Speedup of calibration and pricing with SABR models: from equities to interest rates derivatives

A. M. Ferreiro, J. A. García, J. G. López-Salas and C. Vázquez

Abstract In the more classical models for equities and interest rates evolution, constant volatility is usually assumed. However, in practice the volatilities are not constant in financial markets and different models allowing a varying local or stochastic volatility also appear in the literature. Particularly, we consider here the SABR model that has been first introduced in a paper by Hagan and coworkers, where an asymptotic closed-form formula for the implied volatility of European plain-vanilla options with short maturities is proposed. More recently, different works (Hagan-Lesniewski, Mercurio-Morini and Rebonato) have extended the use of SABR model in the context of LIBOR market models for the evolution of forward rates (SABR-LMM). One drawback of these models in practice comes from the increase of computational cost, mainly due to the growth of model parameters to be calibrated. Additionally, sometimes either it is not always possible to compute an analytical approximation for the implied volatility or its expression results to be very complex, so that numerical methods (for example, Monte Carlo in the calibration process) have to be used. In this work we mainly review some recently proposed global optimization techniques based on Simulated Annealing (SA) algorithms and its implementation on Graphics Processing Units (GPUs) in order to highly speed up the calibration and pricing of different kinds of options and interest rate derivatives. Finally, we present some examples corresponding to real market data.

Key words: SABR volatility models, SABR/LIBOR market models, parallel simulated annealing, GPUs

---

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

1 Introduction

Mathematical models have become of great importance in order to price financial derivatives on different underlying assets. However, in most cases there is no explicit solution to the governing equations, so that accurate robust fast numerical methods are required. Furthermore, financial models usually depend on many parameters that need to be calibrated to market data. As in practice the valuations are required almost in real time, the speed of numerical computations becomes critical and this calibration process must be performed as fast as possible.

In the classical BlackScholes model [1], the underlying asset follows a lognormal process with constant volatility. However, in real markets the volatilities are not constant and they can vary for each maturity and strike (volatility surface). In order to overcome this problem, different local and stochastic volatility models have been introduced. In [7], Hagan, Kumar, Lesniewski and Woodward proposed a stochastic volatility model known as the SABR model (acronym for stochastic, alpha, beta and rho, three of the four model parameters), arguing that sometimes local volatility models could not reproduce market volatility smiles and that their predicted volatility dynamics contradicts market smiles and skews. The forward price of an asset follows, under the assets canonical measure, a CEV type process with stochastic volatility driven by a driftless process. The Brownian motion driving the volatility can be correlated with the one associated to the forward price. The main advantages of the model are the following. Firstly, it is able to correctly capture market volatility smiles. Secondly, its parameters, which play specific roles in the generation of smiles and skews, have an intuitive meaning. Thirdly, the authors obtained an analytical approximation for the implied volatility (known as Hagan formula) through singular perturbation techniques, thus allowing an easy calibration of the model. Finally, it has become the market standard for interpolating and extrapolating prices of plain vanilla options [13]. In [10] Oblój proved that for strikes far from the money and/or long maturities Hagan formula is neither arbitrage free nor a good approximation of implied volatilities. Besides, the author improved the former formula.

The existence of closed-form formula simplifies the calibration of the parameters to fit market data. However, when considering constant parameters (static SABR model), the volatility surface of a set of market data for several maturities cannot be suitably fitted. In order to cope with this problem, SABR model with time dependent parameters (dynamic SABR) is proposed in [7]. Nevertheless, time dependent parameters highly increase the computational cost and it is not always possible to compute an analytical approximation for the implied volatility [3]. In this case, we can use numerical methods (for example, Monte Carlo) in the calibration process.

The standard Libor Market Model (LMM) [2] presents the same drawbacks as the classical Black-Scholes theory. The major disadvantage comes from the assumption of deterministic volatility coefficients that prevents matching cap and swaption volatility smiles and skews observed in the markets. Thus, there has been great research in extending the standard LMM to correctly capture market volatility smiles and skews. Several authors have recently tried to unify SABR and LIBOR market models. In the more standard LIBOR market model, the dynamics of each LIBOR forward rate under the corresponding terminal measure are assumed to be martingales with constant volatility.

In [6], Hagan *et al.* studied the natural extension of both the LMM and the SABR model. They used the technique of low noise expansions in order to produce accurate and workable approximations to swaption volatilities. Mercurio and Morini, arguing that a number of volatility factors lower than the number of state variables is often chosen, proposed in [9] a SABR/LIBOR market model with one single volatility factor. They designed a LIBOR market model starting from the reference SABR dynamics, with the purpose of preserving the SABR closed formula approximation. In [12], Rebonato designed a time-homogeneous SABR-consistent extension of the LMM. More precisely, the author specified financially motivated dynamics for the LMM forward rates and volatilities that match the SABR prices very close. Rebonato also suggested a simple financially justifiable and computationally affordable way to calibrate the model. For sake of brevity, in this work we only focus in the Mercurio and Morini model. Readers interested in the other two models are referred to [5].

