < \cdots < 2.5 < 3 = T_{10}$ in years, with constant periods $\tau = T_{i+1} - T_i = 0.5$.

i	T_i	$F_i(0)$	α_i	i	T_i	$F_i(0)$	α_i
0	0	0.0112	0	3	1.5	0.0126	0.2221
1	0.5	0.0118	0.2366	4	2	0.0130	0.2068
2	1	0.0122	0.2145	5	2.5	0.0135	0.1932

TABLE 1
Hypothetical market data (LIBOR rates and volatilities) used in pricing. Strike rate $K = 0.011$.

The spatial domain is defined by $F^{max} = 0.04$ and $V^{max} = 3.5$, thus upper boundaries were settled between 3 and 4 times the point of interest at which we evaluate the pricing of the interest rate derivative. In the cases where the analytical solution is not available, we first compute reference solutions using the proposed space and time discretizations over classical full grids. These solutions will serve to

assess on the accuracy of the proposed sparse grids methods in space. The designed algorithms were implemented using C++ (GNU C++ compiler 8.3.1) and double precision. Besides, all numerical experiments have been performed in a machine with 16GBytes of RAM and four multicore Intel Xeon CPUs E5-2620 v4 clocked at 2.10GHz, each one with eight cores.

6.1. Numerical results with full grids. The first test to validate the proposed numerical methodologies consists of pricing a caplet without considering stochastic volatility, that is to say, under the classical LMM. This test is a sanity check, since the analytical pricing formula is known for caplets, the so-called Black-Scholes's formula for caplets ([5, equation 1.26]). More precisely, we start pricing the caplet with maturity T_1 and payoff $\tau_1(F_1(T_1) - K)^+$ paid at time T_2, under the data of Table 1, with strike rate $K = 0.011$. The present intrinsic value of the caplet is given by $P(0,T_2)\tau_1(F_1(T_1) - K)^+$. The exact price of this product given by Black-Scholes' formula is 6.058877 basis points (bps, 1 bp $= 10^{-4}$).

In order to price this caplet using the PDE approach presented here it is convenient to consider the terminal probability measure associated with choosing the bond $P(0,T_2)$ as numeraire. Thus, the price of this product is given by the solution of the PDE (2.2) (with $\sigma = 0$) multiplied by $P(0,T_2) = \frac{1}{1+\tau F_0(0)}\frac{1}{1+\tau F_1(0)}$, i.e. $P(0,T_2)u(0,F_1,V)$. Once obtained the PDE solution on the last time slice, the price of the caplet is obtained through interpolation in space, by means of multilinear interpolation, thus maintaining order two in space. In Table 2 full grid solutions are presented for levels from 6 to 13 in space and considering 4, 8, 16 and 256 time steps. The interpolation in space for the last time slice was done in $F_1 = 0.0118$ and $V = 1$. The column labelled as Solution shows the PDE solution in bps, and the column for the error measures the absolute distance of the numerical solution to the exact one, in bps as well. The execution time was measured in seconds in all the experiments in this work. The "grid points" column displays the number of grid points employed in the full space meshes at each time discretization. Since the method is order three in time, few time steps could be considered in real pricing applications. For the space level 13 (for the forward and the volatility) and when using 256 time steps, the method was able to recover the exact solution up to the 8-th decimal digit. Nevertheless, in this case the full grid method required almost three hours, all space meshes in all time slices with more than 67 million points.

Once we have checked the correct behaviour of the full grid method, which will be used in the sparse grid combination technique, we compute full grid reference solutions for financial products without exact prices. They will be used in order to assess the correctness of the upcoming sparse grid combination technique implementation. Therefore, in order to minimize errors due to the time discretization, 256 time steps will be chosen for the rest of the full grid tests in this section. In Table 3 the computed prices of the previous caplet under the stochastic volatility framework are shown.

Next, we deal with the pricing of $T_a \times (T_b - T_a)$ European swaptions. In Table 4, first the results for the 0.5×1 swaption are given. Note that under this full grid framework it is not possible to price this product in reasonable computational times past refinement level 9, due to the high number of involved spatial grid points. Then, the results for the 0.5×1.5 swaption are also shown. Once more, full grid pricing is only achievable on the lower grid levels.

6.2. Numerical results with the standard sparse grid combination technique. In this section, by means of the standard sparse grid combination technique, we price not only the previous caplets and swaptions, but also swaptions involving

