



On an efficient multiple time step Monte Carlo simulation of the SABR model

Álvaro Leitao, Lech A. Grzelak & Cornelis W. Oosterlee

To cite this article: Álvaro Leitao, Lech A. Grzelak & Cornelis W. Oosterlee (2017) On an efficient multiple time step Monte Carlo simulation of the SABR model, Quantitative Finance, 17:10, 1549-1565, DOI: [10.1080/14697688.2017.1301676](https://doi.org/10.1080/14697688.2017.1301676)

To link to this article: <http://dx.doi.org/10.1080/14697688.2017.1301676>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 10 Apr 2017.



Submit your article to this journal [↗](#)



Article views: 438



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)

On an efficient multiple time step Monte Carlo simulation of the SABR model

ÁLVARO LEITAO^{*†‡}, LECH A. GRZELAK^{†§} and CORNELIS W. OOSTERLEE^{†‡}

[†]TU Delft, Delft Institute of Applied Mathematics, Delft, The Netherlands

[‡]Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands

[§]ING, Quantitative Analytics, Amsterdam, The Netherlands

(Received 13 April 2016; accepted 24 February 2017; published online 10 April 2017)

In this paper, we will present a multiple time step Monte Carlo simulation technique for pricing options under the *Stochastic Alpha Beta Rho* model. The proposed method is an extension of the one time step Monte Carlo method that we proposed in an accompanying paper Leitao *et al.* [*Appl. Math. Comput.* 2017, **293**, 461–479], for pricing European options in the context of the model calibration. A highly efficient method results, with many very interesting and nontrivial components, like Fourier inversion for the sum of log-normals, stochastic collocation, Gumbel copula, correlation approximation, that are not yet seen in combination within a Monte Carlo simulation. The present multiple time step Monte Carlo method is especially useful for long-term options and for exotic options.

Keywords: SABR model; Exact simulation; Monte Carlo methods; Copulas; Stochastic collocation; Fourier techniques; Exotic options

JEL Classification: C15, C63

1. Introduction

The Stochastic Alpha Beta Rho (SABR) model (Hagan *et al.* 2002) is an established stochastic differential equation (SDE) model which, in practice, is often used for interest rates and foreign-exchange (FX) modelling. It is based on a parametric local volatility component in terms of a model parameter, β , and reads

$$\begin{aligned} dS(t) &= \sigma(t)S^\beta(t)dW_S(t), & S(0) &= S_0 \exp(rT), \\ d\sigma(t) &= \alpha\sigma(t)dW_\sigma(t), & \sigma(0) &= \sigma_0. \end{aligned} \quad (1)$$

Here, $S(t) = \bar{S}(t) \exp(r(T-t))$ denotes the forward price of the underlying asset $\bar{S}(t)$, with r an interest rate, S_0 the spot price and T the maturity. Further, $\sigma(t)$ represents a stochastic volatility process, with $\sigma(0) = \sigma_0$, $W_S(t)$ and $W_\sigma(t)$ are correlated Brownian motions with constant correlation coefficient ρ (i.e. $W_S W_\sigma = \rho t$). The parameters of the SABR model are $\alpha > 0$ (the volatility of volatility, *vol-vol*), $0 \leq \beta \leq 1$ (the variance elasticity) and ρ (the correlation coefficient). Because of a practically very useful closed-form expression for the implied volatility, Hagan *et al.* (2002) and Obloj (2008), the model has gained its popularity. However, since this expression is derived by perturbation theory, the formula is not always accurate for small strike values or for long time to maturity. By the one time step SABR method in Leitao *et al.* (2017), we

can efficiently calculate accurate implied volatilities for any strike, however, only for short times to maturity (less than two years). This fits well to the context of FX markets.

Here, we extend the one time step Monte Carlo method from Leitao *et al.* (2017), and propose a *multiple time step Monte Carlo simulation technique* for pricing options under the SABR dynamics. We call it the *mSABR method*. With this new simulation approach, we can price options with longer maturities, payoffs relying on the transitional distributions and exotic options (like path-dependent options) under the SABR dynamics, overcoming the limitations of Hagan's formula in the mentioned situations.

Because a robust and efficient SABR simulation scheme may be involved and quite expensive, we aim for using as few time steps as possible. This fact, together with the long-time horizon assumption, implies that we focus on larger time steps. In this context, the important research line is called *exact simulation*, where, rather than Taylor-based simulation techniques, the probability density of the SDE under consideration is highly accurately approximated. Point-of-departure is Broadie and Kaya's exact simulation for the Heston stochastic volatility model (Broadie and Kaya, 2006). That method is, however, time-consuming because of multiple evaluations of Bessel functions and the use of Fourier inversion techniques on each Monte Carlo path. Inspired by this approach, in Smith (2007) and van Hasstrecht and Pelsser (2010), computational

*Corresponding author. Email: A.LeitaoRodriguez@tudelft.nl

improvements were proposed. Furthermore, Andersen (2008) presented two efficient alternatives to the Broadie and Kaya scheme, the so-called Truncated Gaussian and the Quadratic Exponential (QE) schemes.

For the SABR model, exact Monte Carlo simulation is non-trivial because the forward process in (1) is governed by *constant elasticity of variance* (CEV) dynamics (Cox 1996, Schroder 1989). In Islah (2009), the connection between the CEV and a squared Bessel processes was explained, and an analytic approximation for the SABR conditional distribution based on the non-central χ^2 distribution was presented. The expression relies on the stochastic volatility dynamics, but also on the *integrated variance* process (which itself also depends on the volatility).

An (approximately) exact SABR simulation can then be subdivided in to the simulation of the volatility process, the simulation of the integrated variance (conditional on the volatility process) and the simulation of the underlying CEV process (conditional on the volatility and the integrated variance processes). Based on this, Chen *et al.* (2012) proposed a low-biased Monte Carlo scheme with moment-matching (following the techniques of the QE scheme by Andersen (2008)) and a direct inversion approach. For the simulation of the integrated variance, they also employed moment-matching based on conditional moments that were approximated by a small disturbance expansion. In Cai *et al.* (forthcoming), the authors employed direct inversion for the forward distribution and used a Laplace transform inversion technique to simulate the integrated variance. Kennedy and Pham (2014) provided the moments of the integrated variance for specific SABR models (like the normal, log-normal and displaced diffusion SABR models) and derived an approximation of the SABR distribution. In Lord and Farebrother (2014), a summary of different approaches was presented.

In this work, we approximate the distribution of the integrated variance conditional on the volatility dynamics by joining the two involved marginal distributions, i.e. the stochastic volatility and integrated variance distributions, by means of copula techniques. With both marginal distributions available, we use the Archimedean Gumbel copula to define a multivariate distribution which approximates the conditional distribution of the integrated variance given the stochastic volatility.

An approximation of the cumulative distribution function (CDF) of the integrated variance can be obtained by a recursive procedure, as described in Zhang and Oosterlee (2013), originally employed to price arithmetic Asian options. This iterative technique is based on the derivation of the characteristic function of the integrated variance process and Fourier inversion to recover the probability density function (PDF).

The fact that we need to apply recursion and compute the corresponding characteristic function, PDF and CDF of the integrated variance for each Monte Carlo sample at each time step, makes this approach however computationally very expensive (even with the improvements already made in Leitao *et al.* (2017)), like the Heston exact simulation. In order to reduce the use of this procedure as much as possible, highly efficient sampling from the CDF of the integrated variance is proposed, which is based on the so-called *Stochastic Collocation Monte Carlo sampler* (SCMC), see Grzelak *et al.* (2015). The technique employs a sophisticated interpolation (based

on Lagrange polynomials and collocation points) of the CDF under consideration.

As will be seen, the resulting ‘almost exact’ Monte Carlo SABR simulation method that we present here (the mSABR method), contains several interesting (and not commonly used) components, like the Gumbel copula, a recursion plus Fourier inversion to approximate the CDF of the integrated variance and efficient interpolation by means of the SCMC sampler. The proposed mSABR scheme allows for fast and accurate option pricing under SABR dynamics, providing a better ratio of accuracy and computational cost than Taylor-based simulation schemes. Compared to the other approaches mentioned in this introduction, we provide a highly accurate SABR simulation scheme, based on only a few time steps.

The paper is organized as follows. In section 2, SABR model simulation is introduced. In section 3, the different parts of the multiple time step copula-based technique are described, including the derivation of the marginal distributions and the application of the copula. Some numerical experiments are presented in section 4. We conclude in section 5. In the derivation of the CDF of the integrated variance, we need to perform several approximations, which comes with approximation errors. In appendix 1, we list and discuss these errors.

2. ‘Almost exact’ SABR simulation

The SABR model with dependent Brownian motions was already given in (1). The multiple time step Monte Carlo SABR simulation is based on the corresponding SDE system with independent Brownian motions $d\hat{W}_S(t)$ and $d\hat{W}_\sigma(t)$, i.e.

$$\begin{aligned} dS(t) &= \sigma(t)S^\beta(t) \left(\rho d\hat{W}_\sigma(t) + \sqrt{1 - \rho^2} d\hat{W}_S(t) \right), \\ S(0) &= S_0 \exp(rT), \\ d\sigma(t) &= \alpha \sigma(t) d\hat{W}_\sigma(t), \\ \sigma(0) &= \sigma_0. \end{aligned} \quad (2)$$

The forward dynamics are governed by a CEV process. Based on this fact and the work by Schroder (1989) and Islah (2009), an analytic approximation for the CDF of the SABR conditional process has been obtained. For some generic time interval $[s, t]$, $0 \leq s < t \leq T$, assuming $S(s) > 0$, the conditional cumulative distribution for forward $S(t)$ with an absorbing boundary at $S(t) = 0$, given $\sigma(s)$, $\sigma(t)$ and $\int_s^t \sigma^2(z) dz$, is given by

$$\begin{aligned} \Pr \left(S(t) \leq K | S(s) > 0, \sigma(s), \sigma(t), \int_s^t \sigma^2(z) dz \right) \\ = 1 - \chi^2(a; b, c), \end{aligned} \quad (3)$$

where

$$\begin{aligned} a &= \frac{1}{v(t)} \left(\frac{S(s)^{1-\beta}}{(1-\beta)} + \frac{\rho}{\alpha} (\sigma(t) - \sigma(s)) \right)^2, \\ c &= \frac{K^{2(1-\beta)}}{(1-\beta)^2 v(t)}, \\ b &= 2 - \frac{1 - 2\beta - \rho^2(1-\beta)}{(1-\beta)(1-\rho^2)}, \\ v(t) &= (1 - \rho^2) \int_s^t \sigma^2(z) dz, \end{aligned}$$

and $\chi^2(x; \delta, \lambda)$ is the non-central chi-square CDF.

This formula is exact in the case of $\rho = 0$ and constitutes an *approximation* otherwise (because the CEV process is approximated by a shifted process with an approximated initial condition for $\rho \neq 0$ in the derivation). Based on equation (3), an *approximately* exact simulation of SABR model is feasible by inverting the conditional SABR cumulative distribution when the conditional integrated variance is known, see section 2.1.

2.1. SABR Monte Carlo simulation

In order to apply an ‘almost exact’ multiple time step Monte Carlo method for the SABR model, several steps need to be performed, that are described in the following:

- *Simulation of the SABR volatility process*, $\sigma(t)$ given $\sigma(s)$. By equation (2), the stochastic volatility process of the SABR model exhibits a log-normal distribution. The solution is a *geometric Brownian motion*, i.e. the exact simulation of $\sigma(t)|\sigma(s)$ reads
$$\sigma(t) \sim \sigma(s) \exp\left(\alpha \hat{W}_\sigma(t) - \frac{1}{2}\alpha^2 t\right). \quad (4)$$
- *Simulation of the SABR integrated variance process*, $\int_s^t \sigma^2(z)dz|\sigma(s), \sigma(t)$. This conditional distribution is not available in closed form. We will therefore derive an approximation of the conditional distribution of the SABR integrated variance given $\sigma(t)$ and $\sigma(s)$. The integrated variance sampling can be done by simply inverting it.
- *Simulation of the SABR forward price process*. The forward price $S(t)$ can be simulated by inverting the CDF in equation (3). By this, we avoid negative forward prices in the simulation, as an absorbing boundary at zero is considered. There is no analytic expression for the inverse distribution and therefore this inversion has to be computed by means of some numerical approximation.