The main objective of the present work is to efficiently calibrate plain SABR models and SABR/LIBOR market models. As computations based financial analysis should be carried out almost in real-time, an efficient robust and fast optimization algorithm has to be chosen.

In general, swaptions cannot be priced in closed form in the LMM and the main challenge of these works comes from the analytical approximations to price these derivatives. All the previous papers argue that the “brute-force” approach, which consists in calibrating the models using Monte Carlo simulation to price swaptions, is not a practical choice, because each Monte Carlo evaluation is computationally very expensive. However, in this article we propose the use of relatively old Simulated Annealing type algorithms [8], which becomes highly efficient when implemented using High Performance Computing techniques. This combination makes possible the calibration in a reasonable computational time.

2 SABR model

The dynamics of the forward price and its volatility satisfy the system of stochastic differential equations

$$dF(t) = V(t)F(t)^{\beta}dW(t), \qquad F_0 = \hat{f}, \tag{1}$$

$$dV(t) = \nu V(t)dZ(t), \qquad V_0 = \alpha, \tag{2}$$

where $F(t) = S(t)e^{(r-y)(T-t)}$ denotes the *forward* price of the underlying asset S, r being the constant interest rate and y the constant dividend yield. Moreover, $V(t)$ denotes the asset volatility process, dW and dZ are two correlated Brownian motions with constant correlation coefficient ρ (i.e. $dW(t)dZ(t) = \rho dt$) and S_0 is the spot price of the asset. The parameters of the model are: $\alpha > 0$ (the volatility's reference level), $0 \leq \beta \leq 1$ (the variance elasticity), $\nu > 0$ (the volatility of the volatility) and ρ (the correlation coefficient).

This model with constant parameters is known as static SABR model. The main drawback of this static SABR model arises when market data for options with several maturities are considered. In this case, too large errors can appear in the calibration process (see, for example, [3]). In order to overcome this problem, the following dynamic SABR model allows time dependency in some parameters:

$$dF(t) = V(t)F(t)^{\beta} dW(t), \qquad F_0 = \hat{f}, \tag{3}$$

$$dV(t) = \nu(t)V(t)dZ(t), \qquad V(0) = \alpha, \tag{4}$$

where ν and the correlation coefficient ρ are time dependent, i.e. $dW(t)dZ(t) = \rho(t)dt$. As in the static SABR model, the dynamic one also provides an expression to approximate the implied volatility [11],

$$\sigma_{model}(K,\hat{f},T) = \frac{1}{\omega}\left(1 + A_1(T)\ln(K/\hat{f}) + A_2(T)\ln^2(K/\hat{f}) + B(T)T\right), \tag{5}$$

where

$$\begin{aligned}
A_1(T) &= \frac{\beta-1}{2} + \frac{\eta_1(T)\omega}{2}, \\
A_2(T) &= \frac{(1-\beta)^2}{12} + \frac{1-\beta-\eta_1(T)\omega}{4} + \frac{4\nu_1^2(T) + 3(\eta_2^2(T) - 3\eta_1^2(T))}{24}\omega^2, \\
B(T) &= \frac{1}{\omega^2}\left(\frac{(1-\beta)^2}{24} + \frac{\omega\beta\eta_1(T)}{4} + \frac{2\nu_2^2(T) - 3\eta_2^2(T)}{24}\omega^2\right),
\end{aligned}$$

with

$$\begin{aligned}
\nu_1^2(T) &= \frac{3}{T^3}\int_0^T (T-t)^2\nu^2(t)dt, \quad \nu_2^2(T) = \frac{6}{T^3}\int_0^T (T-t)t\nu^2(t)dt, \\
\eta_1(T) &= \frac{2}{T^2}\int_0^T (T-t)\nu(t)\rho(t)dt, \quad \eta_2^2(T) = \frac{12}{T^4}\int_0^T\int_0^t \left(\int_0^s \nu(u)\rho(u)du\right)^2 dsdt.
\end{aligned} \tag{6}$$

The choice of the functions ρ and ν constitutes a very important decision. The values of $\rho(t)$ and $\nu(t)$ have to be smaller for long terms (t large) rather than for short terms (t small). In this work we consider two possibilities with exponential decay:

- **Case I**: It is more classical and corresponds to the choice

$$\rho(t) = \rho_0 e^{-at}, \quad \nu(t) = \nu_0 e^{-bt}. \tag{7}$$

  with $\rho_0 \in [-1,1]$, $\nu_0 > 0$, $a \geq 0$ and $b \geq 0$. In this case, the expressions of the functions ν_1^2, ν_2^2, η_1 and η_2^2, defined by (6), can be exactly calculated and are given by:

$$
\begin{aligned}
v_1^2(T) &= \frac{6v_0^2}{(2bT)^3}\left[\left((2bT)^2/2-2bT+1\right)-e^{-2bT}\right],\\
v_2^2(T) &= \frac{6v_0^2}{(2bT)^3}\left[2(e^{-2bT}-1)+2bT(e^{-2bT}+1)\right],\\
\eta_1(T) &= \frac{2v_0\rho_0}{T^2(a+b)^2}\left[e^{-(a+b)T}-(1-(a+b)T)\right],\\
\eta_2^2(T) &= \frac{6v_0^2\rho_0^2}{T^4(a+b)^4}\left[(a+b)T(1-e^{-2(a+b)T})-2(1-e^{-(a+b)T})^2\right].
\end{aligned}
\tag{8}
$$

