On a one time-step Monte Carlo simulation approach of the SABR model: Application to European options

Álvaro Leitao [a,b,*], Lech A. Grzelak [c,a], Cornelis W. Oosterlee [b,a]

[a] *Delft Institute of Applied Mathematics, TU Delft, Delft, The Netherlands*
[b] *CWI-Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*
[c] *Quantitative Analytics, ING, Amsterdam, The Netherlands*

A B S T R A C T

In this work, we propose a one time-step Monte Carlo method for the SABR model. We base our approach on an accurate approximation of the cumulative distribution function of the time-integrated variance (conditional on the SABR volatility), using Fourier techniques and a copula. Resulting is a fast simulation algorithm which can be employed to price European options under the SABR dynamics. Our approach can thus be seen as an alternative to Hagan's analytic formula for short maturities that may be employed for model calibration purposes.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

The *Stochastic Alpha Beta Rho (SABR)* model [1] is an established SDE system which is often used for interest rates and FX modeling in practice. The model belongs to the so-called *stochastic local volatility (SLV)* models. The idea behind SLV models is that the modeling of volatility is partly done by a local volatility and partly by a stochastic volatility contribution, aiming to preserve the advantages and minimize the disadvantages of the individual models.

In the original paper [1], the authors have provided a closed-form approximation formula for the implied volatility under SABR dynamics. This is important for practitioners, as it can be used highly efficiently within the calibration exercise. However, the closed-form expression is derived by perturbation theory and its applicability is thus not general. The formula is for example not always accurate for small strike values or for long time to maturity options. In [2], Oblój provided a correction to the Hagan formula but the same drawbacks remained. In order to improve the approximation for the SABR implied volatility, different techniques have been proposed in the literature, based on either perturbation theory [3,4], heat kernel expansions [5,6] or model map techniques [7].

The objective of a bank is typically to employ the SABR model for pricing exotic derivatives. Before this pricing can take place, the model needs to be calibrated to European options, and, if possible, to some exotic options as well. It is market practice to use Hagan's formula to calibrate the SABR model, however, since it is only an approximation it will not always result in a sufficient fit to the market. In other words, a SABR Monte Carlo simulation may give different implied volatilities than the Hagan formula.

In this work, we aim to develop a *one time-step* Monte Carlo method, which is accurate for European derivative contracts up to two years. This kind of contract is often traded in foreign-exchange (FX) markets. The present approach may thus

---

* Corresponding author at: Delft Institute of Applied Mathematics, TU Delft, Delft, The Netherlands.
E-mail addresses: A.LeitaoRodriguez@tudelft.nl (Á. Leitao), L.A.Grzelak@tudelft.nl (L.A. Grzelak), c.w.oosterlee@cwi.nl (C.W. Oosterlee).

be used to improve the calibration of the SABR model. In particular, we focus on the efficient simulation of SABR's time-integrated variance, conditional on the volatility dynamics, as this conditional distribution is not available in closed-form. We approximate the distribution by the two marginal distributions involved, i.e. the volatility and time-integrated variance distributions, by using a copula. The copula methodology has been used intensively in the financial world, see, for example, [8], in risk management and option pricing. In the field of risk management, some relevant works are Li [9], Embrechts et al. [10], Cherubini and Luciano [11] or Rosenberg and Schuermann [12]. In the case of pricing multi-asset options, some examples of the use of copulas are Cherubini and Luciano [13] or Rosenberg [14].

The approximation of the *cumulative distribution function (CDF)* of the time-integrated variance is obtained by applying a recursive procedure, as described in [15], which was originally developed to price arithmetic Asian options. The derivation is based on the definition of the characteristic function of the time-integrated variance and the use of Fourier inversion techniques. We adapt the procedure to our problem and improve it in terms of computational costs. Once both marginal distributions are available, we employ the copula technique to get a multivariate distribution which approximates the conditional distribution of the time-integrated variance given the volatility. Several copulas are considered and an analysis of their performance is carried out. The conditional time-integrated variance approximation will be employed in the asset simulation for the complete SABR model.

The paper is organized as follows. In Section 2, the SABR model and the simulation steps are briefly introduced. Section 3 describes the procedure to derive the CDF of SABR's time-integrated variance. The copula approach to approximate the conditional time-integrated variance distribution is explained in Section 4. In Section 5, some numerical experiments are presented. We conclude in Section 6.

The one-step methodology restricts the use of the proposed method to option maturities up to two years and European-type options. In a forthcoming paper, we will generalize the methodology to the *multiple time-step* case.

2. The SABR model

The SABR model is based on a parametric local volatility component in terms of a model parameter, β. The formal definition of the SABR model reads

$$
\begin{aligned}
\mathrm{d}S(t) &= \sigma(t)S^\beta(t)\mathrm{d}W_S(t), & S(0) &= S_0 \exp(rT), \\
\mathrm{d}\sigma(t) &= \alpha\sigma(t)\mathrm{d}W_\sigma(t), & \sigma(0) &= \sigma_0.
\end{aligned}
\tag{1}
$$

where $S(t) = \bar{S}(t)\exp(r(T-t))$ denotes the forward value of the underlying asset $\bar{S}(t)$, with r the interest rate, S_0 the spot price and T the contract's final time. Quantity $\sigma(t)$ denotes the stochastic volatility, with $\sigma(0) = \sigma_0$, $W_S(t)$ and $W_\sigma(t)$ are two correlated Brownian motions with constant correlation coefficient ρ (i.e. $W_S W_\sigma = \rho t$). The open parameters of the model are $\alpha > 0$ (the volatility of the volatility), $0 \leq \beta \leq 1$ (the elasticity) and ρ (the correlation coefficient).

A useful aspect for practitioners is that each model parameter can be assigned to a specific feature of the market observed implied volatility when it is represented against the strike prices, commonly known as a *volatility smile or skew*. The α parameter mainly affects the smile curvature. Exponent β is connected to the forward process distribution, being normal for $\beta = 0$ and log-normal for $\beta = 1$. In terms of the volatility smile, β also affects the curvature. Correlation parameter ρ rotates the implied volatility curve around the *at-the-money* point, obtaining "more pronounced smile" or "more pronounced skew" shapes.

2.1. SABR Monte Carlo simulation

In the first equation of the SABR model (1), the forward asset dynamics are defined as a constant elasticity of variance (CEV) process [16]. Based on the works of Schroder [17] and Islah [18], an analytic expression for the CDF of the SABR conditional process has been obtained. For some $S(0) > 0$, the conditional CDF of $S(t)$ with an absorbing boundary at $S(t) = 0$, and given the volatility, $\sigma(t)$, and the conditional time-integrated variance, $\int_0^t \sigma^2(s)\mathrm{d}s|\sigma(t)$, reads

$$
Pr\left(S(t) \leq X | S(0) > 0, \sigma(t), \int_0^t \sigma^2(s)\mathrm{d}s\right) = 1 - \chi^2(a; b, c),
\tag{2}
$$

where

$$
a = \frac{1}{\nu(t)}\left(\frac{S(0)^{1-\beta}}{(1-\beta)} + \frac{\rho}{\alpha}(\sigma(t) - \sigma(0))\right)^2, \qquad c = \frac{X^{2(1-\beta)}}{(1-\beta)^2\nu(t)},
$$

$$
b = 2 - \frac{1 - 2\beta - \rho^2(1-\beta)}{(1-\beta)(1-\rho^2)}, \qquad\qquad \nu(t) = (1-\rho^2)\int_0^t \sigma^2(s)\mathrm{d}s,
$$

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

It should be noted that this formula is *exact* for the case $\rho = 0$ and results in an *approximation* otherwise. A shifted process with an approximated initial condition is employed in the derivation for $\rho \neq 0$. Based on Eq. (2), an "exact" Monte Carlo simulation scheme for the SABR model can be defined based on inverting the conditional SABR CDF, when the conditional time-integrated variance and the volatility is already simulated. This forms the basis of the one time-step SABR

approach, where exact simulation is required for accuracy reasons when using only one large time-step (meaning no time discretization). Furthermore, it is well known that the convergence ratio of the Monte Carlo method is $1/\sqrt{n}$, with n the number of paths or scenarios.

According to Eq. (2), in order to apply an "exact" one time-step Monte Carlo simulation for the SABR dynamics, we need to perform the following steps (with the terminal time T):

- *Simulation of SABR's volatility.* From Eq. (1), we observe that the volatility process of the SABR model is governed by a log-normal distribution. As is well-known, the solution follows a *geometric Brownian motion*, i.e. the exact simulation of the volatility at terminal time, $\sigma(T)$, reads

$$\sigma(T) \sim \sigma(0) \exp(\alpha \hat{W}_\sigma(T) - \frac{1}{2}\alpha^2 T), \tag{3}$$

where $\hat{W}_\sigma(T)$ is an independent Brownian motion.

- *Simulation of SABR's time-integrated variance, conditional on the terminal value of the volatility, i.e.,* $\int_0^T \sigma^2(s)\mathrm{d}s | \sigma(T)$. Exact simulation of the time-integrated variance in the SABR model is not directly possible since its conditional distribution given the volatility is not known in closed-form. One possibility is to simulate it by a Monte Carlo method, but this implies the use of a nested simulation which can be very expensive or even unaffordable. Another possibility is to find an approximation for the conditional distribution of SABR's time-integrated variance given $\sigma(T)$. The integral $\int_0^T \sigma^2(s)\mathrm{d}s$ can also be approximated by some quadrature rule once $\sigma(0)$ and $\sigma(T)$ (or intermediate values of the volatility process) are available.

- *Simulation of SABR's forward asset process.* We can simulate the forward dynamics by inverting the CDF in Eq. (2). For this, the conditional SABR's time-integrated variance (besides the volatility) is required. Concerning the forward asset simulation, there is no analytic expression for the inverse SABR distribution so the inversion has to be calculated by means of some numerical approximation. Another possibility is to employ SDE discretization schemes, like Taylor-based schemes (Euler, log-Euler, Milstein or full-truncation) or more involved ones (low-bias or QE schemes). However, the use of such schemes gives rise to several issues that have to be managed, like the discretization error, negativity in the asset values and computational cost.

In Algorithm 1, a schematic representation of the complete one time-step Monte Carlo simulation procedure for the SABR model is presented. We include references to the sections of this paper where the respective parts of the SABR simulation are discussed.