Level	4 time steps Solution	Error	Time	8 time steps Solution	Error	Time	Grid points
6	6.084214	2.533666×10^{-2}	0.01	6.082708	2.383026×10^{-2}	0.02	4225
7	6.065957	7.079711×10^{-3}	0.05	6.064274	5.396418×10^{-3}	0.09	16641
8	6.063526	4.648521×10^{-3}	0.15	6.061980	3.102002×10^{-3}	0.27	66049
9	6.060148	1.270262×10^{-3}	0.55	6.058939	6.133732×10^{-5}	0.96	263169
10	6.060300	1.422802×10^{-3}	2.14	6.059077	1.992885×10^{-4}	4.23	1050625
11	6.060237	1.359825×10^{-3}	9.39	6.059021	1.431228×10^{-4}	18.69	4198401
12	6.060236	1.358165×10^{-3}	40.47	6.059019	1.416918×10^{-4}	80.45	16785409
13	6.060226	1.348640×10^{-3}	165.64	6.059011	1.331767×10^{-4}	328.15	67125249
Level	16 time steps Solution	Error	Time	256 time steps Solution	Error	Time	Grid points
6	6.082540	2.366282×10^{-2}	0.04	6.082513	2.363549×10^{-2}	0.47	4225
7	6.064109	5.231082×10^{-3}	0.16	6.064079	5.201370×10^{-3}	1.89	16641
8	6.061870	2.992441×10^{-3}	0.49	6.061840	2.962119×10^{-3}	7.28	66049
9	6.058832	4.556637×10^{-5}	1.97	6.058802	7.582048×10^{-5}	30.77	263169
10	6.058975	9.701419×10^{-5}	8.33	6.058944	6.680796×10^{-5}	131.75	1050625
11	6.058919	4.157937×10^{-5}	37.19	6.058889	1.139059×10^{-5}	593.13	4198401
12	6.058918	4.041366×10^{-5}	163.01	6.058888	1.022896×10^{-5}	2558.92	16785409
13	6.058910	3.191855×10^{-5}	655.84	6.058879	1.735259×10^{-6}	10415.57	67125249

TABLE 2

Full grid method, caplet with expiry T_1, $\sigma = 0$. Prices and errors are shown in bps.

Level	Solution	Time	Grid points	Level	Solution	Time	Grid points
6	6.050103	0.48	4225	10	6.023799	135.29	1050625
7	6.029510	1.93	16641	11	6.023737	597.82	4198401
8	6.026929	7.51	66049	12	6.023734	2557.62	16785409
9	6.023665	31.06	263169	13	6.023725	10505.02	67125249

TABLE 3

Full grid method, caplet with expiry T_1, $\sigma = 0.3$, $\phi_1 = 0.4$, 256 time steps.

	Level	Solution	Time	Grid points	Level	Solution	Time	Grid points
0.5×1	6	13.002003	71.33	274625	8	12.981320	5111.14	16974593
	7	12.984709	590.97	2146689	9	12.980459	43325.53	135005697
0.5×1.5	Level	Solution	Time	Grid points	Level	Solution	Time	Grid points
	3	23.952705	1.70	6561	5	21.577149	474.93	1185921
	4	21.765442	28.45	83521	6	21.486079	8122.91	17850625

TABLE 4

Full grid method, 0.5×1 and 0.5×1.5 swaptions, $\sigma = 0.3$, $\phi_i = 0.4$, $i = 1, 2, 3$, 256 time steps.

more underlying forward interest rates, thus dealing with a high dimensional setting. As usual, we are also interested in the values of these derivatives at the last time cut for the values of the forward rates depicted in Table 1 and $V = 1$, which define the spatial point where the value of the solution of the PDE is computed. In order to obtain the solution given by the sparse grid combination technique at this point, the numerical solution on each grid involved in the combination technique is interpolated at this point with multilinear interpolation. Next, all these values are introduced in the combination technique formula (5.1), thus obtaining the price provided by the standard sparse grid combination technique.

Moreover, sparse grid combination techniques have been implemented to take advantage of shared memory parallel computers. The code was optimized and parallelized using OpenMP framework [45], version 4.5. In order to deal with the previously mentioned load imbalances it is crucial to use a dynamic schedule to assign the involved full grids to threads. In this way, OpenMP assigns one grid to each thread. When the thread finishes, it will be assigned the next mesh that has not been assigned yet. The speedup of the parallelized version is almost equal to the number of available computing cores, in our case 32. This optimal speedup is due to the fact

that communication between processors only takes place at the final step in order to concentrate the solutions over all grids to a single scalar value.

In Table 5 we price the caplet with maturity T_1 under the framework without stochastic volatility, whose exact price was 6.058877 basis points. The results in this Table are to be compared with those of Table 2. The accuracy of each solution is only slightly worse in this case, although the computing time is much lower due the much less number of involved grid points. For example, with the full grid approach, the solution using 256 time steps and refinement level in space 13 employed 10415.57 seconds to achieve an error 1.735259×10^{-6} in basis points, while the standard sparse grid combination technique just needed 569.26 seconds to attain almost the same accuracy, an error of 1.047159×10^{-5}. The reduction in the number of employed grid points is also shown in Table 2.