- **Case II**: A more general case corresponds to the choice (see [3], for details)

$$
\rho(t) = (\rho_0 + q_\rho t)e^{-at} + d_\rho, \quad v(t) = (v_0 + q_v t)e^{-bt} + d_v. \tag{9}
$$

  In this case, the symbolic software package Mathematica allows to calculate exactly the functions v_1^2, v_2^2 and η_1. However, it is not possible to obtain an explicit expression for the function η_2^2, an appropriate quadrature formula has to be used.

2.1 Calibration of the parameters using GPUs

The calibration of the SABR model parameters can be done using the implied volatility formula or the Monte Carlo simulation method. Usually, in trading environments the second one is not used, mainly due to its high execution times. However, if we have a parallel and efficient implementation of the Monte Carlo method, we can consider its usage in the calibration procedure.

In this work, the calibration of the parameters has been done with a Simulated Annealing stochastic global optimization method (see [8], for example). The algorithm consists in an external decreasing temperature loop. At each fixed temperature a Metropolis process, that can be seen as a Markov chain, is performed to compute the equilibrium state at this temperature level. This Markov chain consists of randomly choosing points in the search domain: if the value of the cost function at a new point decreases, the point is accepted; otherwise the point is randomly accepted following the Boltzman criterion, where the probability of accepting points with higher cost function value decreases with temperature. This process is repeated at each temperature level until temperature is low enough. As it is well known in the literature, SA involves a great computational cost.

In order to speed up this algorithm it must be parallelized. In [4], the authors discuss about the parallelization of the SA using GPUs. In next sections 2.1.1 and 2.1.2 we briefly introduce two calibration techniques which are further detailed in [3, 5].

2.1.1 Calibration with Technique I

The idea is that at each temperature level the Markov chains are distributed among the GPU threads. Among all the final reached points of the threads, the one with the minimum cost function value is selected, thus performing a reduction operation. The selected point is the starting one for all the threads in the next temperature level. The process is repeated until reaching a certain value of temperature.

The previous implementation can also be improved using multi-GPUs. In this case, the Markov chains are distributed among GPUs (for example, if we have two GPUs, half of the chains are computed by each GPU, see Figure 1) and at each GPU the chains are distributed among the threads of this particular GPU. Before advancing to the next temperature level the best point must be computed in each GPU and then the best point of all GPUs is computed and used as starting point for all the upcoming threads of the new temperature level (see Figure 1). In order to calibrate models with many parameters, the multi-GPU version becomes more suitable, since the minimization process is very costly.

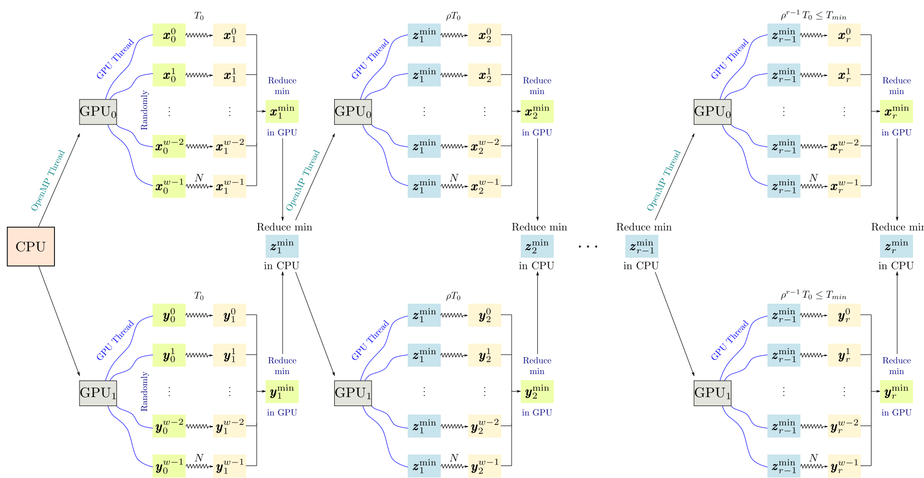

Fig. 1 Sketch of the parallel SA algorithm using two GPUs.

2.1.2 Calibration with Technique II

In this calibration technique the cost function is computed in GPU by a Monte Carlo method. As the Monte Carlo method is carried out inside the GPU, the SA minimization algorithm is run on CPU. In order to use all available GPUs in the system, we propose a CPU SA parallelization using OpenMP [15]. So, each OpenMP SA thread uses a GPU to assess on the Monte Carlo objective function. This approach can be easily extrapolated to a cluster of GPUs using, for example, MPI [14].