The multiple time step Monte Carlo simulation for the SABR model is thus summarized in algorithm 1. Input parameters are the initial conditions (S_0 and σ_0), the maturity time, T , the SABR parameters α , β and ρ , the number of Monte Carlo samples (n) and the number of time steps (m).

Algorithm 1: Multiple time step SABR Monte Carlo simulation.

Data: $S_0, \sigma_0, T, \alpha, \beta, \rho, n, m$.

$S(0) = S_0 \exp(rT), \sigma(0) = \sigma_0$

for Path $p = 1 \dots n$ **do**

for Step $k = 1 \dots m$ **do**

$s = (k-1)\frac{T}{m}$

$t = k\frac{T}{m}$

 Sampling $\sigma(t)$ given $\sigma(s)$.

 Sampling $\int_s^t \sigma^2(z)dz$ given $\sigma(s)$ and $\sigma(t)$.

 Sampling $S(t)$ given $S(s), \sigma(s), \sigma(t)$ and $\int_s^t \sigma^2(z)dz$.

In section 3.3, we propose a procedure to sample $\int_s^t \sigma^2(z)dz|\sigma(s), \sigma(t)$ based on the Gumbel copula. For this, the CDF of the integrated variance given the initial volatility, $\sigma(s)$, (as a marginal distribution) must be derived. In section 3.1, a recursive technique to obtain an approximation of this CDF is

presented. Because we need to apply this recursion to approximate the characteristic function, the PDF and the CDF of $\int_s^t \sigma^2(z)dz|\sigma(s)$ for each sample of $\sigma(s)$ at each time step, this approach is expensive in terms of computational cost. To overcome this drawback, an efficient alternative will be employed here, based on Lagrange interpolation, as in the SCMC (Grzelak *et al.*, 2015). In section 2.2, this methodology is briefly described.

2.2. Stochastic collocation Monte Carlo sampler

In a multiple time step exact simulation Monte Carlo method we need to perform a significant number of computations each time step on each Monte Carlo path. This often hampers the applicability of the exact simulation for systems of SDEs. One of the important components of the multiple time step Monte Carlo SABR simulation is therefore an accurate and highly efficient interpolation, as proposed in the SCMC sampler in Grzelak *et al.* (2015).

In this section, we will introduce the SCMC technique. The details in the SABR conditional integrated variance context will be presented in section 3.2.

The SCMC technique relies on the property that a CDF of a distribution (if it exists) is uniformly distributed. A well-known standard approach to sample from a given distribution, Y , with CDF F_Y reads

$$F_Y(Y) \stackrel{d}{=} U \quad \text{thus} \quad y_n = F_Y^{-1}(u_n),$$

where $\stackrel{d}{=}$ means equality in distributional sense, $U \sim \mathcal{U}([0, 1])$ and u_n are samples from $\mathcal{U}([0, 1])$. The computational cost of this approach highly depends on the cost of the inversion F_Y^{-1} , which is assumed to be expensive.

We therefore consider another, ‘cheap’, random variable X , whose inversion, F_X^{-1} , is computationally much less expensive. In this framework, the following relation holds

$$F_Y(Y) \stackrel{d}{=} U \stackrel{d}{=} F_X(X).$$

The samples y_n of Y , and x_n of X , are thus related via the following inversion,

$$y_n = F_Y^{-1}(F_X(x_n)).$$

However, this does not yet imply any improvement since the number of expensive inversions F_Y^{-1} remains the same. The goal is to compute y_n using a function $g(\cdot) = F_Y^{-1}(F_X(\cdot))$, such that

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

where evaluations of function $g(\cdot)$ do not require many inversions $F_Y^{-1}(\cdot)$.

In Grzelak *et al.* (2015), function $g(\cdot)$ is approximated by means of Lagrange interpolation, which is a well-known interpolation also used in the Uncertainty Quantification (UQ) context. The result is a polynomial, $g_{N_Y}(\cdot)$, which approximates function $g(\cdot) = F_Y^{-1}(F_X(\cdot))$, and the samples y_n can be obtained by employing $g_{N_Y}(\cdot)$ as

$$y_n \approx g_{N_Y}(x_n) = \sum_{i=1}^{N_Y} y_i \ell_i(x_n), \quad \ell_i(x_n) = \prod_{j=1, j \neq i}^{N_Y} \frac{x_n - \bar{x}_j}{\bar{x}_i - \bar{x}_j},$$

where x_n is a vector of samples from X and \bar{x}_i and \bar{x}_j are so-called *collocation points*. N_Y represents the number of collocation points and y_i the exact inversion value of F_Y at the collocation point \bar{x}_i , i.e. $y_i = F_Y^{-1}(F_X(\bar{x}_i))$. By applying this interpolation, the number of inversions is reduced and, with only N_Y expensive inversions $F_Y^{-1}(F_X(\bar{x}_i))$, we can generate any number of samples by evaluating the polynomial $g_{N_Y}(x_n)$. This constitutes the 1D version of SCMC sampler in [Grzelak et al. \(2015\)](#).

According to the definition of the SCMC technique, a crucial aspect for the computational cost is parameter N_Y : the fewer collocation points are required, the more efficient will be the sampling procedure. The collocation points must be *optimally* chosen in a way to minimize their number. For any random variable X , the optimal collocation points will be based on the moments of X . The optimal collocation points are here chosen to be Gauss quadrature points that are defined as the zeros of the related orthogonal polynomial. This approach leads to a stable interpolation under the probability distribution of X . The complete procedure to compute the collocation points is described in [Grzelak et al. \(2015\)](#).

In section 3.2, the application of SCMC technique to generate samples from the CDF of $\int_s^t \sigma^2(z) dz | \sigma(s)$ is presented. Since we deal with a conditional distribution, the 2D version of SCMC needs to be used.

3. Components of the mSABR method

In this section, we will discuss the different components of the mSABR method. For simplicity, hereafter, we denote the SABR's integrated variance process by $Y(s, t) := \int_s^t \sigma^2(z) dz$. We will explain how to efficiently sample the integrated variance given the initial and the final volatility, i.e. $Y(s, t) | \sigma(t)$, $\sigma(s)$, as well as its use in a complete SABR simulation. Since the distribution is not available in closed form, some approximations need to be made. In section 3.3, we propose an accurate sampling method based on copula theory ([Sklar, 1959](#)), which is employed to approximate the required conditional distributions. The copula relies on the availability of the *marginal* distributions to simulate the *joint* distribution. As the marginal distributions, $Y(s, t) | \sigma(s)$ and $\sigma(t) | \sigma(s)$ appear as the natural choices. In section 3.1, a procedure to recover the CDF of the integrated variance process given the initial volatility is presented.

3.1. CDF of $\int_s^t \sigma^2(z) dz | \sigma(s)$ using the COS method

We present a technique to approximate the CDF of $Y(s, t) | \sigma(s)$, i.e. $F_{Y | \sigma(s)}$. We will work in the log-space, so an approximated CDF of $\log Y(s, t) | \log \sigma(s)$, $F_{\log Y | \log \sigma(s)}$, will be derived. We have employed this technique in the one time step SABR method ([Leitao et al. 2017](#)), but the multiple time step version is more involved. We approximate $Y(s, t)$ by its discrete equivalent, i.e.

$$Y(s, t) := \int_s^t \sigma^2(z) dz \approx \sum_{j=1}^M \Delta t \sigma^2(t_j) =: \hat{Y}(s, t) \quad (5)$$

where M is the number of intermediate or discrete time points,[†] $\Delta t = \frac{t-s}{M}$ and $t_j = s + j \Delta t$, $j = 1, \dots, M$. $\hat{Y}(T)$ is subsequently transformed to the logarithmic domain, with

$$F_{\log \hat{Y} | \log \sigma(s)}(x) = \int_{-\infty}^x f_{\log \hat{Y} | \log \sigma(s)}(y) dy, \quad (6)$$

and $f_{\log \hat{Y} | \log \sigma(s)}$ the PDF of $\log \hat{Y}(s, t) | \log \sigma(s)$.

Density $f_{\log \hat{Y} | \log \sigma(s)}$ is, in turn, found by approximating the associated characteristic function, $\phi_{\log \hat{Y} | \log \sigma(s)}$, and applying a Fourier inversion procedure. The characteristic function and the desired PDF of $\log \hat{Y}(T) | \log \sigma(s)$ form a *Fourier pair*. Based on the work in [Benhamou \(2002\)](#) and [Zhang and Oosterlee \(2013\)](#), we can define a recursive procedure to recover the characteristic function of $f_{\log \hat{Y} | \log \sigma(s)}$.

3.1.1. Recursive procedure to recover $\phi_{\log \hat{Y} | \log \sigma(s)}$. We start by defining the sequence,

$$R_j = \log \left(\frac{\sigma^2(t_j)}{\sigma^2(t_{j-1})} \right), \quad j = 1, \dots, M, \quad (7)$$

where R_j is the logarithmic increment of $\sigma^2(t)$ between t_j and t_{j-1} , $j = 1, \dots, M$. As the volatility process follows log-normal dynamics, and increments of Brownian motion are independent and identically distributed, the R_j are also independent and identically distributed, i.e. $R_j \stackrel{d}{=} R$. In addition, the characteristic function of R_j is well known and reads, $\forall u, j$,

$$\phi_{R_j}(u) = \phi_R(u) = \exp(-i u \alpha^2 \Delta t - 2 u^2 \alpha^2 \Delta t), \quad (8)$$

with α as in (1) and $i = \sqrt{-1}$ the imaginary unit. By the definition of R_j in equation (7), we write $\sigma^2(t_j)$ as

$$\sigma^2(t_j) = \sigma^2(t_0) \exp(R_1 + R_2 + \dots + R_j). \quad (9)$$

At this point, a backward recursion procedure in terms of R_j will be defined by which we can recover $\phi_{\log \hat{Y} | \log \sigma(s)}(u)$. We define

$$Y_1 = R_M, \quad Y_j = R_{M+1-j} + Z_{j-1}, \quad j = 2, \dots, M. \quad (10)$$

with $Z_j = \log(1 + \exp(Y_j))$.

By equations (9) and (10), the discrete integrated variance can be expressed as follows:

$$\hat{Y}(s, t) = \sum_{i=1}^M \sigma^2(t_i) \Delta t = \Delta t \sigma^2(s) \exp(Y_M). \quad (11)$$

From equation (11) and by applying the definition of characteristic function, we determine $\phi_{\log \hat{Y} | \log \sigma(s)}$, as follows:

[†]So, we have the m large time steps, for the multiple time step Monte Carlo simulation, and we have M intermediate dates for the integrated variance, $M \gg m$. We suggest to prescribe Δt , and use $M = \frac{t-s}{\Delta t}$, in order to avoid over-discretization issues. This makes the value of M dependent on the interval size, $t - s$.

$$\begin{aligned}
\phi_{\log \hat{Y} | \log \sigma(s)}(u) &= \mathbb{E}[\exp(iu \log \hat{Y}(s, t)) | \log \sigma(s)] \\
&= \exp\left(iu \log(\Delta t \sigma^2(s))\right) \mathbb{E}[\exp(iu Y_M) | \log \sigma(s)] \\
&= \exp\left(iu \log(\Delta t \sigma^2(s))\right) \phi_{Y_M}(u). \tag{12}
\end{aligned}$$

We have reduced the computation of $\phi_{\log \hat{Y} | \log \sigma(s)}$ to the computation of ϕ_{Y_M} . As Y_M is defined recursively, its characteristic function can be obtained by a recursion as well. According to the definition of the (backward) sequence Y_j in equation (10), the initial and recursive characteristic functions are given by the following expressions,

$$\begin{aligned}
\phi_{Y_1}(u) &= \phi_{R_M}(u) = \phi_R(u) = \exp(-iu\alpha^2 \Delta t - 2u^2\alpha^2 \Delta t), \\
\phi_{Y_j}(u) &= \phi_{R_{M+1-j}}(u) \phi_{Z_{j-1}}(u) = \phi_R(u) \phi_{Z_{j-1}}(u), \tag{13}
\end{aligned}$$

where equation (8) is used in both expressions and the independence of R_{M+1-j} and Z_{j-1} is also employed. By definition, the characteristic function of Z_{j-1} reads

$$\phi_{Z_{j-1}}(u) := \int_{-\infty}^{\infty} (\exp(x) + 1)^{iu} f_{Y_{j-1}}(x) dx.$$