	4 time steps			8 time steps			
Level	Solution	Error	Time	Solution	Error	Time	Grid points
6	6.063081	4.202988×10^{-3}	0.02	6.053883	4.994378×10^{-3}	0.02	385
7	6.120850	6.197245×10^{-2}	0.02	6.110998	5.212012×10^{-2}	0.02	833
8	6.067165	8.287799×10^{-3}	0.02	6.059333	4.557323×10^{-4}	0.03	1793
9	6.061071	2.192917×10^{-3}	0.03	6.056314	2.563871×10^{-3}	0.05	3841
10	6.061681	2.803747×10^{-3}	0.07	6.059090	2.119811×10^{-4}	0.12	8193
11	6.059735	8.569127×10^{-4}	0.20	6.058133	7.450762×10^{-4}	0.35	17409
12	6.060156	1.278608×10^{-3}	0.58	6.058787	9.054229×10^{-5}	1.21	36865
13	6.060211	1.333075×10^{-3}	2.43	6.058951	7.311969×10^{-5}	4.53	77825
14	6.060223	1.345681×10^{-3}	9.00	6.058998	1.206373×10^{-4}	17.90	163841
	16 time steps			256 time steps			
Level	Solution	Error	Time	Solution	Error	Time	Grid points
6	6.052778	6.099826×10^{-3}	0.02	6.052738	6.139440×10^{-3}	0.05	385
7	6.110144	5.126590×10^{-2}	0.03	6.110118	5.124050×10^{-2}	0.09	833
8	6.059007	1.291280×10^{-4}	0.04	6.058984	1.066738×10^{-4}	0.22	1793
9	6.056190	2.687266×10^{-3}	0.07	6.056164	2.713620×10^{-3}	0.71	3841
10	6.059029	1.514420×10^{-4}	0.20	6.058998	1.205596×10^{-4}	2.31	8193
11	6.058034	8.433494×10^{-4}	0.64	6.058003	8.741697×10^{-4}	8.47	17409
12	6.058698	1.800588×10^{-4}	2.33	6.058667	2.108407×10^{-4}	34.02	36865
13	6.058852	2.520960×10^{-5}	8.99	6.058822	5.565743×10^{-5}	141.73	77825
14	6.058897	1.981956×10^{-5}	35.69	6.058867	1.047159×10^{-5}	569.26	163841

TABLE 5

Sparse grid combination technique, caplet with expiry T_1, $\sigma = 0$.

Next, in Table 6 the results for the previous caplet under the stochastic volatility framework are shown. These results are to be compared with those of Table 3. Then, Tables 7 and 8 show the prices given by the standard sparse grid combination technique for 0.5×1 and 0.5×1.5 swaptions under stochastic volatility. These results are to be compared with those of Table 4. Clearly, the standard sparse grid combination technique outperforms the full grid approach. Besides, the sparse method is able to cope with higher resolution levels, thus allowing to price successfully the 0.5×1.5 swaption. Note that this was not possible with the full grid approach, see Table 4.

Level	Solution	Time	Level	Solution	Time
6	6.057668	0.06	11	6.022082	8.40
7	6.095685	0.10	12	6.023257	33.96
8	6.025848	0.25	13	6.023595	141.41
9	6.018222	0.70	14	6.023693	569.07
10	6.021834	2.32			

TABLE 6

Sparse grid combination technique, caplet with expiry T_1, $\sigma = 0.3$, $\phi_1 = 0.4$, 256 time steps.

Level	Solution	Time	Grid points	Level	Solution	Time	Grid points
8	12.311172	0.67	8705	12	13.205324	89.80	219137
9	13.024747	2.00	19713	13	12.993536	360.72	483329
10	13.616333	6.77	44289	14	12.971783	1399.46	1060865
11	13.525821	24.42	98817	15	12.973900	5755.68	2318337

TABLE 7

Sparse grid combination technique, 0.5×1 *swaption,* $\sigma = 0.3$*,* $\phi_1 = \phi_2 = 0.4$*, 256 time steps.*

	8 time steps		256 time steps		
Level	Solution	Time	Solution	Time	Grid points
12	21.935448	8.50	21.936574	271.54	1064961
13	21.842901	31.75	21.844522	998.51	2439169
14	21.609183	119.97	21.610055	3809.31	5550081
15	21.707363	461.28	21.708001	14866.96	12554241
16	21.519917	1838.36	21.516402	59010.15	28246017
17	21.483062	7315.95	21.478930	235912.03	63242241

TABLE 8

Sparse grid combination technique, 0.5×1.5 *swaption* $\sigma = 0.3$*,* $\phi_1 = \phi_2 = \phi_3 = 0.4$*.*

Finally, in Tables 9 and 10, 0.5×2 and 0.5×2.5 swaptions are priced under stochastic volatility. The curse of dimensionality makes impossible to price these products with full grid approaches. In order to speedup the convergence of the sparse grid method, a useful technique is to consider a computational domain such that the point of interest is in the neighbourhood of the center of the domain. This strategy easily improves sparse grid results. In fact, in that region is where the sparse grid contains more points. Indeed, the central point belongs to all non degenerated grids involved in the standard sparse grid combination technique. The improvement in accuracy can be observed in Table 10, where the upper boundaries of the forward rates, F^{max}, were shrunk from 0.04 to 0.02.

	4 time steps		8 time steps	
Level	Solution	Time	Solution	Time
14	35.341806	180.34	35.346408	360.57
15	34.388334	669.85	34.425087	1335.40
16	32.115380	2561.87	32.122101	5133.81
17	30.639336	10058.18	30.641664	20076.39
18	30.881086	40097.11	30.918448	80268.57
19	30.822037	239746.17	30.797087	479681.87

TABLE 9

Sparse grid combination technique, 0.5×2 *swaption,* $\sigma = 0.3$*,* $\phi_1 = \ldots = \phi_4 = 0.4$*.*

In order to price interest rate derivatives involving more underlying forward rates using this approach, the proposed algorithm should be implemented to run on a cluster of processors (distributed memory machines). Since the communications between processors is minimal, the technique scales optimally. This extra layer of parallelism would bring also a further reduction on the previous execution times, thus allowing to stress the method with higher resolution levels.