Algorithm 1: SABR's one time-step Monte Carlo simulation.

Data: $S_0, \sigma_0, \alpha, \beta, \rho, T, n$.
for *Path* $p = 1 \ldots n$ **do**
   Simulation of $\hat{W}_\sigma(t) \sim \mathcal{N}(0, \sqrt{T})$.
   Simulation of $\sigma(T) | \sigma(0)$ (see Eq. (3)).
   Simulation of $\int_0^T \sigma^2(s)\mathrm{d}s | \sigma(T)$ (see Section 4.3).
   Simulation of $S(T) | S(0), \sigma(T), \int_0^T \sigma^2(s)\mathrm{d}s$ (see Section 4.4).

In the above steps of the SABR model exact simulation, the challenging part is the simulation of the time-integrated variance, $\int_0^T \sigma^2(s)\mathrm{d}s$, conditional on $\sigma(T)$ (and $\sigma(0)$). Sometimes moment-matching techniques can be used. Here, however, we propose a computationally efficient approximation, based on a copula multi-variate distribution, to simulate the conditional distribution of $\int_0^T \sigma^2(s)\mathrm{d}s$ given the volatility $\sigma(T)$. Simulation by means of a copula technique requires the CDF of the involved marginal distributions. Although the distribution of $\sigma(T)$ is known, the distribution of $\int_0^T \sigma^2(s)\mathrm{d}s$ needs to be approximated. We propose a method based on a Fourier technique, which was already employed for Asian options in [15]. In Section 3, this derivation is presented in detail.

Hereafter, for notational convenience, we will use $Y(T) := \int_0^T \sigma^2(s)\mathrm{d}s$.

3. CDF of SABR's time-integrated variance

In this section, we present a procedure to approximate the CDF of the time-integrated variance, $Y(T)$. Since we will work in the log-space, an approximation of the CDF of $\log Y(T)$, $F_{\log Y(T)}$, will be derived. First of all, we approximate $Y(T)$ by its discrete analog, i.e.

$$\int_0^T \sigma^2(s)\mathrm{d}s \approx \sum_{j=1}^{M} \sigma^2(t_j)\Delta t =: \hat{Y}(T), \tag{4}$$

where M is the number of discrete time points[1], i.e., $t_j = j\Delta t$, $j = 1, \ldots, M$ and $\Delta t = \frac{T}{M}$. $\hat{Y}(T)$ is subsequently transformed to the logarithmic domain, where we aim to find an approximation of $F_{\log \hat{Y}(T)}$, i.e.

$$F_{\log \hat{Y}(T)}(x) = \int_{-\infty}^{x} f_{\log \hat{Y}(T)}(y) \mathrm{d}y, \tag{5}$$

with $f_{\log \hat{Y}(T)}$ the *probability density function* (PDF) of $\log \hat{Y}(T)$. $f_{\log \hat{Y}(T)}$ is, in turn, found by approximating the associated characteristic function, $\phi_{\log \hat{Y}(T)}$, and applying a Fourier inversion procedure. The characteristic function and the desired PDF of $\log \hat{Y}(T)$ form a so-called Fourier pair. Based on the works in [15,19], we develop a recursive procedure to recover the characteristic function of $f_{\log \hat{Y}(T)}$. We start by defining the sequence,

$$R_j = \log \left(\frac{\sigma^2(t_j)}{\sigma^2(t_{j-1})} \right) = \log \left(\sigma^2(t_j) \right) - \log \left(\sigma^2(t_{j-1}) \right), \tag{6}$$

where R_j is the logarithmic increment of $\sigma^2(t)$ between t_j and t_{j-1}, $j = 1, \ldots, M$. As mentioned, in the SABR model context the volatility process follows log-normal dynamics. Since 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, as a normal increment, is well known and reads, $\forall u, j$,

$$\phi_{R_j}(u) = \phi_R(u) = \exp(-iu\alpha^2 \Delta t - 2u^2\alpha^2 \Delta t), \tag{7}$$

with α as in (1) and i the imaginary unit.

By Eq. (6), we can write $\sigma^2(t_j)$ as

$$\sigma^2(t_j) = \sigma^2(t_0) \exp(R_1 + R_2 + \cdots + R_j), \tag{8}$$

because

$$\sigma^2(t_j) = \sigma^2(t_0) \exp \left(\log \left(\frac{\sigma^2(t_1)}{\sigma^2(t_0)} \right) + \log \left(\frac{\sigma^2(t_2)}{\sigma^2(t_1)} \right) + \cdots + \log \left(\frac{\sigma^2(t_j)}{\sigma^2(t_{j-1})} \right) \right)$$
$$= \sigma^2(t_0) \exp \left(\log \left(\frac{\sigma^2(t_j)}{\sigma^2(t_0)} \right) \right).$$

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

$$\begin{aligned} Y_1 &= R_M, \\ Y_j &= R_{M+1-j} + Z_{j-1}, \quad j = 2, \ldots, M. \end{aligned} \tag{9}$$

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

Using the definitions in Eqs. (8) and (9), the approximated time-integrated variance can be written as

$$\hat{Y}(T) = \sum_{i=1}^{M} \sigma^2(t_i) \Delta t = \Delta t \sigma_0^2 \exp(Y_M). \tag{10}$$

From Eq. (10), we determine $\phi_{\log \hat{Y}(T)}(u)$, as follows

$$\begin{aligned} \phi_{\log \hat{Y}(T)}(u) &= \mathbb{E}[\exp(iu \log \hat{Y}(T))] \\ &= \mathbb{E}\left[\exp \left(iu \log \left(\Delta t \sigma_0^2 \right) + iuY_M \right) \right] \\ &= \exp \left(iu \log \left(\Delta t \sigma_0^2 \right) \right) \phi_{Y_M}(u). \end{aligned} \tag{11}$$

We have reduced the computation of $\phi_{\log \hat{Y}(T)}$ to the computation of ϕ_{Y_M}. As Y_M is defined recursively, its characteristic function, ϕ_{Y_M}, can be obtained recursively as well. In Section 3.1, a procedure for obtaining an approximation for $\phi_{Y_M}(u)$, i.e. $\hat{\phi}_{Y_M}(u)$, is described. According to the definition of the (backward) sequence Y_j in Eq. (9), the required initial and recursive characteristic functions are given by the following expressions,

$$\begin{aligned} \phi_{Y_1}(u) &= \phi_{R_M}(u) = \exp(-iu\alpha^2 \Delta t - 2u^2\alpha^2 \Delta t), \\ \phi_{Y_j}(u) &= \phi_{R_{M+1-j}}(u) \phi_{Z_{j-1}}(u) = \phi_R(u)\phi_{Z_{j-1}}(u), \end{aligned} \tag{12}$$

where Eq. (7) is used in both expressions and the independence of R_{M+1-j} and Z_{j-1} is also employed.

After the computation of $\phi_{Y_M}(u)$, and thus of $\phi_{\log \hat{Y}(T)}(u)$, the next step is to compute the PDF of $\log \hat{Y}(T)$ by means of the COS method [20]. For that, we need to determine the support of $\log \hat{Y}(T)$, which is denoted by interval $[\hat{a}, \hat{b}]$. It can be

---

[1] These time points are not to be confused with the Monte Carlo time steps. We will have only one Monte Carlo time-step. M is the number of points for the discrete approximation of $Y(T)$.

calculated by employing the relation

$$\log(\hat{Y}(T)) = \log\left(\Delta t \sigma_0^2 \exp(Y_M)\right) = \log\left(\Delta t \sigma_0^2\right) + Y_M.$$

An alternative to the COS method is the so-called SWIFT method [21], based on Shannon wavelets. This technique is efficient especially when long time horizons are considered. For the present application, however, the COS method appears superior in terms of computational time. The use of the one-step SABR simulation will be restricted to exercise times up to two years (details in Section 4.2).

We recover $f_{\log \hat{Y}(T)}$ by employing the COS expression, as follows

$$f_{\log \hat{Y}(T)}(x) \approx \frac{2}{\hat{b} - \hat{a}} \sum_{k=0}^{N-1\prime} C_k \cos\left((x - \hat{a})\frac{k\pi}{\hat{b} - \hat{a}}\right), \tag{13}$$

with

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

and

$$\phi_{\log \hat{Y}(T)}\left(\frac{k\pi}{\hat{b} - \hat{a}}\right) = \exp\left(i\frac{k\pi}{\hat{b} - \hat{a}}\log\left(\Delta t \sigma_0^2\right)\right)\phi_{Y_M}\left(\frac{k\pi}{\hat{b} - \hat{a}}\right)$$

$$\approx \exp\left(i\frac{k\pi}{\hat{b} - \hat{a}}\log\left(\Delta t \sigma_0^2\right)\right)\hat{\phi}_{Y_M}\left(\frac{k\pi}{\hat{b} - \hat{a}}\right),$$

where N is the number of COS terms and the prime $\prime$ and $\Re$ symbols in (13) indicate division of the first term in the summation by two and taking the real part of the complex-valued expressions in the brackets, respectively.