2.2 Numerical results

We consider market data corresponding to the EUR/USD exchange rate. The EUR/USD spot rate is $S_0 = 1.2939$ US dollars quoted in December of 2011. In Figure 2, the whole volatility surface at maturities 3, 6, 12 and 24 months is shown. Note that the dynamic SABR model captures correctly the volatility skew. The mean relative error is 2.441714×10^{-2} and the maximum relative error is 6.954307×10^{-2}. For one GPU the speedup is around 240, while for two GPUs is near 451. More details can be found in [3].

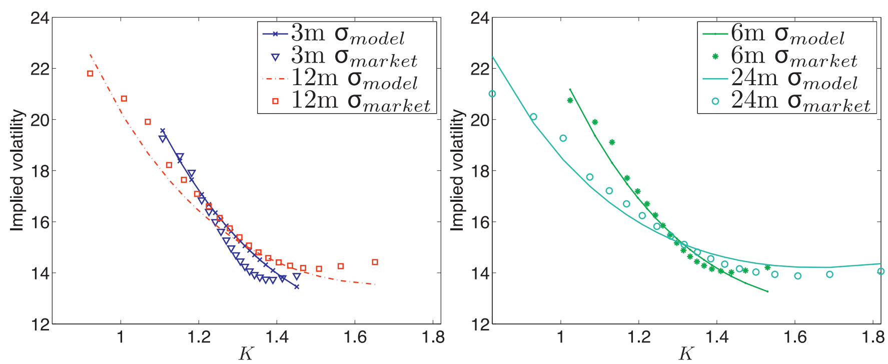

Fig. 2 EURUSD. Dynamic SABR. σ_{model} vs σ_{market} for the whole volatility surface.

3 SABR/LIBOR Mercurio & Morini model

When adding the SABR model, the forward rates and volatility processes satisfy the following coupled dynamics

$$
\begin{aligned}
dF_i(t) &= V_i(t)F_i(t)^{\beta_i} dW_i(t), \\
dV_i(t) &= \sigma_i V_i(t) dZ_i(t).
\end{aligned}
$$

We note that if the interest rate derivative only depends on one particular forward rate, then it is convenient to use the corresponding terminal measure. However, when derivatives depend on several forward rates, a common measure needs to be used.

Thus, in the case of pricing complex derivatives a change of measure produces the appearance of drift terms in forward rates and volatilities dynamics.

In the Mercurio & Morini model [9], the existence of a lognormal common volatility process to all forward rates is assumed, while each forward rate F_i $i = 1,\dots,M$ satisfies a particular SDE. More precisely, we have

$$dF_i(t) = \mu^{F_i}(t)dt + \alpha_i V(t)F_i(t)^{\beta} dW_i(t), \tag{10}$$

$$dV(t) = \sigma V(t)dZ(t), \tag{11}$$

with the associated correlations denoted by

$$\mathbb{E}[dW_i(t)\cdot dW_j(t)] = \rho_{i,j}dt, \quad \mathbb{E}[dW_i(t)\cdot dZ(t)] = \phi_i dt,$$

and the initial given values $V(0) = 1$ and $F_i(0)$. Thus, the correlation structure is given by the block-matrix

$$P = \begin{bmatrix} \rho & \phi \\ \phi^{\top} & 1 \end{bmatrix},$$

where the submatrix $\rho = (\rho_{i,j})$ represents the correlations between the forward rates F_i and F_j and the vector $\phi = (\phi_1,\dots,\phi_M)^{\top}$ includes the correlations between the forward rates F_i and the instantaneous volatility V.

More precisely, if we introduce the bank-account numeraire $\beta(t)$, defined by

$$\beta(t) = \prod_{j=0}^{i-1} \left(1 + \Delta t F_j(T_j)\right) \quad \text{if } t \in [T_i, T_{i+1}],$$

then, under the associated spot probability measure, the drift terms of the processes defined in (10) are

$$\mu^{F_i}(t) = \alpha_i V(t)F_i(t)^{\beta} \sum_{j=h(t)}^{i} \frac{\tau_j \rho_{i,j} \alpha_j V(t) F_j(t)^{\beta}}{1 + \tau_j F_j(t)},$$

where $h(t)$ denotes the index of the first unfixed F_i, i.e.,

$$h(t) = j, \text{ if } t \in [T_{j-1}, T_j). \tag{12}$$

The implied volatility for this model can be computed from Hagan second order approximation formula [10]:

$$\sigma\big(K, F_i(0)\big) \approx \frac{\alpha_i}{F_i(0)^{(1-\beta)}} \times \Bigg\{ 1 - \frac{1}{2}(1-\beta-\phi_i\sigma\omega_i)\cdot \ln\Big(\frac{K}{F_i(0)}\Big) + \frac{1}{12}\Big((1-\beta)^2 + (2-3\phi_i^2)\sigma^2\omega_i^2 + 3\big((1-\beta)-\phi_i\sigma\omega_i\big)\Big)\cdot \Big[\ln\Big(\frac{K}{F_i(0)}\Big)\Big]^2 \Bigg\}, \tag{13}$$