PDF $f_{Y_{j-1}}$ is not known. To approximate it, the Fourier cosine series expansion on $f_{Y_{j-1}}$ is applied. We first truncate the integration range to $[a, b]$. The calculation of integration boundaries a and b follows the cumulant-based approach as described in [Zhang and Oosterlee \(2013\)](#) and [Fang and Oosterlee \(2008\)](#). After truncation of the integral and by applying a cosine series expansion to $f_{Y_{j-1}}$, we have

$$\begin{aligned}
\phi_{Z_{j-1}}(u) &\approx \frac{2}{b-a} \sum_{l=0}^{N-1} B_l \int_a^b (\exp(x) + 1)^{iu} \\
&\quad \times \cos\left((x-a) \frac{l\pi}{b-a}\right) dx =: \hat{\phi}_{Z_{j-1}}(u), \\
B_l &= \Re\left(\hat{\phi}_{Y_{j-1}}\left(\frac{l\pi}{b-a}\right) \exp\left(-ia \frac{l\pi}{b-a}\right)\right), \tag{14}
\end{aligned}$$

with N the number of cosine expansion elements, and where

$$\begin{aligned}
\hat{\phi}_{Y_1}(u) &:= \phi_R(u), \\
\hat{\phi}_{Y_j}(u) &:= \phi_R(u) \hat{\phi}_{Z_{j-1}}(u).
\end{aligned}$$

We wish to compute $\hat{\phi}_{Y_M}(\frac{k\pi}{b-a})$, for $k = 0, \dots, N-1$. Considering equation (14), we rewrite $\hat{\phi}_{Y_j}(u) = \phi_R(u) \hat{\phi}_{Z_{j-1}}(u)$ in matrix-vector form, as follows:

$$\Phi_{j-1} = \mathcal{M} A_{j-1}, \tag{15}$$

where

$$\begin{aligned}
\Phi_{j-1} &= [\Phi_{j-1}(k)]_{k=0}^{N-1}, \\
\Phi_{j-1}(k) &= \hat{\phi}_{Z_{j-1}}(u_k), \\
\mathcal{M} &= [\mathcal{M}(k, l)]_{k=0}^{N-1}, \\
\mathcal{M}(k, l) &= \int_a^b (\exp(x) + 1)^{iu_k} \cos((x-a)u_l) dx, \\
A_j &= \frac{2}{b-a} [A_j(l)]_{l=0}^{N-1}, \\
A_j(l) &= \Re\left(\hat{\phi}_{Y_{j-1}}(u_l) \exp(-iau_l)\right), \tag{16}
\end{aligned}$$

with

$$u_z = \frac{z\pi}{b-a}. \tag{17}$$

By the recursion procedure in equation (15), we obtain the approximation $\hat{\phi}_{Y_M}$ of the characteristic function ϕ_{Y_M} of Y_M .

3.1.2. Numerical approximation of integral. For the approximation in equation (15), we must compute matrix \mathcal{M} in equation (16) highly efficiently. The fast computation of the integral occurring for each element of \mathcal{M} is a key aspect for the performance of the mSABR method. The number of elements in matrix \mathcal{M} can be large, as it corresponds to the number of elements in the cosine series expansion, i.e. N^2 . The efficiency also depends on the required accuracy. In this work, we take advantage of the performed analysis in [Leitao et al. \(2017\)](#). Among several options, the authors have shown that a numerical approximation based on a *piecewise linear approximation* provides a good balance between performance and accuracy. Following this approximation, we rewrite the considered integral as follows:

$$\begin{aligned}
I_{\mathcal{M}} &:= \int_a^b \exp(iu_k h(x)) \cos((x-a)u_l) dx, \\
h(x) &= \log(\exp(x) + 1), \tag{18}
\end{aligned}$$

with u_k and u_l as in (17). Although function $h(x)$ is not linear, it is smooth and monotonic (see figure 1). Hence, the proposed numerical technique is to define sub-intervals of the integration range $[a, b]$ in which $h(x)$ is almost constant or linear, i.e.

$$I_{\mathcal{M}} = \sum_{j=1}^L \int_{a_j}^{b_j} \exp(iu_k h(x)) \cos((x-a)u_l) dx,$$

so that we can perform a first-order expansion in each sub-interval $[a_j, b_j]$, $j = 1, \dots, L$. In order to guarantee continuity of the approximation, $h(x)$ is approximated by a linear function in each interval as follows:

$$h(x) \approx c_{1,j} + c_{2,j}x, \quad x \in [a_j, b_j].$$

This gives us an approximation $\hat{I}_{\mathcal{M}}$, as follows

$$\hat{I}_{\mathcal{M}} = \sum_{j=1}^L \exp(iu_k c_{1,j}) \underbrace{\int_{a_j}^{b_j} \exp(iu_k c_{2,j}x) \cos((x-a)u_l) dx}_{I_j}. \tag{19}$$

In each sub-interval $[a_j, b_j]$, we need to determine a simple integral, I_j , which can be computed analytically.

The optimal integration points a_j and b_j in equation (19) are found by differentiation of $h(x)$ w.r.t. x , i.e.

$$H(x) := \frac{\partial h(x)}{\partial x} = \frac{\exp(x)}{\exp(x) + 1} = \frac{1}{1 + \exp(-x)},$$

giving a relation between the derivative of $h(x)$ and the logistic distribution. This representation indicates that $H(x)$ is the CDF of the logistic distribution, with a so-called *location parameter* $\mu = 0$ and a *scale parameter* $\eta = 1$. In order to determine the optimal integration points so that their positions correspond to the logistic distribution, we need to compute the quantiles. The quantiles of the logistic distribution are analytically available and given by

$$q(p) = \log\left(\frac{p}{1-p}\right).$$

The algorithm for the optimal integration points, given integration interval $[a, b]$ and the number of intervals L , is then as follows:

- Determine an interval in the probability space: $P_h = [H(a), H(b)]$.
- Divide the interval P_h equally into L parts: $P_h^j = [p_j, p_{j+1}]$, $j = 0 \dots L-1$.
- Determine a_j and b_j by calculating the quantiles, $a_j = q(p_j)$ and $b_j = q(p_{j+1})$, $j = 0 \dots L-1$.

The algorithm above ensures that integration points are optimally distributed (see figure 1(a)). However, for the particular problem, the integration interval can be rather large. As the integration points are typically concentrated near the function's curved part, the errors in the outer sub-intervals dominate the overall error and more points at the beginning and at the end of the interval are required to reduce this error. A more evenly distribution of the integration points can be achieved by introducing a scale parameter, $\eta \neq 1$, which implies that the integration points correspond to a CDF with increased variance. In figure 1(b), the case of $\eta = 3$ is presented. When $\eta \neq 1$, the expression for the quantiles reads,

$$q(p) = \eta \log \left(\frac{p}{1-p} \right).$$

Through the experiments in [Leitao et al. \(2017\)](#), we concluded that the choice $\eta = 3$ provided accurate results in this context.

3.1.3. Recovering $f_{\log \hat{Y} | \log \sigma(s)}$ by COS method. Once the approximation of ϕ_{Y_M} , $\hat{\phi}_{Y_M}$, has been efficiently derived, we can recover $f_{\log \hat{Y} | \log \sigma(s)}$ from $\phi_{\log \hat{Y} | \log \sigma(s)}$ by employing the COS method ([Fang and Oosterlee 2008](#)), as follows:

$$f_{\log \hat{Y} | \log \sigma(s)}(x) \approx \frac{2}{\hat{b} - \hat{a}} \sum_{k=0}^{N-1'} C_k \cos \left((x - \hat{a}) \frac{k\pi}{\hat{b} - \hat{a}} \right), \quad (20)$$

with

$$C_k = \Re \left(\phi_{\log \hat{Y} | \log \sigma(s)} \left(\frac{k\pi}{\hat{b} - \hat{a}} \right) \exp \left(-i \frac{\hat{a}k\pi}{\hat{b} - \hat{a}} \right) \right),$$

and

$$\begin{aligned} \phi_{\log \hat{Y} | \log \sigma(s)} \left(\frac{k\pi}{\hat{b} - \hat{a}} \right) &= \exp \left(i \frac{k\pi}{\hat{b} - \hat{a}} \log \left(\Delta t \sigma^2(s) \right) \right) \phi_{Y_M} \left(\frac{k\pi}{\hat{b} - \hat{a}} \right) \\ &\approx \exp \left(i \frac{k\pi}{\hat{b} - \hat{a}} \log \left(\Delta t \sigma^2(s) \right) \right) \hat{\phi}_{Y_M} \left(\frac{k\pi}{\hat{b} - \hat{a}} \right), \end{aligned}$$

where N is the number of COS terms, $[\hat{a}, \hat{b}]$ is the support[†] of $\log \hat{Y}(s, t) | \log \sigma(s)$ and the prime ' and \Re symbols in equation (20) mean division of the first term in the summation by two and taking the real part of the complex-valued expressions in the brackets, respectively. CDF $F_{\log \hat{Y} | \log \sigma(s)}$ can be obtained by integration, plugging the approximated $f_{\log \hat{Y} | \log \sigma(s)}$ from equation (20) into equation (6).

The characteristic function $\phi_{\log \hat{Y} | \log \sigma(s)}$ and PDF $f_{\log \hat{Y} | \log \sigma(s)}$ need to be derived for each sample of $\log \sigma(s)$

and this makes the computation of $F_{\log \hat{Y} | \log \sigma(s)}$ (and also its subsequent inversion) very expensive. In order to overcome this issue, the SMC technique (see section 2.2) will be employed approximating these rather expensive computations by means of an accurate interpolation.

3.2. Efficient sampling of $\log \hat{Y} | \log \sigma(s)$

By employing the SMC technique, instead of directly computing $F_{\log \hat{Y} | \log \sigma(s)}$ for each sample of $\log \sigma(s)$, we only have to compute $F_{\log \hat{Y} | \log \sigma(s)}$ at the collocation points. In general, only a few collocation points are sufficient to obtain accurate approximations, which is a well-known fact from the UQ research field. This fact allows us to drastically reduce the computational cost of sampling the required distribution.

For the problem at hand, we require samples from the integrated variance conditional on the initial volatility, $\log \hat{Y}(s, t) | \log \sigma(s)$. Therefore, we need to make use of the *2D version of the SMC technique*. Two levels of collocation points need to be chosen, one for each dimension. If we denote them by $N_{\hat{Y}}$ and N_{σ} , respectively, the resulting number of inversions equals $N_{\hat{Y}} \cdot N_{\sigma}$. The formal definition of the 2D SMC technique applied to our context reads

$$\begin{aligned} y_n | v_n &\approx g_{N_{\hat{Y}}, N_{\sigma}}(x_n) \\ &= \sum_{i=1}^{N_{\hat{Y}}} \sum_{j=1}^{N_{\sigma}} F_{\log \hat{Y} | \log \sigma(s)=\bar{v}_j}^{-1} (F_X(\bar{x}_i)) \ell_i(x_n) \ell_j(v_n), \quad (21) \end{aligned}$$

where x_n are the samples from $X \sim \mathcal{N}(0, 1)$, which is used as the cheap variable, and v_n the samples of $\log \sigma(s)$; \bar{x}_i and \bar{v}_j are the collocation points for approximating variables $\log Y$ and $\log \sigma(s)$, respectively. The Lagrange polynomials ℓ_i and ℓ_j are defined by

$$\ell_i(x_n) = \prod_{k=1, k \neq i}^{N_{\hat{Y}}} \frac{x_n - \bar{x}_k}{\bar{x}_i - \bar{x}_k}, \quad \ell_j(v_n) = \prod_{k=1, k \neq j}^{N_{\sigma}} \frac{v_n - \bar{v}_k}{\bar{v}_j - \bar{v}_k}.$$

The collocation points \bar{x}_i and \bar{v}_j are calculated based on the moments of the random variables. The two involved variables in the application of the 2D SMC technique are the collocation variables, $X \sim \mathcal{N}(0, 1)$, and $\log \sigma(s) \sim \mathcal{N}(\log \sigma_0 + \frac{1}{2}\alpha^2 s, \alpha\sqrt{s})$. The moments of a normal variable, $X \sim \mathcal{N}(\mu_X, s_X)$ are analytically available, and are given by

$$\mathbb{E}[X^p] = \begin{cases} 0 & \text{if } p \text{ is odd,} \\ s_X^{\frac{p}{2}}(p-1)!! & \text{if } p \text{ is even,} \end{cases}$$

where p is the moment order and the expression $!!$ represents the *double factorial*. In order to compute the optimal collocation points, \bar{v}_j , the precalculated collocation points by means of the $\log \sigma(s)$ moments need to be shifted according to the mean, i.e. $\log \sigma_0 + \frac{1}{2}\alpha^2 s$.

