TU Delft

Delft University of Technology

Sparse grid method for highly efficient computation of exposures for xVA

Grzelak, Lech A.

DOI
10.1016/j.amc.2022.127446

Publication date
2022

Document Version
Final published version

Published in
Applied Mathematics and Computation

Citation (APA)
Grzelak, L. A. (2022). Sparse grid method for highly efficient computation of exposures for xVA. *Applied Mathematics and Computation*, *434*, Article 127446. https://doi.org/10.1016/j.amc.2022.127446

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Applied Mathematics and Computation 434 (2022) 127446

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Sparse grid method for highly efficient computation of exposures for xVA

Lech A. Grzelak[a,b,*]

[a] Mathematical Institute, Utrecht University, Utrecht, the Netherlands
[b] Financial Engineering, Rabobank, Utrecht, the Netherlands

ARTICLE INFO

Article history:
Received 29 April 2021
Revised 23 May 2022
Accepted 24 July 2022
Available online 10 August 2022

Keywords:
Stochastic collocation
SC
xVA
Valuation adjustment
Expected exposures
Smolyak's sparse grids
Chebyshev polynomials
Clenshaw–Curtis

ABSTRACT

Every "x"-adjustment in the so-called xVA financial risk management framework relies on the computation of exposures. Considering thousands of Monte Carlo paths and tens of simulation steps, a financial portfolio needs to be evaluated numerous times during the lifetime of the underlying assets. This is the bottleneck of every simulation of xVA.

In this article, we explore numerical techniques for improving the simulation of exposures. We aim to decimate the number of portfolio evaluations, particularly for large portfolios involving multiple, correlated risk factors. The usage of the Stochastic Collocation (SC) method Grzelak et al. (2019)[, together with Smolyak's (1963), Judd et al. (2014) sparse grid extension, allows for a significant reduction in the number of portfolio evaluations, even when dealing with many risk factors. The proposed model can be easily applied to any portfolio and size.We report that for a realistic portfolio comprising linear and non-linear derivatives, the expected reduction in the portfolio evaluations may exceed 6000 times, depending on the dimensionality and the required accuracy. We give illustrative examples and examine the method with realistic multi-currency portfolios consisting of interest rate swaps and swaptions.

© 2022 The Author. Published by Elsevier Inc.
This is an open access article under the CC BY license
(http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Since the Basel Committee introduced their requirements for the credit exposures computation [22], many banks needed to reform their pricing infrastructure. Calculation of risk indicators like expected exposure (EE) or potential future exposures (PFE) is crucial to assess the safety of a bank's positions against market movements in the future. From the pricing perspective, however, these calculations require substantial computational effort. Both indicators need hypothetical scenarios using models calibrated to the financial market. These scenarios represent potential movements of the risk factors in the future. Therefore, to "measure" the bank's exposure, the portfolios need to be evaluated for many future scenarios [12,23], which is a highly intensive task, especially for large portfolios involving thousands of trades that depend on different risk factors.

For large financial institutions, like banks and hedge/pension funds, most of the products are linear, non-exotic. This is particularly true since illiquid derivatives are often capital intensive due to regulatory requirements. Although a portfolio

* Correspondence to: Mathematical Institute, Utrecht University, Utrecht, the Netherlands.
E-mail address: L.A.Grzelak@uu.nl

https://doi.org/10.1016/j.amc.2022.127446
0096-3003/© 2022 The Author. Published by Elsevier Inc. This is an open access article under the CC BY license
(http://creativecommons.org/licenses/by/4.0/)

of linear products is straightforward to value, because of the volume of trades (often exceeding hundreds of thousands of trades), the complete xVA computation may take multiple hours. Especially if a portfolio involves many long-dated swaps with daily compounding or averaging.[1]

Evaluation of xVA poses a significant challenge to researchers and practitioners to improve the required computational time. Although the problem exists for more than a decade already, no genuine progress in efficiency has been achieved. The common approach to enhance the speed focuses on reducing the number of Monte Carlo paths or exposure (monitoring) dates. Either of these choices is at the cost of quality and stability of the ultimate results. An alternative way to improving computational speed is significant investments in hardware, like multi-core CPUs/GPUs [1,11].

This article focuses on the efficient computation of exposures, where many risk factors rule out traditional PDE-based techniques. Alternative approaches to improve the exposure computation are also known. Deep learning, for example, is discussed in Andersson and Oosterlee [2]. A combination of deep learning, GPUs, and forward/backward SDEs can be found in Crépey et al. [6]. The approach proposed in this article provides a novel addition to the methods known in the literature.

Despite all these efforts to improve computation speed, fast evaluation of exposures for a portfolio is still open, especially when dealing with computationally intensive calculations of the related sensitivities. This article aims to take a step forward in reducing the computational effort for simulating exposures. The technique presented is also well-suited for parallelization and thus for GPU computation.

We will develop a highly accurate and fast numerical scheme. For this, we will employ the Stochastic Collocation (SC) method,[2] developed in Grzelak et al. [15], as an efficient approach for approximating distribution functions. The distribution function of interest is then expanded as a polynomial in terms of a random variable that is cheap to sample from at given collocation points, and interpolation occurs between these points. Stochastic collocation points have a specific meaning, i.e., they represent critical features of the probability distribution of interest. The SC method enables us to generate samples from a complex distribution by mainly using interpolation efficiently.

Although in the 1D (one risk factor) case, the number of collocation points, n_1, is typically small,[3] an extensive system of, let us say, d risk factors would require n_1^d collocation points. Thus, it is subject to the so-called curse of dimensionality. Therefore, for large systems of SDEs, as often seen in the xVA context, this is not desirable. However, this number can be reduced using the sparse grid approach, introduced by Smolyak [24]. The sparse grid algorithm in Smolyak's work is concentrated on multidimensional integration and high-dimensional interpolation. The significant advantage of the algorithm is that the number of grid points does not grow exponentially with the dimension, but only polynomially, meaning that the Smolyak method is not subject to the curse of dimensionality. The algorithm has been associated with sparse grids, hyperbolic cross approximation, sparse tensor products, and various applications [26]. In particular, variants of Smolyak's algorithm have been employed in the computation of high-dimensional integrals, in the numerical solution of PDEs and SDEs, and uncertainty quantification [25].

It is also worth mentioning that, recently, the reduction of the number of option valuations using polynomial interpolation has been addressed with Chebyshev interpolation in Gaß et al. [8]. The collocation method with Lagrange interpolations for arbitrage-free option pricing is discussed in Grzelak and Oosterlee [14]. Practical aspects of fast portfolio evaluation using interpolation for xVA are covered in Laris and Ruiz [19], and in Glau et al. [10] the application of Chebyshev interpolation for exposure calculation for the 1D case of Bermudan interest rate swaptions has been presented. Moreover, alternatives to the sparse grid approach for dealing with multi-dimensionality problems in the context of derivatives pricing exist. Recently, a low-rank tensor approximation gained increasing interest in applications to derivative pricing [9,29]. The comparative study of the two methods in the context of uncertainty quantification is presented in Loukrezis et al. [21]. It was shown that the SC method on sparse grids appears to be computationally more efficient, at the cost of accuracy.

The present article is organized as follows: in Section 2, we introduce the Stochastic Collocation (SC) method and discuss exposure computation for portfolios depending on single and multi-factor models together with Smolyak's sparse grid algorithm. Section 3 is a numerical section where the developed method is applied to realistic portfolios of interest rate swaps in multi-currencies. Section 4 focuses on implementation details and improvements. Discussion of error analysis and convergence is covered in Section 5. Concluding remarks are in Section 6.

2. Computations of exposures with the SC method

Mathematically, the positive and negative exposures, $E^+(t,\mathbf{X}(t))$, $E^-(t,\mathbf{X}(t))$, are defined as,

$$E^+(t,\mathbf{X}(t)) := \max(V(t,\mathbf{X}(t)),0), \quad E^-(t,\mathbf{X}(t)) := \max(-V(t,\mathbf{X}(t)),0), \tag{2.1}$$

with

$$V(t,\mathbf{X}(t)) = \mathbb{E}^{\mathbb{Q}}\left[\sum_{j=1}^{L}\frac{M(t)}{M(T_j)}H(T_j,\mathbf{X}(T_j))\Big|\mathcal{F}(t)\right], \quad T_j > t, \tag{2.2}$$

[1] Daily compounding of rates can be seen, among others, in SOFR, ESTR, FF, Brazilian market.

[2] Generally, the term *collocation* denotes techniques that estimate deterministic or stochastic variables by finding a linear predictor from a finite set of observations.

[3] Typically it varies from 3 to 6 to guarantee high-quality approximation (see Grzelak et al. [15] for more details).

where $V(t,\mathbf{X}(t))$ represents the discounted value of a payoff $H(T_j,\mathbf{X}(T_j))$ at time t. The payments are taking place at T_j, $j=1,\ldots,L$, with $T_j>t$, thus only outstanding payments are considered in the exposure computation. $\mathbf{X}(t)$ indicates a *risk factor* on which the derivative $H(\cdot)$ depends, and $M(t)$ stands for the money-savings account.

In practice, the exposures are computed per netting set, often involving hundreds of trades, thus with a netted portfolio involving $\bar{M}$ trades and d different risk factors, the value $V(\cdot)$ is given by:

$$V(t,\mathbf{X}(t)) := \sum_{i=1}^{\bar{M}} V_i(t,\mathbf{X}(t)), \quad \mathbf{X}(t)=[X_1(t),\ldots,X_d(t)]^{\mathrm{T}}. \tag{2.3}$$

Each of the risk factors in $\mathbf{X}(t)$ represents a stochastic process that influences the value of the portfolio. These can be interest rates in different currencies, stocks, inflation, foreign exchange, or commodities. The number of risk factors typically varies in time and depends on the portfolio composition.

Every component of xVA will depend on the exposure computation for any exposure date T_i. In a general setting, it can be represented as follows:

$$\begin{aligned}\mathrm{xVA}(t_0) &= \int_{t_0}^{T} \mathbb{E}^{\mathbb{Q}}\left[\frac{M(t_0)}{M(t)}\chi\left(t,V(t,\mathbf{X}(t))\right)\Big|\mathcal{F}(t_0)\right]\mathrm{d}t \\ &\approx \sum_{k=1}^{N_T} \mathbb{E}^{\mathbb{Q}}\left[\frac{M(t_0)}{M(T_k)}\chi\left(T_k,V(T_k,\mathbf{X}(T_k))\right)\Big|\mathcal{F}(t_0)\right]\Delta t,\end{aligned} \tag{2.4}$$

with some generic function of exposures $\chi(t,x)$ and a discretization grid, $T_1,\ldots,T_{N_T}$. In the case of CVA, for example,[4] $\chi(t,x)$ reads $\chi(t,x)=(1-R_c)E^+(t,x)f_D(t)$, with $E^+(t,x)$ defined in (2.1) and $f_D(t)$ being the default probability,[5] R_c the recovery rate.

In order to estimate the value of the portfolio $V(t,\mathbf{X}(t))$ with N_p simulated Monte Carlo paths, the portfolio needs to be evaluated N_p-times at each exposure date T_k. For a portfolio involving multiple risk factors, the number of paths can be in the range of tens of thousands. From a computational perspective, a low number of portfolio evaluations is desired.

In this article, we propose a method that focuses on a drastic reduction of the number of portfolio evaluations. The proposed method relies on the approximation of the multi-dimensional portfolio $V(t,\mathbf{X}(t))$ by an approximating function, $\widetilde{g}(\{V\}_{i_1,\ldots,i_d},\mathbf{X}(t))$, where $\{V\}_{i_1,\ldots,i_d} := V(t,\{\mathbf{X}\}_{i_1,\ldots,i_d})$. Function $\widetilde{g}(\cdot)$ is built based on only a few evaluations of the portfolio $V(t,\{\mathbf{X}\}_{i_1,\ldots,i_d})$ on the set of so-called "collocation points", $\{\mathbf{X}\}_{i_1,\ldots,i_d}$, obtained from the SC method (SC) [15]. Intuitively, these collocation points can be understood as *optimal* quadrature points that describe the underlying random variable. The idea behind the collocation method is, given the uncertain factors $\mathbf{X}(t)$, to determine the collocation points, $\{\mathbf{X}\}_{i_1,\ldots,i_d}$, being the zeros of an orthogonal polynomial based on variable $\mathbf{X}(t)$.

Once the approximating function $\widetilde{g}(\cdot)$ is determined then the computation of xVA is done as follows:

$$\mathrm{xVA}(t_0) \approx \sum_{k=1}^{N_T} \mathbb{E}^{\mathbb{Q}}\left[\frac{M(t_0)}{M(T_k)}\chi\left(T_k,\widetilde{g}(\{V\}_{i_1,\ldots,i_d},\mathbf{X}(T_k))\right)\Big|\mathcal{F}(t_0)\right]\Delta t, \tag{2.5}$$

with $\{V\}_{i_1,\ldots,i_d} := V(T_k,\{\mathbf{X}\}_{i_1,\ldots,i_d})$. Function $\widetilde{g}(\cdot)$ does not require portfolio evaluations for every Monte Carlo path. The portfolio evaluation takes place only at the "optimal" points $\{\mathbf{X}\}_{i_1,\ldots,i_d}$ that are determined based on the SC method. Once the approximating function $\widetilde{g}(\cdot,\mathbf{X}(t))$ is established, it is evaluated for all Monte Carlo paths. This computation, then, is extremely cheap, as function $\widetilde{g}(\cdot)$ would typically have a polynomial form constructed by the SC method.