The CDF $F_{\log \hat{Y}(T)}$, as in Eq. (5), is obtained by integrating the corresponding approximated PDF from Eq. (13), as follows

$$F_{\log \hat{Y}(T)}(x) = \int_{-\infty}^{x} f_{\log \hat{Y}(T)}(y)\mathrm{d}y$$

$$\approx \int_{\hat{a}}^{x} \frac{2}{\hat{b} - \hat{a}} \sum_{k=0}^{N-1\prime} C_k \cos\left((y - \hat{a})\frac{k\pi}{\hat{b} - \hat{a}}\right)\mathrm{d}y. \tag{14}$$

3.1. Characteristic function of Y_M

In the previous section, we defined a procedure to compute an approximation of the CDF of $Y(T)$. This computation relies on the efficient calculation of the characteristic function of Y_M. In this section, we explain the recursive procedure for approximating $\phi_{Y_M}(u)$ based on the initial characteristic function, $\phi_{Y_1}(u)$, the characteristic function of the recursive process, $\phi_R(u)$, and the equality $\phi_{Y_j}(u) = \phi_R(u)\phi_{Z_{j-1}}(u)$ (see Eqs. (6) and (12)). Note that this iterative computation needs to be done only once. By definition, the characteristic function of Z_{j-1} reads

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

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 restrict the integration range to $[a, b]$. The calculation of integration boundaries a and b follows the cumulant-based approach as described in [15,20]. After truncation of the integral, we have

$$\hat{\phi}_{Z_{j-1}}(u) = \int_{a}^{b} (\exp(x) + 1)^{iu} f_{Y_{j-1}}(x)\mathrm{d}x. \tag{15}$$

Now we can apply a cosine series expansion to $f_{Y_{j-1}}$, so that $\hat{\phi}_{Z_{j-1}}(u)$ becomes

$$\hat{\phi}_{Z_{j-1}}(u) \approx \frac{2}{b - a} \sum_{l=0}^{N-1\prime} B_l \int_{a}^{b} (\exp(x) + 1)^{iu} \cos\left((x - a)\frac{l\pi}{b - a}\right)\mathrm{d}x,$$

$$B_l = \Re\left(\hat{\phi}_{Y_{j-1}}\left(\frac{l\pi}{b - a}\right)\exp\left(-ia\frac{l\pi}{b - a}\right)\right), \tag{16}$$

with N the number of cosine expansion elements, $\hat{\phi}_{Y_1}(u) = \phi_R(u)$ and $\hat{\phi}_{Y_j}(u) = \phi_R(u)\hat{\phi}_{Z_{j-1}}(u)$.

We wish to compute $\hat{\phi}_{Y_M}(\frac{k\pi}{\hat{b} - \hat{a}})$, for $k = 0, \ldots, N-1$. Considering Eq. (16), we rewrite $\hat{\phi}_{Y_j}(u) = \phi_R(u)\hat{\phi}_{Z_{j-1}}(u)$ in matrix–vector form, as follows

$$\Phi_{j-1} = \mathcal{M}A_{j-1}, \tag{17}$$

where

$$
\Phi_{j-1} = \left[\Phi_{j-1}(k)\right]_{k=0}^{N-1}, \quad \Phi_{j-1}(k) = \hat{\phi}_{Z_{j-1}}(u_k),
$$

$$
\mathcal{M} = [\mathcal{M}(k,l)]_{k=0}^{N-1}, \quad \mathcal{M}(k,l) = \int_a^b (\exp(x)+1)^{iu_k} \cos((x-a)u_l)\mathrm{d}x, \tag{18}
$$

$$
A_j = \frac{2}{b-a}\left[A_j(l)\right]_{l=0}^{N-1}, \quad A_j(l) = \Re\left(\hat{\phi}_{Y_{j-1}}(u_l)\exp(-iau_l)\right),
$$

with

$$
u_z = \frac{z\pi}{b-a}. \tag{19}
$$

By the recursion procedure in Eq. (17), we obtain the approximation $\hat{\phi}_{Y_M}$. Before using this approximation, we must compute matrix $\mathcal{M}$ in Eq. (18) highly efficiently. This matrix $\mathcal{M}$ has to be computed only once. Its elements are given by

$$
I_{\mathcal{M}} := \int_a^b (\exp(x)+1)^{iu_k} \cos((x-a)u_l)\mathrm{d}x. \tag{20}
$$

The efficient computation of the integral in Eq. (20) is a key aspect for the performance of the whole procedure. 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. We discuss the numerical treatment in Section 3.2.

3.2. Treatment of integral $I_{\mathcal{M}}$

The efficient calculation of the integral in Eq. (20) is crucial for an efficient computation of $\hat{\phi}_{Y_M}(u)$ (and thus of $F_{\log \hat{Y}(T)}$). In this section, we propose several ways to deal with $I_{\mathcal{M}}$. In Section 3.3, an analysis in terms of accuracy and efficiency for each of the approximations is carried out.

Analytic solution based on $_2F_1$. Integral $I_{\mathcal{M}}$ in (20) can be rewritten as

$$
I_{\mathcal{M}} = \int_a^b e^{iu_k g(x)} \cos((x-a)u_l)\mathrm{d}x, \quad g(x) = \log((\exp(x)+1)), \tag{21}
$$

with u_k and u_l as in (19).

By the Euler formula, integral $I_{\mathcal{M}}$ becomes

$$
I_{\mathcal{M}} = \frac{1}{2}\left(I^+ + I^-\right), \quad I^{\pm} = \int_a^b \exp(i(u_k g(x) \pm (x-a)u_l))\mathrm{d}x.
$$

Integrals I^+ and I^- can be solved analytically in terms of the hypergeometric function $_2F_1$. The solution reads

$$
I^{\pm} = \mp \frac{ie^{\pm iu_l(x-a)}}{u_l} \, _2F_1(-iu_k, \pm iu_l; 1 \pm iu_l; -\exp(x)).
$$

Unfortunately, multiple evaluations of the hypergeometric function make the computation rather time-consuming.

Clenshaw–Curtis quadrature. Regarding numerical approximation methods, quadrature rules can be applied to solve integral $I_{\mathcal{M}}$. In our case, the elements in Eq. (20) can, for example, be approximated numerically by the Clenshaw–Curtis quadrature, i.e.,

$$
\int_a^b (\exp(x)+1)^{iu_k} \cos((x-a)u_l)\mathrm{d}x \approx (\mathcal{D}^T v)^T w, \tag{22}
$$

where the matrix element $\mathcal{D}(k,m)$ and the vectors v and w can be computed by the expressions

$$
\mathcal{D}(k,m) = \frac{2}{n_q} \cos\left(\frac{(m-1)(k-1)\pi}{n_q/2}\right) \cdot \begin{cases} 1/2 & \text{if } m = \{1, n_q/2+1\}, \\ 1 & \text{otherwise}, \end{cases}
$$

$$
v := \left(1, \frac{2}{1-4}, \frac{2}{1-16}, \ldots, \frac{2}{1-(n_q-2)^2}, \frac{2}{1-n_q^2}\right)^T,
$$

$$
w_m := h\left(\cos\left(\frac{(m-1)\pi}{n_q}\right)\right) + h\left(-\cos\left(\frac{(m-1)\pi}{n_q}\right)\right),
$$

$$
h(x) = \frac{b-a}{2}\cos\left(\left(\frac{b-a}{2}x + \frac{a+b}{2} - a\right)u_l\right)\left(\exp\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) + 1\right)^{iu_k},
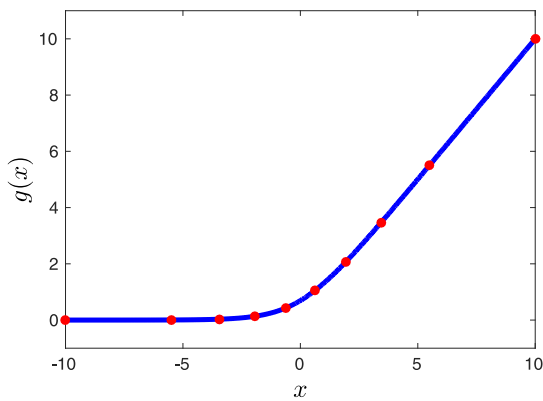$$

with $k, m = 1, \ldots, (n_q/2+1)$ and with n_q the number of terms in the quadrature. This was the approach taken in [15], with computational complexity $O(n_q N^2)$.
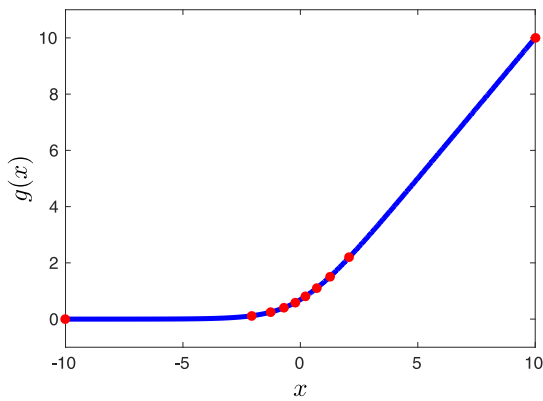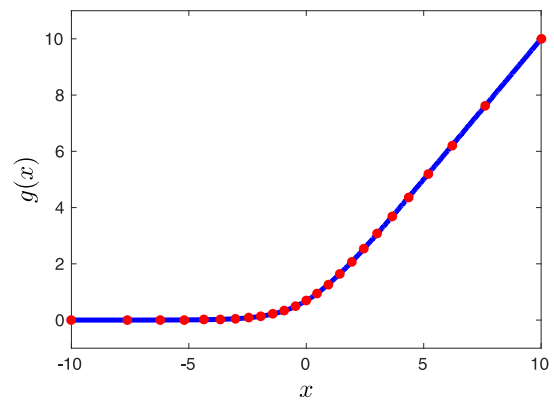
(a) $L = 10$ and $s = 1$
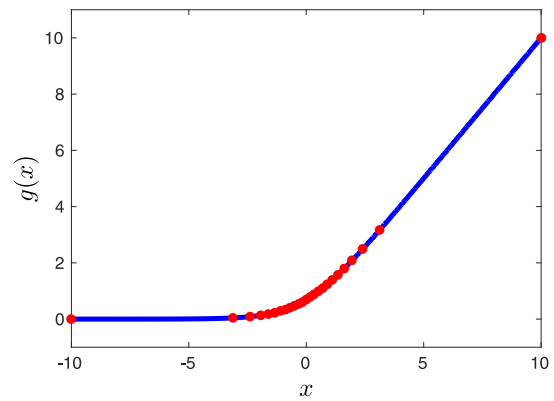
(b) $L = 25$ and $s = 1$

(c) $L = 10$ and $s = 3$

(d) $L = 25$ and $s = 3$

Fig. 1. Optimal integration points. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Piecewise linear approximation. In a second numerical approximation, we start with $I_\mathcal{M}$ in (21) and notice that although function $g(x)$ (the blue curve in Fig. 1) is not linear, it is smooth and monotonic. Hence, the numerical technique proposed is to define sub-intervals of the integration range $[a, b]$ in which $g(x)$ is constant or almost linear, i.e.,

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

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

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

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

$$\hat{I}_\mathcal{M} = \sum_{j=1}^{L} \exp(iu_k c_{1,j}) \underbrace{\int_{a_j}^{b_j} \exp(iu_k c_{2,j}x) \cos((x - a)u_l) dx}_{I_L}. \tag{23}$$

In each sub-interval $[a_j, b_j]$, we need to determine a simple integral. I_L in Eq. (23) is known analytically and is given by

$$I_L = \frac{\exp(ia_j c_{2,j} u_k)\big(-ic_{2,j} u_k \cos((a - a_j)u_l) + u_l \sin((a - a_j)u_l)\big)}{(u_l - c_{2,j} u_k)(u_l + c_{2,j} u_k)}$$

$$+ \frac{i \exp(ib_j c_{2,j} u_k)\big(c_{2,j} u_k \cos((a - b_j)u_l) + iu_l \sin((a - b_j)u_l)\big)}{(u_l - c_{2,j} u_k)(u_l + c_{2,j} u_k)}.$$

Table 1
MSE vs. L for $\hat{I}_{\mathcal{M}}$ in Eq. (23).