We first test numerically the accuracy of the SMC technique for our problem. In figure 2, two experiments are presented. In figure 2(a), we compare the samples obtained by direct inversion and by the SMC technique. The fit is highly accurate, already with only seven collocation points. In figure 2(b), the empirical CDFs are depicted, where the excellent match is confirmed.

[†]It can be calculated given the support of Y_M , $[a, b]$.

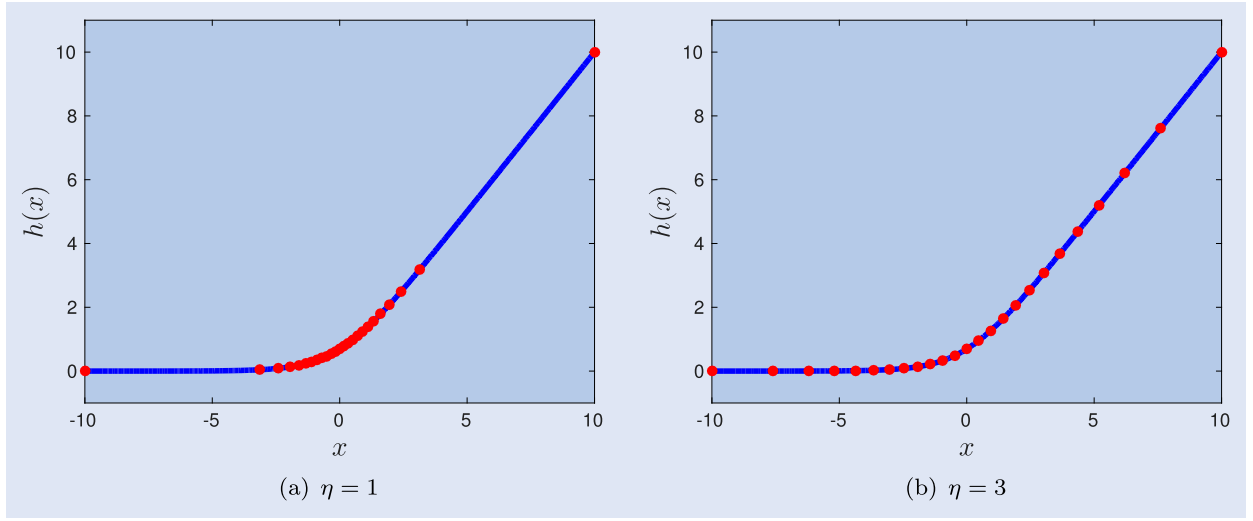


Figure 1. Optimal integration points. $L = 25$.

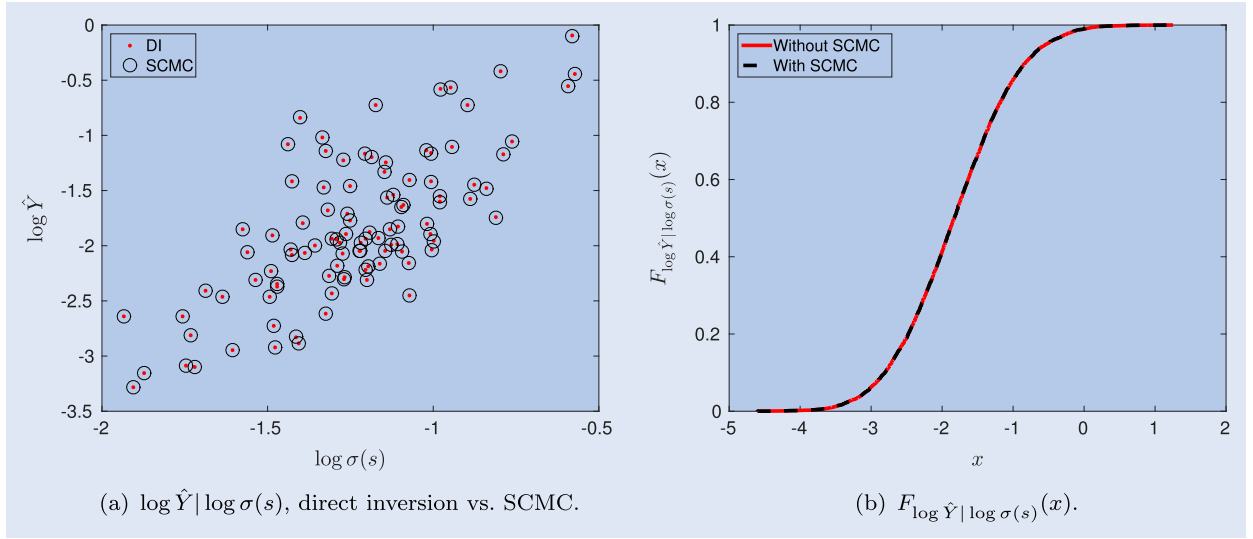


Figure 2. (a) Points cloud $\log \hat{Y} | \log \sigma(s)$ by direct inversion (DI) sampling (red dots) and SMC sampling (black circles). (b) Empirical CDFs of $\log Y | \log \sigma(s)$ with and without using SMC.

In order to show the performance of SMC technique for the simulation of $\log \hat{Y} | \log \sigma(s)$, the execution times of generating different numbers of samples are presented in table 1.

In appendix 1, the different sources of error due to the approximations made in the sections 3.1 and 3.2 are analysed.

3.3. Copula-based simulation of $\int_s^t \sigma^2(z) dz | \sigma(t), \sigma(s)$

In this section, we present an algorithm for the simulation of the integrated variance given $\sigma(t)$ and $\sigma(s)$ by means of a copula. In order to obtain the joint distribution, we require the marginal distributions. In our case, the required CDFs are $F_{\log Y | \log \sigma(s)}$ and $F_{\log \sigma(t) | \log \sigma(s)}$. In section 3.1, an approximated CDF of $\log Y | \log \sigma(s)$, $F_{\log \hat{Y} | \log \sigma(s)}$, has been derived. Since $\sigma(t)$ follows a log-normal distribution, by definition, $\log \sigma(t)$ is normally distributed, and the conditional process $\log \sigma(t)$ given $\log \sigma(s)$, follows a normal distribution,

$$\log \sigma(t) | \log \sigma(s) \sim \mathcal{N} \left(\log \sigma(s) - \frac{1}{2} \alpha^2 (t-s), \alpha \sqrt{t-s} \right). \quad (22)$$

3.3.1. Pearson's correlation coefficient. For any copula some measure of the correlation between the marginal distributions is needed. We will employ the Pearson correlation coefficient for $\log Y(s, t)$ and $\log \sigma(t)$. For this quantity, an approximated analytic formula can be derived. By definition, we have

$$\begin{aligned} \mathcal{P}_{\log Y, \log \sigma(t)} &= \text{corr} \left[\log \int_s^t \sigma^2(z) dz, \log \sigma(t) \right] \\ &= \frac{\text{cov} \left[\log \int_s^t \sigma^2(z) dz, \log \sigma(t) \right]}{\sqrt{\text{var} \left[\log \int_s^t \sigma^2(z) dz \right] \text{var} [\log \sigma(t)]}}. \end{aligned}$$

Table 1. SCMC time in seconds.

Samples	Without SCMC	With SCMC		
		$N_{\hat{Y}} = N_{\sigma} = 3$	$N_{\hat{Y}} = N_{\sigma} = 7$	$N_{\hat{Y}} = N_{\sigma} = 11$
100	1.0695	0.0449	0.0466	0.0660
10000	16.3483	0.0518	0.0588	0.0798
1000000	1624.3019	0.2648	0.5882	1.0940

We employ the following approximation

$$\log \int_s^t \sigma^2(z) dz \approx \int_s^t \log \sigma^2(z) dz = 2 \int_s^t \log \sigma(z) dz.$$

where the logarithm and the integral are interchanged. Since the log function is concave, this approximation forms a lower bound for the true value. This can be seen by applying Jensen's inequality, i.e.

$$\log \int_s^t \sigma^2(z) dz \geq \int_s^t \log \sigma^2(z) dz.$$

It has been numerically shown that, under the model settings with an interval size $t - s$ less than two years, the results based on this approximation are highly satisfactory (further details in [Leitao et al. \(2017\)](#)). The correlation coefficient can then be approximated by

$$\mathcal{P}_{\log Y(T), \log \sigma(t)} \approx \frac{\text{cov} \left[\int_s^t \log \sigma(z) dz, \log \sigma(t) \right]}{\sqrt{\text{var} \left[\int_s^t \log \sigma(z) dz \right] \text{var} [\log \sigma(t)]}}.$$

To compute the covariance, we need $\mathbb{E} \left[\left(\int_s^t \log \sigma(z) dz \right) \log \sigma(t) \right]$, $\mathbb{E} \left[\int_s^t \log \sigma(z) dz \right]$ and $\mathbb{E} [\log \sigma(t)]$. From equation (4), the last expectation as well as $\text{var} [\log \sigma(t)]$ are known. We find that

$$\log \sigma(t) = \log \sigma_0 - \frac{1}{2} \alpha^2 t + \alpha W(t) =: \eta(t) + \alpha W(t),$$

and

$$\begin{aligned} \int_s^t \log \sigma(z) dz &= \log \sigma_0(t-s) - \frac{1}{4} \alpha^2 (t^2 - s^2) + \alpha \int_s^t W(z) dz \\ &=: \gamma(s, t) + \alpha \left(tW(t) - sW(s) - \int_s^t z dW(z) \right). \end{aligned}$$

Based on these equations, using

$$\mathbb{E} [\log \sigma(T)] = \eta(t), \quad \mathbb{E} \left[\int_s^t \log \sigma(z) dz \right] = \gamma(s, t),$$

we obtain the following expressions for the remaining expectation,

$$\begin{aligned} &\mathbb{E} \left[\left(\int_s^t \log \sigma(z) dz \right) \log \sigma(t) \right] \\ &= \mathbb{E} \left[\left(\gamma(s, t) + \alpha \left(tW(t) - sW(s) - \int_s^t z dW(z) \right) \right) \right. \\ &\quad \left. \times (\eta(t) + \alpha W(t)) \right] \\ &= \eta(t) \gamma(s, t) + \alpha^2 \left(t^2 - s^2 - \mathbb{E} \left[\int_s^t z dW^2(z) \right] \right) \\ &= \eta(t) \gamma(s, t) + \frac{1}{2} \alpha^2 (t^2 - s^2). \end{aligned}$$