6.3. Numerical results with the modified sparse grid combination technique. Our last set of numerical experiments aims at showing that the modified sparse grid technique defined by (5.2) is able to improve the performance (accuracy and computing time) of the standard sparse grid combination technique given by expression (5.1), specially in moderately high dimensions.

$F^{max} = 0.04$						
	2 time steps		4 time steps		8 time steps	
Level	Solution	Time	Solution	Time	Solution	Time
16	54.923235	3812.17	53.634032	7565.44	53.512820	15084.20
17	39.023559	14245.68	41.296346	28565.06	41.328455	56819.84
18	39.780626	55108.12	38.377139	109843.14	38.823376	220825.17
19	41.230567	285159.96	41.970631	570320.92	41.599974	1140639.84
$F^{max} = 0.02$						
	2 time steps		4 time steps		8 time steps	
Level	Solution	Time	Solution	Time	Solution	Time
16	42.250960	3812.17	43.016757	7565.44	42.777426	15084.20
17	42.116312	14245.68	41.406882	28565.06	41.842625	56819.84
18	42.991274	55108.12	42.746717	109843.14	42.737987	220825.17
19	42.446354	285159.96	42.892002	570320.92	42.838119	1140639.84

TABLE 10

Sparse grid combination technique, 0.5×2.5 *swaption,* $\sigma = 0.3$*,* $\phi_1 = \ldots = \phi_5 = 0.4$*.*

As in the previous cases, we start with the sanity test of the pricing of the caplet with expiry T_1 under the classical LMM. Table 11 gathers the behaviour of the modified technique when pricing this caplet. Firstly, we compare Table 11 with Table 5 originated with the standard combination technique. With $\psi = 1$, the modified technique is able to obtain an accuracy of 1.047853×10^{-5} with level equal 12 in less than a hundred seconds. In contrast, the standard sparse grid technique required a higher refinement level of 14 and employed more than five hundred seconds to obtain a similar accuracy. Also note that with $\psi = 2$ and the refinement level 10, the modified combination technique is able to get better results, an error of 7.625043×10^{-6} in just over 25 seconds. Moreover, while the obtained order of convergence in space for the standard combination technique is slightly worse than two, with this modified method is almost two when = 2. The comparison with full grid method results shown in Table 2 could be summarized by noting that with $\psi = 2$ the modified sparse grid technique is able to obtain an error less than 1.735259×10^{-6} in less than five hundred seconds, while the full grid approach needed almost 2.9 hours.

	$\psi = 1$				$\psi = 2$			
n	Solution	Error	Time	#points	Solution	Error	Time	#points
7	6.056324	2.55×10^{-3}	0.20	2817	6.057952	9.25×10^{-4}	0.70	10241
8	6.058978	1.00×10^{-4}	0.44	6145	6.058684	1.93×10^{-4}	2.04	22529
9	6.058005	8.72×10^{-4}	1.61	13313	6.058822	5.51×10^{-5}	7.08	49153
10	6.058666	2.11×10^{-4}	6.35	28673	6.058870	7.62×10^{-6}	26.06	106497
11	6.058822	5.52×10^{-5}	24.62	61441	6.058874	4.08×10^{-6}	105.78	229377
12	6.058867	1.04×10^{-5}	96.95	131073	6.058877	9.15×10^{-7}	478.10	491521
13	6.058873	4.45×10^{-6}	473.16	278529	6.058877	3.17×10^{-7}	1991.75	1048577
14	6.058877	1.07×10^{-6}	1968.99	589825	6.058878	9.91×10^{-8}	7868.34	2228225

TABLE 11

Modified sparse grid combination technique, caplet with maturity T_1*,* $\sigma = 0$*,* 256 *time steps,* $F^{max} = 0.04$*.*

Finally, Tables 12 and 13 show the results for 4 and 6 dimensional PDEs in space, respectively. These Tables are to be compared with the corresponding Tables 8 and 10 generated with the standard combination technique. We observe that the higher the dimensionality of the problem the lower ψ should be, otherwise the curse of dimensionality will appear soon again. Table 12 shows that with $\psi = 1$ the modified method is able to obtain in just 15 minutes a similar accuracy to the one obtained with the standard approach in more than 246 minutes. With $\psi = 2$ we observe that 5

decimal digits are stabilized in the modified combination technique. Regarding Table 13, with $\psi = 1$ three decimal digits are stabilized already in level 13. Also note that for the 6 dimensional PDEs in space reported in Table 13, in our machine the modified method is not able to go further level 12 with $\psi = 2$ in a reasonable computational time due to the curse of dimensionality. Nevertheless, the accuracy recovered for level 12 is remarkable. Finally, notice that adding points entails better performance than increasing the level of the sparse grid.