In order to measure the quality of the approximations, we will consider discounted expected (positive) exposures defined for the exposure date, T_k, as:

$$EE(t_0,T_k) = \mathbb{E}^{\mathbb{Q}}\left[\frac{M(t_0)}{M(T_k)}E^+(T_k,\mathbf{X}(T_k))\Big|\mathcal{F}(t_0)\right], \tag{2.6}$$

with $M(t)$ is the money-savings account where positive exposures $E^+(T_k,\mathbf{X}(T_k))$ are defined in (2.1) which, for the approximating function $\widetilde{g}(\cdot)$, reads:

$$E^+(T_k,\mathbf{X}(T_k)) := \max(V(T_k,\mathbf{X}(T_k)),0) \approx \max(\widetilde{g}(V(T_k,V(T_k,\{\mathbf{X}\}_{i_1,\ldots,i_d}),\mathbf{X}(T_k)),0).$$

Another metric to measure the quality is the so-called *Potential Future Exposure* (PFE) which measures the maximum credit exposure calculated at some confidence level. The measure can be associated with measuring the quality of the approximating function $\widetilde{g}(\cdot)$ in the tails of the distribution of $V(t,\mathbf{X}(t))$, at $t=T_k$. PFE, at time t, i.e. $\mathrm{PFE}(t_0,t)$, is defined as a quantile of the positive exposure $E^+(t,\mathbf{X}(t))$,

$$\mathrm{PFE}(t_0,t) = \inf\{x\in\mathbb{R} : p \le F_{E^+(t,\mathbf{X}(t))}(x)\}, \tag{2.7}$$

[4] In these illustrative examples we assume no Wrong-Way-Risk, however, the methodology would stay intact even when these assumptions were relaxed.

[5] Default probability $f_D(t)$ depends on a particular "x" in xVA and may involve multiple counterparties. Moreover, its values will depend on the whole discretized interval $[t,t+\Delta t]$.

where $F_{E^+(t,\mathbf{X}(t))}(x)$ is the CDF of positive exposures observed at time t. Coefficient p represents the certainty level, i.e., the quantile level.

The SC method is considered an accurate simulation technique to approximate these quantities; while only a few portfolio evaluations are required to get sufficient accuracy. Typically, the number of points is low and varies from 3 to 6 per risk factor, independently of the number of trades in the underlying portfolio. Once the number of risk factors increases, it is no longer beneficial to apply the standard SC method,[6] but we have to switch to a sparse-grid approach, where the number of portfolio evaluations will not suffer from the curse of dimensionality. In the follow-up section, we will provide details on the computational cost associated with exposure computation under the SC method.

Remark (Machine Learning and Exposure)**.** One may consider building the approximating function $g(\cdot)$ as a process of supervised learning of the portfolio value $V(t,\mathbf{X}(t))$. In the standard machine learning methods, supervised learning involves expensive (portfolio) evaluations to determine the landscape of the objective function in the off-line stage. The method presented in this article may significantly improve the computations in such an expensive off-line stage. A similar idea of a so-called Compression-Decompression technique has been recently presented in Liu et al. [20] in the context of Monte Carlo simulation using deep learning.

2.1. Optimal points in low and high dimensions

Let us start with some background on the collocation method. The proposed method was used to approximate an *expensive* random variable Y utilizing a *cheap* variable X. An approximation is made based on the inversion of the CDF of Y at only a small set of collocation points, being the zeros of an orthogonal polynomial based on variable X.

Since any CDF is uniformly distributed, we have $F_Y(Y) \overset{\mathrm{d}}{=} F_X(X)$. From this representation, realizations of Y, y_n, and X, x_n, are connected via the following inversion relation,

$$y_n = F_Y^{-1}(F_X(x_n)) =: q_Y(F_X(x_n)), \tag{2.8}$$

with $q_Y(p) = \min\{y \in \mathbb{R} : F_Y(y) \geq p\}$ indicating a quantile function. The target is to determine an alternative relationship that does not require many of the "expensive" inversions $F_Y^{-1}(\cdot)$ for all samples of X. The task is thus to find an approximation for function $g(\cdot) = F_Y^{-1}(F_X(\cdot))$ such that

$$F_X(x) = F_Y(g(x)) \quad \text{and} \quad Y \overset{\mathrm{d}}{=} g(X),$$

where evaluations of function $g(\cdot)$ do not require the inversions $F_Y^{-1}(\cdot)$. With a mapping $g(\cdot)$ determined, the CDFs $F_X(x)$ and $F_Y(g(x))$ are not only equal in the distributional sense but also element-wise [15].

Sampling from random variable Y can be decomposed into sampling from a cheap random variable X and a transformation to Y via $g(\cdot)$, i.e., $y_n = g(x_n)$. It is important to choose $g(\cdot)$ to be a basic function. To find a proper mapping function, we need to extract some information from Y.

An efficient method for sampling from variable Y in terms of variable X is obtained by defining $g(\cdot)$ to be a *polynomial expansion*, i.e.

$$y_n \approx \widetilde{g}(x_n) = \sum_{i=1}^{n_1} y_i \psi_i(x_n), \quad \psi_i(x_n) = \prod_{j=1, i\neq j}^{n_1} \frac{x_n - x_j}{x_i - x_j}, \tag{2.9}$$

where x_n is a sample from X and x_i are so-called *collocation points*, y_i is the exact evaluation at collocation point x_i in (2.8), i.e., $y_i = F_Y^{-1}(F_X(x_i))$ in (2.9) and $\psi_i(x_n)$ represent unidimensional basis functions. As mentioned earlier, y_i, is simply the $F_X(x_i)$-quantile thus, in essence, Eq. (2.9) represents an approximation of a random variable Y through interpolation of its quantiles. There are multiple ways of choosing the collocation points x_i. In the standard SC [15] method the "optimal" points are the quadrature points of random variable X. One can also consider these points to be determined by equally spaced quantiles, $p_1, \ldots, p_{n_1}$ with $x_i := q_X(p_i) = \min\{x \in \mathbb{R} : F_X(x) \geq p_i\}$.

Knowing the principles behind the SC method, we need to adapt it to our problem at hand; namely, the approximation of the portfolio given a particular stochastic driver X. A direct application of the SC method, in Eq. (2.8), would require the availability of the portfolio distribution. Unfortunately, such a distribution is not available and can only be determined based on simulation- thus, precisely what we wish to avoid. Therefore, the analogy between cheap and expensive variables in the SC method is not applicable in reducing portfolio evaluations. On the other hand, we know that the value of a portfolio $V(t,X)$ is a function of the stochastic variable X; therefore, we can utilize the SC method to interpolate the value of a portfolio $V(t,x_i)$, $i = 1, \ldots, n_1$, where x_i's are the collocation points, determined based on stochastic quantity, $X(t)$.

The SC method can be extended to multiple inputs using a tensor product of quadrature points, i.e. for d-dimensions we have:

$$\widetilde{g}(x_1, \ldots, x_d) = \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} y_{j_1,\ldots,j_d} \psi_{j_1,\ldots,j_d}(x_1, \ldots, x_d), \tag{2.10}$$

[6] The standard SC method is associated with a Cartesian grid built based on the collocation points constructed based on the corresponding basis for orthogonal polynomials.

where $x_1, \ldots, x_d$ are the collocation for each dimension, $1, \ldots, d$. Although the representation (2.10) constitutes a natural extension of the 1D case, once the number of uncertain parameters increases, approximations based on these tensor product grids become inefficient because the number of collocation points in a tensor grid grows exponentially in dimension, i.e., $n_1 \cdot \ldots \cdot n_d$. In such a scenario, one may consider sparse tensor product spaces as first proposed by Smolyak [24]. More precisely, the Smolyak sparse grid SC method for approximating statistical quantities is used here to reduce the exponential increase of the number of tensor product quadrature points in d dimensions. A linear combination of tensor product operators $p^{|\mathbf{w}|}$, with $|\mathbf{w}| = w_1 + \ldots + w_d$, see [18,28], is given by:

$$\widetilde{g}(x_1, \ldots, x_d) = \sum_{\max(d,\mu+1) \leq |\mathbf{w}| \leq d+\mu} (-1)^{d+\mu-|\mathbf{w}|} \binom{d-1}{d+\mu-|\mathbf{w}|} p^{|\mathbf{w}|}(x_1, \ldots, x_d), \tag{2.11}$$

with μ the sparse grid level often referred to as the *deepness* or *density* parameter, and where the tensor product operator is defined by:

$$p^{|\mathbf{w}|}(x_1, \ldots, x_d) = \sum_{|\mathbf{w}| = w_1 + \ldots + w_d} p^{w_1, w_2, \ldots, w_d}(x_1, \ldots, x_d), \tag{2.12}$$

with

$$p^{w_1, w_2, \ldots, w_d}(x_1, \ldots, x_d) = \sum_{j_1=1}^{m(w_1)} \ldots \sum_{j_d=1}^{m(w_d)} b_{j_1, \ldots, j_d} \psi_{j_1}(x_1) \ldots \psi_{j_d}(x_d), \tag{2.13}$$

where $m(j_k) = 2^{j_k} + 1$ for $j_k \geq 2$ and $m(1) \equiv 1$ representing the number of basis functions in dimension j_k. Functions $\psi_{j_k}(x_k)$ are the j_k'th unidimensional basis functions and $b_{j_1, \ldots, j_d}$ corresponds to polynomial coefficients, where $x_j \in [-1, 1]^d$ are the collocation points for dimension j. These coefficients are constructed in such a way that the approximating polynomial $\widetilde{g}(\cdot)$ exactly matches the original function $g(\cdot)$ in all the points of the constructed grid.

As noted in Judd et al. [18], the classical Smolyak representation in (2.11) involves nested sets, i.e., many elements of the summation contain repeated elements that increase in dimension d and level μ. To further improve the computational efficiency, an alternative formulation, based on disjoint sets, has been proposed [18]:

$$\widetilde{g}(x_1, \ldots, x_d) = \sum_{d \leq |\mathbf{w}| \leq d+\mu} q^{|\mathbf{w}|}(x_1, \ldots, x_d), \tag{2.14}$$

with a tensor product operator:

$$q^{|\mathbf{w}|}(x_1, \ldots, x_d) = \sum_{|\mathbf{w}| = w_1 + \ldots + w_d} q^{w_1, w_2, \ldots, w_d}(x_1, \ldots, x_d), \tag{2.15}$$

and where

$$q^{w_1, w_2, \ldots, w_d}(x_1, \ldots, x_d) = \sum_{j_1=m(w_1-1)+1}^{m(w_1)} \ldots \sum_{j_d=m(w_d-1)+1}^{m(w_d)} b_{j_1, \ldots, j_d} \psi_{j_1}(x_1) \ldots \psi_{j_d}(x_d), \tag{2.16}$$

with $\psi_{j_1}(x_1) \ldots \psi_{j_d}(x_d)$ representing a product of unidimensional basis functions and other settings are as for representation in (2.11). The benefit of the presentation above is that there are no repeated terms across the different functions $p^{w_1, w_2, \ldots, w_d}(x_1, \ldots, x_d)$. Coefficients $b_{j_1, \ldots, j_d}$ must be constructed so that the approximating polynomial $\widetilde{g}(\cdot)$ exactly matches the true function $g(\cdot)$ at the collocation points. As discussed in Judd et al. [18], there are essentially two choices for the computation of the coefficients $b_{j_1, \ldots, j_d}$, analytically involving nested sums or numerically by a numerical solution of a Lagrange interpolation problem given by solving the following algebraic system of equations:

$$\begin{bmatrix} g(x_1) \\ g(x_2) \\ \ldots \\ g(x_d) \end{bmatrix} = \begin{bmatrix} \widetilde{g}(x_1) \\ \widetilde{g}(x_2) \\ \ldots \\ \widetilde{g}(x_d) \end{bmatrix} = \begin{bmatrix} \psi_1(x_1) & \ldots & \psi_d(x_1) \\ \psi_1(x_2) & \ldots & \psi_d(x_d) \\ & \ddots & \\ \psi_1(x_d) & \ldots & \psi_d(x_d) \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_d \end{bmatrix}, \tag{2.17}$$

where $\psi_k : [-1, 1]^d \to \mathbb{R}$ for $k = 1, \ldots, d$ are $d-$dimensional basis functions. The representation above is very flexible as it allows for different choices of collocation points $x_1, \ldots, x_d$ and basis functions ψ_k. The sparse grid construction is based on a unidimensional basis function. A popular choice for the construction of these basis functions is found in the Chebyshev polynomials [17] and their extrema,[7] also known as the Clenshaw–Curtis points [5], defined by a recursive relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2,$$

[7] The article relies on the Smolyak grid, where the computed interpolation coefficients are used on a universal Lagrange interpolation technique. We use families of orthogonal basis functions to ensure the numerical stability of a solution to the Lagrange inverse problem. Although we use the Chebyshev polynomials in the article, it is not a limitation.

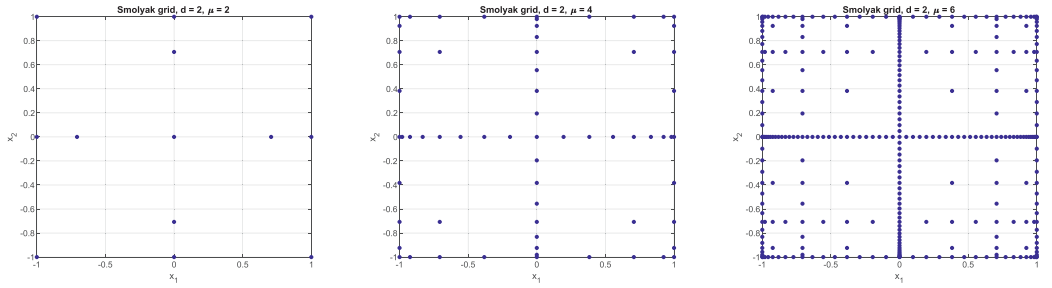

Fig. 1. Collocation points for the $d = 2$ case with varying level parameter $\mu = 2, 4, 6$.