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

The optimal integration points a_j and b_j in (23) are found by differentiation of $g(x)$ w.r.t. x, i.e.

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

giving a relation between the derivative of $g(x)$ and the logistic distribution. This representation indicates that $G(x)$ is the CDF of the logistic distribution with a location parameter $\mu = 0$ and a scale parameter $s = 1$. 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 as follows:

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

The algorithm above ensures that integration points are optimally distributed. In Fig. 1a and b, we present the optimal integration points for $L = 10$ and $L = 25$, respectively. The points are well-positioned in the region of interest, as expected.

However, for our specific problem, the integration interval can be rather large. As the integration points are densely concentrated in 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 balanced distribution of the integration points can be achieved by introducing a scale parameter, $s \neq 1$, which implies that the integration points correspond to a CDF with increased variance. In Fig. 1c and d, the distributions of the points for $s = 3$ are shown. In the case of $s \neq 1$, the expression for the quantiles reads,

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

The impact of the optimal point redistribution can be seen in Table 1. The mean squared error (MSE), in absolute value, of the piecewise linear approximation is presented for a fixed interval $[-10, 30]$ and a fixed number of elements in the cosine expansion, $N = 150$, considering different numbers of sub-intervals and different integration point locations (equidistant, optimal with $s = 1$ and optimal with $s = 3$). We observe that, for the problem at hand, the optimal distribution with $s > 1$ gives better accuracy.

3.3. Error and performance analysis

In the sections above, we have proposed a method to compute the approximated CDF for SABR's time-integrated variance. In this section, we perform an error analysis, indicating how to reduce or bound the errors. As usual, there is a relation between the error (or accuracy) and the performance and we cannot discuss them separately. Therefore, we will take into account the computational effort here. The sources of error in our approximation include ϵ_D, the discretization of $Y(T)$ in Eq. (4), ϵ_T, the truncation in Eqs. (14) and (15), ϵ_C, due to the cosine expansion in Eqs. (13) and (16) and ϵ_M, the numerical computation of $I_{\mathcal{M}}$ in Eq. (20).

Error ϵ_D. This error occurs because the SABR's time-integrated variance is approximated by its discrete equivalent, i.e. $Y(T) \approx \hat{Y}(T)$. Note that the process $Y(t) = \int_0^t \sigma^2(s)ds$ is the solution of the SDE

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

which, by employing the same time discretization as for $\hat{Y}(T)$ in Eq. (4), can be written as

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

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 our discrete approximation $\hat{Y}(T)$ as follows

$$\sum_{j=1}^{M} \sigma^2(t_{j-1})\Delta t = \sum_{j=1}^{M} \left(\bar{Y}(t_j) - \bar{Y}(t_{j-1})\right) = \bar{Y}(t_M) - \bar{Y}(0) = \hat{Y}(T).$$

(a) Error ϵ_D (b) $F_{\log \hat{Y}(T)}(x)$
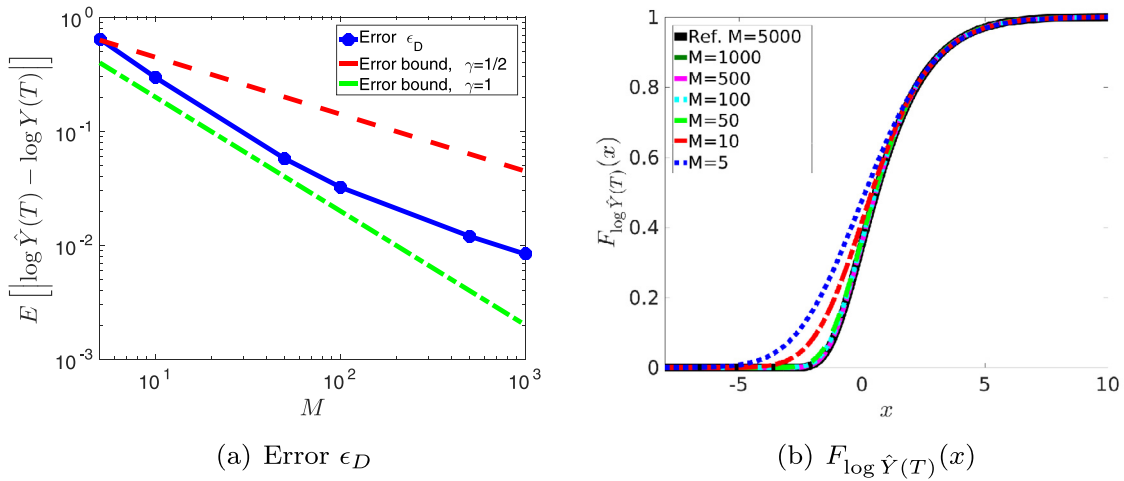
Fig. 2. Convergence analysis in terms of the number of discrete points, M.

The error (or convergence) of the Euler–Maruyama discretization scheme has been intensively studied in the literature (see [22], for example). Two errors can be considered when a stochastic process is discretized: *strong* and *weak*. It was proved by Kloeden and Platen in [22] that, under some conditions of regularity, the Euler–Maruyama scheme has strong convergence with order $\gamma = \frac{1}{2}$ and weak convergence with order $\gamma = 1$. In our particular case, we focus on the strong error between $Y(T)$ and $\hat{Y}(T)$. Then, the error ϵ_D can be computed and bounded by employing the usual strong convergence expression as

$$\epsilon_D := \mathbb{E}\big[\big|\hat{Y}(T) - Y(T)\big|\big] \leq C\Big(\frac{T}{M}\Big)^{\gamma},$$

for some constant $C > 0$.

In our actual experiments, we consider $\log \hat{Y}(T)$ and $\log Y(T)$. This will reduce the error further. In order to numerically show the order of convergence of our approximation, in Fig. 2a the strong error[2] of $\log \hat{Y}(T)$ with respect to the number of discrete time points, M, is depicted. Additionally, an "upper" ($C = 1$ and $\gamma = \frac{1}{2}$) and "lower" ($C = 1$ and $\gamma = 1$) bounds are presented as a reference. In order to further test the convergence, the resulting CDFs, $F_{\log \hat{Y}(T)}$, are presented in Fig. 2b, again varying the number of discrete time points, M. Both experiments perform as expected.

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

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

where H corresponds to $\log \hat{Y}(T)$ and Y_j, according to Eqs. (14) and (15), respectively. By a sufficiently large integration range $[a, b]$, this error can be reduced and does not dominate the overall error.

Error ϵ_C. The convergence due to the number of terms N in the cosine series expansion was derived in [20]. For any $f(y|x) \in \mathbf{C}^{\infty}[a, b]$, 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 PDF of the time-integrated variance is a very smooth function, ϵ_C will decay exponentially with respect to N.

Error ϵ_M. We have described two different numerical methods to compute integral $I_{\mathcal{M}}$: Clenshaw–Curtis quadrature and the piecewise linear approximation.

Regarding the Clenshaw–Curtis quadrature in Eq. (22) as stated in [15], the error can be bounded by $O((2n_q)^{-m}/m)$ for an m-times differentiable integrand, with n_q the number of quadrature terms. When m is bounded, we achieve algebraic convergence while, otherwise, the error converges exponentially with respect to n_q. In Eq. (20), the integrand of $I_{\mathcal{M}}$ belongs to $\mathbf{C}^{\infty}[a, b]$ thus, by Clenshaw–Curtis quadrature, error ϵ_M can be reduced with exponential convergence in n_q.

---

[2] As $\log Y(T)$ is unknown, the reference value is computed by using a very fine discretization, i.e. $M = 5000$.

When the approximation in Eq. (23) is employed, we observe a reduction of the error by a factor four, when going from $L = 25$ to $L = 50$ integration points in Table 1. However, with $L = 100$ the error reduces more drastically due to the choice of optimal integration points.

3.3.1. Performance analysis

We perform an efficiency analysis considering the terms that control the evaluated errors. Error ϵ_T is not taken into account here, as we always employ a sufficiently large integration interval.

Error ϵ_D can be reduced by increasing M in Eq. (4). We find that the value of M hardly affects the computational cost, so it can be increased without a significant impact on the performance.

Error ϵ_C is governed by the number of series expansion terms, N. As ϵ_C exhibits an exponential decay w.r.t. N, a relatively small value ($N \approx 100$) already results in highly accurate approximations. Larger N values have a great impact on the performance because $I_{\mathcal{M}}$ is calculated $N \times N$ times. As this calculation dominates the computational cost, the execution time increases by $O(N^2)$.

Error ϵ_M can be controlled by the number of quadrature terms, n_q, in the case of the Clenshaw–Curtis method or by the number of sub-intervals, L, when using the piecewise linear approximation. Again, the impact can be significant because $I_{\mathcal{M}}$ is computed many times. To achieve a similar order of accuracy ($\sim 10^{-7}$) in a generic interval $[a, b]$, the piecewise linear approximation is approximately three times faster than the Clenshaw–Curtis quadrature. Therefore, we will use the piecewise linear approximation in the following.