	$\psi = 1$			$\psi = 2$		
Level	Solution	Time	Grid points	Solution	Time	Grid points
9	21.511031	6.11	114689	21.519347	107.59	763905
10	21.297979	19.02	262145	21.491243	328.03	1765377
11	21.600032	60.94	593921	21.485699	1051.35	4038657
12	21.595254	209.27	1335297	21.466814	3807.11	9158657
13	21.472738	806.72	2981889	21.464192	13921.97	20611073

TABLE 12

Modified sparse grids combination technique, swaption 0.5×1.5, $\sigma = 0.3$, $\phi_i = \phi_2 = \phi_3 = 0.4$, 16 *time steps,* $F^{max} = 0.04$.

	$\psi = 1$		$\psi = 2$	
Level	Solution	Time	Solution	Time
12	42.624837	3255.48	42.723635	420696.70
13	42.828046	10476.64	–	–
14	42.686665	34859.98	–	–
15	42.702808	121471.65	–	–

TABLE 13

Modified sparse grid combination technique, swaption 0.5×2.5, $\sigma = 0.3$, $\phi_1 = \ldots = \phi_5 = 0.4$, 4 *time steps,* $F^{max} = 0.02$.

7. Conclusions. In this work we have mainly developed a new numerical methodology which combines high order time discretization algorithms with a sparse grids modified combination technique to solve high dimensional PDE problems arising in finance. More precisely, we have focused on the numerical solution of the PDE formulation proposed in [33] for pricing a large variety of interest rate derivatives, when the underlying forward rates follow a SABR-LMM model. For this purpose, we have proposed the use of high order in time AMFR-W methods, thus allowing the use of larger time steps. Moreover, a suitable splitting of the involved operators additionally contributes to the computational time reduction for a given accuracy. As the PDE problem becomes high dimensional in space when the particular interest rate derivative requires the consideration of a large number of forward rates (each one giving rise to one spatial dimension), the application of AMFR-W methods on sparse grids with combination technique turns out to be very efficient to obtain the pricing in reasonable computational times. As illustrated in the section of numerical results, parallel implementations of the algorithms based on OpenMP framework lead to a significant speed up of the computations. As indicated, an appropriate load imbalances management provides an optimal speed up, which is almost equal to the number of available computer cores. All computer implementations have been carried out from scratch. Another relevant innovative aspect comes from the suitable consideration of new homogeneous Neumann boundary conditions, instead of Dirichlet ones in [33]. This consideration avoids the numerical difficulties associated to the presence of boundary layers in the outflow boundaries when the parameter β is not

zero, specially in the advection dominated regime. Moreover, they motivate the introduction of a modified combination technique to cope with a certain decrease in the accuracy of the standard combination technique, which mainly comes from the inaccuracy of approximations obtained with some degenerated grids included in the sparse grids combination expression. Numerical results also illustrate the advantages of the proposed modified combination technique with respect to the standard version.

Although this article focuses on the PDE formulation of the very relevant financial problem of pricing interest rate derivatives, the proposed methodology can be applied to a large variety of models involving high dimension PDE formulations which arise not only in finance but also in other disciplines in sciences and engineering. For example, in finance high dimension PDE problems related to the pricing of basket options or the computation of the XVA associated to portfolios could be considered. In computational biology, the same happens with problems related to gene networks or synthetic biology.

Appendix A. Appendix.

Proof of Lemma 3.1: Since $M_i - m_i \geq 0$, $j_i - m_i \geq 0$, for all i, it is clear that $\vartheta(m_1, m_2, \ldots, m_N) \leq \vartheta(\mathbf{j}) \leq \vartheta(M_1, M_2, \ldots, M_N)$, for all $\mathbf{j} \in \mathcal{I}_N$. It is easy to see that $\vartheta(m_1, m_2, \ldots, m_N) = m_1$ and $\vartheta(M_1, M_2, \ldots, M_N) = M_T + m_1 - 1$.

On the other hand, calling $Q_l := M_l - m_l \geq 0$, if $\vartheta(\mathbf{j}) = \vartheta(\mathbf{k})$ for $\mathbf{j}, \mathbf{k} \in \mathcal{I}_N$,

$$q_1 = -\sum_{l=2}^{N} \left(q_l \prod_{r=1}^{l-1} (Q_r + 1) \right), \qquad q_l := j_l - k_l, \, \forall 1 \leq l \leq N, \tag{A.1}$$

where the differences q_l are integers that satisfy $|q_l| \leq Q_l$, $1 \leq l \leq N$. Let us suppose that $N > 2$ (when $N = 2$ is much simpler). From (A.1), we obtain that

$$q_1 = -\left(q_2 + \sum_{l=3}^{N} \left(q_l \prod_{r=2}^{l-1} (Q_r + 1) \right) \right) (Q_1 + 1). \tag{A.2}$$