where $\omega_i = \bar{\alpha}_i{}^{-1} F_i(0)^{(1-\beta)}$, $\bar{\alpha}_i = \alpha_i \left[\mathrm{e}^{\int_0^{T_i} M_i(s)ds} \right]$ and $M_i(t) = -\sigma \sum_{j=h(t)}^{i} \frac{\tau_j \phi_j \alpha_j F_j(0)^\beta}{1+\tau_j F_j(0)}$.

For the correlations, we consider the following functional parameterizations:

$$\rho_{i,j} = \eta_1 + (1-\eta_1)\exp[-\lambda_1 |T_i - T_j|]. \tag{14}$$

3.1 Model calibration

We consider the market prices of caplets and swaptions and we pose the corresponding global optimization problems to calibrate the model parameters. Model parameters are calibrated in two stages, firstly to caplets and secondly to swaptions. We note that model parameters can be classified into two categories (volatility and correlation parameters). The volatility parameters are $\boldsymbol{x} = (\phi_i, \sigma, \alpha_i)$ and the correlation ones $\boldsymbol{y} = (\eta_1, \lambda_1)$. According to this classification, the cost functions to be minimized in the calibration process are the following:

- Function to calibrate the market prices of caplets:

$$f_c(\boldsymbol{x}) = \sum_{i=1}^{M} \sum_{j=1}^{numK} \Big(\sigma\big(K_j, F_i(0)\big) - \sigma_{market}\big(K_j, F_i(0)\big)\Big)^2 (\boldsymbol{x}),$$

  where σ is given by Hagan formula (13) with the parameters (**??**), σ_{market} are the market smiles and $\boldsymbol{x}$ is the vector containing the volatility parameters of the model. Moreover, M and $numK$ denote the number of maturities and strikes of the caplets, respectively. In order to minimize this function we use the previous calibration technique I.
- Function to calibrate the market prices of swaptions:

$$f_s(\boldsymbol{y}) = \sum_{i=1}^{numSws} \left(S_{\mathrm{Black}}(swaption_i) - S_{MC}(swaption_i)\right)^2 (\boldsymbol{y}),$$

  where $swaption_i$ denotes the i-th swaption, S_{Black} is the Black formula for swaptions and $S_{MC}(swaption_i)$ is the value of the *i-th* swaption computed using Monte Carlo method. Moreover, $\boldsymbol{y}$ denotes the vector containing the correlation parameters of the model and *numSws* is the number of swaptions. So as to optimize this function we employ the former calibration technique II.

3.2 Numerical results

Market data correspond to the 6 month EURIBOR rate (see [5] for details). We show in Table 1 the smiles of the forward rates and in Table 2 the smiles of the swap rates.

Table 1 Smiles of forward rates. Fixing dates (first column) and moneyness (first row).

	-80%	-60%	-40%	-20%	0%	20%	40%	60%	80%
21-05-12	142.61%	117.05%	97.26%	82.58%	72.29%	70.89%	69.49%	68.08%	66.67%
21-11-12	112.74%	99.23%	88.27%	79.62%	73.03%	71.95%	70.87%	69.77%	68.69%
21-05-13	91.55%	83.75%	77.09%	71.50%	67.93%	67.10%	66.41%	65.88%	65.49%
21-11-13	64.82%	60.95%	57.08%	53.21%	52.49%	51.34%	50.61%	50.30%	50.46%
21-05-14	66.96%	61.84%	56.69%	52.43%	50.32%	48.72%	47.70%	47.14%	46.97%
21-11-14	69.20%	62.75%	56.30%	51.65%	48.19%	46.19%	44.91%	44.12%	43.66%
21-05-15	71.49%	63.67%	55.92%	50.89%	46.19%	43.83%	42.32%	41.35%	40.64%
21-11-15	73.89%	64.61%	55.54%	50.13%	44.25%	41.56%	39.84%	38.71%	37.78%
21-05-16	76.34%	65.56%	55.16%	49.39%	42.40%	39.43%	37.54%	36.26%	35.15%
21-11-16	78.90%	66.53%	54.78%	48.65%	40.61%	37.38%	35.34%	33.94%	32.68%
21-05-17	81.50%	67.50%	54.41%	47.94%	38.93%	35.47%	33.30%	31.81%	30.42%
21-11-17	84.24%	68.50%	54.03%	47.22%	37.29%	33.63%	31.36%	29.78%	28.28%
21-05-18	87.02%	69.50%	53.67%	46.53%	35.74%	31.92%	29.55%	27.90%	26.32%

Table 2 Smiles of swap rates. Maturities (first column) and moneyness (first row).