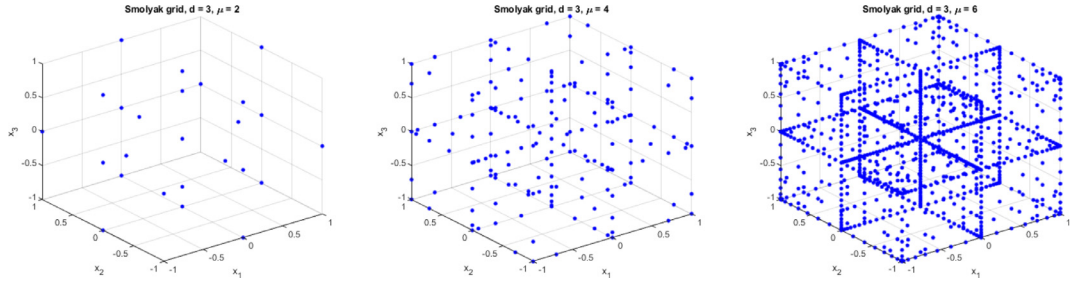

Fig. 2. Collocation points for the $d = 3$ case with varying level parameter $\mu = 2, 4, 6$.

where $T_0(x) = 1$, and $T_1(x) = x$. The Chebyshev polynomial of degree $N-1$ has N extrema given by: $x_i = -\cos(\pi(i-1)/N-1)$ for $i = 1, \ldots, N$. The approximation error of the Chebyshev is known to be polynomial/exponentially convergent under the supreme norm for Lipschitz/analytic functions [21].

The sparse tensor product grids for high dimensions are built upon Clenshaw–Curtis abscissas as they are particularly efficient (since the resulting sparse grids are nested). This hierarchical sampling property allows for reusing the samples when increasing the order when a more accurate model's response is required. In the original literature, we can find works that report the higher accuracy of Clenshaw–Curtis points than of the corresponding number of Gauss quadrature points [27].

An illustration of the collocation points for a 2D and 3D case is presented in Figs. 1 and 2. As we can see, the distribution of the points strongly depends on the level μ. Under the sparse grid method, the number of grid points does not grow exponentially as it does for the tensor product but grows polynomially with dimension d. In a later section, we will investigate its impact on the quality of the results[8]

Remark (High and Low Dimension Methods)**.** The main difference between the low- and high-dimensional methods is that in the low-dimensional case, the collocation points may be chosen based on the underlying risk factors, i.e., the SC method determines the optimal quadrature points based on the distribution, while in a high-dimensional case the collocation points are solely determined based on the sparse grid algorithm. Once the grid points in the $d-$dimensional space $[-1, 1]^d$ are determined, the grid is appropriately scaled to match the domain of the risk factors $\mathbf{X}(t)$. Details on the scaling will be given in Section 4.1.

In Table 1, an overview of the required number of portfolio evaluations depending on dimensionality, d, and grid level, μ, is presented. The results are very encouraging. For example, in the 2D case, we only need between 5 (for $\mu = 1$) and 29 (for $\mu = 3$) portfolio evaluations while in, for example, the 5D case, we only require 11 or 241 evaluations for $\mu = 1$ and $\mu = 3$, respectively. Considering that the "industrial" practice for xVA is to run between 4000 and 25,000 scenarios, the method makes a significant improvement, especially for a portfolio that does not depend on a very high number of risk factors. It is desirable to keep the level parameter, μ, as low as possible from the computational perspective. As it will be presented, the SC model with Smolyak's sparse grid performs accurately in estimating EEs and PFEs, already for $\mu = 2$ or $\mu = 3$.

As a rule of thumb, we distinguish two variants for determining the collocation points: with low dimensionality, $d \leq 2$, the collocation points will be established based on the simulated risk factor $\mathbf{X}(t)$ (standard collocation method). In the high-dimensional case, $d > 2$, these points are established based on the sparse grid method of Smolyak.

Remark. Table 1 illustrates the gain in the portfolio evaluations; however, it is also worth mentioning that the evaluation of the approximating function $\widetilde{g}$, although swift, is not cost-free, i.e., it requires the evaluation of the interpolating function for all the simulated paths. This, however, is independent of the size of the portfolio considered.

[8] Although the formulation may seem somewhat involved, open-source computed code libraries exist. An open-source library for efficient Smolyak grid construction can be found, for example, at https://github.com/EconForge/Smolyak.

Table 1
The required number of portfolio evaluations per exposure date T_k, depending on the approximating algorithm, model configuration, tensor product and sparse grid: comparison of grid points as a function of dimension. n_i indicates the number of collocation points in the i'th dimension and d the number of risk factors that need to be simulated for exposure computation. μ represents the deepness of the grid.

d	Tensor product for $n_i = 3$	Tensor product for $n_i = 4$	Tensor product $n_i = 5$	Smolyak grid			
				$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$
1	3	4	5	3	5	9	–
2	9	16	25	5	13	29	65
3	27	64	125	7	25	69	177
4	81	256	625	9	41	137	401
5	243	1024	3125	11	61	241	801
6	729	4096	15,625	13	85	389	1457
7	2187	16,384	78,125	15	113	589	2465
8	6561	65,536	390,625	17	145	849	3937

Table 2
The total number of portfolio evaluations depending on the parameter μ. The speed-up number reflects the reduction of portfolio evaluations against Monte Carlo simulation with $25k$ paths. In the experiment, we considered $N_T = 75$ exposure dates, and where the error is defined as error $= \frac{1}{N_T}\sum_{k=1}^{N_T} |f(T_k) - \widetilde{f}(T_k)|/f(T_k)1_{f(T_k)\neq 0}$, for $f(T_k)$ being EE or PFE and where $\widetilde{f}(T_k)$ indicates the SC approximation.

$d = 7$	Total # of portfolio eval.	Speed-up vs. Monte Carlo	Error		
			EE	$\text{PFE}_{0.95}$	$\text{PFE}_{0.99}$
$\mu = 1$	75×15	1666	3.9686	2.6203	0.3865
$\mu = 2$	75×113	221	0.1023	0.0720	0.0248
$\mu = 3$	75×589	42	0.0313	0.0101	0.0166
Monte Carlo	$75 \times 25,000$	–	–	–	–

Table 3
The total number of trade evaluations, #, depending on the number of the collocation points, n_1. The speed-up number reflects the reduction of portfolio evaluations against Monte Carlo simulation with $25k$ paths. Number of valuations represents the product of the number of exposure dates: T_k, $N_T = 75$, times the number of collocation points, n_1.

	Total # of sub-portfolio eval.	Speed-up vs. Monte Carlo	Error		
			EE	$\text{PFE}_{0.95}$	$\text{PFE}_{0.99}$
$n_1 = 2$	75×2	12,500	0.6048	0.5346	0.5841
$n_1 = 3$	75×3	8333	0.1432	0.0011	0.1605
$n_1 = 4$	75×4	6250	0.0074	0.0757	0.0088
$n_1 = 5$	75×5	5000	0.0071	0.0289	0.0223
$n_1 = 6$	75×6	4167	0.0014	0.0024	0.0126
Monte Carlo	$75 \times 25,000$	–	–	–	–

2.2. Construction of the approximating portfolio

Given the method described in the previous section, we specify the approximating portfolio construction. Every exposure computation relies on stochastic risk factors that need to be simulated, typically using a big time-step Monte Carlo simulation. In the multi-dimensional case, the risk factor $\mathbf{X}(t)$ is described by the SDEs:

$$\mathrm{d}\mathbf{X}(t) = \alpha(t, \mathbf{X}(t))\mathrm{d}t + \sigma(t, \mathbf{X}(t))\mathrm{d}\mathbf{W}(t), \tag{2.18}$$

with $\alpha(t, \mathbf{X}(t))$ being a drift, $\sigma(t, \mathbf{X}(t))$ is the volatility and $\mathbf{W}(t)$ represents multi-dimensional correlated Brownian motion. From the computational perspective, this step of the exposure simulation is just a fraction of the total portfolio evaluation. Discussion on efficient simulation of multidimensional SDEs can be found in Oosterlee and Grzelak [23], for example.

Once we have simulated the risk factors, the next step is to determine the approximation of the portfolio, i.e.: the value of a portfolio, at any exposure date T_k, which is approximated as:

$$V(T_k, \mathbf{X}(T_k)) \approx \widetilde{g}\Big(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k)\Big), \tag{2.19}$$

where $V(T_k, \mathbf{X}(T_k))$ stands for the original portfolio and $\widetilde{g}\Big(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k)\Big)$ is the approximation of the portfolio based on the portfolio values evaluated only at the collocation points $\{V\}_{i_1,\ldots,i_d} := V(t, \{\mathbf{X}\}_{i_1,\ldots,i_d})$ with $\{\mathbf{X}\}_{i_1,\ldots,i_d} = \{x_{1,1}, x_{1,2}, \ldots x_{2,1}, x_{2,2}, \ldots\}$, and where $x_{i,j}$ is the j'th collocation point for the i'th risk factor.

Note that the approach presented above slightly differs from the original SCMC method in Grzelak et al. [15], i.e., a direct application of the SCMC model would require the construction of the approximating polynomial based on quantiles of an *expensive* random variable, which in the current setting would be represented by the portfolio priced at the exposure date,

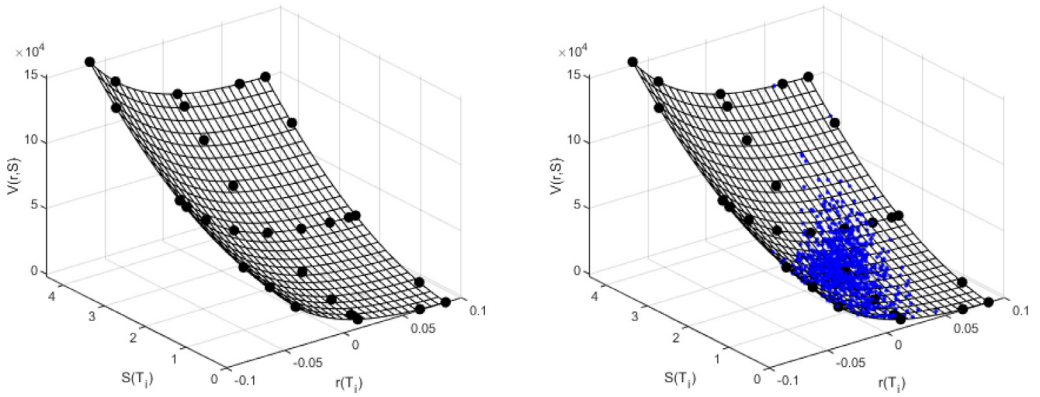

Fig. 3. The construction of the sparse grid for portfolio evaluation. LHS: black dots indicate the collocation values and the grid constructed using Lagrange interpolation. RHS: blue dots illustrate the portfolio values based on the estimated grid for simulated risk factors $\mathbf{X} = [r(t), S(t)]$.

T_k. Such quantiles are not available as they would require evaluation of the portfolio for all Monte Carlo paths- which we wish to avoid. Instead, we build an approximating grid based solely on the underlying risk factors.

Finally, once the approximating function $\widetilde{g}\left(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k)\right)$ is built, we evaluate it for all Monte Carlo paths defined in (2.18). This function evaluation does not require pricing of a portfolio, so the computational cost is low.

In Fig. 3, an illustrative example is presented. In the experiment, we consider a portfolio comprising 20 interest rate swaps and a stock. Thus, a case with two risk factors, $\mathbf{X}(T_k) = [r(T_k), S(T_k)]^{\mathrm{T}}$ observed at a time T_k. In the LHS figure marked with black dots, we see the set of collocation points, $\{V\}_{i_1,i_2}$, that is determined based on the Smolyak sparse grid. Based on these grid points, an interpolated surface of the portfolio values $\widetilde{g}(\{V\}_{i_1,i_2})$ is built with the Lagrange interpolation. The RHS figure illustrates the evaluation of the approximating portfolio using the simulated risk factors, $\widetilde{g}\left(\{V\}_{i_1,i_2}, \mathbf{X}(T_k)\right)$. The procedure described above needs to be repeated for all exposure dates: T_k, $k = 1, \ldots, N_T$, in Eq. (2.19). Once function $\widetilde{g}(\cdot)$ is determined for every T_k and evaluated for all Monte Carlo scenarios, it can be used to calculate risk measures as exposure, potential future exposure, etc.

Remark (Correlation)**.** Since the proposed algorithm focuses on approximating the underlying portfolio, the method does not affect the structure of the underlying stochastic processes, i.e., Monte Carlo paths stay the same; thus, the model correlation structure stays intact.

A detailed discussion on the error of the approximation, convergence, and the computational cost will be discussed in Section 5, while the complete simulation algorithm is described in Appendix A.

3. Practical application and numerical experiments

This section provides a detailed description of using the SC method to approximate portfolios and, therefore, get exposure profiles efficiently. We start with a single currency portfolio comprising multiple interest rate swaps, thus a 1D case with a single risk factor. Later we will discuss portfolios depending on multiple risk factors, i.e., containing derivatives like interest rate swaps in different currencies. We also focus on multi-factor models, like the Hull–White 2 factor model, and we assess the performance when valuing portfolios containing swaptions. [9]

3.1. Portfolio with single currency interest rate products, 1D case

In this Section, we analyze a portfolio that solely depends on a single risk factor; it can be, for example, a portfolio consisting only of EUR swaps.[10] As the industry-standard, we consider the 1D Hull–White (HW) model as the driver of the short rate process $r(t)$, with the dynamics given by:

$$\mathrm{d}r(t) = \lambda(\theta(t) - r(t))\mathrm{d}t + \eta \mathrm{d}W^{\mathbb{Q}}(t), \tag{3.1}$$

and parameters λ and η, term structure function $\theta(t)$ and a Brownian motion under the risk-neutral measure $W^{\mathbb{Q}}(t)$. Function $\theta(t)$ is defined in terms of the zero-coupon bonds, $P_{mrkt}(0,t)$, quoted in the market and is given by:

$$\theta(t) = \frac{1}{\lambda}\frac{\partial}{\partial t} f^r(0,t) + f(0,t) + \frac{\eta^2}{2\lambda^2}\left(1 - \mathrm{e}^{-2\lambda t}\right), \tag{3.2}$$

[9] In the numerical experiment, the pricing has been performed using Python 3.9 on Intel(R) Core (TM) i5-6800 K CPU @ 3.6 Ghz.