Therefore, q_1 is a multiple of the positive integer $(Q_1 + 1)$. Moreover, as $-Q_1 \leq q_1 \leq Q_1$, so necessarily q_1 must be zero. Therefore, because of (A.2), we obtain

$$q_2 = -\sum_{l=3}^{N} \left(q_l \prod_{r=2}^{l-1} (Q_r + 1) \right),$$

which is the same formula as in (A.1), although starting from $r = 2$ instead of $r = 1$. Applying a similar procedure as for q_1, we get that necessarily $q_2 = 0$. Inductively, we obtain that $q_l = 0$, $i = 1, 2, \ldots, N$, so $\mathbf{j} = \mathbf{k}$ and the map ϑ is injective. Since clearly the two sets $\mathcal{I}_N$ and $\{m_1, m_1 + 1, \ldots, M_T + m_1 - 1\}$ have the same number M_T of elements, ϑ is a bijection. □

As a consequence of Lemma 3.1, for all $J \in \{m_1, m_1 + 1, \ldots, M_T + m_1 - 1\}$, there exists a unique multi-index $\mathbf{j} \in \mathcal{I}_N$ given by $\mathbf{j} = \vartheta^{-1}(J)$. In practice, it is necessary to compute this inverse when we manipulate finite differences. An efficient way to calculate it is to use the `modulo` operation, i.e., $(a \,\texttt{mod}\, n)$ is the remainder of the Euclidean division of a by n.

LEMMA A.1. *For every integer $J \in \{m_1, m_1+1, \ldots, M_T+m_1-1\}$, the components of the unique multi-index $\mathbf{j} = (j_1, \ldots, j_N) = \vartheta^{-1}(J) \in \mathcal{I}_N$ can be written as $j_i =$*

716 $m_i + b_i$, $i = 1, \ldots, N$ *where the integers* b_i *satisfy*

$$
\begin{aligned}
&c_1 = J - m_1, && b_1 = c_1 \,\mathtt{mod}\, (M_1 - m_1 + 1),\\
&c_i = \frac{c_{i-1} - b_{i-1}}{M_{i-1} - m_{i-1} + 1}, && b_i = c_i \,\mathtt{mod}\, (M_i - m_i + 1), \quad i = 2, \ldots, N.
\end{aligned}
\tag{A.3}
$$

717

Algorithm A.1 Procedure to solve linear systems of type $(I - \nu \Delta t \mathcal{A}_i)K = G$

Define a matrix Q of dimension M_i; $K = G$
if $i = 1$ **then**
  $\hat{L}_r = M_r, \quad r = 2, \ldots, N$
else if $i \geq 2$ **then**
  $\hat{L}_r = M_{r-1}$, $r = 2, \ldots, i$, $\hat{L}_r = M_r$, $r = i+1, \ldots, N$
end if
$\hat{L} = \prod_{r=2}^{N} \hat{L}_r = \prod_{r \neq i} M_r$
for $I = 1, \ldots, \hat{L}$ **do**
  $Q = 0$
  $(k_2, \ldots, k_N) = \vartheta_{N-1}^{-1}(I)$; $j_r = k_{r+1}$, $r = 1, \ldots, i-1$; $j_r = k_r$, $r = i+1, \ldots, N$
  for $j_i = 1, \ldots, M_i$ **do**
    $\mathbf{j} = (j_1, \ldots, j_i, \ldots, j_N)$; $J = \vartheta_0(\mathbf{j})$; $R(j_i) = G(J)$

$$
P = \begin{cases} \dfrac{(d_i)_J}{h_i^2} = \dfrac{\alpha_i^2}{2} j_i^2 F_{N, j_N}^2, & \text{if } 1 \leq i \leq N-1 \\ \dfrac{(d_N)_J}{h_N^2} = \dfrac{\sigma^2}{2} j_N^2 & \text{if } i = N \end{cases}
$$

  $Q(j_i, j_i) = -2P$
  if $j_i \geq 2$ **then**

$$
Q(j_i, j_i - 1) = \begin{cases} P & \text{if } j_i \leq M_i - 1 \\ 2P & \text{if } j_i = M_i \end{cases}
$$

  end if
  if $j_i \leq M_i - 1$ **then**
    $Q(j_i, j_i + 1) = P$
  end if
  end for
  Solve $(I_{L_i} - \nu \Delta t Q)X = R$
  for $j_i = 1, \ldots, M_i$ **do**
    $\mathbf{j} = (j_1, \ldots, j_i, \ldots, j_N)$, $J = \vartheta_0(\mathbf{j})$; $K(J) = X(j_i)$
  end for
end for

Proof of Lemma A.1: With the same notation as in the proof of Lemma 3.1, $b_l = j_l - m_l \in \{0, 1, \ldots, Q_l\}$ for all $l = 1, \ldots, N$, equation (3.2) turns into

$$
c_1 := J - m_1 = b_1 + \sum_{l=2}^{N} b_l \prod_{r=1}^{l-1} (Q_r + 1) = b_1 + (Q_1 + 1) \left(b_2 + \sum_{l=3}^{N} b_l \prod_{r=2}^{l-1} (Q_r + 1) \right),
$$

then $b_1 = c_1 \,\mathtt{mod}\, (Q_1 + 1) \in \{0, 1, \ldots, Q_1\}$, $j_1 = b_1 + m_1$, and

$$
c_2 := \frac{c_1 - b_1}{Q_1 + 1} = b_2 + \sum_{l=3}^{N} b_l \prod_{r=2}^{l-1} (Q_r + 1).
$$