4. Simulation of $Y(T)|\sigma(T)$: copula approach

The approximated CDF of the marginal distribution of $Y(T)$ (in the log-space, see Section 3) is employed to simulate $Y(T)|\sigma(T)$ by means of a *copula*. A d-dimensional copula is a joint distribution function on a unit hyper-cube $[0, 1]^d$ with d marginal distributions. A distribution with a copula contains essentially the same information as the joint distribution, like the marginal properties and the dependence structure, but this information can be decoupled. The importance of copulas is represented by two parts of the Sklar theorem, introduced by Sklar [23]. The first part states that a copula can be derived from any joint distribution function, and the second part states that any copula can be combined with any set of marginal distributions to result in a multivariate distribution function. More formally, Sklar's theorem reads

- Let F be a joint distribution function and F_j, $j = 1, \ldots, d$, be the marginal distributions. Then there exists a copula C: $[0, 1]^d \to [0, 1]$ such that

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)),$$

  for all $x_j \in [-\infty, \infty]$. Moreover, if the margins are continuous, then C is unique; otherwise C is uniquely determined on $Ran(F_1) \times \cdots \times Ran(F_d)$, where $Ran(F_j) = F_j([-\infty, \infty])$ is the range of F_j.
- The converse is also true. That is, if C is a copula and $F_1, \ldots, F_d$ are univariate distribution functions, then the multivariate function defined by the preceding equation is a joint distribution function with marginal distributions F_j, $j = 1, \ldots, d$.

4.1. Families of copulas

Copulas can be classified into so-called *fundamental*, *implicit* and *explicit* copulas. Fundamental copulas are copulas that represent either perfect positive dependence, independence or perfect negative dependence. Implicit copulas are copulas extracted from well-known multivariate distributions, like normal, Gaussian or Student **t** distributions, that do not have closed-form expressions. Explicit or Archimedean copulas have simple closed-form expressions. The Clayton, Frank and Gumbel copulas are the most commonly employed ones.

Here, we briefly describe these copulas, considering $F_1 \ldots, F_d \in [0, 1]^d$ as the marginal distributions:

- *Gaussian*: given a correlation matrix, $\mathcal{R} \in \mathbb{R}^{d \times d}$, the Gaussian copula with parameter $\mathcal{R}$ is defined by

$$\mathcal{C}_{\mathcal{R}}(F_1 \ldots, F_d) = \Phi_{\mathcal{R}}(\Phi^{-1}(F_1), \ldots, \Phi^{-1}(F_d)),$$

  where Φ is the distribution function of a standard normal random variable and $\Phi_{\mathcal{R}}$ is the d-variate standard normal distribution with zero mean vector and covariance matrix $\mathcal{R}$.
- *Student* **t**: given correlation matrix $\mathcal{R} \in \mathbb{R}^{d \times d}$ and the degrees of freedom parameter $\nu \in (0, \infty)$, the **t** copula is defined by

$$\mathcal{C}_{\mathcal{R}, \nu}(F_1 \ldots, F_d) = \mathbf{t}_{\mathcal{R}, \nu}(\mathbf{t}_{\nu}^{-1}(F_1), \ldots, \mathbf{t}^{-1}(F_d)),$$

  where $\mathbf{t}_{\nu}$ and $\mathbf{t}_{\mathcal{R}, \nu}$ are univariate and the d-variate Student **t** distributions, respectively.
- *Archimedean*: is governed by the representation

$$\mathcal{C}_{\theta}(F_1 \ldots, F_d) = \psi_{\theta}^{-1}(\psi_{\theta}(F_1) + \cdots + \psi_{\theta}(F_d)),$$

  where $\psi_{\theta}: [0, 1] \to [0, \infty]$ is the so-called generator function which is supposed to be a continuous, strictly decreasing and convex function such that $\psi_{\theta}(1) = 0$, $\psi_{\theta}(0) = \infty$ and its inverse ψ_{θ}^{-1} is monotonic on $[0, \infty)$. A variety of Archimedean copulas can be defined, like the following well-known copulas.

The *Clayton* copula, for which the generator function is given by $\psi_\theta(u) = \theta^{-1}(u^{-\theta} - 1)$, $\theta > 0$, reads

$$C_\theta(F_1 \ldots, F_d) = \left(\sum_{i=1}^{d} F_i^{-\theta} - d + 1 \right)^{-1/\theta},$$ (24)

The *Frank* copula, with generator function given by $\psi_\theta(u) = -\log\left(\frac{\exp(-\theta u) - 1}{\exp(-\theta) - 1} \right)$, reads

$$C_\theta(F_1 \ldots, F_d) = -\frac{1}{\theta} \log \left(1 + \frac{\prod_{i=1}^{d}(\exp(-\theta F_i) - 1)}{(\exp(-\theta) - 1)^{d-1}} \right),$$ (25)

being $\theta \in \mathbb{R} \backslash \{0\}$ for $d = 2$ and $\theta > 0$ for $d \geq 3$.

The *Gumbel* copula, with generator function $\psi_\theta(u) = (-\log(u))^\theta$, $\theta > 0$, reads

$$C_\theta(F_1 \ldots, F_d) = \exp \left(- \left(\sum_{i=1}^{d} (-\log(F_i))^\theta \right)^{1/\theta} \right),$$ (26)

In order to calibrate parameter θ in (24), (25) or (26) a relation between θ and the rank correlation coefficient Kendall's τ can be employed. This relation is available in closed-form for the Clayton copula, i.e. $\theta = 2\tau/(1 - \tau)$, and the Gumbel copula, i.e. $\theta = 1/(1 - \tau)$. There is no analytic expression of θ for the Frank copula.

To apply any of these copulas, an approximation for the correlation between the involved distributions needs to be determined. Here, Pearson's correlation coefficient, $\mathcal{P}$, is chosen since it is employed directly in the Gaussian and Student **t** copulas and a relation with Kendall's τ exits

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

In Section 4.2, an approximation for Pearson's coefficient between $\log Y(T)$ and $\log \sigma(T)$ of the SABR dynamics is derived.

4.2. Correlation: $\log Y(T)$ and $\log \sigma(T)$

For the problem at hand, we need to derive an expression for Pearson's correlation coefficient $\mathcal{P}_{\log Y(T), \log \sigma(T)}$. By definition, we have

$$\mathcal{P}_{\log Y(T), \log \sigma(T)} = \text{corr} \left[\log \int_0^T \sigma^2(s)ds, \log \sigma(T) \right] = \frac{\text{cov} \left[\log \int_0^T \sigma^2(s)ds, \log \sigma(T) \right]}{\sqrt{\text{var} \left[\log \int_0^T \sigma^2(s)ds \right] \text{var}[\log \sigma(T)]}}.$$

We employ the following approximation

$$\log \int_0^T \sigma^2(s)ds \approx \int_0^T \log \sigma^2(s)ds = 2 \int_0^T \log \sigma(s)ds.$$

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.

$$\log \int_0^T \sigma^2(s)ds \geq \int_0^T \log \sigma^2(s)ds.$$

We will numerically show that, under our model settings with the option contract's time to maturity of less than two years, this is a highly satisfactory approximation. The correlation coefficient can now be computed as

$$\mathcal{P}_{\log Y(T), \log \sigma(T)} \approx \frac{\text{cov} \left[\int_0^T \log \sigma(s)ds, \log \sigma(T) \right]}{\sqrt{\text{var} \left[\int_0^T \log \sigma(s)ds \right] \text{var}[\log \sigma(T)]}}.$$

The quantity $\text{var}[\log \sigma(T)]$ is known by Eq. (3), and we derive expressions the other two quantities. Starting with the covariance, we have

$$\text{cov} \left[\int_0^T \log \sigma(s)ds, \log \sigma(T) \right] = \mathbb{E} \left[\left(\int_0^T \log \sigma(s)ds \right) \log \sigma(T) \right] - \mathbb{E} \left[\int_0^T \log \sigma(s)ds \right] \mathbb{E}[\log \sigma(T)].$$

From Eq. (3), we find

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

and

$$\int_0^T \log \sigma(s)\mathrm{d}s = T \log \sigma_0 - \frac{1}{4}\alpha^2 T^2 + \alpha \int_0^T W(s)\mathrm{d}s =: \gamma(T) + \alpha \int_0^T (T-s)\mathrm{d}W(s).$$

Based on these equations, with

$$\mathbb{E}[\log \sigma(T)] = \eta(T), \quad \mathbb{E}\left[\int_0^T \log \sigma(s)\mathrm{d}s\right] = \gamma(T),$$

we obtain the following expressions for the required expectation,

$$\mathbb{E}\left[\left(\int_0^T \log \sigma(s)\mathrm{d}s\right)\log \sigma(T)\right] = \mathbb{E}\left[\left(\gamma(T) + \alpha \int_0^T (T-s)\mathrm{d}W(s)\right)(\eta(T) + \alpha W(T))\right]$$

$$= \gamma(T)\eta(T) + \alpha^2 \mathbb{E}\left[\int_0^T \mathrm{d}W(s)\int_0^T (T-s)\mathrm{d}W(s)\right]$$

$$= \gamma(T)\eta(T) + \alpha^2 \mathbb{E}\left[\int_0^T (T-s)\mathrm{d}s\right] = \gamma(T)\eta(T) + \frac{1}{2}\alpha^2 T^2.$$

In order to determine the variance of $\int_0^T \log \sigma(s)\mathrm{d}s$, we compute

$$\mathbb{E}\left[\left(\int_0^T \log \sigma(s)\mathrm{d}s\right)^2\right] = \mathbb{E}\left[\left(\gamma(T) + \alpha \int_0^T (T-s)\mathrm{d}W(s)\right)^2\right]$$

$$= \gamma^2(T) + \alpha^2 \mathbb{E}\left[\int_0^T (T-s)^2\mathrm{d}s\right]$$

$$= \gamma^2(T) + \frac{1}{3}\alpha^2 T^3,$$

and the variance reads

$$\mathrm{var}\left[\int_0^T \log \sigma(s)\mathrm{d}s\right] = \mathbb{E}\left[\left(\int_0^T \log \sigma(s)\mathrm{d}s\right)^2\right] - \left(\mathbb{E}\left[\int_0^T \log \sigma(s)\mathrm{d}s\right]\right)^2 = \frac{1}{3}\alpha^2 T^3.$$

Pearson's correlation coefficient is then obtained as

$$\mathcal{P}_{\log Y(T),\log \sigma(T)} \approx \frac{\mathbb{E}\left[\left(\int_0^T \log \sigma(s)\mathrm{d}s\right)\log \sigma(T)\right] - \mathbb{E}\left[\int_0^T \log \sigma(s)\mathrm{d}s\right]\mathbb{E}[\log \sigma(T)]}{\sqrt{\mathrm{var}\left[\int_0^T \log \sigma(s)\mathrm{d}s\right]\mathrm{var}[\log \sigma(T)]}}$$

$$= \frac{\gamma(T)\eta(T) + \frac{1}{2}\alpha^2 T^2 - \gamma(T)\eta(T)}{\sqrt{\left(\frac{1}{3}\alpha^2 T^3\right)\left(\alpha^2 T\right)}} = \frac{\frac{1}{2}\alpha^2 T^2}{\sqrt{\frac{1}{3}\alpha^4 T^4}} = \frac{\sqrt{3}}{2}.$$

The approximation obtained is a *constant value*. We will show numerically that, for the problems at hand, this appears to be a very reasonable approximation. The correlation between $\log Y(T)$ and $\log \sigma(T)$ is affected by the maturity time T, and by the *volatility-of-volatility* parameter α. In Fig. 3, the empirical and approximated (red grid) correlations are depicted for typical values of T and α. Because we focus on short maturity options, we restrict $T \in [0, 2]$ and $\alpha \in (0, 1]$. The experiment shows that, only in rather extreme cases ($\alpha > 0.7$), the differences between the empirical and approximated correlation increase, but remain within five basis points. The approximation is therefore sufficiently accurate for our purposes, since a small error in the correlation (of less than ten basis points) does not affect the performed simulation significantly.