[10] Swaps are financial products that enable their holders to swap two sets of interest rate payments.

where the instantaneous forward rate is given by:

$$f^r(0,t) = -\frac{\partial}{\partial t}\log P_{mrkt}(0,t).$$

The Hull–White process in (3.1) allows for a large time step Monte Carlo simulation with the following *exact* formula:

$$r(t) = r(s)\mathrm{e}^{-\lambda(t-s)} + \lambda\int_s^t \theta(u)\mathrm{e}^{-\lambda(t-z)}\mathrm{d}z + \frac{\eta}{\sqrt{2\lambda}}\mathrm{e}^{-\lambda(t-s)}W^{\mathbb{Q}}(\mathrm{e}^{2\lambda(t-s)}-1), \tag{3.3}$$

for any $t > s$.

The number of trades does not affect the method discussed, so we consider generic $\bar{M}$ interest rate (IR) swaps in this portfolio, i.e.:

$$V(t,r(t)) = \sum_{i=1}^{\bar{M}} V_i(t,r(t)), \tag{3.4}$$

with $V_i(t,r(t))$ being the i'th IR swap evaluated at time t. The benefit of using the HW model (3.1) is that at any future time $t > t_0$, the value of the i'th IR swap with payments at $\{T_{j+1}, T_{j+2}, \ldots, T_{\bar{M}_i}\}$ is known analytically and is given by:

$$V_i(t,r(t)) = N_i\sum_{k=j+1}^{\bar{M}_i}\tau_k P(t,T_k)(\ell_k(t)-K_i)\mathbb{1}_{t\leq T_k}, \quad \text{with} \quad \ell_k(t) = \frac{1}{\tau_k}\left(\frac{P(t,T_{k-1})}{P(t,T_k)}-1\right), \tag{3.5}$$

where N_i stands for the notional amount, K_i is the fixed rate and where the ZCB, $P(t,T)$, is defined as Oosterlee and Grzelak [23]:

$$P(t,T) = \mathbb{E}^{\mathbb{Q}}\left[\mathrm{e}^{-\int_t^T r(z)\mathrm{d}z}\middle|\mathcal{F}(t)\right] = \mathrm{e}^{A(t,T)+B(t,T)r(t)}, \tag{3.6}$$

with,

$$B(t,T) = \frac{1}{\lambda}\left(\mathrm{e}^{-\lambda(T-t)}-1\right), \tag{3.7}$$

$$A(t,T) = \lambda\int_t^T \theta(z)B(z,T)\mathrm{d}z + \frac{\eta^2}{4\lambda^3}\left[\mathrm{e}^{-2\lambda(T-t)}\left(4\mathrm{e}^{\lambda(T-t)}-1\right)-3+2\lambda(T-t)\right]. \tag{3.8}$$

Because of the affinity of the Hull–White model, the evaluation of a swap at any time t only depends on the interest rate, $r(t)$, which is highly beneficial.

Given the xVA(t_0) discretization in (2.5), the portfolio in (3.4) needs to be simulated at every time $T_1, \ldots, T_{N_T}$. In a standard approach to get the exposures at each exposure date, T_k, portfolio $V(T_k, r(T_k))$ needs to be evaluated at each path of the stochastic process $r(T_i)$. Typically, to get convergence of the solution, many simulated Monte Carlo paths are used. Often the number of paths is limited by the hardware, especially for large portfolios.

We define the approximating function $\widetilde{g}(V(T_k,\{r_j(T_k)\}_{j=1}^{n_1}), r(T_k))$, which will only require portfolio evaluation at the n_1 collocation points,

$$\{r_j(T_k)\}_{j=1}^{n_1} := \{r_j(T_k), \ldots, r_{n_1}(T_k)\}. \tag{3.9}$$

One needs to decide which technique for the computation of the SC points will be used. Determining collocation points for a Gaussian process is a straight-forward exercise, i.e., the collocation points, $r_{i,j}$, are given by:

$$r_j(T_k) = \mathbb{E}[r(T_k)] + \sqrt{\mathbb{V}\mathrm{ar}[r(T_k)]}x_j^{\mathcal{N}(0,1)}, \quad j = 1, \ldots, n_1,$$

where $x_j^{\mathcal{N}(0,1)}$'s are the collocation points from a standard normal and where the mean and the variance are given by,

$$\mathbb{E}[r(t)|\mathcal{F}(t_0)] = r_0\mathrm{e}^{-\lambda t} + \lambda\int_0^t \theta(z)\mathrm{e}^{-\lambda(t-z)}\mathrm{d}z, \quad \text{and} \quad \mathbb{V}\mathrm{ar}[r(t)|\mathcal{F}(t_0)] = \frac{\eta^2}{2\lambda}\left(1-\mathrm{e}^{-2\lambda t}\right).$$

The collocation points for a standard normal, $x_j^{\mathcal{N}(0,1)}$, are known analytically and are tabulated in Grzelak et al. [15]. These points are also related to the abscissas of the Gauss–Hermite quadrature. The difference between the Gauss–Hermite abscissas $\{x_j^H\}_{j=1}^{n_1}$ and the collocation points $\{x_j^{\mathcal{N}(0,1)}\}_{j=1}^{n_1}$, with $X\sim\mathcal{N}(0,1)$, is the weight function, i.e., we use the normal distribution while Gauss–Hermite quadrature is based on the function e^{-x^2}. The relation between the abscissas is given by $x_j^H = x_j^{\mathcal{N}(0,1)}/\sqrt{2}$, so once we have the abscissas from the Gauss–Hermite quadrature, they can be used in the collocation method.

Once we employ the standard collocation method, the approximating portfolio function could be generated using Lagrange polynomial:

$$\widetilde{g}(V(T_k,\{r_j(T_k)\}_{j=1}^{n_1}), r(T_k)) = \sum_{j=1}^{n_1} V(T_k, r_j(T_k))\psi_j(r(T_k)), \tag{3.10}$$

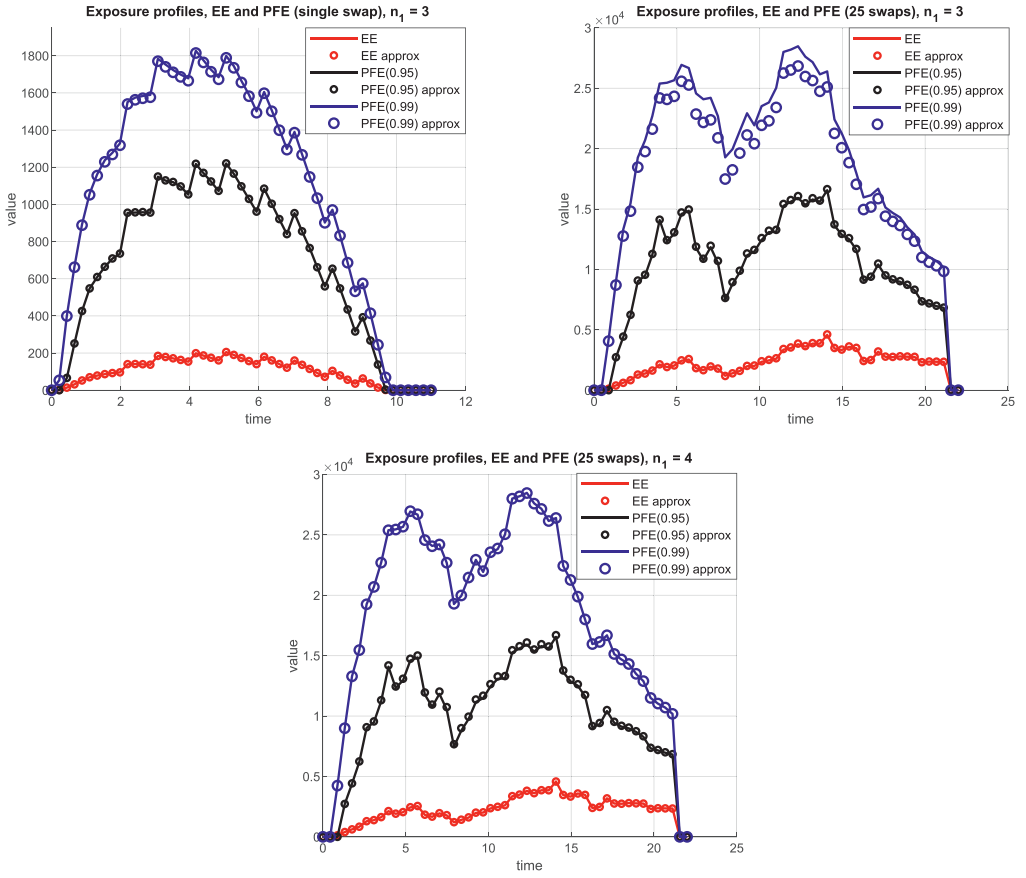

Fig. 4. Portfolio with a single IR swap for $n_1 = 3$ (first figure). Portfolio with 25 swaps for $n_1 = 3$ (second figure) and portfolio with $n_1 = 4$ (third figure).

where $r_j(T_k), r_l(T_k)$ are the collocation points defined in (3.9), $V(T_k, r_j(T_i))$ are the portfolio evaluations for a given expiry date, T_k, evaluated at the collocation points, $r_j(T_k)$, and where $\psi_j(r(T_k))$ is defined as in (2.9).

Given the representation in (3.10), we can evaluate function $\widetilde{g}(\cdot)$ at all stochastic paths of the underlying process, $r(T_i)$. The Lagrange polynomial's particular choice is not obligatory; the reader can decide employing different interpolation routines. However, a benefit of Lagrange representation is that its monomial basis representation allows very efficient polynomial evaluation, even for a large number of samples.

In the numerical experiment, we consider discounted expected (positive) exposures and potential future exposures defined for the exposure date, T_k, specified as:

$$\mathrm{EE}(t_0, T_k) = \mathbb{E}^{\mathbb{Q}}\left[\frac{M(t_0)}{M(T_k)} E^+(T_k, r(T_k)) \Big| \mathcal{F}(t_0)\right],$$

$$\mathrm{PFE}(t_0, T_k) = \inf\{x \in \mathbb{R} : p \leq F_{E^+(T_k, r(T_k))}(x)\},$$

with positive exposures, $E^+(T_k, r(T_k))$, defined in (2.1) for which the approximating function $\widetilde{g}(\cdot)$, reads:

$$E^+(T_k, r(T_k)) \approx \max(\widetilde{g}(V(T_k, \{r_j(T_k)\}_{j=1}^{n_1}), r(T_k)), 0),$$

and where $M(t)$ under the HW model represents the money-savings account defined by $\mathrm{d}M(t) = r(t)M(t)\mathrm{d}t$.

To illustrate the approximation quality, we consider two portfolios: one with a single IR swap with a maturity of 10 years and one with 25 different IR swaps with varying maturities. In the experiment we consider 50 equally spaced exposure dates, i.e. $T_1 = 0.5, T_2 = 1, \ldots, T_{50} = 25$.

In Fig. 4 (first figure), the numerical results for the computer simulation are presented.[11] The figure shows EE's and PFE's for different confidence levels, p. We consider only $n_1 = 3$ collocation points in the experiment, thus only 3 portfolio evaluations per exposure date. The results are highly satisfying. Moreover, for 10k Monte Carlo paths and 50 exposure dates, we have reduced the number of portfolio evaluations from 500k to only 150. Thus, the number of portfolio evaluations is reduced by more than a factor of 3000.

[11] The reported numerical results were obtained with typical model parameters observed in 2020.

Fig. 4 (second figure) shows that in the second portfolio case, the quality in the tail for PFE's, represented with $p = 0.99$, can be further improved. As illustrated in Fig. 4 (third figure), the error for a 99% PFE quantile can be reduced to almost zero once the number of collocation points is increased to $n_1 = 4$. The increase of collocation points has also increased the number of portfolio evaluations from 150 to 200. Then, the reduction of the number of portfolio evaluations is 2500. Discussion on the error propagation depending on the number of the collocation points takes place in Section 5.

3.2. Multi-D case: portfolios with multiple risk factors

Although the example in Section 3.1 is illustrative and confirms the high potential of the collocation method, portfolios typically involve multiple risk factors, for example, swaps in different currencies, foreign exchange forwards, stocks in different currencies, or other products.

Here, we consider a representative portfolio consisting of swaps in d_c different currencies. This implies a problem of $d = 2d_c + 1$ dimensions: d_c processes for foreign interest rates, $r_k(t)$, for $k = 1, \ldots, d_c$, d_c processes for foreign exchange process that exchange a foreign amount of money to a base currency, $y_k(t)$, and 1 for an interest rate in a base currency, $r_b(t)$. This setting can be easily further extended with stochastic volatility or multi-factor processes for interest rates. However, we note that although advanced models for modeling foreign exchange rates with implied volatility smile and skew exist [13], here, we consider the FX model to be based on a hybrid, the Black–Scholes Hull–White model [23].

The system of SDEs for the FX model, under the risk-neutral measure of a base currency, reads:

$$\mathrm{d}y_k(t) = (r_b(t) - r_k(t))y_k(t)\mathrm{d}t + \sigma_{y,k}y_k(t)\mathrm{d}W_k^y(t), \tag{3.11}$$

with r_b representing the base currency (3.1) and with $k = 1, \ldots, d_c$ indicating foreign currencies, where we consider a full matrix of correlations with the following correlation coefficients,

$$\mathrm{d}W_k^y(t)\mathrm{d}W^b(t) = \rho_{k,b}^y\mathrm{d}t, \quad \mathrm{d}W_k^y(t)\mathrm{d}W_j(t) = \rho_{k,j}^y\mathrm{d}t, \quad \mathrm{d}W^b(t)\mathrm{d}W_k(t) = \rho_k\mathrm{d}t,$$

and where the convexity adjusted term structure for foreign short rate processes is given by $\hat{\theta}_k(t) = \theta_k(t) - \eta_k\sigma_{y,k}\rho_{k,k}^y$ with $\theta(t)$ defined in (3.2). Although the short-rate process, $r_k(t)$, contains a convexity correction in the drift, the model belongs to the affine class of processes, and the zero-coupon bonds can be priced as in (3.6). Thus, all the linear interest rate products can be expressed as linear combinations of zero-coupon bonds. The representation of the system in (3.11) allows for large time step Monte Carlo simulation with the following solution for each FX process:

$$y_k(t) = y_k(s)\exp\left(\int_s^t \left(r_b(z) - r_k(z) - \frac{1}{2}\sigma_{y,k}^2\right)\mathrm{d}t + \sigma_{y,k}(W_k^y(t) - W_k^y(s))\right),$$

and large-time steps for $r_b(t)$ and $r_k(t)$, as presented in (3.3).