Clearly, we can apply again the `mod` operation to c_2, obtaining $b_2 = c_2 \,\texttt{mod}\, (Q_2 + 1) \in \{0, 1, \ldots, Q_2\}$ and $j_2 = b_2 + m_2$. The proof is completed by iteration. □

In order to help the reader interested in computing every directional linear system of type $(I - \nu \Delta t \mathcal{A}_i)K = G$ of dimension M, $i = 1, \ldots, N$, the procedure to solve them is presented in Algorithm A.1, when the $i-$direction and the right-hand side vector G are given. We must observe that a new bijection ϑ_{N-1} of type (3.2) is used there for multi-indices of dimension $N-1$ with $m_i = 1$, $M_i = \hat{L}_{i+1}, i = 1, \ldots, N-1$.

REFERENCES

[1] S. Achatz, *Higher order sparse grid methods for elliptic partial differential equations with variable coefficients*, Computing, 71 (2003), pp. 1–15.

[2] A. Alonso, R. Bermejo, M. Pájaro, and C. Vázquez, *Numerical analysis of a method for a partial integro-differential equation model in regulatory gene networks*, Mathematical Models and Methods in Applied Sciences, 10 (2018), pp. 2069–2095.

[3] A. Arciniega and E. Allen, *Extrapolation of difference methods in option valuation*, Applied Mathematics and Computation, 135 (2004), pp. 165–186.

[4] I. Beyna, *Interest Rate Derivatives*, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin Heidelberg, 2013.

[5] D. Brigo and F. Mercurio, *Interest Rate Model – Theory and Practice*, Springer Finance, Springer-Verlag, Berlin Heidelberg, 2007.

[6] H. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica, 13 (2004), pp. 147–269.

[7] H. J. Bungartz, M. Griebel, D. Röschke, and C. Zenger, *Pointwise convergence of the combination technique for the Laplace equation*, East-West Journal of Numerical Mathematics, 2 (1994), pp. 21–45.

[8] H. J. Bungartz, M. Griebel, D. Röschke, and C. Zenger, *A proof of convergence for the combination technique for the Laplace equation using tools of symbolic computation*, Mathematics and Computers in Simulation, 41 (1996), pp. 595–605.

[9] C. Chiarella and B. Kang, *The evaluation of American compound option prices under stochastic volatility and stochastic interest rates*, Journal of Computational Finance, 17 (2013), pp. 71–92.

[10] B. Düring, C. Hendricks, and J. Miles, *Sparse Grid High-Order ADI Scheme for Option Pricing in Stochastic Volatility Models*, in Novel Methods in Computational Finance, M. Ehrhardt, M. Günther, and E. J. W. ter Maten, eds., Springer International Publishing, Switzerland, 2017, ch. 16, pp. 295–312.

[11] A. Gerisch and J. G. Verwer, *Operator splitting and approximate factorization for taxis-diffusion-reaction models*, Applied Numerical Mathematics, 42 (2002), pp. 159–176.

[12] S. González-Pinto, E. Hairer, D. Hernández-Abreu, and S. Pérez-Rodríguez, *AMF-type W-methods for parabolic problems with mixed derivatives*, SIAM Journal on Scientific Computing, 40 (2018), pp. A2905–A2929.

[13] S. González-Pinto, E. Hairer, D. Hernández-Abreu, and S. Pérez-Rodríguez, *PDE-W-methods for parabolic problems with mixed derivatives*, Numerical Algorithms, 78 (2018), pp. 957–981.

[14] S. González-Pinto, D. Hernández-Abreu, and S. Pérez-Rodríguez, *Rosenbrock-type methods with inexact AMF for the time integration of advection-diffusion-reaction PDEs*, Journal of Computational and Applied Mathematics, 262 (2014), pp. 304–321.

[15] S. González-Pinto, D. Hernández-Abreu, and S. Pérez-Rodríguez, *W-methods to stabilize standard explicit Runge-Kutta methods in the time integration of advection-diffusion-reaction PDEs*, Journal of Computational and Applied Mathematics, 316 (2017), pp. 143–160.

[16] S. González-Pinto, D. Hernández-Abreu, and S. Pérez-Rodríguez, *AMFR-W-methods for parabolic problems with mixed derivates. Applications to the Heston model*, Journal of Computational and Applied Mathematics, (2019). Available online since 8 October 2019, DOI: 10.1016/j.cam.2019.112518.

[17] S. González-Pinto, D. Hernández-Abreu, S. Pérez-Rodríguez, and R. Weiner, *A family of three-stage third order AMF-W-methods for the time integration of advection-diffusion-reaction PDEs*, Applied Mathematics and Computation, 274 (2016), pp. 565–584.

[18] M. Griebel, *A parallelizable and vectorizable multi-level algorithm on sparse grids*, in Parallel Algorithms for Partial Differential Equations. Proceedings of the Sixth GAMM-Seminar, Kiel, January 19–21, 1990, W. Hackbusch, ed., Braunschweig, Vieweg, 1991.