For the variance of $\int_s^t \log \sigma(z) dz$, we first compute

$$\begin{aligned} &\mathbb{E} \left[\left(\int_s^t \log \sigma(z) dz \right)^2 \right] \\ &= \mathbb{E} \left[\left(\gamma(s, t) + \alpha \left(tW(t) - sW(s) - \int_s^t z dW(z) \right) \right)^2 \right] \\ &= \gamma^2(s, t) + \alpha^2 \mathbb{E} \left[(tW(t) - sW(s))^2 \right] \\ &\quad + \alpha^2 \mathbb{E} \left[\left(\int_s^t z dW(z) \right)^2 \right] \\ &\quad - 2\alpha^2 \mathbb{E} \left[(tW(t) - sW(s)) \int_s^t z dW(z) \right], \end{aligned}$$

where

$$\begin{aligned} &\mathbb{E} \left[(tW(t) - sW(s))^2 \right] \\ &= t^2 \mathbb{E} [W(t)^2] + s^2 \mathbb{E} [W(s)^2] - 2st \mathbb{E} [W(s)W(t)] \\ &= t^3 + s^3 - 2s^2t, \\ &\mathbb{E} \left[\left(\int_s^t z dW(z) \right)^2 \right] \\ &= \int_s^t z^2 dz = \frac{1}{3} (t^3 - s^3), \\ &\mathbb{E} \left[(tW(t) - sW(s)) \int_s^t z dW(z) \right] \\ &= t \mathbb{E} \left[W(t) \int_s^t z dW(z) \right] - s \mathbb{E} \left[W(s) \int_s^t z dW(z) \right] \\ &= t \mathbb{E} \left[\int_s^t dW(z) \int_s^t z dW(z) \right] - s \mathbb{E} \left[\int_s^s dW(z) \int_s^t z dW(z) \right] \\ &= \frac{1}{2} t (t^2 - s^2), \end{aligned}$$

and the variance reads

$$\begin{aligned} \text{var} \left[\int_s^t \log \sigma(z) dz \right] &= \mathbb{E} \left[\left(\int_s^t \log \sigma(z) dz \right)^2 \right] - \left(\mathbb{E} \left[\int_s^t \log \sigma(z) dz \right] \right)^2 \\ &= \alpha^2 \left(t^3 + s^3 - 2s^2t + \frac{1}{3}(t^3 - s^3) - t(t^2 - s^2) \right) \\ &= \alpha^2 \left(\frac{1}{3}t^3 + \frac{2}{3}s^3 - s^2t \right). \end{aligned}$$

An approximation of the Pearson correlation coefficient is then obtained as follows:

$$\begin{aligned} \mathcal{P}_{\log Y, \log \sigma(t)} &\approx \frac{\eta(t)\gamma(s, t) + \frac{1}{2}\alpha^2(t^2 - s^2) - \eta(t)\gamma(s, t)}{\sqrt{\alpha^2 \left(\frac{1}{3}t^3 + \frac{2}{3}s^3 - s^2t \right) (\alpha^2 t)}} \\ &= \frac{t^2 - s^2}{2\sqrt{\left(\frac{1}{3}t^4 + \frac{2}{3}ts^3 - t^2s^2 \right)}}. \end{aligned} \quad (23)$$

Some copulas do not employ, in their definition, Pearson's coefficient as the correlation measure. For example, Archimedean copulas employ the Kendall's τ rank correlation coefficient. However, a relation between both Pearson's and Kendall's τ coefficients is available,

$$\mathcal{P} = \sin \left(\frac{\pi}{2} \tau \right).$$

3.3.2. Gumbel copula. In this work, we will use Archimedean Gumbel copula. The Gaussian copula may seem a natural choice, as $\sigma(t)$ follows a log-normal distribution and $Y(T) = \int_s^t \sigma^2(z) dz$ can be seen as summation of squared log-normal processes. However, in [Leitao et al. \(2017\)](#), we have empirically shown that the Gumbel copula performs most accurately in this context for a variety of SABR parameter values. The formal definition of the Archimedean Gumbel copula, considering $F_1 \dots, F_d \in [0, 1]^d$ as the marginal distributions, reads

$$C_\theta(F_1 \dots, F_d) = \exp \left(- \left(\sum_{i=1}^d (-\log(F_i))^\theta \right)^{1/\theta} \right), \quad (24)$$

where $(-\log(\cdot))^\theta, \theta > 0$ is the so-called *generator function* of the Gumbel copula. In order to calibrate parameter θ , a relation between θ and the rank correlation coefficient Kendall's τ can be employed, i.e. $\theta = 1/(1 - \tau)$.

So, we have derived approximations for all the components required to apply the copula-based technique for the integrated variance simulation. The algorithm to sample $\int_s^t \sigma^2(z) dz$ given $\sigma(t)$ and $\sigma(s)$ then consists of the following steps:

- (1) Determine $F_{\log \sigma(t) | \log \sigma(s)}$ by equation (22).
- (2) Determine $F_{\log \hat{Y} | \log \sigma(s)}$ by equation (6).
- (3) Determine the correlation between $\log Y(s, t)$ and $\log \sigma(t)$ by equation (23).
- (4) Generate correlated uniform samples, $U_{\log \sigma(t) | \log \sigma(s)}$ and $U_{\log \hat{Y} | \log \sigma(s)}$ from the Gumbel copula by equation (24).
- (5) From $U_{\log \sigma(t) | \log \sigma(s)}$, invert the CDF $F_{\log \sigma(t) | \log \sigma(s)}$ to get the samples $\tilde{\sigma}_n$ of $\log \sigma(t) | \log \sigma(s)$. This

procedure is straightforward since the normal distribution inversion is analytically available.

- (6) From $U_{\log \hat{Y} | \log \sigma(s)}$, invert the CDF $F_{\log \hat{Y} | \log \sigma(s)}$ to get the samples \tilde{y}_n of $\log \hat{Y} | \log \sigma(s)$. We propose an inversion procedure based on linear interpolation. First, we evaluate the CDF function at some discrete points. Then, the insight is that, by rotating the CDF under consideration, we can interpolate over probabilities. This is possible when the CDF function is a smoothly increasing function. The interpolation polynomial provides the quantiles of the original distribution from some given probabilities. Since $F_{\log \hat{Y} | \log \sigma(s)}$ is indeed a smooth and increasing function, the interpolation-based inversion is definitely applicable. This procedure together with the use of 2D SMC sampler (see section 3.2) results in a fast and accurate inversion.
- (7) The samples σ_n of $\sigma(t) | \sigma(s)$ and y_n of $Y(s, t) = \int_s^t \sigma^2(z) dz | \sigma(t), \sigma(s)$ are obtained by simply taking exponentials as follows:

$$\sigma_n = \exp(\tilde{\sigma}_n), \quad y_n = \exp(\tilde{y}_n).$$

3.4. Simulation of $S(t)$ given $S(s), \sigma(s), \sigma(t)$ and $\int_s^t \sigma^2(z) dz$

We complete the mSABR method by the conditional sampling of $S(t)$. The most commonly used techniques can be classified in two categories: direct inversion of the SABR distribution function given in equation (3) and moment-matching approaches. The direct inversion procedure has a higher computational cost because of the evaluation of the non-central χ^2 distribution. However, some recent developments make this computation affordable. [Chen et al. \(2012\)](#) proposed a forward asset simulation based on a combination of moment-matching (Quadratic Gaussian) and enhanced direct inversion procedures. We employ this technique also here. Note however that, for some specific values of β , the simulation of the conditional $S(t)$ given $S(s), \sigma(s), \sigma(t)$ and $\int_s^t \sigma^2(z) dz$ enables analytic expressions.

3.4.1. Case $\beta = 0$. For $\beta = 0$, it is easily seen from equation (2) that the asset process (in integral form) becomes

$$S(t) = S(s) + \rho \int_s^t \sigma(z) d\hat{W}_\sigma(z) + \sqrt{1 - \rho^2} \int_s^t \sigma(z) d\hat{W}_S(z),$$

where

$$\int_s^t \sigma(z) d\hat{W}_\sigma(z) = \frac{1}{\alpha} (\sigma(t) - \sigma(s)), \quad (25)$$

and

$$\int_s^t \sigma(z) d\hat{W}_S(z) | \sigma(t), \sigma(s) \sim \mathcal{N} \left(0, \sqrt{\int_s^t \sigma^2(z) dz} \right). \quad (26)$$

Since this allows negative asset prices, the simulation of $S(t)$ with $\beta = 0$ must be corrected. If the price at the initial time is less or equal to zero, i.e. $S(s) \leq 0$, the price $S(t)$ must remain zero for all time. Otherwise,

$$S(t) = \max(S(t), 0).$$

This correction is performed to be consistent within the whole simulation procedure, since the distribution of the 'exact' SABR

simulation in equation (3) relies on the condition $S(s) > 0$. If negative values are permitted, this must be handled in a different way (see Hagan *et al.* (2014) or Antonov *et al.* (2015), for example).

3.4.2. Case $\beta = 1$. In the case of $\beta = 1$, the asset dynamics in equation (1) become log-normal and the solution is given by

$$S(t) = S(s) \exp \left(-\frac{1}{2} \int_s^t \sigma^2(z) dz + \rho \int_s^t \sigma(z) d\hat{W}_\sigma(z) + \sqrt{(1-\rho^2)} \int_s^t \sigma(z) d\hat{W}_S(z) \right).$$

If we take the log-transform,

$$\log \left(\frac{S(t)}{S(s)} \right) = -\frac{1}{2} \int_s^t \sigma^2(z) dz + \rho \int_s^t \sigma(z) d\hat{W}_\sigma(z) + \sqrt{(1-\rho^2)} \int_s^t \sigma(z) d\hat{W}_S(z),$$

and by considering again equations (25) and (26), we obtain the distribution of $\log \left(\frac{S(t)}{S(s)} \right) | \int_s^t \sigma^2(z) dz, \sigma(t), \sigma(s)$, which reads

$$\mathcal{N} \left(-\frac{1}{2} \int_s^t \sigma^2(z) dz + \frac{\rho}{\alpha} (\sigma(t) - \sigma(s)), \sqrt{(1-\rho^2) \int_s^t \sigma^2(z) dz} \right). \quad (27)$$

By employing equation (27), the asset dynamics $S(t)$ can be sampled from

$$S(t) \sim S(s) \exp \left(-\frac{1}{2} \int_s^t \sigma^2(z) dz + \frac{\rho}{\alpha} (\sigma(t) - \sigma(s)) + X \sqrt{(1-\rho^2) \int_s^t \sigma^2(z) dz} \right), \quad (28)$$

where X is the standard normal.

3.4.3. Case $\beta \neq 0, \beta \neq 1$. As mentioned before, for the generic case of $\beta \in (0, 1)$, we employ the enhanced inversion of the SABR asset price distribution in equation (3) as introduced in Chen *et al.* (2012). We briefly summarize this approach. The asset simulation is performed either by a moment-matched quadratic Gaussian approximation or by an enhanced direct inversion. The authors in Chen *et al.* (2012) proposed a threshold value to choose the suitable technique among these two. The moment-matching approach relies on the fact that, for $S(s) \gg 0$, the distribution function in equation (3) can be approximated by

$$\Pr \left(S(t) \leq K | S(s) > 0, \sigma(s), \sigma(t), \int_s^t \sigma^2(z) dz \right) \approx \chi^2(c; 2-b, a).$$

By definition, the mean, μ and the variance, κ^2 , of a generic non-central chi-square distribution $\chi^2(x; \delta, \lambda)$ are $\mu := \delta + \lambda$ and $\kappa^2 := 2(\delta + 2\lambda)$, respectively. A variable $Z \sim \chi^2(x; \delta, \lambda)$

is accurately approximated by a quadratic Gaussian distribution, as follows

$$Z \stackrel{d}{\approx} d(e + X)^2, \quad X \sim \mathcal{N}(0, 1),$$

with

$$\psi := \frac{\kappa^2}{\mu^2}, \quad e^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1} - 1} \sqrt{2\psi^{-1} - 1} \geq 0, \\ d = \frac{\mu}{1 + e^2}.$$

Applying this to the case in equation (3), we can sample the conditional asset dynamics, $0 \leq s < t$, by

$$S(t) = \left((1-\beta)^2 (1-\rho^2) d(e + X)^2 \int_s^t \sigma^2(z) dz \right)^{\frac{1}{2(1-\beta)}}.$$

The moment-matching approximation is safely and accurately applicable when $0 < \psi \leq 2$ and $\mu \geq 0$ (Chen *et al.* 2012). Otherwise, a direct inversion procedure must be employed. A root-finding Newton method is then used. In order to reduce the number of Newton iterations (and the expensive evaluation of the non-central chi-square CDF), the first iterations are based on an approximated formula for the non-central chi-square distribution, which is based on the normal CDF and derived by Sankaran (1963). Then, the obtained value is employed as the initial guess for the Newton algorithm. The result is a significant reduction in the number of iterations and, hence, in the computational cost. Furthermore, this root-finding procedure consists of only basic operations, so that the whole procedure can be easily vectorized, leading to a further efficiency improvement.