		-80%	-60%	-40%	-20%	0%	20%	40%	60%	80%
1 year	21/05/2012	122.30%	102.40%	87.12%	76.45%	70.40%	66.47%	64.20%	63.03%	62.56%
	21/11/2012	102.86%	89.97%	79.85%	72.49%	67.90%	64.58%	62.16%	60.39%	59.19%
	21/05/2013	95.64%	83.17%	73.42%	66.40%	62.10%	59.03%	56.84%	55.26%	54.18%
	21/11/2013	88.11%	76.06%	66.69%	60.00%	56.00%	53.18%	51.22%	49.84%	48.87%
2 years	21/05/2012	111.50%	91.60%	76.32%	65.65%	59.60%	55.67%	53.40%	52.23%	51.76%
	21/11/2012	89.66%	76.77%	66.65%	59.29%	54.70%	51.38%	48.96%	47.19%	45.99%
	21/05/2013	82.94%	70.47%	60.72%	53.70%	49.40%	46.33%	44.14%	42.56%	41.48%
	21/11/2013	77.81%	65.76%	56.39%	49.70%	45.70%	42.88%	40.92%	39.54%	38.57%
3 years	21/05/2012	106.40%	86.50%	71.22%	60.55%	54.50%	50.57%	48.30%	47.13%	46.66%
	21/11/2012	83.66%	70.77%	60.65%	53.29%	48.70%	45.38%	42.96%	41.19%	39.99%
	21/05/2013	78.34%	65.87%	56.12%	49.10%	44.80%	41.73%	39.54%	37.96%	36.88%
	21/11/2013	73.61%	61.56%	52.19%	45.50%	41.50%	38.68%	36.72%	35.34%	34.37%
4 years	21/05/2012	101.90%	82.00%	66.72%	56.05%	50.00%	46.07%	43.80%	42.63%	42.16%
	21/11/2012	80.26%	67.37%	57.25%	49.89%	45.30%	41.98%	39.56%	37.79%	36.59%
	21/05/2013	75.24%	62.77%	53.02%	46.00%	41.70%	38.63%	36.44%	34.86%	33.78%
	21/11/2013	70.91%	58.86%	49.49%	42.80%	38.80%	35.98%	34.02%	32.64%	31.67%
5 years	21/05/2012	96.15%	74.25%	58.83%	49.88%	47.40%	45.74%	44.61%	43.76%	43.05%
	21/11/2012	89.58%	68.82%	54.14%	45.54%	43.00%	39.36%	37.33%	36.15%	35.37%
	21/05/2013	83.91%	64.51%	50.71%	42.51%	39.90%	36.48%	34.59%	33.50%	32.76%
	21/11/2013	79.13%	61.09%	48.17%	40.37%	37.70%	34.50%	32.74%	31.75%	31.05%

3.2.1 Calibration to caplets

The calibrated parameters are shown in Table 3. The execution time was 9.165 seconds: the mono-GPU SA employed 9.124 s (the cost function was evaluated roughly 112 million times) and the Nelder-Mead local optimization algorithm consumed the remaining time.

In Table 4, market vs. model volatilities for the smiles of F_1 to F_4 and the moneyness -40% to 40% are shown. The mean relative error considering all smiles is 3.11×10^{-2}.

Figure 3 shows the model fitting to the smiles of the first four forward rates.

Table 3 Mercurio & Morini model, calibration to caplets with SABR formula (13): calibrated parameters.

	ϕ_i	α_i		ϕ_i	α_i
F_1	-0.7549	0.0888	F_8	-0.3661	0.0696
F_2	-0.2309	0.0842	F_9	-0.4770	0.0683
F_3	0.0666	0.0817	F_{10}	-0.5760	0.0693
F_4	0.1698	0.0662	F_{11}	-0.6615	0.0682
F_5	0.0302	0.0635	F_{12}	-0.7380	0.0682
F_6	-0.1098	0.0684	F_{13}	-0.8044	0.0669
F_7	-0.2417	0.0667			
$\sigma = 0.5986$					

Table 4 Calibration to caplets, σ_{market} vs. σ_{model}.

Moneyness	Smile of F_1			Smile of F_2		
	σ_{market}	σ_{model}	$\frac{\lvert\sigma_{market}-\sigma_{model}\rvert}{\sigma_{market}}$	σ_{market}	σ_{model}	$\frac{\lvert\sigma_{market}-\sigma_{model}\rvert}{\sigma_{market}}$
-40%	97.26	102.19	5.07×10^{-2}	88.27	89.59	1.50×10^{-2}
-20%	82.58	90.71	9.85×10^{-2}	79.62	81.81	2.75×10^{-2}
0%	72.29	81.16	1.23×10^{-1}	73.03	75.77	3.74×10^{-2}
20%	70.89	73.55	3.76×10^{-2}	71.95	71.47	6.69×10^{-3}
40%	69.49	67.88	2.31×10^{-2}	70.87	68.91	2.77×10^{-2}
Moneyness	Smile of F_3			Smile of F_4		
	σ_{market}	σ_{model}	$\frac{\lvert\sigma_{market}-\sigma_{model}\rvert}{\sigma_{market}}$	σ_{market}	σ_{model}	$\frac{\lvert\sigma_{market}-\sigma_{model}\rvert}{\sigma_{market}}$
-40%	77.09	77.13	4.45×10^{-4}	57.08	55.98	1.92×10^{-2}
-20%	71.50	71.99	6.92×10^{-3}	53.21	52.54	1.26×10^{-2}
0%	67.93	68.27	5.11×10^{-3}	52.49	50.39	4.00×10^{-2}
20%	67.10	65.96	1.69×10^{-2}	51.34	49.53	3.51×10^{-2}
40%	66.41	65.07	2.03×10^{-2}	50.61	49.97	1.27×10^{-2}