A cross-currency portfolio consisting of swaps in d_c different currencies is then given by:

$$V(t, \mathbf{X}(t)) = \sum_{i=1}^{M_b} V_i^b(t, r_b(t)) + \sum_{k=1}^{d_c} y_k^b(t) \sum_{i=1}^{M_k} V_i^k(t, r_k(t)), \tag{3.12}$$

with the following state vector, $\mathbf{X}(t) = [r_b(t), y_1^b(t), \ldots, y_{d_c}^b(t), r_1(t), \ldots, r_{d_c}(t)]^{\mathrm{T}}$, and where the first sum indicates the sum over all swaps under the base currency, the second term indicates the summation over all swaps under foreign currencies that are *exchanged* with $y_k^b(t)$ to the base currency. The value of the portfolio in (3.12) is expressed in the base currency. In a realistic scenario involving multiple currencies, the complexity of the exposure computation has increased. A portfolio with d_c foreign currencies would require a tensor grid of $d = 2d_c + 1$ dimensions. This, as presented in Table 1, may lead to a higher number of portfolio evaluations than one would expect with the Monte Carlo simulation. We will apply the SC method with Smolyak's sparse grid algorithm to reduce the number of computations.

Following the procedure described in Section 2.2, we approximate the portfolio, for each exposure date, T_k, with:

$$V(T_k, \mathbf{X}(T_k)) \approx \widetilde{g}\left(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k)\right), \tag{3.13}$$

where function $\widetilde{g}(\cdot)$ is built based on Smolyak's sparse grid method presented in Eq. (2.14) and where $\{V\}_{i_1,\ldots,i_d}$ are the portfolio values evaluated at the collocation points based on Smolyak's sparse grid.

To illustrate the performance of the method, we consider the base currency to be EUR and a portfolio consisting of many swaps in Euros (EUR), British Pounds (GBP), and Polish Złoty (PLN). Such a problem requires a 7-dimensional system of SDEs in (3.11):

$$\mathbf{X}(t) = [r_e(t), y_{\$}^e(t), y_{£}^e(t), y_{zł}^e(t), r_{\$}(t), r_{£}(t), r_{zł}(t)]^{\mathrm{T}}, \tag{3.14}$$

where each process is calibrated to individual market and is then used in the system's simulation of SDEs in (3.11). In the experiment, we consider the case where all the processes are correlated. The correlation coefficients can be estimated based on historical data or estimated from correlation products that are occasionally present in the OTC market.

In the experiment, we consider $N_T = 75$ equally spaced exposure dates, i.e., $T_1, T_2, \ldots, T_{N_T}$, and a Monte Carlo simulation with 25k stochastic paths. Details regarding model parameters used in the experiments are presented in Appendix B.

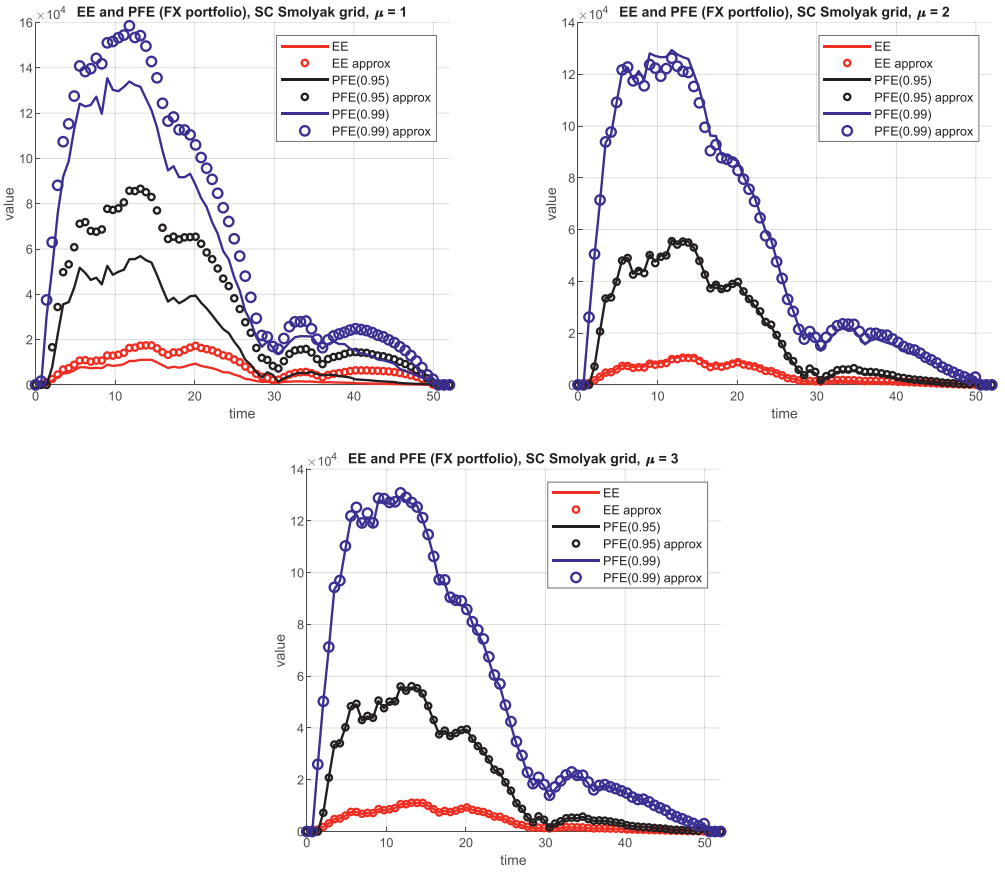

Fig. 5. Expected Exposures, PFE for different significance level (0.95,0.99) depending on the parameter $\mu = 1, 2, 3$.

As previously, we compare the results from a full-blown Monte Carlo simulation against the approximation in (3.13). The graphical representation of the numerical results for the expected positive exposures and PFEs for different significance levels is presented in Fig. 5. The results vary depending on the sparse grid parameter μ, which defines the grid-level.

The visual inspection shows that highly satisfactory results are obtained, already for $\mu = 2$. This particular choice requires only 113 portfolio evaluations per expiry date, T_k. Details regarding the error defined as a relative difference are presented in Table 2. We conclude that the method performs well, and a significant computational gain is guaranteed. In the next section, further improvements for the methods are discussed.

3.3. Divide and conquer: further reduction of portfolio evaluations

In Section 2.1, in Table 1, we have shown a relation between the dimensionality and the number of portfolio evaluations. The application of the sparse grid techniques facilitates an improvement compared to the tensor product. However, it is also clear that with the increasing dimensionality, the benefits of the SC method are reduced, i.e., although Table 1 shows that the number of collocation points grows only polynomially in the dimension, d for $d > 10$, the number of grid points may exceed the typical number of Monte Carlo paths, especially for high parameter value for μ.

Although some portfolios may depend on multiple risk factors, individual trades typically do not. Even trades such as cross-currency swaps will highly unlikely depend on many currencies- this is an important observation as it allows us to the number of reduce portfolio evaluations further. By dividing the portfolio into sub-portfolios, we can *linearize* the problem and further reduce the numerical complexity. For example, a grid with $d = 8$ and level $\mu = 3$, according to Table 1, would require 849 grid points. If we divide the portfolio into two sub-portfolios for $d = 4$, we only will need 137 evaluations per trade, and with four sub-portfolios with $d = 2$, only 29 evaluations yielding a huge computational gain.[12]

Remark (Sub-portfolios and Netting)**.** The division into sub-portfolio has no significant impact on the netting effects. The SC method is only used to approximate the value of a sub-portfolio, which later is evaluated for all the underlying risk factors.

[12] Note that reported evaluations are per sub-portfolio, however since these portfolios are mutually disjoint none of the trades is evaluated more than once.

Let us look at a realistic portfolio discussed in Section 3.2, which consists of swaps in different currencies. The strategy proposed above would yield the following portfolio decomposition:[13]

$$V(T_k, \mathbf{X}(T_k)) = \overline{V}^b(T_k, r_b(T_k)) + \sum_{j=1}^{d_c} y_j^b(t)\overline{V}^j(T_k, r_j(T_k))$$
$$\approx g_b\left(\{\overline{V}^b\}_i, r(T_k)\right) + \sum_{j=1}^{d_c} y_j^b(t) g_j\left(\{\overline{V}^j\}_i, r_j(T_k)\right), \quad i = 1, \ldots, n_1, \tag{3.15}$$

where $\mathbf{X}(t)$ is defined in (3.14) and where $\overline{V}^b(\cdot)$ and $\overline{V}^j(\cdot)$ indicate portfolios in the base currency b and foreign currency j, respectively, and are defined as:

$$\overline{V}^b(T_k, r_b(T_k)) := \sum_{i=1}^{M_b} V_i^b(T_k, r_b(T_k)) \quad \text{and} \quad \overline{V}^j(T_k, r_j(T_k)) := \sum_{l=1}^{M_j} V_l^j(T_k, r_j(T_k)). \tag{3.16}$$

The representation above makes up a further improvement of the SC method for computing exposures. By approximating each of the d_j sub-portfolios, we can separate the value of a portfolio from the cross-currency variable, y_j^b, for $j = 1 \ldots, d_c$. This procedure has reduced the dimensionality from a 7D problem to a sum of 1D problems.

In Table 3, the numerical results are presented. We observe that by dividing the portfolio into sub-portfolios, a significant valuation improvement is achieved, i.e., we only require 4 to 6 evaluations per exposure date to guarantee errors that are even smaller than those reported in Table 2. Compared with a brute-force Monte Carlo pricing, the speed improvement varies from a factor of 6000 for $n_1 = 4$ to 4000 for $n_1 = 6$ collocation points.

The decomposition into a sum of 1D problems, as described above, may not always be possible in an actual portfolio as it depends on the type of derivatives, e.g., with cross-currency swaps depending on three currencies, we would expect sums of 3D portfolios.

Remark (Parallelization)**.** The encouraging results presented in this section can be further improved: each sub-portfolio can be evaluated independently, allowing for parallelization at the sub-portfolio level. Although relatively cheap, one can also parallelize the evaluation of function $\widetilde{g}(\cdot)$ at each Monte Carlo path facilitating even further speed gains.

3.4. Portfolios with non-linear products driven by multi-factor models

In the previous section, we have presented the benefits of dividing the pricing problems into pieces. Unfortunately, the same strategy cannot be applied to problems that depend on multiple factor models. When determining risk limits, it is desired to determine PFEs based on multi-factor models. Such models mimic the yield curve dynamics time much better than single-factor models. In this section, we consider such a case.

This section considers an extension of the case discussed in Section 3.3- a portfolio consisting of linear and non-linear derivatives driven by multi-factor processes. In particular, we consider the portfolio defined in (3.15) with additional, non-linear products, namely- several interest rate swaptions with the varying strike, maturity and notional values. Furthermore, swaptions are added for each currency, and the portfolio is ensured to be balanced, i.e., there is no dominance of either interest rate swaps or swaptions that potentially could skew the results.

In the experiment, each of the interest rate processes is driven by the 2-factor Hull–White model [16], $r(t) = x^r(t) + y^r(t) + \psi^r(t)$ with $x^r(t)$ and $y^r(t)$ being OU processes with the initial and long-term mean equal to 0 and where $\psi^r(t)$ is a term-structure (details are presented in Brigo and Mercurio [4]). Swaptions, contrary to interest rate swaps, are computationally more expensive as they require numerical integration and optimization for every Monte Carlo realization $\{x^r(t_i), y^t(t_i)\}$ (the details regarding the pricing of swaptions under the two-factor models are, for reader's convenience, in Appendix C)).

Under the two-factor model the approximating formula is as follows,

$$V(T_k, \mathbf{X}(T_k)) \approx g_b\left(\{\overline{V}^b\}_{i_1,i_2}, x^r(T_k), y^r(T_k)\right) + \sum_{j=1}^{d_c} y_j^b(t) g_j\left(\{\overline{V}^j\}_{i_1,i_2}, x_j^r(T_k), y_j^r(T_k)\right), \tag{3.17}$$

where $i_1 = 1, \ldots, n_1$, $i_2 = 1, \ldots, n_2$, the state vector $\mathbf{X}(t)$ now includes a pair of $x^r(t)$, $y^r(t)$ per each currency. Both $\overline{V}^b(\cdot)$ and $\overline{V}^j(\cdot)$ are the portfolios in each currency, as defined similarly as in (3.16), with the exception that these portfolios explicitly depend on the underlying stochastic factors, $x^r(t)$ and $y^r(t)$, and additionally include swaptions.

As in the single factor Hull–White interest rate model, the model's performance is excellent. In Table 4, convergence results are presented. We report that already, for $\mu = 2$, high-quality results are obtained. For two-dimensional problems, $d = 2$, with $\mu = 2$, require only 13 portfolio evaluations (see Table 1), thus resulting in a drastic reduction in the computational time.

Fig. 6 illustrates the quality for the approximation for different exposure dates. Again, already for $\mu = 2$ we report no significant errors along the lifetime of the portfolio.

[13] We consider here n_1 collocation points for every dimension, $j = 1, \ldots, d_c$.

Table 4
The error associated with the sparse approximation of the "non-linear" portfolio of derivatives. The settings of the experiment are chosen described in Table 2. The number of portfolio evaluations is as in Table 3.