[19] M. Griebel, *The combination technique for the sparse grid solution of PDE's on multiprocessor machines*, Parallel Processing Letters, 02 (1992), pp. 61–70.
[20] 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 Albegra, P. de Groen and R. Beauwens, eds., Brussels, April 1991, Elsevier, Amsterdam, 1992, pp. 263–281.
[21] P. Hagan and A. Lesniewski, *LIBOR market model with SABR style stochastic volatility*, Working paper, available at http://lesniewski.us/papers/working/SABRLMM.pdf, (2008).
[22] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1996.
[23] M. Hegland, J. Garcke, and V. Challis, *The combination technique and some generalisations*, Linear Algebra and its Applications, 420 (2007), pp. 249–275.
[24] C. Hendricks, M. Ehrhardt, and M. Günther, *High-order ADI schemes for diffusion equations with mixed derivatives in the combination technique*, Applied Numerical Mathematics, 101 (2016), pp. 36–52.
[25] C. Hendricks, M. Ehrhardt, and M. Günther, *Error Splitting Preservation for High Order Finite Difference Schemes in the Combination Technique*, Numerical Mathematics: Theory, Methods and Applications, 10 (2017), pp. 689–710.
[26] C. Hendricks, C. Heuer, M. Ehrhardt, and M. Günther, *High-order ADI finite difference schemes for parabolic equations in the combination technique with application in finance*, Journal of Computational and Applied Mathematics, 316 (2017), pp. 175–194.
[27] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2003.
[28] K. in 't Hout and C. Mishra, *Stability of ADI schemes for multidimensional diffusion equations with mixed derivative terms*, Applied Numerical Mathematics, 74 (2013), pp. 83–94.
[29] J. Lang and J. G. Verwer, *W-methods in optimal control*, Numerische Mathematik, 124 (2013), pp. 337–360.
[30] C. Leentvaar and C. Oosterlee, *Pricing multi-asset options with sparse grids and fourth order finite differences*, in Numerical Mathematics and Advanced Applications, A. B. de Castro, D. Gómez, P. Quintela, and P. Salgado, eds., Berlin, Heidelberg, 2006, Springer Berlin Heidelberg, pp. 975–983.
[31] C. Leentvaar and C. Oosterlee, *On coordinate transformation and grid stretching for sparse grid pricing of basket options*, Journal of Computational and Applied Mathematics, 222 (2008), pp. 193–209. Special Issue: Numerical PDE Methods in Finance.
[32] J. López-Salas and C. Vázquez, *Sparse Grid Combination Technique for Hagan SABR/LIBOR Market Model*, in Novel Methods in Computational Finance, M. Ehrhardt, M. Günther, and E. J. W. ter Maten, eds., Springer International Publishing, Switzerland, 2017, ch. 27, pp. 477–500.
[33] J. López-Salas and C. Vázquez, *PDE formulation of some SABR/LIBOR market models and its numerical solution with a sparse grid combination technique*, Computers & Mathematics with Applications, 75 (2018), pp. 1616–1634.
[34] A. Lyashenko and F. Mercurio, *Looking forward to backward looking rates: A modelling framework for term rates replacing LIBOR*, Working paper, available at https://ssrn.com/abstract=3330240, (2020).
[35] F. Mercurio, *LIBOR market model with stochastic basis*, Risk Magazine, 12 (2012).
[36] F. Mercurio and M. Morini, *No-Arbitrage dynamics for a tractable SABR term structure Libor Model*, Modeling Interest Rates: Advances in Derivatives Pricing, Risk Books (2009), (2009).
[37] J. Rang and L. Angermann, *New Rosenbrock W-Methods of Order 3 for Partial Differential Algebraic Equations of Index 1*, BIT Numerical Mathematics, 45 (2005), pp. 761–787.
[38] R. Rebonato and R. White, *Linking caplets and swaptions prices in the LMM-SABR model*, The Journal of Computational Finance, 13 (2009), pp. 19–45.
[39] C. Reisinger, *Analysis of linear difference schemes in the sparse grid combination technique*, IMA Journal of Numerical Analysis, 33 (2012), pp. 544–581.
[40] C. Reisinger and G. Wittum, *Efficient Hierarchical Approximation of High-Dimensional Option Pricing Problems*, SIAM Journal on Scientific Computing, 29 (2007), pp. 440–458.
[41] T. Steihaug and A. Wolfbrandt, *An Attempt to Avoid Exact Jacobian and Nonlinear Equations in the Numerical Solution of Stiff Differential Equations*, Mathematics of Computation, 33 (1979), pp. 521–534.
[42] P. J. van der Houwen and B. P. Sommeijer, *Approximate factorization fr time-dependent*

partial differential equations, Journal of Computational and Applied Mathematics, 128 (2001), pp. 447–466. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[43] A. Zeiser, *Fast Matrix-Vector Multiplication in the Sparse-Grid Galerkin Method*, Journal of Scientific Computing, 47 (2010), pp. 328–346.

[44] C. Zenger, *Sparse grids*, in Parallel Algorithms for Partial Differential Equations. Proceedings of the Sixth GAMM-Seminar, Kiel, January 19–21, 1990, W. Hackbusch, ed., Braunschweig, Vieweg, 1991.

[45] OpenMP web page: http://www.openmp.org.