4.3. Sampling Y(T)|σ(T)

The following steps describe the complete procedure for the simulation of $Y(T)$ given $\sigma(T)$ by using a bivariate copula approach:

1. Determine $F_{\log \sigma(T)}$ (analytically) and $F_{\log \hat{Y}(T)}$ via Eq. (14).
2. Define the bivariate copula distribution by employing the marginal distributions and the correlation approximation (Section 4.2).
3. Generate correlated uniform samples, $U_{\log \sigma(T)}$ and $U_{\log \hat{Y}(T)}$ by means of the bivariate copula distribution.
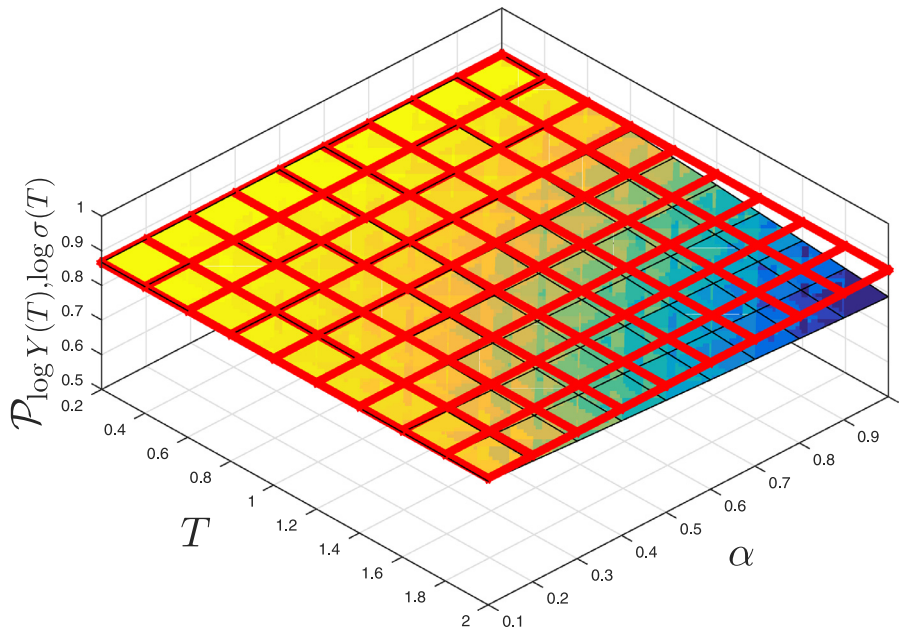
Fig. 3. Pearson's correlation coefficient: empirical (surface) vs. approximation (red grid). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. From $U_{\log \sigma(T)}$ and $U_{\log \hat{Y}(T)}$ invert the original marginals, $F_{\log \sigma(T)}$ and $F_{\log \hat{Y}(T)}$. The resulting quantiles should be correlated under the chosen correlation coefficient.
5. Finally, the samples of $Y(T)|\sigma(T)$ are obtained by taking exponentials.

The inversion procedure in the fourth step is straightforward in the case of $\sigma(T)$. The case of $\hat{Y}(T)$ is more involved, and we propose a direct inversion procedure based on linear interpolation. The insight is that, by rotating CDF $F_{\log \hat{Y}(T)}$, we can interpolate over probabilities instead of x-values. This is possible since $F_{\log \hat{Y}(T)}$ is a smoothly increasing function. The interpolation polynomial provides the quantiles of the original distribution, $F_{\log \hat{Y}(T)}$, from the probabilities $U_{\log \hat{Y}(T)}$. This inversion procedure is very fast (at almost no cost) and sufficiently accurate.

4.4. Simulation of S(T) given S(0), σ(T) and $\int_0^T \sigma^2(s)ds$

We complete the one-step SABR simulation by the conditional sampling of $S(T)$. The most commonly used techniques can be classified in two categories: direct inversion of SABR's CDF given in Eq. (2) 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. In [24], Chen et al. 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 $S(0)$, $\sigma(T)$ and $\int_0^T \sigma^2(s)ds$ enables analytic expressions.

4.4.1. Case $\beta = 0$
For $\beta = 0$, it is easily seen from Eq. (1) that the asset process (in integral form) becomes

$$S(T) = S(0) + \rho \int_0^T \sigma(s)d\hat{W}_\sigma(s) + \sqrt{1 - \rho^2} \int_0^T \sigma(s)d\hat{W}_S(s).$$

where $d\hat{W}_S(t)$ and $d\hat{W}_\sigma(t)$ independent Brownian motions and with

$$\int_0^T \sigma(s)d\hat{W}_\sigma(s) = \frac{1}{\alpha}(\sigma(T) - \sigma(0)), \tag{27}$$

and

$$\int_0^T \sigma(s)d\hat{W}_S(s)|\sigma(T) \sim \mathcal{N}\left(0, \sqrt{\int_0^T \sigma^2(s)ds}\right). \tag{28}$$

4.4.2. Case $\beta = 1$

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

$$S(T) = S(0) \exp \left(-\frac{1}{2} \int_0^T \sigma^2(s) \mathrm{d}s + \rho \int_0^T \sigma(s) \mathrm{d}\hat{W}_\sigma(s) + \sqrt{(1-\rho^2)} \int_0^T \sigma(s) \mathrm{d}\hat{W}_S(s) \right),$$

where we can consider again Eqs. (27) and (28). The asset dynamics $S(T)$ can be thus sampled from

$$S(T) \sim S(0) \exp \left(-\frac{1}{2} \int_0^T \sigma^2(s) \mathrm{d}s + \frac{\rho}{\alpha}(\sigma(T) - \sigma(0)) + Z\sqrt{(1-\rho^2) \int_0^T \sigma^2(s) \mathrm{d}s} \right), \tag{29}$$

where Z is the standard normal.

4.4.3. 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 Eq. (2) as introduced in [24]. 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 [24] proposed a threshold value to choose the suitable technique among these two. The moment-matching approach relies on the fact that, for $S(0) \gg 0$, the distribution function in Eq. (2) can be approximated by

$$Pr\left(S(t) \leq X | S(0) > 0, \sigma(t), \int_0^t \sigma^2(s)\mathrm{d}s \right) \approx \chi^2(c; 2 - b, a).$$

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 $\mathcal{V} \sim \chi^2(x; \delta, \lambda)$ is accurately approximated by a quadratic Gaussian distribution, as follows

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

with

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

Applying this to our case in Eq. (2), we can sample the conditional asset dynamics, $T > 0$, by

$$S(T) = \left((1-\beta)^2(1-\rho^2)d(e + Z)^2 \int_0^T \sigma^2(s)\mathrm{d}s \right)^{\frac{1}{2(1-\beta)}}.$$

The moment-matching approximation is safely and accurately applicable when $0 < \psi \leq 2$ and $\mu \geq 0$ [24]. 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 [25]. 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. The resulting sampling procedure is robust and efficient.

4.4.4. Martingale correction

As pointed out in [24,26], the exact simulation of the asset price can, in certain stochastic volatility models, give rise to loss of the martingale property (because of the approximation of a continuous process by its discrete equivalent). This is especially seen, when the size of the 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. As we deal with one *large* time-step, a martingale correction needs to be applied. We incorporate a simple but effective numerical martingale correction, as follows

$$S(T) = S(T) - \frac{1}{n}\sum_{p=1}^n S_p(T) + \mathbb{E}[S(T)],$$

$$= S(T) - \frac{1}{n}\sum_{p=1}^n S_p(T) + S_0,$$

where $S_p(\cdot)$ represents the forward asset value for the *p*-th Monte Carlo path.

Note, however, that, for practical SABR parameters, the martingale error is very small.

Table 2
Data sets.

	S_0	σ_0	α	β	ρ	T
Set I	1.0	0.5	0.4	0.7	0.0	2
Set II	0.05	0.1	0.4	0.0	−0.8	0.5
Set III	0.04	0.4	0.8	1.0	−0.5	2

Table 3
MSE of $\hat{\lambda}(u) - \lambda(u)$.

	Clayton	Frank	Gumbel
Set I	1.3469×10^{-3}	2.9909×10^{-4}	5.1723×10^{-5}
Set II	1.0885×10^{-3}	2.1249×10^{-4}	8.4834×10^{-5}
Set III	2.1151×10^{-3}	7.5271×10^{-4}	2.6664×10^{-4}

5. Numerical experiments

In this section, different numerical experiments will be carried out. In Section 5.1, we compare the different copulas from Section 4.1 for the simulation of $Y(T)|\sigma(T)$. After that, in Section 5.2, we employ the best fitting copulas in a SABR pricing experiment. We consider several representative SABR data sets with special characteristics, like a zero correlation (Set I), a normal SABR model (Set II) and a high *volatility-of-volatility* SABR set (Set III). The parameter values are shown in Table 2. Other parameters in our one-step SABR method include:

- Number of discrete time points: $M = 1,000$.
- Number of COS elements: $N = 150$.
- Number of intervals for piecewise linear approximation of $I_{\mathcal{M}}$: $L = 100$.