	Error		
	EE	$PFE_{0.95}$	$PFE_{0.99}$
$\mu = 1$	0.2518	0.2553	0.2932
$\mu = 2$	0.0266	0.0111	0.0177
$\mu = 3$	0.0213	0.0178	0.0250

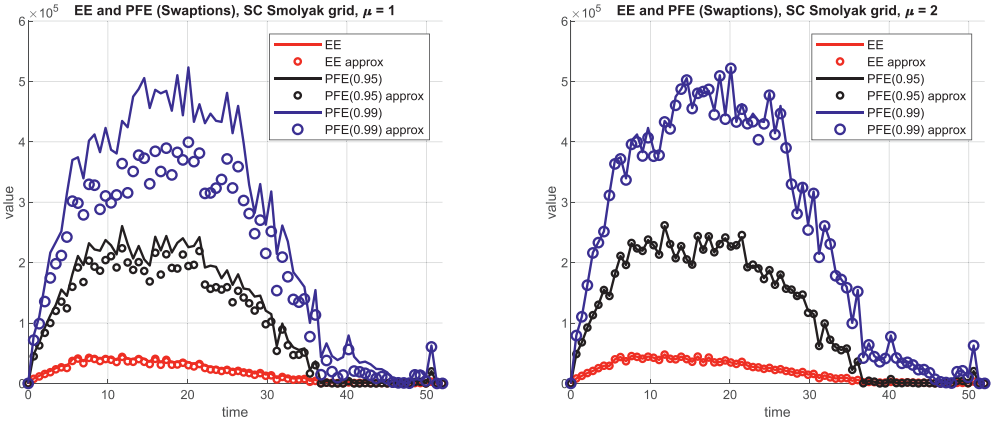

Fig. 6. Portfolio with interest rate swaps and swaptions in different currencies. Expected Exposures, PFE for different significance level (0.95,0.99) depending on the parameter $\mu = 1, 2$.

We conclude that under a multi-factor setting, the sparse grid methodology for approximating a portfolio consisting of linear and non-linear derivatives, there is no deterioration observed compared to a portfolio solely consisting of linear products. Furthermore, our experiments suggest that the method can be applied to even more exotic, possibly callable, derivatives.

4. The SC method: implementation details and improvements

This section focuses on details and improvements regarding the SC method when applied to portfolio evaluations. Here, we discuss a domain scaling for Smolyak's grid, optimal collocation points, and adaptive grids for multi-D cases.

4.1. Domain scaling

Since Smolyak's grid is constructed in a d-dimensional cube, $[-1, 1]^d$, a transformation of the grid needs to take place. It is a distinct feature compared to the SC method for low dimensions where the grid is built from the optimal points based on the quadrature points (see Section 2.1). Each dimension, i, is stretched with the following transformation:

$$x_i = \frac{1}{2}u_i(ub_i - lb_i) + lb_i + \frac{1}{2}(ub_i - lb_i),$$

where u_i is the initial grid point in $[-1, 1]$ and lb_i and ub_i are the lower and upper bound, respectively. Both quantiles, lb_i and ub_i, can be computed either based on Monte Carlo paths or by using analytical properties of the underlying distribution of $X(t)$. Using Monte Carlo paths for a process $X(t)$, one can either obtain $lb_i = \min X(T_k)$ and $lb_i = \max X(T_k)$ at the exposure date T_k or compute quantiles at a certain level α:

$$lb_i = \min\{y \in \mathbb{R} : F_{X(T_k)}(y) \geq 1 - \alpha\}, \tag{4.1}$$

$$ub_i = \min\{y \in \mathbb{R} : F_{X(T_k)}(y) \geq \alpha\}. \tag{4.2}$$

Fig. 7 illustrates the stretched grid using (4.1) and (4.2) applied to a 2D case. The figure shows the grid stretched for correlated variables X_1 and X_2. We have, for $\alpha = 0.95$, obtained the following lower and upper bounds: $lb_1 = -0.003$, $lb_2 = 1.055$, $ub_1 = 0.023$ and $ub_2 = 1.356$. A similar effect is achieved for other processes in the underlying system of SDEs.

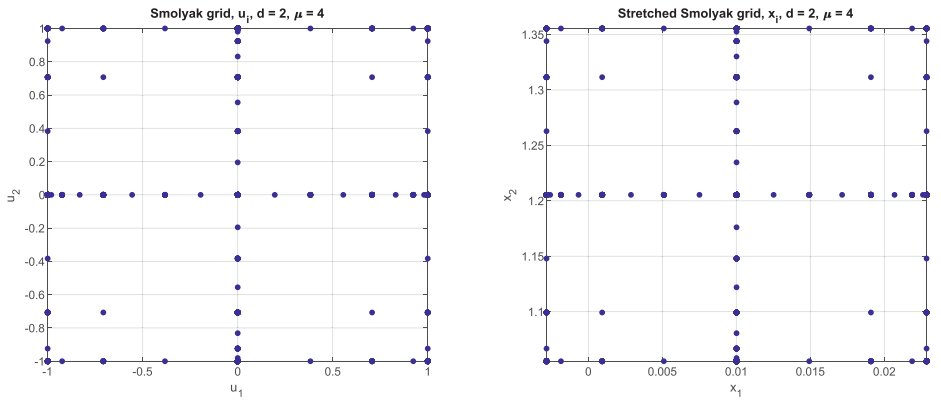

Fig. 7. Smolyak's grid stretching based on quantiles. LHS: original grid. RHS: stretched grid.

4.2. Sub-optimal collocation points

Although the collocation points under the Smolyak's algorithm are given and do not depend on the properties of the underlying processes,[14] this is not the case for the *standard* SC method, described in Section 2.1. These collocation points are determined based on the moments of the underlying stochastic model. This article focused on the most commonly used process for assets. We dealt with either normal or lognormal distributions; however, when the underlying processes are complex, one would need to compute moments of such a process to establish the corresponding collocation points. This could be troublesome for two reasons: the moments are not always available in closed form, and moments obtained from Monte Carlo simulation may give rise to inaccurate results (the collocation method requires an LU decomposition of the so-called Gram matrix constructed by n_i^2 moments, n_i being the number of collocation points in the i'th dimension).

To address the issue described above, one can calculate the optimal collocation points based on the normally distributed kernel process, i.e., for a random variable ξ for which the moments are not available, the collocation points are computed by:

$$\xi_i = F_{\xi}^{-1}(F_{\mathcal{N}(0,1)}(x_i^{\mathcal{N}(0,1)})).$$

If we consider the CDF, $F_{\xi}(\cdot)$, of a continuous random variable ξ, the mapping $y = F_{\xi}(x)$ is bijective and $F_{\xi}(x)$ is strictly increasing, so is $F_{\xi}^{-1}(y)$. This implies that the argument x can be obtained by the inverse interpolation of $F_{\xi}(y)$ against y, which can be done at essentially *no cost*. When the random variable ξ is obtained from a Monte Carlo simulation, $F_{\xi}(\cdot)$ can be estimated from the empirical cumulative distribution function.

4.3. Anisotropic and adaptive grids

When dealing with sparse grids and valuation of portfolios, each of the dimensions is treated equally (the same number of collocation points in each dimension). This may not be desired, as some of the risk factors may be dominant in a portfolio risk profile. Anisotropic sparse grids address this problem and allow for unequal distribution of the collocation points over different dimensions. Discussions on this subject are well-covered in Judd [17].

Now, let us look at another aspect of the grid construction process. Figs. 1–3, show that the SC method generates a rectangular grid of collocation points. This may be sub-optimal, especially for correlated processes where the realizations would cluster. In such a multidimensional case where the underlying model consists of d correlated risk factors, we use a multidimensional version of the normal collocation points. Thus, the grid of points would be determined by a multidimensional Gaussian copula estimated based on the risk factors $\mathbf{X}(t)$.

For d risk factors, $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_d(t)]^{\mathrm{T}}$, with their corresponding number of collocation points $n_1, n_2, \ldots, n_d$, we have:

$$\begin{aligned}
\hat{x}_{i_1} &= \mathbf{L}_{1,1} x_{i_1}^{\mathcal{N}(0,1)}, \\
\hat{x}_{i_2} | \hat{x}_{i_1} &= \mathbf{L}_{2,1} \hat{x}_{i_1} + \mathbf{L}_{2,2} x_{i_2}^{\mathcal{N}(0,1)}, \\
\hat{x}_{i_3} | \hat{x}_{i_1}, \hat{x}_{i_2} &= \mathbf{L}_{3,1} \hat{x}_{i_1} + \mathbf{L}_{3,2} \hat{x}_{i_2} + \mathbf{L}_{3,3} x_{i_3}^{\mathcal{N}(0,1)},
\end{aligned} \tag{4.3}$$

where $x_{i_j}^{\mathcal{N}(0,1)}$ indicates the j'th collocation point from a standard normal and where $\mathbf{L}_{i,j}$ is the (i,j)'th element of the Cholesky decomposition, $\mathbf{C} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ with $\mathbf{C}$ being the correlation matrix. In order to compute all the collocation points, we

[14] Except for the domains, as discussed in Section 4.1.

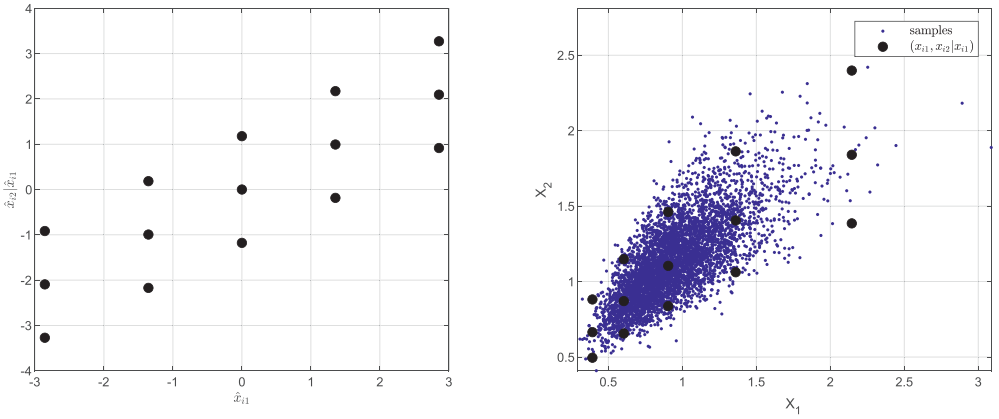

Fig. 8. LHS: 2D collocation points $\hat{x}_{i1}$ and $\hat{x}_{i2}|\hat{x}_{i1}$ for $\rho = 0.6$. RHS: Scatter plot for two lognormal random variables $X_1 \sim e^{Z_1}$ and $X_2 \sim e^{Z_2}$ where $Z_1 \sim \mathcal{N}(-0.1, 0.3)$, $Z_2 \sim \mathcal{N}(0.2, 0.2)$, $N_1 = 5$, $N_2 = 3$ and $\rho_{Z_1,Z_2} = 0.6$ and the corresponding collocation points x_{i1} and $x_{i2}|x_{i1}$.

require the following inversions:

$$\begin{aligned} x_{i_1} &= F_{X_1}^{-1}(F_{\mathcal{N}(0,1)}(\hat{x}_{i_1})), \\ x_{i_2}|x_{i_1} &= F_{X_2}^{-1}(F_{\mathcal{N}(0,1)}(\hat{x}_{i_2}|\hat{x}_{i_1})), \\ x_{i_3}|x_{i_1}, x_{i_2} &= F_{X_3}^{-1}(F_{\mathcal{N}(0,1)}(\hat{x}_{i_3}|\hat{x}_{i_1}, \hat{x}_{i_2})). \end{aligned} \tag{4.4}$$

In Fig. 8, an illustrative example of the grid distribution in 2D is presented. The LHS figure shows the grid distribution given in (4.3), while the RHS figure corresponds to the actual grid points from Eq. (4.4).

Given the representation above the approximation for the portfolio value is expressed by:

$$\widetilde{g}(V(T_k, \{\mathbf{X}\}_{i_1,\ldots,i_d}), \mathbf{X}(T_k)) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} V(T_i, x_{i_1}; x_{i_2}|x_{i_1}; \ldots; x_{i_d}|x_{i_1}, \ldots, x_{i_{d-1}}) \hat{\psi}(\mathbf{X}(T_k)),$$

where $\hat{\psi}(\mathbf{X}(T_k))$ is defined in (3.10).

5. Error analysis and convergence

This section focuses on the error analysis of the approximating function $\widetilde{g}(\cdot)$. In particular, we analyze the impact of the approximation on a portfolio containing interest rate swaps, possibly in different currencies and driven by the BSHW model defined in (3.11). Such a portfolio is of particular interest in the industry. The result below gives us essential insight into the distributional properties of such a portfolio.

Lemma 5.1 (Distribution of a portfolio of multi-currency swaps)**.** *Under the multi-currency BSHW model, defined in (3.11), a portfolio, at the exposure date T_k, consisting of interest rate swaps in d_c foreign currencies, is distributed as a linear combination of lognormally distributed random variables*

$$\mathbf{V}(T_k, \mathbf{X}(T_k)) \sim \sum_{\ell \in \Omega} c_\ell e^{Z_\ell(T_k)}, \tag{5.1}$$

with a constant $c_\ell \in \mathbb{R}$, where $Z_\ell(T_k) \sim \mathcal{N}(m_\ell, \Sigma_\ell^2)$, for a certain constant mean, m_ℓ, variance, Σ_ℓ^2, and where Ω represents the set of all ZCBs under all underlying currencies.