As in the case of $\beta = 0$, the absorbing boundary at zero needs to be prescribed to avoid negative prices.

The resulting sampling procedure is robust and efficient.

3.4.4. Martingale correction. As already pointed by Andersen (2008) and Chen *et al.* (2012), the almost exact simulation of the asset price in some stochastic volatility models can produce a loss of the martingale property, due to the approximation of a continuous process by its discrete equivalent. This is especially seen, when the size of time step is large and for certain configurations of the SABR parameters, like small β values, close-to-zero initial asset values S_0 , high *vol-vol* parameter α or large initial volatility σ_0 . In order to overcome this issue, we incorporate to the mSABR method the use of a simple but effective numerical martingale correction, as follows:

$$S(t) = S(t) - \frac{1}{n} \sum_{i=1}^n S_i(t) + \mathbb{E}[S(t)], \\ = S(t) - \frac{1}{n} \sum_{i=1}^n S_i(t) + S_0,$$

where $S_i(t)$ represents the i -th Monte Carlo sample. Note, however, that, for practical SABR parameters, the martingale error is very small and can be easily controlled by increasing the number of overall time steps m .

4. Numerical experiments

In this section, we benchmark the mSABR method by pricing options under the SABR dynamics. We will consider several parameter configurations, extracted from the literature (see table 2), a zero-correlation set as in Grzelak *et al.* (2015) (set I), a set under log-normal dynamics as in Chen *et al.* (2012) (set II), a medium long time to maturity set as in Antonov *et al.* (2013) (set III) and a more extreme set as in Antonov *et al.* (2015) (set IV). The strikes, K_i , are chosen following the expression

$$K_i(T) = S(0) \exp(0.1 \times T \times \delta_i),$$

$$\delta = \{-1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5\},$$

introduced by Piterburg (2005).

Other parameters include:

- Step size in equation (5): $\Delta t = 5 \times 10^{-5}$.
- Number of terms in equations (14) and (20): $N = 150$.
- Number of intervals for piecewise linear approximation in equation (19): $L = 100$.
- Number of collocation points in equation (21): $N_{\hat{Y}} = N_{\sigma} = 3$.
- Number of samples: $n = 1.000.000$.

The experiments were performed on a computer with CPU Intel Core i7-4720HQ 2.6 GHz and RAM memory of 16 Gigabytes. The employed software package was Matlab r2015b.

As usual, the European option prices are obtained by averaging the maximum of the forward asset values at maturity minus the strike and zero, i.e. $\max(S(T) - K_i(T), 0)$ (call option). Subsequently, the Black–Scholes formula is inverted to determine the *implied volatilities*. Note that, once the forward asset is simulated by the mSABR method, we can price options at multiple strikes simultaneously.

4.1. Convergence test

We perform a convergence study in terms of the number of time steps, m and, subsequently, in terms of the number of samples, n . Set I is considered suitable for this experiment, since an analytic solution is available by Antonov *et al.* (2013) when $\rho = 0$, and it can be used as a reference. Also, as equation (3) is employed to sample the forward asset dynamics, the case $\rho = 0$ results in an *exact simulation*.

In table 3, the implied volatilities obtained by the mSABR method for several choices of the number of time steps m are presented. We observe by the errors in basis points (bp) that, by increasing the number of time steps, the technique converges very rapidly. Furthermore, only a few time steps are required to obtain accurate results: with only $m = 4$ time steps (one per year) the error is already less than ten basis points and with $m = 4T$ (the time step size is a quarter of year), the error remains below one basis points. Compared to the *low-bias SABR simulation scheme* proposed in Chen *et al.* (2012), the mSABR method is more accurate, since it relies on the *exact* simulation of the integrated variance, whereas the low-bias scheme approximates the integrated variance by a small disturbance expansion and moment-matching techniques. That scheme performs worse when bigger time steps are considered.

We also perform a convergence test in the number of samples, n . According to the previous experiment and in order to guarantee that the errors do not increase by the time discretization, we set $m = 4T$. In table 4, the implied volatilities for increasing n are presented. By the *relative error* (RE), we can see that, as expected, the error is approximately reduced by a factor of $1/\sqrt{n}$. Furthermore, we wish to show how the number of samples affects the variance of the mSABR method. In figure 3, the standard deviations[†] for several choices of n are presented. Two strikes are evaluated, one *in-the-money* strike, K_2 , and one *out-of-the-money* strike,[‡] K_6 . An identical behaviour can be observed for all other strikes as well. Again, we see a fast convergence and variance reduction in terms of n .

4.2. Performance test

Next to the convergence of the mSABR method in terms of m and n , another important aspect is the computational cost. Specifically, we will measure the execution times of an option pricing experiment using different alternative Monte Carlo-based methods and compare them with the mSABR method. Again, parameter Set I is employed since a reference value is available for this case. We consider a plain Monte Carlo Euler (MC Euler) discretization scheme for the forward asset, $S(t)$. Note that the absorbing boundary at zero must be handled with care. Since our main contribution here is the exact simulation of the integrated variance, it is natural to also compare the performance with other, less involved, techniques. We therefore also present a comparison between the mSABR method and two techniques where the integrated variance is either approximated by $\int_s^t \sigma^2(z) dz \approx \sigma^2(s)(t - s)$ (left rectangular rule) or $\int_s^t \sigma^2(z) dz \approx \frac{1}{2}(\sigma^2(t) + \sigma^2(s))(t - s)$ (trapezoidal rule). We denote these alternatives as the *Y-Euler* and *Y-trpz* schemes, respectively. The simulation of $S(t)$ is, in both cases, carried out as explained in section 3.4. In table 5, we present execution times (in seconds) of the MC Euler, *Y-Euler*, *Y-trpz* and mSABR schemes, to achieve a certain level of accuracy (measured as the error of the implied volatilities in basis points). In addition, the required number of time steps is presented in parentheses to provide an insight in the efficiency. Furthermore, the speedup obtained by our method is included in table 6. We observe that the mSABR method is superior in terms of the ratio between computational cost and accuracy. The accuracy of *Y-Euler* or *Y-trpz* can be improved by adding intermediate time steps, but then also the computational cost increases.

4.3. ‘Almost exact’ SABR simulation by varying ρ

As a next experiment we test the stability of the mSABR method when correlation parameter ρ varies from negative to positive values. As stated before, equation (3) is only exact for $\rho = 0$. For that, we use Set II since the conditional forward asset simulation enables a closed-form solution (given by equation (28)) for $\beta = 1$. The considered values are $\rho = \{-0.5, 0.0, 0.5\}$. Following the outcome of the previous experiment, we set $m = 4T$. As a reference, a Monte Carlo (MC)

[†]Computed by performing 100 realizations.

[‡]We consider European call options.

Table 2. Data sets.

	S_0	σ_0	α	β	ρ	T
Set I (Grzelak <i>et al.</i> 2015)	0.5	0.5	0.4	0.5	0.0	4
Set II (Chen <i>et al.</i> 2012)	0.04	0.2	0.3	1.0	-0.5	5
Set III (Antonov <i>et al.</i> 2013)	1.0	0.25	0.3	0.6	-0.5	20
Set IV (Antonov <i>et al.</i> 2015)	0.0056	0.011	1.080	0.167	0.999	1

Table 3. Implied volatility, increasing m : Antonov vs. mSABR. Set I.

Strikes	K_1	K_2	K_3	K_4	K_5	K_6	K_7
Antonov	73.34%	71.73%	70.17%	N/A	67.23%	65.87%	64.59%
$m = T/4$	73.13%	71.75%	70.41%	69.11%	67.85%	66.64%	65.48%
Error(bp)	-21.51	2.54	24.38	N/A	61.71	76.66	89.26
$m = T/2$	73.30%	71.78%	70.29%	68.86%	67.49%	66.17%	64.93%
Error(bp)	-4.12	4.94	12.71	N/A	25.48	30.40	34.73
$m = T$	73.25%	71.67%	70.14%	68.66%	67.24%	65.89%	64.62%
Error(bp)	-9.56	-5.93	-2.79	N/A	0.92	2.21	3.17
$m = 2T$	73.32%	71.71%	70.16%	68.65%	67.22%	65.85%	64.55%
Error(bp)	-2.08	-1.56	-1.20	N/A	-1.65	-2.35	-3.36
$m = 4T$	73.34%	71.73%	70.18%	68.67%	67.24%	65.87%	64.58%
Error(bp)	0.15	0.58	0.78	N/A	0.43	0.04	-0.48

Table 4. Implied volatility, increasing n : Antonov vs. mSABR. Set I.

Strikes	K_1	K_2	K_3	K_4	K_5	K_6	K_7
Antonov	73.34%	71.73%	70.17%	N/A	67.23%	65.87%	64.59%
$n = 10^2$	67.29%	65.55%	63.84%	62.20%	60.63%	59.01%	57.65%
RE	8.24×10^{-2}	8.61×10^{-2}	9.01×10^{-2}	N/A	9.82×10^{-2}	1.04×10^{-1}	1.07×10^{-1}
$n = 10^4$	73.41%	71.87%	70.36%	68.91%	67.51%	66.19%	64.94%
RE	9.65×10^{-4}	1.94×10^{-3}	2.75×10^{-3}	N/A	4.08×10^{-3}	4.93×10^{-3}	5.48×10^{-3}
$n = 10^6$	73.34%	71.73%	70.18%	68.67%	67.24%	65.87%	64.58%
RE	2.04×10^{-5}	8.08×10^{-5}	1.11×10^{-4}	N/A	6.39×10^{-5}	6.07×10^{-6}	7.43×10^{-5}

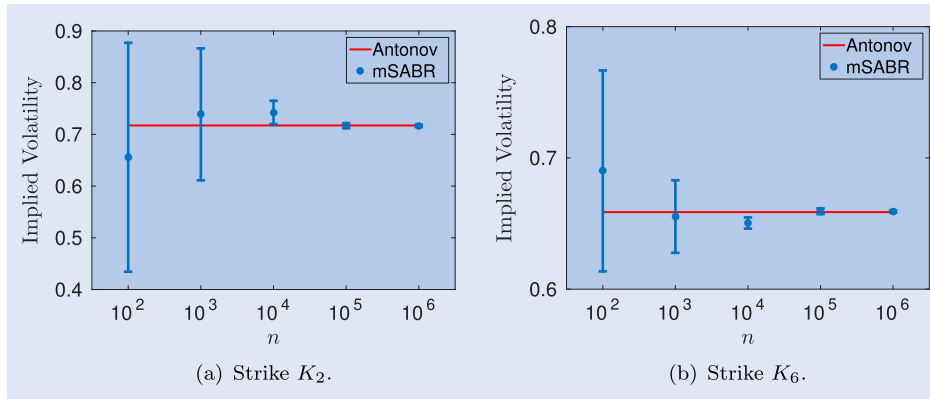


Figure 3. Convergence of the mSABR method.

simulation with a very fine Euler time discretization (1000T time steps) in the forward asset process is employed. In table 7, the resulting implied volatilities are provided. The differences between the Monte Carlo volatilities and the ones given by the mSABR method are within 10 basis points for all choices of ρ .

4.4. Pricing barrier options

We will price barrier options with the mSABR method. The *up-and-out* call option is considered here, with the barrier level, B , $B > S_0$, $B > K_i$. The price of this type of barrier option reads

Table 5. Execution times and time steps, m (parentheses).

Error	<100 bp	<50 bp	<25 bp	<10 bp
MC Euler	6.85(200)	10.71(300)	27.42(800)	42.90(1200)
Y-Euler	2.18(4)	6.55(16)	11.85(32)	45.12(128)
Y-trpz	2.17(3)	4.24(8)	7.25(16)	14.47(32)
mSABR	3.46(1)	2.98(2)	3.72(3)	4.89(4)

Table 6. Speedups provided by the mSABR method.