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

5.1. Copula approach analysis

As $Y(T) = \int_0^T \sigma^2(s)ds$ is a summation of squared log-normal processes, Gaussian-based copulas may fit well. However, we also consider Archimedean copulas as these may be superior in this case. In order to choose an optimal copula, we assess a so-called *goodness-of-fit (GOF)* criterion for copulas. There are different GOF tests in the literature, see an overview in [27], for example. These measures are based on a comparison between some empirical accurate approximations (that are usually computationally expensive) and the analyzed copula. According to the work of Durrleman et al. in [28], we split the analysis, first evaluating the Archimedean copulas and subsequently, after choosing the optimal copula from this class, performing an analysis of the remaining copulas. The GOF testing for Archimedean copulas is a graphic procedure based on Kendall's processes, proposed by Genest and Rivest [29] and Barbe et al. [30]. They defined a function K by

$$K(u) = \Pr[\mathcal{C}(U_1 \dots, U_d) \leq u]$$

where $\mathcal{C}$ is the Archimedean copula and $u \in [0, 1]$. A non-parametric estimate[3] of K is available as

$$\hat{K}(u) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[\vartheta_i \leq u]}, \quad \vartheta_i = \frac{1}{n-1} \sum_{j=1}^{n} \mathbf{1}_{[x_1^j < x_1^i, \dots, x_d^j < x_d^i]}$$

with n the number of samples and x_d^i the ith sample of the dth marginal distribution. Since we deal with a bivariate case, we have $d = 2$. The graphic procedure of the GOF test consists in comparing the distance $\hat{\lambda}(u) := u - \hat{K}(u)$ and the *empirical* distance, $\lambda(u)$, obtained by using an empirical estimation of $K(u)$.

In Fig. 4, the distances $\hat{\lambda}(u)$ between three Archimedean copulas (Clayton, Frank and Gumbel) for the graphical GOF test are depicted. The empirical $\lambda(u)$ (black line) is employed as the reference. The experiment is performed for each data set in Table 2. As the measurable quantity, the mean squared error (MSE) of $\hat{\lambda}(u) - \lambda(u)$ is presented in Table 3. We use $n = 100,000$ simulations.

From the GOF results for the Archimedean copulas, we find that the Gumbel copula fits best in our framework. Once we have determined the most suitable Archimedean copula, we perform a GOF test including the Gaussian, Student **t** and Gumbel copulas. For an objective comparison, also a measurable GOF technique is applied. Among the several families of

---

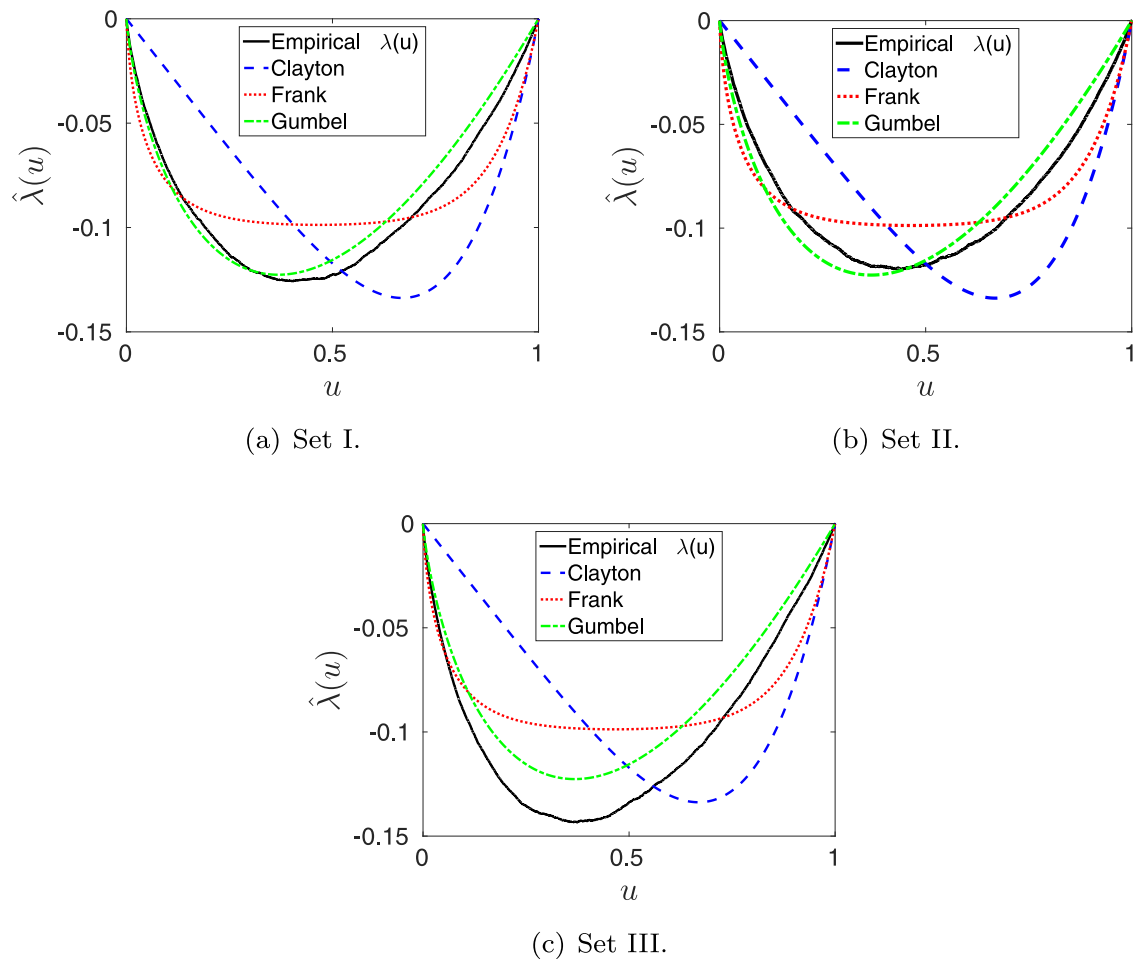[3] An analytic solution is known, but impracticable in terms of computational effort.

(a) Set I.

(b) Set II.

(c) Set III.

Fig. 4. Archimedean GOF test: $\hat{\lambda}(u)$ vs. empirical $\lambda(u)$.

Table 4
Generic GOF: D_2.

	Gaussian	Student **t** ($\nu = 5$)	Gumbel
Set I	5.0323×10^{-3}	5.0242×10^{-3}	3.8063×10^{-3}
Set II	3.1049×10^{-3}	3.0659×10^{-3}	4.5703×10^{-3}
Set III	5.9439×10^{-3}	6.0041×10^{-3}	4.3210×10^{-3}

available generic GOF tests in the literature, in [27], Berg suggests that methods that rely on the so-called *Deheuvels or empirical copula* [31] give reliable results. Hence, we will perform a test which is based on the distances between the Deheuvels copula, $\mathcal{C}_d$, and the analyzed copula $\mathcal{C}$ (see [28]). By using the discrete L^2 norm, this GOF measure reads

$$D_d(\mathcal{C}_d, \mathcal{C}) = \|\mathcal{C}_d - \mathcal{C}\|_{L^2},$$

where d is the number of random variables to build the copula, $d = 2$. For this GOF test, we reduce the number of samples to $n = 1,000$ due to the computational (time and memory) cost of the Deheuvels copula calculations. The other parameters remain the same. In Table 4, the obtained distances, D_2, between the tested copulas (Gaussian, Student **t** and Gumbel) and the Deheuvels copula are shown.

According to the GOF results, the three copulas perform very similarly. When longer maturities are considered, the Gumbel copula exhibits smallest errors in the GOF tests. In terms of speed, the Gaussian copula is around three times faster than the Gumbel copula, although the impact in the overall method is very small. The Student **t** copula is discarded since its accuracy and performance are very similar to the Gaussian copula and the calibration of the ν parameter adds extra complexity. Therefore, as a general strategy, we conclude that the Gumbel copula is the most robust choice. When short maturities are considered, the Gaussian copula may be a satisfactory alternative.
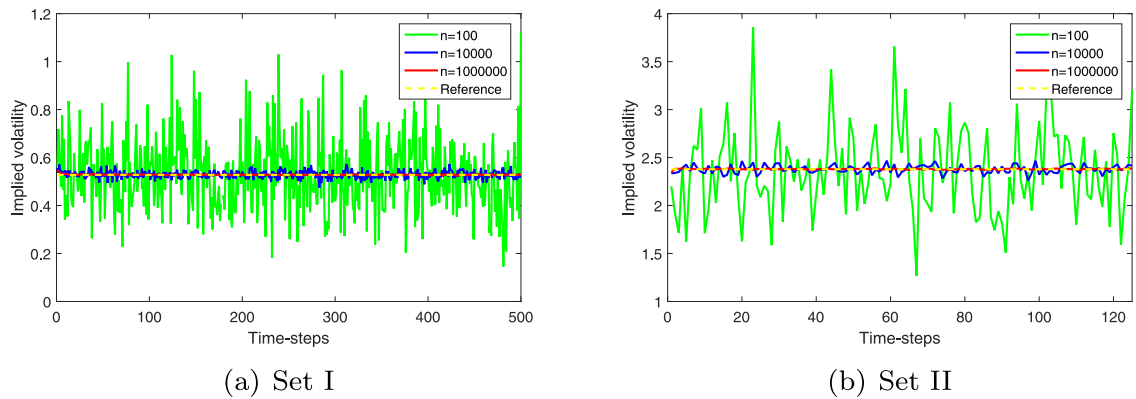
(a) Set I   (b) Set II

Fig. 5. Monte Carlo Milstein convergence test. Strike X_1.

Table 5
Convergence in n: mean and standard deviation of the error (basis points) and time (seconds).

	$n = 1000$	$n = 10,000$	$n = 100,000$	$n = 1,000,000$
		Gaussian (Set I, X_1)		
Error	519.58(204.02)	132.39(68.03)	37.42(16.55)	16.23(7.66)
Time	0.3386	0.3440	0.3857	0.5733
		Gumbel (Set I, X_1)		
Error	151.44(199.36)	−123.76(86.33)	34.14(17.03)	11.59(6.58)
Time	0.3492	0.3561	0.3874	0.6663

5.2. Pricing European options by the one-step SABR method

In Section 2.1, the Monte Carlo simulation of the SABR model was described. Now that all method components have been defined, we wish to test the one-step SABR method by pricing European call options.