Proof. Under the Hull–White model, every zero-coupon bond is log-normally distributed, i.e.,

$$P(t,T) = \exp(A(t,T) + B(t,T)r(t)) \sim \log\mathcal{N}\left(\mu_P, \sigma_P^2\right), \tag{5.2}$$

with a certain mean parameter, μ_P, variance, σ_P^2, where $r(t)$ is the short-rate at time t and $A(t,T)$ and $B(t,T)$ are time-dependent functions defined in (3.8) and (3.7). As a result, an interest rate swap, as presented in (3.5) is, after some simplifications, given as a linear combination of zero-coupon bonds,

$$V(t, r(t)) = N\left(P(t,T_j) - P(t,T_M)\right) - NK \sum_{k=j+1}^{M} \tau_k P(t,T_k), \tag{5.3}$$

for a certain strike, K, a notional amount, N, and swap payment dates $T_{j+1}, \ldots, T_M$. The interest rates swap defined in (5.3) can be recognized as a linear combination of lognormally distributed random variables:

$$V(t, r(t)) = \sum_{k=j}^{M} a_k \mathrm{e}^{\mu_k + \sigma_k r_b(T_k)} =: \sum_{k=j}^{M} a_k Y_k, \tag{5.4}$$

for some constant parameter a_k and where $Y_k \sim \log\mathcal{N}\left(\mu_k, \sigma_k^2\right)$. The same analogy holds for a portfolio consisting of interest rate swaps, even in the case of a portfolio involving multiple currencies, driven by the SDEs in (3.11),

$$V(t, \mathbf{X}(t)) = \sum_{i=1}^{M_b} V_i^b(t, r_b(t)) + \sum_{k=1}^{d_c} y_k^b(t) \sum_{i=1}^{M_k} V_i^k(t, r_k(t)) \overset{\mathrm{d}}{=} \sum_{\ell \in \Omega} c_\ell \mathrm{e}^{Z_\ell(T_k)}, \tag{5.5}$$

where c_l is a constant in $\mathbb{R}$ and $Z_l(T_k) \sim \log\mathcal{N}\left(m_\ell, \Sigma_\ell^2\right)$ for some m_ℓ and Σ_ℓ^2.■

From the representation above, we conclude that under the BSHW model in (3.11), a portfolio consisting of swaps under different currencies can be represented as a linear combination of correlated log-normally distributed random variables. This also means that to estimate the error, we need to look at the SC method and its quality in approximating a linear combination of lognormals in (5.5). Unfortunately, even the problem of the sum of lognormals is unresolved. We refer to an overview of attempts and approximations in Dufresne [7]. Let us now proceed with the error estimates for the 1D and multi-D cases.

In the 1D or 2D instances in which the collocation points x_i correspond to the zeros of an orthogonal polynomial, the following equality, because of the connection to Gauss quadrature, in L^2, holds:

$$\int_{\mathbb{R}} (V(T_k, x) - \widetilde{g}(T_k, x))^2 f_X(x) \mathrm{d}x = \sum_{i=1}^{n_1} (V(T_k, x_i) - \widetilde{g}(T_k, x_i))^2 \omega_i + \epsilon_{n_1} = \epsilon_{n_1}, \tag{5.6}$$

where $V(T_k, x) - \widetilde{g}(T_k, x)$ represents the difference between the exact portfolio and the SC approximated function, $f_X(x)$ is the weight function, and where ω_i, for $i > 0$, are the quadrature points. When the Gauss–Hermite quadrature is used with n_1 collocation points, the approximation error of the CDF can be estimated as,

$$\epsilon_{n_1} = \frac{n_1! \sqrt{\pi}}{2^{n_1}} \frac{\Psi^{(2n_1)}(\hat{\xi})}{(2n_1)!},$$

where

$$\Psi(x) := (V(T_k, x) - \widetilde{g}(T_k, x))^2 = \left(\frac{1}{n_1!} \frac{\mathrm{d}^{n_1} g(T_k, x)}{\mathrm{d}x^{n_1}} \Big|_{x=\hat{\xi}} \prod_{i=1}^{n_1} (x - x_i) \right)^2. \tag{5.7}$$

Since the approximations proposed in this article are also used to estimate PFEs at different significance levels, we also need to assess the so-called *tail risk*. Since functions $V(T_k, x)$, and $\widetilde{g}(T_k, x)$ agree at the collocation points, the upper bound for a risk factor $X(T_k)$ is given by Grzelak et al. [15]:

$$\mathbb{E}\Big[(V(T_k, X(T_k)) - \widetilde{g}(T_k, X(T_k)))^2 | X > x_*\Big] \leq \frac{1}{\mathbb{P}[X(T_k) > x_*]} \frac{n_1! \sqrt{\pi}}{2^{n_1}} \frac{\Psi^{(2n_1)}(\hat{\xi})}{(2n_1)!}, \tag{5.8}$$

with $\Psi(x)$ defined in (5.7). In a special case, when the risk factor is normally distributed, $X(T_k) \sim \mathcal{N}(0, 1)$, and by performing integration by parts twice, one can show that for $x_* > 0$:

$$\mathbb{P}[X(T_k) > x_*] \geq \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x_*^2/2} \left(\frac{1}{x_*} - \frac{1}{x_*^3} \right), \tag{5.9}$$

so that the error for $x_* > 1$ is bounded by:

$$\mathbb{E}\Big[(V(T_k, X(T_k)) - \widetilde{g}(T_k, X(T_k)))^2 | X(T_k) > x_*\Big] \leq \pi \sqrt{2} \mathrm{e}^{x_*^2/2} \frac{x_*^3}{x_*^2 - 1} \frac{n_1!}{2^{n_1} (2n_1)!} \Psi^{(2n_1)}(\hat{\xi}).$$

The factorial $(2n_1)!$ in the expression above is dominant and thus, for a smooth function $\Psi(\xi)$, we obtain:

$$\lim_{n_1 \to \infty} \mathbb{E}\Big[(V(T_k, X(T_k)) - \widetilde{g}(T_k, X(T_k)))^2 | X > x_*\Big] = 0.$$

Now, let us investigate the error propagation under sparse grid methods. The quality of sparse grid methods comes with strict requirements on the smoothness of the high-D functions and bounded mixed derivatives, i.e., the convergence of the method is specified for a particular *regularity* parameter r, which is defined for function spaces as follows,

$$F_d^r := \left\{ f : \omega^d \to \mathbb{R}; \ \frac{\partial^\alpha f(\mathbf{x})}{\partial \mathbf{x}^\alpha} \ \text{exists and} \ \left|\left| \frac{\partial^\alpha f(\mathbf{x})}{\partial \mathbf{x}^\alpha} \right|\right|_\infty \leq 1; \ \text{if} \ \ \alpha \leq r \right\}. \tag{5.10}$$

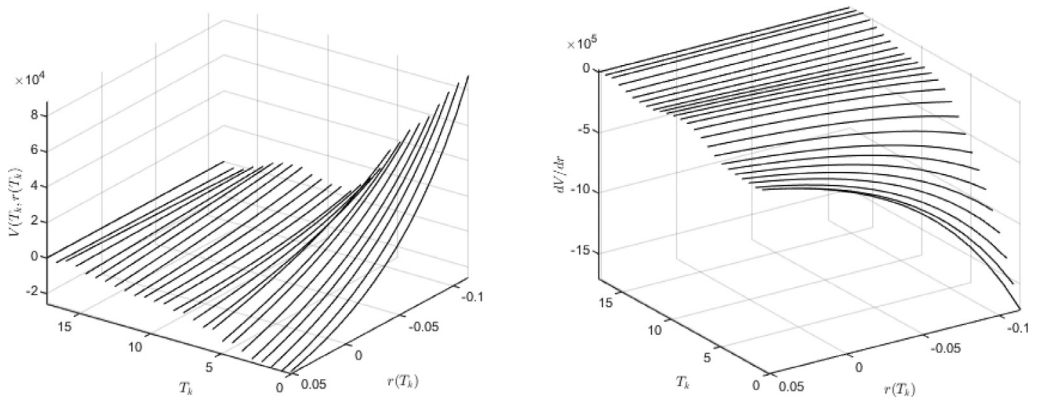

Fig. 9. LHS: The portfolio of swaps, $\overline{V}^b(T_k, r_b(T_k))$ for $k = 1, \ldots, N_T$, in the base currency as a function of exposure date, T_k, defined in (3.15) and interest rate, $r(T_k)$. RHS: Derivative of the portfolio with respect to interest rates.

Then, according to Barthelmann et al. [3] for $V(T_k, \mathbf{x}) \in F_d^r$ and a particular finite domain, ω, the corresponding Smolyak quadrature rule of degree μ has the asymptotic convergence rate of

$$\sup_{V(T_k,\mathbf{x})\in F_d^r} \left|\left|V(T_k, \mathbf{x}) - \widetilde{g}(T_k, \mathbf{x})\right|\right|_\infty \leq \frac{c_{d,r}}{n^r(d,\mu)} (\log n(d,\mu))^{(r+2)(d-1)+1}, \tag{5.11}$$

where r is the regularity parameter defined in (5.10), $n(d, \mu)$ represents the total number of grid points used in the grid construction (see Table 1), and $c_{d,r}$ is a constant that only depends on dimension, d, and the regularity parameter, r. The error bound in (5.11) shows the relation between the number of grid points and smoothness of the pricing function.

The condition defined for the partial derivatives in (5.10) requires that a continuous function f is contracting and is associated with a satisfied Lipschitz condition. Lemma 5.10 shows that the value of a multi-currency portfolio with interest rate swaps driven by the BSHW model in (3.11) is of an exponential form. The exponential function, $e^{\mathbf{x}}$, is not globally Lipschitz continuous as it becomes arbitrarily steep for $\mathbf{x} \to +\infty$ or $\mathbf{x} \to -\infty$. However, any continuously differentiable function is locally Lipschitz, as continuous functions are locally bounded, therefore its gradient is locally bounded.

In Fig. 9, the base currency portfolio, defined in Section 3.2, and its derivatives to the underlying risk factor are presented. We observe a rather smooth profile of the value for different exposure dates, T_k. However, the portfolio's value increases exponentially for negative interest rates, especially for $T_k \to 0$. Therefore, one would expect more grid points to be needed to compensate for the exponential growth of the portfolio. However, from a practical perspective, such a scenario is unrealistic as it would imply highly negative rates for short maturities. On the other hand, in the computation of xVA, exposures are multiplied with default probabilities that for $T_k \to 0$ are very low; therefore, the overall impact, even for significant error, is minimal.

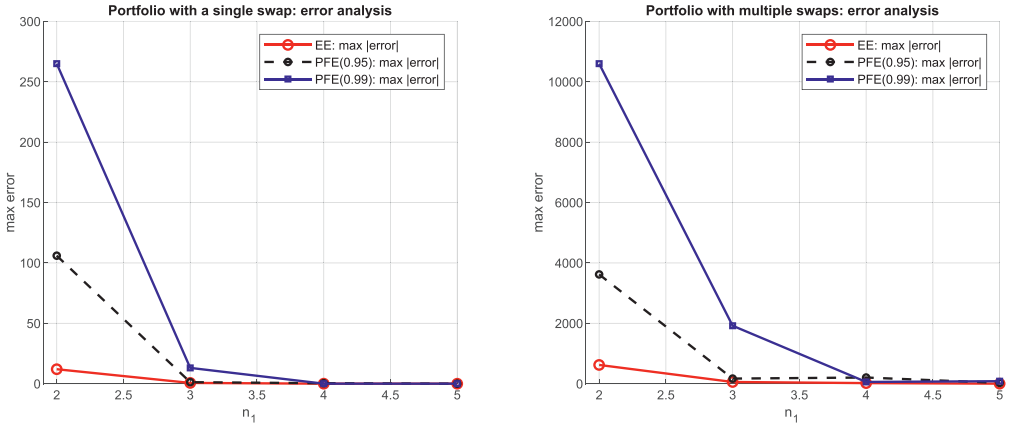

Fig. 10. Error is defined as $\max_k |f(T_k) - \widetilde{f}(T_k)|$, where $f(T_k)$ is EE or PFE, and where $\widetilde{f}(T_k)$ corresponds to the SC approximation. The details regarding the portfolios under consideration are defined in Section 3.1. LHS: single swap portfolio. RHS: multi-swap case.

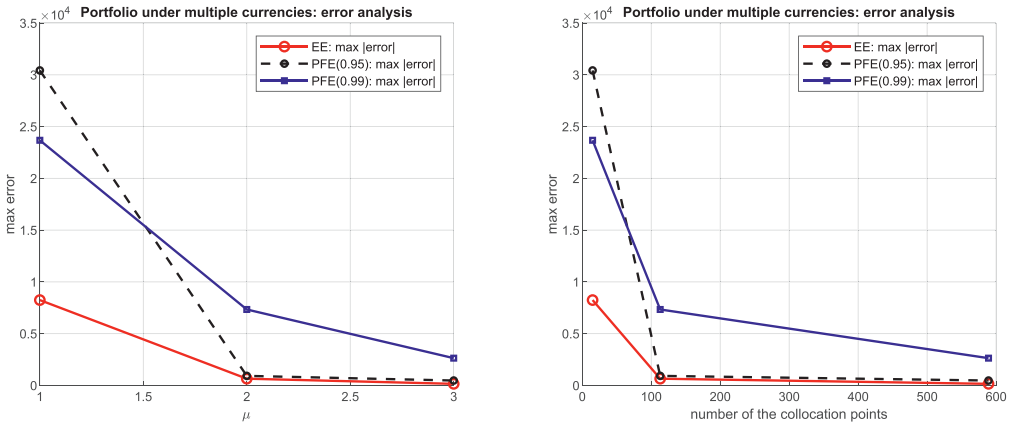

Fig. 11. Error as a function of level, μ, is defined as $\max_k |f(T_k) - \widetilde{f}(T_k)|$ where $f(T_k)$ is EE or PFE, and where $\widetilde{f}(T_k)$ corresponds to the SC approximation. The details regarding the portfolios under consideration are defined in Section 3.2.

5.1. Convergence with a numerical experiment

This section analyzes the convergence of the SC method depending on the number of collocation points, n_1 (1D case), and the level parameter, μ (7D case), based on the portfolio cases discussed in Section 3. The dependence of the parameter μ on the number of the collocation points is presented in Table 1.

Fig. 10 illustrates the convergence results for the portfolio based on the Lagrange interpolation with the collocation points determined based on the quadrature rule. We report excellent results: already $n_1 = 4$ guarantees high accuracy. In Fig. 11, the multi-D case is considered. As expected, the fastest convergence is obtained for EE, while a higher number of grid points is necessary for a satisfactory convergence in tails of the distribution, PFEs.

6. Conclusions

In this article, we have discussed the SC method extended with the sparse grid algorithm of Smolyak and its application to efficient exposure computation in financial risk management. We can drastically reduce the number of portfolio evaluations for multi-currency portfolios. The proposed algorithm is beneficial for large portfolios involving many risk factors. The model can be easily generalized to any portfolio and size. Our numerical experiments have shown that for a realistic portfolio consisting of linear and non-linear derivatives, the expected reduction in the portfolio evaluations may exceed 6000 times, depending on the number of risk factors and required accuracy. We have discussed the convergence aspects, and several realistic examples were given.