3.2.2 Calibration to swaptions

The calibrated parameters are $\eta_1 = 0.779175$ and $\lambda_1 = 2.722489$. Using two GPUs the execution time was approximately 2 hours (by using a cluster of GPUs, computing time could be substantially reduced).

In Table 5, some market vs. model swaption prices are shown. The mean absolute error considering all market swaptions is 5.50×10^{-2}.

Table 5 Calibration to swaptions, S_{Black} vs. S_{MC}, prices in %.

Moneyness	0.5×1 swaptions			1×1 swaptions		
	S_{Black}	S_{MC}	$\lvert S_{Black} - S_{MC}\rvert$	S_{Black}	S_{MC}	$\lvert S_{Black} - S_{MC}\rvert$
-40%	0.4866	0.4870	4.00×10^{-4}	0.5917	0.5870	4.70×10^{-3}
-20%	0.3562	0.3670	1.08×10^{-2}	0.4661	0.4699	3.80×10^{-3}
0%	0.2356	0.2478	1.22×10^{-2}	0.3467	0.3517	5.00×10^{-3}
20%	0.1363	0.1427	6.40×10^{-3}	0.2394	0.2422	2.80×10^{-3}
40%	0.0680	0.0657	2.30×10^{-3}	0.1517	0.1514	3.00×10^{-4}
Moneyness	1.5×1 swaptions			2×1 swaptions		
	S_{Black}	S_{MC}	$\lvert S_{Black} - S_{MC}\rvert$	S_{Black}	S_{MC}	$\lvert S_{Black} - S_{MC}\rvert$
-40%	0.7357	0.6872	4.85×10^{-2}	0.8184	0.7465	7.19×10^{-2}
-20%	0.5908	0.5516	3.92×10^{-2}	0.6603	0.5959	6.44×10^{-2}
0%	0.4536	0.4170	3.66×10^{-2}	0.5118	0.4469	6.49×10^{-2}
20%	0.3277	0.2951	3.26×10^{-2}	0.3754	0.3137	6.17×10^{-2}
40%	0.2213	0.1957	2.56×10^{-2}	0.2587	0.2078	5.09×10^{-2}

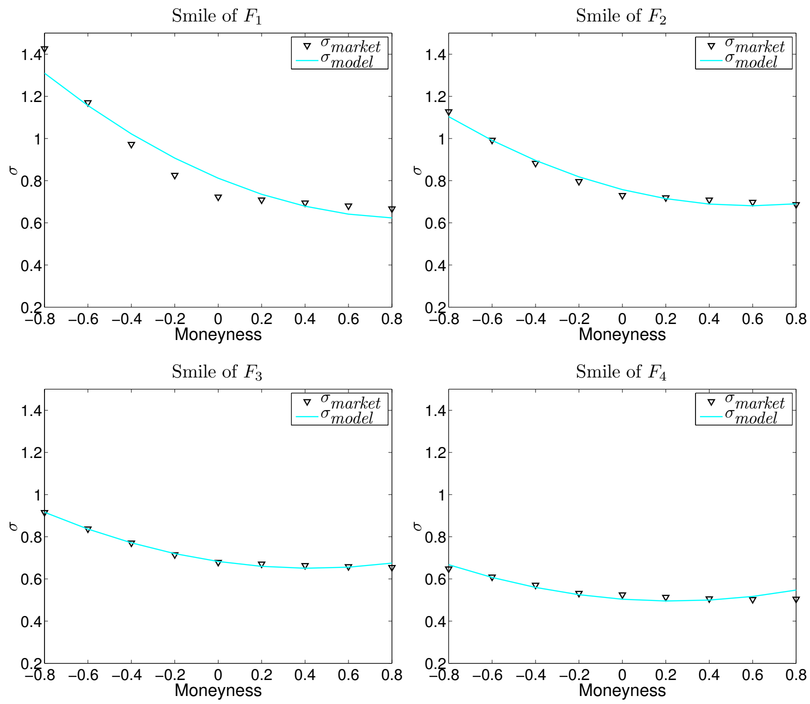

Fig. 3 σ_{market} vs. σ_{model}, smiles of $F_1, \ldots, F_4$.