Error	<100 bp	<50 bp	<25 bp	<10 bp
MC Euler	1.98	3.59	7.37	8.77
Y-Euler	0.63	2.19	3.18	9.22
Y-trpz	0.62	1.42	1.94	2.95

Table 7. Implied volatility, varying ρ : Euler Monte Carlo vs. mSABR. Set II.

Strikes	K_1	K_2	K_3	K_4	K_5	K_6	K_7
$\rho = -0.5$							
MC	22.17%	21.25%	20.38%	19.57%	18.88%	18.33%	17.95%
mSABR	22.21%	21.28%	20.39%	19.58%	18.88%	18.32%	17.94%
Error(bp)	3.59	2.86	1.78	0.95	-0.19	-0.96	-1.10
$\rho = 0.0$							
MC	21.35%	20.96%	20.71%	20.63%	20.71%	20.96%	21.34%
mSABR	21.35%	20.95%	20.69%	20.60%	20.68%	20.93%	21.32%
Error(bp)	0.04	-1.04	-2.51	-3.02	-3.33	-3.19	-2.56
$\rho = 0.5$							
MC	19.66%	20.04%	20.61%	21.34%	22.20%	23.14%	24.16%
mSABR	19.59%	19.96%	20.54%	21.28%	22.15%	23.11%	24.11%
Error(bp)	-6.93	-7.36	-6.77	-5.53	-4.35	-3.76	-4.05

$$V_i(K_i, B, T) = \exp(-rT) \mathbb{E} \left[(S(T) - K_i) \mathbb{1}(\max_{0 \leq t_k \leq T} S(t_k) > B) \right],$$

where $\mathbb{1}(\cdot)$ represents the indicator function and t_k are the predefined times where the barrier condition (whether or not it is hit) is checked.

As a reference, a Monte Carlo method with very fine Euler time discretization is employed, as in the previous experiment. The number of time steps for the mSABR scheme is again set to $m = 4T$. In tables 8, 9 and 10, the obtained prices are presented for the parameter sets I, II and III, respectively. The prices are multiplied by a factor of 100. Also, the *mean squared error* (MSE) is shown. We define the MSE as

$$\text{MSE} = \frac{1}{7} \sum_{i=1}^7 \left(V_i^{MC}(K_i, B, T) - V_i^{mSABR}(K_i, B, T) \right)^2$$

where $V_i^{MC}(K_i, B, T)$ and $V_i^{mSABR}(K_i, B, T)$ are the barrier option prices provided by Monte Carlo method and by the mSABR method, respectively.

The resulting accuracy is very satisfactory. As expected, higher option prices are obtained for bigger barrier levels, B , and/or lower strikes, K_i .

4.5. Negative interest rates

The SABR model is very popular in the interest rate context. One of the most important current features in this market is the occurrence of negative interest rate values and strikes. Approaches dealing with this issue have appeared in the literature, like Antonov *et al.* (2015) or Hagan *et al.* (2014). To apply the mSABR scheme in the case of negative interest rates, we will use the *shifted SABR model* by Schlenkrich *et al.* (2014). The shifted SABR model is defined as follows:

$$\begin{aligned} dS(t) &= \sigma(t)(S(t) + \theta)^\beta dW_S(t), \\ S(0) &= (S_0 + \theta) \exp(rT), \end{aligned}$$

where $\theta > 0$ is a displacement, or shift, in the underlying. The volatility process, $\sigma(t)$, remains invariant (see equation (1)). Parameter θ can be seen as the maximum level of negativity that is expected in the rates. This generalization of the SABR model is widely used by market practitioners due to its simplicity, and, precisely, the advantage of keeping the existing simulation schemes. In mSABR, the assumption $S(t) > 0$ is required to employ the method.

In order to test the mSABR scheme in the context of negative interest rates, we employ the set IV. The SABR model parameters were obtained by Antonov *et al.* (2015) after a

Table 8. Pricing barrier options with mSABR: $V_i(K_i, B, T) \times 100$. Set I.

Strikes	K_1	K_2	K_3	K_4 $B = 1.2$	K_5	K_6	K_7
MC	6.2868	5.4577	4.6330	3.8263	3.0551	2.3358	1.6869
mSABR	6.3264	5.4890	4.6566	3.8436	3.0677	2.3452	1.6953
MSE	5.3196×10^{-8}						
$B = 1.5$							
MC	10.2271	9.1568	8.0718	6.9856	5.9142	4.8753	3.8899
mSABR	10.2249	9.1507	8.0609	6.9702	5.8960	4.8572	3.8737
MSE	1.8884×10^{-8}						
$B = 1.8$							
MC	13.5740	12.3571	11.1118	9.8513	8.5910	7.3477	6.1403
mSABR	13.5915	12.3686	11.1174	9.8507	8.5851	7.3380	6.1276
MSE	1.0851×10^{-8}						

Table 9. Pricing barrier options with mSABR: $V_i(K_i, B, T) \times 100$. Set II.

Strikes	K_1	K_2	K_3	K_4 $B = 0.08$	K_5	K_6	K_7
MC	1.1702	0.9465	0.7268	0.5215	0.3423	0.1996	0.0987
mSABR	1.1724	0.9486	0.7285	0.5226	0.3428	0.1997	0.0986
MSE	1.8910×10^{-10}						
$B = 0.1$							
MC	1.3099	1.0766	0.8462	0.6290	0.4367	0.2794	0.1626
mSABR	1.3092	1.0761	0.8456	0.6282	0.4355	0.2782	0.1618
MSE	7.5542×10^{-11}						
$B = 0.12$							
MC	1.3521	1.1168	0.8841	0.6644	0.4695	0.3093	0.1891
mSABR	1.3518	1.1166	0.8838	0.6639	0.4686	0.3080	0.1880
MSE	6.3648×10^{-11}						

Table 10. Pricing barrier options with mSABR: $V_i(K_i, B, T) \times 100$. Set III.

Strikes	K_1	K_2	K_3	K_4 $B = 2.0$	K_5	K_6	K_7
MC	29.1174	23.4804	17.2273	10.7825	5.0203	1.1750	0.0036
mSABR	29.2346	23.5828	17.3086	10.8327	5.0385	1.1805	0.0036
MSE	4.8146×10^{-7}						
$B = 2.5$							
MC	41.3833	34.5497	26.8311	18.6089	10.7281	4.4893	0.9434
mSABR	41.3394	34.5097	26.7948	18.5747	10.6943	4.4546	0.9320
MSE	1.2131×10^{-7}						
$B = 3.0$							
MC	48.5254	41.1652	32.7980	23.7807	14.9344	7.5364	2.6692
mSABR	48.5008	41.1515	32.7888	23.7655	14.9097	7.5117	2.6549
MSE	3.6201×10^{-8}						

calibration process to real data[†] under the shifted SABR model. Note that this configuration corresponds to an extreme situation with high values of the correlation, ρ , and *vol-vol*, α . As a

shift parameter, $\theta = 0.02$ is chosen. We compare the *normal implied volatilities*[‡] obtained by employing Monte Carlo with very fine (1000T) time discretization (as a reference) and the

[†]1Y15Y Swiss franc swaption from 10 February 2015.

[‡]Determined using the *normal* Bachelier model instead of the *log-normal* Black–Scholes model.

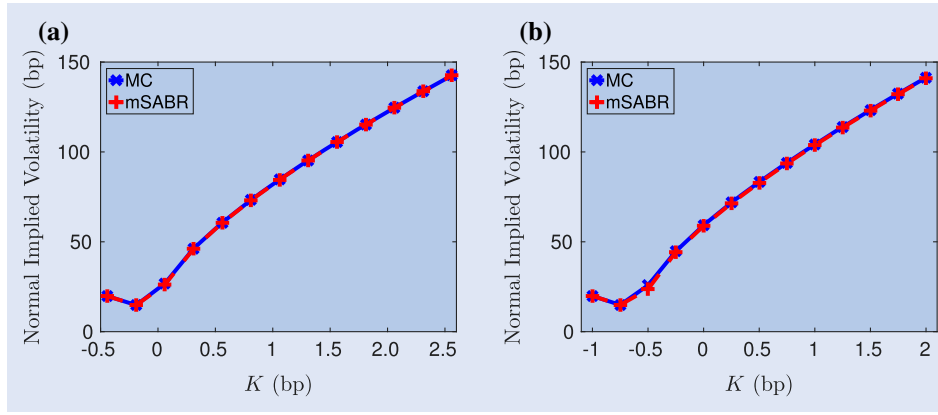


Figure 4. The mSABR method in the context of negative interest rates.

Table 11. Execution times and time steps, m (parentheses).

Error	< 25 bp	< 10 bp	< 5 bp	< 1 bp
Y-Euler	2.47(4)	9.64(16)	14.32(24)	19.25(32)
Y-trpz	1.32(2)	4.99(8)	9.75(16)	14.52(24)
mSABR	1.61(1)	2.57(2)	3.52(3)	4.57(4)

Table 12. Speedups provided by the mSABR method.

Error	< 25 bp	< 10 bp	< 5 bp	< 1 bp
Y-Euler	1.57	3.75	4.07	4.21
Y-trpz	0.82	1.94	2.77	3.17

mSABR method with $m = 4T$. In figure 4(a), these curves are depicted and a perfect fit is observed. The strikes, K are chosen to permit negative values. We perform an even more extreme experiment, by setting $S_0 = 0$. The resulting normal implied volatilities are presented in figure 4(b), again with an excellent fit.

Due to the complexity of the negative interest rates experiment and this particular set of parameters ($\rho \approx 1$), we wish to test the performance of our method under these conditions. In tables 11 and 12 the execution times obtained are presented. Again, the small number of time steps due to the ‘almost’ exact simulation of the mSABR method provides an important gain in performance.

5. Conclusions

In this work, we have proposed an accurate and robust multiple time step Monte Carlo method to simulate the SABR model with only a few time steps. The mSABR method employs many nontrivial components that have not been seen before in combination within a Monte Carlo simulation. A copula methodology to generate samples from the conditional SABR’s integrated variance process has been introduced. The marginal distribution of the integrated variance has been derived by employing Fourier techniques. We have dealt with an increasing computational complexity, due to the multiple time steps and the almost exact simulation, by applying an interpolation based on stochastic collocation. This has resulted in an

efficient SABR simulation scheme. We have numerically shown the convergence and the stability of the method. In terms of convergence, the mSABR method requires only very few time steps to achieve high accuracy, due to the focus on almost exact simulation (in contrast to the low-bias SABR scheme [Chen et al. 2012](#)). This fact impacts the performance, obtaining an excellent ratio between accuracy and computational cost. As a multiple time step method, it can be employed to price path-dependent options. As an example, a pricing experiment for barrier options has been carried out. Furthermore, we have shown that the mSABR scheme is also suitable in the context of negative interest rates, in combination with a shifted model.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The first author is supported by the EU in the FP7-PEOPLE-2012-ITN Program [Agreement number 304617] (FP7 Marie Curie Action, Project Multi-ITN STRIKE—Novel Methods in Computational Finance).

References

- Andersen, L.B.G., Efficient simulation of the Heston stochastic volatility model. *J. Comput. Finance*, 2008, **11**, 1–22.
- Antonov, A., Konikov, M. and Spector, M., SABR spreads its wings. *Risk Mag.*, 2013, 58–63.
- Antonov, A., Konikov, M. and Spector, M., The free boundary SABR: Natural extension to negative rates. *Risk Mag.*, August, 2015.
- Benhamou, E., Fast Fourier transform for discrete Asian options. *J. Comput. Finance*, 2002, **6**, 49–68.
- Broadie, M. and Kaya, O., Exact simulation of stochastic volatility and other affine jump diffusion processes. *Oper. Res.*, 2006, **54**, 217–231.
- Cai, N., Song, Y. and Chen, N., Exact simulation of the SABR model. *Oper. Res.*, forthcoming.
- Chen, B., Oosterlee, C.W. and van der Weide, H., A low-bias simulation scheme for the SABR stochastic volatility model. *Int. J. Theor. Appl. Finance*, 2012, **15**, 1250016-1–1250016-37.
- Cox, J.C., Note on option pricing I: Constant elasticity of variance diffusions. *J. Portfolio Manage.*, 1996, **22**, 15–17.
- Fang, F. and Oosterlee, C.W., A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 2008, **31**, 826–848.