Appendix A. The SC algorithm for exposure computation

Algorithm 1 Construction and evaluation of the approximating portfolio.

procedure SC
  Calibrate the underlying SDEs, $\mathbf{X}(t) = [X_1(t), \ldots, X_d(t)]^{\mathrm{T}}$, as defined in~(3.11).
  Simulate multi-dimensional process $\mathbf{X}(t)$ using Monte Carlo.
  for T_k with $k \in \{1, \ldots, N_T\}$ **do**
    Given the simulated Monte Carlo paths determine, for each $X_i(T_k)$, the interpolation domain, as described in Section 4.1.
    Build the SC grid using the Smolyakʹ algorithm, see Section 2.1.
    Evaluate portfolio $V(T_k, V(\mathbf{X}(T_k))$ at the grid points $\{X\}_{i_1,\ldots,i_d}$ obtained in the previous step and obtain $\{V\}_{i_1,\ldots,i_d}$.
    Using Smolyakʹ interpolation build the approximating function $\widetilde{g}\big(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k)\big)$, as described in~(2.19).
    Evaluate function $\widetilde{g}(\{V\}_{i_1,\ldots,i_d}, \mathbf{X}(T_k))$ for all the Monte Carlo paths.
    Compute EEs, PFEs etc. as defined in~(2.6) and~(2.7).

Appendix B. Multi-D case: details regarding model configuration

$$\mathbf{C} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.65 & 0.7 & 0.75 & 0.8 \\ 0.5 & 1 & 0.45 & 0.35 & 0.5 & 0.5 & 0.6 \\ 0.5 & 0.45 & 1 & 0.5 & 0.5 & 0.5 & 0.7 \\ 0.65 & 0.35 & 0.5 & 1 & 0.5 & 0.5 & 0.5 \\ 0.7 & 0.5 & 0.5 & 0.5 & 1 & 0.5 & 0.58 \\ 0.75 & 0.5 & 0.5 & 0.5 & 0.5 & 1 & 0.55 \\ 0.8 & 0.6 & 0.7 & 0.5 & 0.58 & 0.55 & 1 \end{bmatrix}, \tag{B.1}$$

$P_1(0,t) = \exp(-0.01t)$, $P_2(0,t) = \exp(-0.015t)$, $P_3(0,t) = \exp(-0.02t)$, $\lambda_b = 0.003$, $\lambda_1 = 0.003$, $\lambda_2 = 0.002$, $\lambda_3 = 0.001$, $\eta_b = 0.01$, $\eta_1 = 0.01$, $\eta_2 = 0.02$, $\eta_3 = 0.003$, $y_1(t_0) = 1.2$, $y_2(t_0) = 0.86$, $y_3(t_0) = 4.59$. Considered portfolio consisted of about 30 randomly chosen interest rate swaps.

Appendix C. Pricing of swaptions under the Gaussian 2-factor model

Under the Gaussian 2 Factor model, the short rate process, $r(t)$, is defined as: $r(t) = x^r(t) + y^r(t) + \psi^r(t)$, where processes $x^r(t)$ and $y^r(t)$ are defined by the following system of SDEs:

$$\begin{aligned} \mathrm{d}x^r(t) &= -\lambda_1 x^r(t)\mathrm{d}t + \eta_1 \mathrm{d}W_1(t), \quad x^r(t_0) = 0, \\ \mathrm{d}y^r(t) &= -\lambda_2 y^r(t)\mathrm{d}t + \eta_2 \mathrm{d}W_2(t), \quad y^r(t_0) = 0, \end{aligned}$$

with $\mathrm{d}W_1(t)\mathrm{d}W_2(t) = \rho \mathrm{d}t$, and where $\psi^r(t)$ is known in a closed form Brigo and Mercurio [4], which is an explicit function of the zero-coupon bonds, $P_{Mrkt}(t_0,T)$, available in the market. Given simulated realizations $\{x^r(t_i), y^r(t_i)\}$ one can establish a yield curve, as a function of T, in terms of zero-coupon bonds,

$$P(t_i,T) = \exp\Big(A(t_i,T) - \boxed{x^r(t_i)}B_1(t_i,T) - \boxed{y^r(t_i)}B_2(t_i,T)\Big),$$

where

$$A(t_i,T) = \log\frac{P_{Mrkt}(t_0,T)}{P_{Mrkt}(t_0,t_i)} - \frac{1}{2}\big(V^2(0,T) + V^2(0,t_i)\big),$$

and $B_j(t_i,T) = \frac{1}{\lambda_j}(1 - \mathrm{e}^{-\lambda_j(T-t_i)})$, $B_{1,2}(t_i,T) = \frac{1}{\lambda_1+\lambda_2}\mathrm{e}^{-(T-t_i)(\lambda_1+\lambda_2)}$, with $j = \{1,2\}$, and where

$$\begin{aligned} V^2(t_i,T) = &\frac{\eta_1^2}{\lambda_1^2}\Big(T - t_i - B_1(t_i,T) - \frac{1}{2}\lambda_2 B_2^2(t_i,T)\Big) + \frac{\eta_2^2}{\lambda_2^2}\Big(T - t_i - B_2(t_i,T) - \frac{1}{2}\lambda_1 B_2^2(t_i,T)\Big) \\ &+ \frac{2\eta_1\eta_2\rho}{\lambda_1\lambda_2}(T - t_i - B_1(t_i,T) - B_2(t_i,T) + B_{1,2}(t_i,T)). \end{aligned}$$

Then the price of a European payer swaption with swaption expiry T on an interest rate swap, with notional N, fixed rate, K and a set of pay-dates $\mathcal{T} = \{T_{\alpha+1}, \ldots, T_\beta\}$:

$$V(t_0,\mathcal{T},N,K) = NP_M(t_0,T)\int_{\mathbb{R}}\tilde{\omega}(x)\Big(\Phi(-h_1(x) - \sum_{i=\alpha+1}^{\beta}\kappa_i(x)\mathrm{e}^{\psi_i(x)}\Phi(-h_2(x)))\Big)\mathrm{d}x, \tag{C.1}$$

with the weight function, $\tilde{\omega}(x)$, given by:

$$\tilde{\omega}(x) = \frac{1}{\tilde{\eta}_1\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\tilde{\mu}_1)^2}{2\tilde{\eta}_1^2}},$$

and where functions $h_1(x)$ and $h_2(x)$ are defined as:

$$h_1(x) = \frac{\bar{x} - \tilde{\mu}_2}{\tilde{\eta}_2\sqrt{1-\tilde{\rho}^2}} - \frac{\tilde{\rho}(x - \tilde{\mu}_1)}{\tilde{\eta}_1\sqrt{1-\tilde{\rho}^2}}, \quad h_2(x) = h_1(x) + B_2(T,T_i)\tilde{\eta}_2\sqrt{1-\tilde{\rho}^2}.$$

Constant $\bar{x}$ is the solution of the following equation,

$$\sum_{i=\alpha+1}^{\beta}\kappa_i(x)\mathrm{e}^{-\bar{x}B_2(T,T_i)} = 1,$$

with

$$\kappa_i(x) = c_i\mathrm{e}^{A(T,T_i) - xB_1(T,T_i)}, \quad c_i = K\tau_i, \quad \text{for} \quad \alpha < i < \beta, \quad \text{and} \quad c_\beta = 1 + K\tau_\beta.$$

Finally, $\psi(x)$, and $\tilde{\mu}_1$ $\tilde{\mu}_2$ are:

$$\psi_i(x) = -B_2(T,T_i)\left(\tilde{\mu}_2 - \frac{1}{2}(\tilde{\eta}_2^2(1-\tilde{\rho}^2))B_2(T,T_i) + \tilde{\rho}\tilde{\eta}_2\frac{1}{\tilde{\eta}_1}(x-\tilde{\mu}_1)\right),$$

$$\tilde{\mu}_1 = \frac{\eta_1^2}{2\lambda_1^2}(1-\mathrm{e}^{-2\lambda_1 T}) + \frac{\eta_1\eta_2\rho}{\lambda_2}B_{1,2}(0,T) - \left(\frac{\eta_1^2}{\lambda_1} + \frac{\eta_1\eta_2\rho}{\lambda_2}\right)B_1(0,T),$$

$$\tilde{\mu}_2 = \frac{\eta_2^2}{2\lambda_2^2}(1-\mathrm{e}^{-2\lambda_2 T}) + \frac{\eta_1\eta_2\rho}{\lambda_1}B_{1,2}(0,T) - \left(\frac{\eta_2^2}{\lambda_2} + \frac{\eta_1\eta_2\rho}{\lambda_1}\right)B_2(0,T),$$

with

$$\tilde{\eta}_j = \eta_j\sqrt{\frac{1-\mathrm{e}^{-2\lambda_j T}}{2\lambda_j}}, \quad \text{for} \quad j\in\{1,2\}, \quad \text{and} \quad \tilde{\rho} = \frac{\eta_1\eta_2\rho}{\tilde{\eta}_1\tilde{\eta}_2}B_{1,2}(0,T).$$

References

[1] L. Abbas-Turki, S. Crépey, B. Diallo, XVA principles, nested Monte Carlo strategies, and GPU optimizations, IJTAF 21 (6) (2018), doi:10.1142/S0219024918500309.
[2] K. Andersson, C. Oosterlee, A deep learning approach for computations of exposure profiles for high-dimensional Bermudan options, Available at arXiv:2003.01977 (2019).
[3] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Adv. Comput. Math. 12 (2000) 273–288, doi:10.1023/A:1018977404843.
[4] D. Brigo, F. Mercurio, Interest Rate Models- Theory and Practice: With Smile, Inflation and Credit, 2nd ed., Springer Finance, 2007.
[5] C. Clenshaw, A. Curtis, A method for numerical integration on an automatic computer, Numer. Math. 2 (197) (1960).
[6] S. Crépey, R. Hoskinson, B. Saadeddine, Balance sheet XVA by deep learning and GPU, Available https://math.maths.univ-evry.fr/crepey/papers (2019).
[7] D. Dufresne, Sums of Lognormals, Technical report, Centre for Actuarial Sciences, University of Melbourne, 2008.
[8] M. Gaß, K. Glau, M. Mirco, M. Maximilian, Chebyshev interpolation for parametric option pricing, Financ. Stoch. 22 (3) (2018) 701–731, doi:10.1007/s00780-018-0361-y.
[9] K. Glau, F. Kressner, F. Statti, Low-rank tensor approximation for Chebyshevinterpolation in parameteric option pricing, SIAM J. Financ. Math. 11 (3) (2020) 897–927.
[10] K. Glau, R. Pachon, C. Pötz, Speed-up credit exposure calculations for pricing and risk management, Quant. Financ. 21 (2020) 481–499.
[11] A. Green, XVA: Credit, Funding and Capital Valuation Adjustments, Wiley & Sons Ltd, 2015.
[12] J. Gregory, Counterparty Credit Risk: The New Challenge for Global Financial Markets, John Wiley & Sons, 2010.
[13] L. Grzelak, C. Oosterlee, On cross-currency models with stochastic volatility and correlated interest rates, Appl. Math. Financ. 19 (1) (2012) 1–35.
[14] L. Grzelak, C. Oosterlee, From arbitrage to arbitrage-free implied volatilities, J. Comput. Financ. 20 (3) (2016) 31–49.
[15] L. Grzelak, J. Witteveen, M. Suárez-Taboada, C. Oosterlee, The stochastic collocation Monte Carlo sampler: highly efficient sampling from "expensive" distributions, Quant. Financ. 19 (2) (2019) 339–356, doi:10.1080/14697688.2018.1459807.
[16] J. Hull, A. White, Numerical procedures for implementing term structure models II: two-factor models, J. Deriv. 2 (1994) 37–47.
[17] K. Judd, Numerical Methods in Economics, MIT Press, Cambridge, MA, 1998.
[18] K. Judd, L. Maliar, S. Maliar, R. Valero, Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain, J. Econ. Dyn. Control 44 (2014) 92–123. http://www.sciencedirect.com/science/article/pii/S0165188914000621.
[19] M. Laris, I. Ruiz, Chebyshev methods for ultra-efficient risk calculations, Available at SSRN 3165563 (2018).
[20] S. Liu, L. Grzelak, C. Oosterlee, The seven-league scheme: deep learning for large time step Monte Carlo simulations of stochastic differential equations, Risks 10 (3) (2022), doi:10.3390/risks10030047. https://www.mdpi.com/2227-9091/10/3/47.
[21] D. Loukrezis, U. Römer, T. Casper, S.Schöps, H.D. Gersem, High dimensional uncertainty quantification for an electrothermal field problem using stochastic collocation on sparse grids and tensor train decompositions, Int. J. Numer. Model. 31 (2) (2017) 1–8.
[22] Besel Committee on Banking Supervision, Annex 4 to "International convergence of capital measurement and capital standards - a revised framework" (2005).
[23] C. Oosterlee, L. Grzelak, Mathematical Modeling and Computation in Finance, World Scientific, 2019.
[24] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Dokl. Akad. Nauk SSSR 4 (1963) 240–243.
[25] M. Suárez-Taboada, J. Witteveen, L. Grzelak, C. Oosterlee, Uncertainty quantification and Heston model, in: P. Quintela, P. Barral, D. Gómez, F. Pena, J. Rodríguez, P. Salgado, M. Vázquez-Méndez (Eds.), Progress in Industrial Mathematics at ECMI 2016, Springer International Publishing, Cham, 2017, pp. 153–159.
[26] R. Tempone, S. Wolfers, Smolyak's algorithm: a powerful black box for the acceleration of scientific computations, in: J. Garcke, D. Pflüger, C. Webster, G. Zhang (Eds.), Sparse Grids and Applications—Miami 2016, 2018, pp. 201–228.
[27] L. Trefethen, Is Gauss quadrature better than Clenshaw–Curtis? SIAM Rev. 50 (67–87) (2008).
[28] D. Xiu, J. Hesthaven, High order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (2005) 1118–1139.
[29] M. Zeron, I. Ruiz, Tensoring Dynamic Sensitivities and Dynamic Initial Margin, Risk Magazine, 2021.