Figure 4 shows the model fitting to the first four swaption market prices.

In [5], a comparative analysis of the SABR/LIBOR models proposed by Hagan, Mercurio & Morini and Rebonato is presented. The model with the best performance is the Mercurio & Morini one, since it is the easiest to calibrate, it achieves the best fit to the swaption market prices and it results the fastest one in the pricing with Monte Carlo simulation.

Note that the speedup with GPUs of the Monte Carlo calibration techniques can be applied to more complex products, for example CMS spread options which contain more information on the smile structure and the correlation of LIBOR rates.

References

1. Black, F., Scholes, M.: The pricing of options and corporate liabilities. Journal of Political Economy **81**, 637–654 (1973)
2. Brace, A., Gatarek, D., Musiela, M.: The Market model of interest rate dynamics. Mathematical Finance **7**(2), 127–155 (1997)
3. Fernández, J.L., Ferreiro, A.M., García, J.A., López-Salas, J.G., Vázquez, C.: Static and dynamic SABR stochastic volatility models: calibration and option pricing using GPUs. Mathematics and Computers in Simulation **94**, 55–75 (2013)

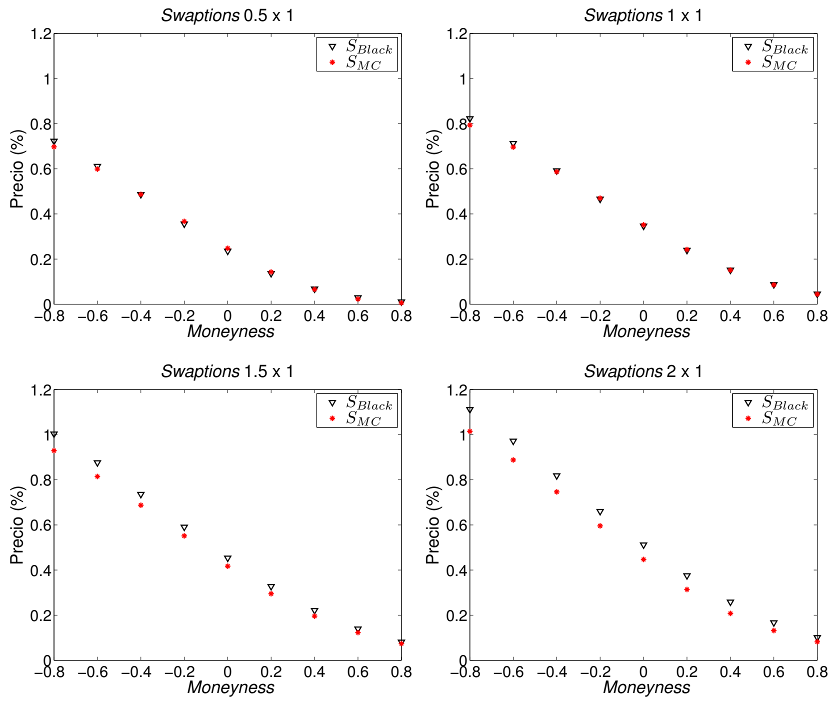

Fig. 4 S_{Black} vs. S_{MC}, $\{0.5,\dots,2\} \times 1$ swaptions.

4. Ferreiro, A.M., García, J.A., López-Salas, J.G., Vázquez, C.: An efficient implementation of parallel simulated annealing algorithm in GPUs. Journal of Global Optimization **57**(3), 863–890 (2013)
5. Ferreiro, A.M., García, J.A., López-Salas, J.G., Vázquez, C.: SABR/LIBOR market models: Pricing and calibration for some interest rate derivatives. Applied Mathematics and Computation **242**, 65–89 (2014)
6. Hagan, P., Lesniewski, A.: LIBOR market model with SABR style stochastic volatility. Working paper, available at http://lesniewski.us/papers/working/SABRLMM.pdf (2008)
7. Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E.: Managing Smile Risk. Wilmott Magazine (2002)
8. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
9. Mercurio, F., Morini, M.: No-Arbitrage dynamics for a tractable SABR term structure Libor Model. Modeling Interest Rates: Advances in Derivatives Pricing, Risk Books (2009)
10. Oblój, J.: Fine-tune your smile: Correction to Hagan *et al.* Wilmott Magazine (2008)
11. Osajima, Y.: The Asymptotic Expansion Formula of Implied Volatility for Dynamic SABR Model and FX Hybrid Model. Capital Markets: Asset Pricing and Valuation e-Journal (2007)
12. Rebonato, R.: A time-homogeneous SABR-consistent extension of the LMM. Risk (2008)
13. Rebonato, R., Mckay, K., White, R.: The SABR/LIBOR Market Model. Pricing, Calibration and Hedging for Complex Interest-Rate Derivatives, first edn. John Wiley & Sons (2009)
14. MPI: A message-passing interface standard. Message passing interface forum. http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html
15. OpenMP web page: http://openmp.org