- Grzelak, L.A., Witteveen, J.A.S., Suárez-Taboada, M. and Oosterlee, C.W., The stochastic collocation Monte Carlo sampler: Highly efficient sampling from "expensive" distributions. 2015. Available at SSRN: <http://ssrn.com/abstract=2529691>
- Hagan, P.S., Kumar, D., Lesniewski, A.S. and Woodward, D.E., Managing smile risk. *Wilmott Mag.*, January, 2002, 84–108.
- Hagan, P.S., Kumar, D., Lesniewski, A.S. and Woodward, D.E., Arbitrage-free SABR. *Wilmott Mag.*, January, 2014, 60–75.
- Islah, O., Solving SABR in exact form and unifying it with LIBOR market model. 2009. Available at SSRN: <http://ssrn.com/abstract=1489428>
- Kennedy, J.E. and Pham, D., On the Approximation of the SABR with mean reversion model: A probabilistic approach. *Appl. Math. Finance*, 2014, **21**, 451–481.
- Kloeden, P.E. and Platen, E., *Numerical solution of stochastic differential equations*, 1992 (Springer-Verlag: Berlin).
- Leitao, A., Grzelak, L.A. and Oosterlee, C.W., On a one time-step Monte Carlo simulation approach of the SABR model: Application to European options. *Appl. Math. Comput.*, 2017, **293**, 461–479.
- Lord, R. and Farebrother, A., Fifty shades of SABR simulation. Presentation at 10th Fixed Income Conference, WBS Training, Barcelona, Spain, 2014. Available online at: <http://www.rogerlord.com/fiftyshadesabrwb.pdf>
- Obloj, J., Fine-tune your smile: Correction to Hagan et al. 2008. Available at arXiv: <http://arxiv.org/abs/0708.0998v3>
- Piterbarg, V., A multi-currency model with FX volatility skew. 2005. Available at SSRN: <http://ssrn.com/abstract=685084>
- Sankaran, M., Approximations to the non-central chi-square distribution. *Biometrika*, 1963, **50**, 199–204.
- Schlenkrich, S., Miemiec, A. and Wolff-Siemssen, T., Low strike extrapolation for SABR. *Wilmott Mag.*, November, 2014.
- Schroder, M., Computing the constant elasticity of variance option pricing formula. *J. Finance*, 1989, **44**, 211–219.
- Sklar, A., Fonctions de répartition à n dimensions et leurs marges [N-dimensional distribution functions and their margins]. *Publ. Inst. Statist. Univ. Paris*, 1959, **8**, 229–231.
- Smith, R.D., An almost exact simulation method for the Heston model. *J. Comput. Finance*, 2007, **11**, 115–125.
- van Hasstrecht, A. and Pelsser, A.A., Efficient, almost exact simulation of the Heston stochastic volatility model. *Int. J. Theor. Appl. Finance*, 2010, **13**, 1–43.
- Zhang, B. and Oosterlee, C.W., Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions. *SIAM J. Financ. Math.*, 2013, **4**, 399–426.

Appendix 1. Error analysis of $\int_s^t \sigma^2(z) dz | \sigma(s)$ simulation

A.1. Errors in the CDF of $\int_s^t \sigma^2(z) dz | \sigma(s)$

In section 3.1, we have proposed a method to compute the approximated CDF for the conditional integrated variance, i.e. $Y(s, t) | \sigma(s)$. Here, we perform an error analysis, indicating how to reduce or bound the errors. The sources of error in the approximation include ϵ_D , the discretization of $Y(s, t)$ in equation (5), ϵ_T , the truncation in equations (14) and (20), ϵ_C , due to the cosine expansion in equations (14) and (20) and ϵ_M , the numerical computation of I_M in equation (19).

A.1.1. Error ϵ_D . This error occurs because SABR's time-integrated variance is approximated by its discrete equivalent, i.e. $Y(s, t) \approx \hat{Y}(s, t)$. Note that the process $Y(s, t) = \int_s^t \sigma^2(z) dz$ is the solution of the SDE

$$dY(s, z) = \sigma^2(z) dz, \quad Y(s, s) = 0,$$

which, by employing the same time discretization as for $\hat{Y}(s, t)$ in Equation (5), can be written as follows:

$$\bar{Y}(s, t_j) - \bar{Y}(s, t_{j-1}) = \sigma^2(t_{j-1}) \Delta t.$$

This is essentially the Euler–Maruyama discretization scheme. Indeed, it is easy to see that, if we sum the right-hand side of the equality

over the M discrete time points, we recover the discrete approximation $\hat{Y}(s, t)$, as follows:

$$\begin{aligned} \sum_{j=1}^M \sigma^2(t_{j-1}) \Delta t &= \sum_{j=1}^M (\bar{Y}(s, t_j) - \bar{Y}(s, t_{j-1})) \\ &= \bar{Y}(s, t_M) - \bar{Y}(s, t_0) = \hat{Y}(s, t). \end{aligned}$$

The error (or convergence) of the Euler–Maruyama discretization scheme has been intensively studied in the literature (see, e.g. Kloeden and Platen 1992). Two errors can be defined when a stochastic process is discretized: *strong* and *weak*. It was proved by Kloeden and Platen (1992) that, under some conditions of regularity, the Euler–Maruyama scheme has strong convergence of order $\gamma = \frac{1}{2}$ and weak convergence of order $\gamma = 1$. We focus on the strong error between $Y(s, t)$ and $\hat{Y}(s, t)$. Then, error ϵ_D can be computed and bounded by employing the usual strong convergence expression as

$$\epsilon_D := \mathbb{E} \left[\left| \hat{Y}(s, t) - Y(s, t) \right| \right] \leq C \left(\frac{t-s}{M} \right)^\gamma,$$

for some constant $C > 0$.

In the actual experiments, we consider $\log \hat{Y}(s, t)$ and $\log Y(s, t)$. This will reduce the error further. In Leitao et al. (2017), we have presented numerical results that confirm the expected behaviour of this error.

A.1.2. Error ϵ_T . The use of cosine expansions and the COS method implies a truncation of the integration range to interval $[a, b]$. This gives rise to a truncation error, which is defined by

$$\epsilon_T(H) := \int_{\mathbb{R} \setminus [a, b]} f_H(y) dy,$$

where H corresponds to Y_j and $\log \hat{Y}(s, t)$, according to equations (14) and (20), respectively. By a sufficiently large integration range $[a, b]$, this error can be reduced and does not dominate the overall error.

A.1.3. Error ϵ_C . The convergence due to the number of terms N in the cosine series expansion was derived in Fang and Oosterlee (2008). For any $f(y|x) \in \mathcal{C}^\infty[a, b]$, ϵ_C can be bounded by

$$|\epsilon_C(N, [a, b])| \leq P^*(N) \exp(-(N-1)\nu),$$

being $\nu > 0$ a constant and $P^*(N)$ a term which varies less than exponentially with respect to N . As the integrated variance is a very smooth function, ϵ_C will decay exponentially with respect to N .

A.1.4. Error ϵ_M . When the piecewise approximation in equation (19) is employed, the convergence in terms of the number of sub-intervals, L , is numerically tested. In table A1, we observe a reduction of the error by a factor four, when going from $L = 25$ to $L = 50$ integration points. However, with $L = 100$ the error reduces more drastically due to the choice of optimal integration points.

A.2. Error in the application of SCMC to $\int_s^t \sigma^2(z) dz | \sigma(s)$

In Grzelak et al. (2015), where SCMC sampler was proposed, an error analysis of the technique was carried out. Here we summarize the most important results related to our context and refer the reader to the original paper for further details.

To measure the error which results from the collocation method in a more general case (see section 2.2), we can consider either the difference between functions $g(X)$ and $g_N(X)$ (X the cheap variable) or the error associated with the approximated CDF. The first type of error is due to Lagrange interpolation, so the corresponding error estimate is well known. The error $\epsilon(\xi_N)$ with N_Y collocation points is given by the standard Lagrange interpolation reads

Table A1. MSE vs. L for $\hat{I}_{\mathcal{M}}$ in equation (19).

	$L = 25$	$L = 50$	$L = 75$	$L = 100$
Equidistant	2.4075×10^{-3}	2.6743×10^{-4}	3.1064×10^{-4}	2.9547×10^{-6}
Optimal ($\eta = 1$)	1.3273×10^{-2}	2.5343×10^{-3}	9.5959×10^{-4}	4.7953×10^{-4}
Optimal ($\eta = 3$)	3.4837×10^{-5}	1.1445×10^{-6}	2.1506×10^{-7}	6.7114×10^{-8}

$$\epsilon_X(\xi_n) = |g(\xi_n) - g_{N_Y}(\xi_n)| = \left| \frac{1}{N_Y!} \frac{\partial^{N_Y} g(x)}{\partial x^{N_Y}} \Big|_{x=\xi} \prod_{i=1}^{N_Y} (\xi_n - \bar{x}_i) \right|, \quad (\text{A1})$$

where \bar{x}_i are the collocation points, and $\hat{\xi} \in [\min(\mathbf{x}), \max(\mathbf{x})]$, $\mathbf{x} = (\bar{x}_1, \dots, \bar{x}_{N_Y})^T$. The error can be bounded by defining ξ as the point where $|\partial^{N_Y} g(x)/\partial x^{N_Y}|$ has its maximum. A small probability of large errors *in the tails* can be observed by deriving the error $\epsilon_U(\xi_n)$, by substituting the uniformly distributed random variable u_n in the previous equation, using $\xi_n = F_X^{-1}(u_n)$,

$$\begin{aligned} \epsilon_U(u_n) &= |g(F_X^{-1}(u_n)) - g_{N_Y}(F_X^{-1}(u_n))| \\ &= \left| \frac{1}{N_Y!} \frac{\partial^{N_Y} g(x)}{\partial x^{N_Y}} \Big|_{x=\xi} \prod_{i=1}^{N_Y} (F_X^{-1}(u_n) - \bar{x}_i) \right|. \end{aligned}$$

A ‘close-to-linear’ relation between the involved stochastic variables, meaning $\partial^{N_Y} g(x)/\partial x^{N_Y}$ being small, gives a small approximation error. On the other hand, when approximating the CDF of the expensive variable Y , we have $F_Y(y) = F_Y(g(x)) \approx F_Y(g_{N_Y}(x))$, which is exact at the collocation points \bar{x}_i ,

$$F_Y(y_i) = F_Y(g(\bar{x}_i)) = F_Y(g_{N_Y}(\bar{x}_i)).$$

A deeper analysis can be found in the original paper of SCMC (Grzelak *et al.*, 2015), including a theoretical convergence in L^2 and the convergence within the distribution tails.

We have chosen X to be standard normal, and $Y = \log \hat{Y} | \log \sigma(s)$ can be seen as the logarithm of a sum of log-normal variables. Therefore, it is a ‘close-to-normal’ distribution which can be efficiently approximated by a linear combination of standard normal distributions. In order to measure the error when SCMC is applied to the sampling of $\int_s^t \sigma^2(z) dz | \sigma(s)$ (see section 3.2), we can consider the difference between the functions $g(x) = F_{\log \hat{Y} | \log \sigma(s)}^{-1}(F_X(x))$ and $g_{N_{\hat{Y}}, N_{\sigma}}(x)$ in equation (21). By following the same derivation as in equation (A1), we have

$$\begin{aligned} \epsilon_{\hat{Y}}(\xi_n) &= |g(\xi_n) - g_{N_{\hat{Y}}, N_{\sigma}}(\xi_n)| \\ &= \left| \frac{1}{N_{\hat{Y}}!} \frac{\partial^{N_{\hat{Y}}} g(x)}{\partial x^{N_{\hat{Y}}}} \Big|_{x=\xi} \prod_{i=1}^{N_{\hat{Y}}} (\xi_n - \bar{x}_i) \prod_{j=1}^{N_{\sigma}} (v_n - \bar{v}_j) \right|, \end{aligned}$$

where $N_{\hat{Y}}$ and N_{σ} are the number of collocation points in each dimension and \bar{v}_j are the collocation points corresponding to the conditional random variable $\log \sigma(s)$.