For the experiment, we consider the data sets in Table 2, the strike values X_i are chosen following the expression by Piterbarg [32],

$$X_i(T) = S(0) \exp(0.1 \times T \times \delta_i),$$
$$\delta_i = -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5.$$

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

We will compare the implied volatilities obtained by means of different approaches: Hagan's formula, Monte Carlo simulation with a Milstein discretization and the one-step SABR method. Milstein-based Monte Carlo simulation is employed here as a reference for our one time-step simulation. In [24], the authors have shown that the Milstein scheme is relatively stable when applied for the SABR model compared to the Euler or log-Euler schemes. In Fig. 5, the convergence of the Milstein Monte Carlo simulation is numerically shown. Fig. 5a gives convergence results for Set I, where an analytic solution is given by Antonov [7] ($\rho = 0$). In Fig. 5b, the Monte Carlo convergence for Set II is depicted (in [33], Korn and Tang provide a reference price for $\beta = 0$). In order to have a reliable reference, we set $n = 1,000,000$ paths and $250 \times T$ time-steps for the pricing experiment.

Now, we analyze the accuracy and the performance of the one-step SABR method with the Gaussian and Gumbel copulas. In Table 5, the convergence of our method when the number of samples, n is increased is empirically shown. We present the means and the standard deviations of the error in basis points between the implied volatilities given by our one-step method and the reference price (Antonov) when Set I and X_1 are employed. We observe a reduction in the error (both mean and standard deviation) according to the expected Monte Carlo ratio ($1/\sqrt{n}$). Regarding the computational time, also in Table 5, the execution times of the one-step SABR method are shown. We can see that, up to $n = 100,000$, the number of simulations does not affect the performance. Also when larger numbers of samples are required, the computational time of the one time-step method does not increase drastically. As stated before, sampling with Gumbel copula is somewhat slower than with the Gaussian copula, but the overall difference is negligible. The parameter M, number of discrete time points, is not varied here because, as stated in Section 3.3.1, it has no significant impact on the execution time and can be set to a very high value.

Table 6
One-step SABR method with Gaussian and Gumbel copulas – implied volatility: differences in basis points.

Strikes	X_1	X_2	X_3	X_4	X_5	X_6	X_7
			Set I (reference: Antonov [7])				
Hagan	55.07	52.34	50.08	N/A	47.04	46.26	45.97
MC	23.50	21.41	19.38	N/A	16.59	15.58	14.63
Gaussian	16.23	20.79	24.95	N/A	33.40	37.03	40.72
Gumbel	11.59	15.57	19.12	N/A	25.41	28.66	31.79
			Set II (reference: Korn [33])				
Hagan	−558.82	−492.37	−432.11	−377.47	−327.92	−282.98	−242.22
MC	5.30	6.50	7.85	9.32	10.82	12.25	13.66
Gaussian	9.93	9.98	10.02	10.20	10.57	10.73	11.04
Gumbel	−9.93	−9.38	−8.94	−8.35	−7.69	−6.83	−5.79
			Set III (reference: MC Milstein)				
Hagan	287.05	252.91	220.39	190.36	163.87	141.88	126.39
Gaussian	16.10	16.76	16.62	15.22	13.85	12.29	10.67
Gumbel	6.99	3.79	0.67	−2.27	−5.57	−9.79	−14.06

To further test the one-step SABR method, in Table 6, the differences (in basis points) between the obtained implied volatilities with different considered methods and several strikes are presented. The number of paths is set to $n = 1,000,000$.

As the pricing experiments show, our copula-based one-step method achieves a very high accuracy. Furthermore, since it is a one time-step technique, it is a fast ($\approx 0.5\,$s) alternative for pricing of European options up to maturities of two years under the SABR model. Our one-step approach works particularly well in the case of low strike values and higher volatilities, that pose some problems for the Hagan implied volatility formula.

6. Conclusions

In this work, an efficient method to obtain samples of the SABR dynamics based on the time-integrated variance has been developed. The technique employs a Fourier method to derive the CDF of the time-integrated variance. By means of copulas, the conditional sampling technique is obtained. Its application gives us a fast and accurate *one time-step* Monte Carlo method for the SABR model simulation. By numerical experiments, we have shown that our method does not suffer from well-known problems of the Hagan formula in the case of small strike values and higher volatilities.

A generalization of our one-step SABR method to a multiple time-step version will be presented in our future work. This will allow us to deal with problems with longer maturities (more than two years) and also with more involved exotic options (early-exercise and path-dependent options). We need to introduce new method components to deal with the increasing computational complexity in that case.

Acknowledgment

References

[1] P.S. Hagan, D. Kumar, A.S. Lesniewski, D.E. Woodward, Managing smile risk, Wilmott Mag. (January, 2002) 84–108.
[2] J. Obloj, Fine-tune your smile: correction to Hagan et al., 2013, Available at http://arxiv.org/abs/0708.0998v3.
[3] P.S. Hagan, A.S. Lesniewski, D.E. Woodward, Probability distribution in the SABR model of stochastic volatility, In proceedings: Large Deviations and Asymptotic Methods in Finance 2005 (2015) 1–35, doi:10.1007/978-3-319-11605-1_1.
[4] Q. Wu, Series expansion of the SABR joint density, Math. Financ. 22 (2) (2012) 310–345, doi:10.1111/j.1467-9965.2010.00460.x.
[5] P. Henry-Labordère, A general asymptotic implied volatility for stochastic volatility models, 2005, Available at http://arxiv.org/pdf/cond-mat/0504317v2.pdf.
[6] L. Paulot, Asymptotic implied volatility at the second order with application to the SABR model, 2009, Available at http://ssrn.com/abstract=1413649.
[7] A. Antonov, M. Konikov, M. Spector, SABR spreads its wings, Risk Mag. (July, 2013) 58–63.
[8] U. Cherubini, S. Mulinacci, F. Gobbi, S. Romagnoli, Dynamic copula methods in finance, vol. 625, John Wiley & Sons, 2011.
[9] D.X. Li, On default correlation: a copula function approach, J. Fixed Income 9 (2000) 43–54, doi:10.3905/jfi.2000.319253.
[10] P. Embrechts, A. McNeil, D. Straumann, Correlation: pitfalls and alternatives, Risk Mag. 12 (1999) 69–71.
[11] U. Cherubini, E. Luciano, Value-at-risk trade-off and capital allocation with copulas, Econ. Notes 30 (2) (2001) 235–256.
[12] J.V. Rosenberg, T. Schuermann, A general approach to integrated risk management with skewed, fat-tailed risks, J. Financ. Econ. 79 (3) (2006) 569–614.
[13] U. Cherubini, E. Luciano, Bivariate option pricing with copulas, Appl. Math. Financ. 9 (2) (2002) 69–85.
[14] J.V. Rosenberg, Nonparametric pricing of multivariate contingent claims, J. Deriv. 10 (3) (2003) 9–26, doi:10.3905/jod.2003.319198.
[15] B. Zhang, C.W. Oosterlee, Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions, SIAM J. Financ. Math. 4 (1) (2013) 399–426.
[16] J.C. Cox, Note on option pricing I: constant elasticity of variance diffusions, J. Portf. Manag. 22 (1996) 15–17.

[17] M. Schroder, Computing the constant elasticity of variance option pricing formula, J. Financ. 44 (1) (1989) 211–219, doi:10.1111/j.1540-6261.1989.tb02414.x.

[18] O. Islah, Solving SABR in exact form and unifying it with LIBOR market model, 2009, Available at http://ssrn.com/abstract=1489428.

[19] E. Benhamou, Fast Fourier transform for discrete Asian options, J. Comput. Financ. 6 (1) (2002) 49–68.

[20] F. Fang, C.W. Oosterlee, A novel pricing method for European options based on Fourier-cosine series expansions, SIAM J. Sci. Comput. 31 (2008) 826–848.

[21] L. Ortiz-Gracia, C.W. Oosterlee, A highly efficient Shannon wavelet inverse Fourier technique for pricing European options, SIAM J. Sci. Comput. 38 (1) (2016) 118–143.

[22] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer–Verlag, 1992.

[23] A. Sklar, Fonctions de répartition à n dimensions et leurs marges, Publications de l'Institut de Statistique de l'Université de Paris 8 (1959) 229–231.

[24] B. Chen, C.W. Oosterlee, H. van der Weide, A low-bias simulation scheme for the SABR stochastic volatility model, Int. J. Theor. Appl. Financ. 15 (2) (2012) 1250016-1–1250016-37.

[25] M. Sankaran, Approximations to the non-central chi-square distribution, Biometrika 50 (1–2) (1963) 199–204, doi:10.1093/biomet/50.1-2.199.

[26] L.B.G. Andersen, Efficient simulation of the Heston stochastic volatility model, J. Comput. Financ. 11 (3) (2008) 1–22.

[27] D. Berg, Copula goodness-of-fit testing: an overview and power comparison, Eur. J. Financ. 15 (7–8) (2009) 675–701, doi:10.1080/13518470802697428.

[28] V. Durrleman, A. Nikeghbali, T. Roncalli, Which copula is the right one?, 2000, Available at SSRN: http://ssrn.com/abstract=1032545.

[29] C. Genest, L.-P. Rivest, Statistical inference procedures for bivariate Archimedean copulas, J. Am. Stat. Assoc. 88 (423) (1993) 1034–1043.

[30] P. Barbe, C. Genest, K. Ghoudi, B. Rémillard, On Kendall's process, J. Multivar. Anal. 58 (2) (1996) 197–229.

[31] P. Deheuvels, La fonction de dépendance empirique et ses propriétés, un test non paramétrique d'indépendance, Bulletin de l'Académie Royale de Belgique, Classe des Sciences (5) 65 (1979) 274–292.

[32] V. Piterbarg, Smiling hybrids, Risk Mag. 19 (5) (2006) 66–70.

[33] R. Korn, S. Tang, Exact analytical solution for the normal SABR model, Wilmott Mag. 2013 (66) (2013) 64–69, doi:10.1002/wilm.10235.