REPOSITORIO UNIVERSIDADE CORUÑA

UNIVERSIDADE DA CORUÑA

This is an ACCEPTED VERSION of the following published document:

Ferreiro, A.M., García, J.A., López-Salas, J.G., Vázquez, C. (2016). Efficient Calibration and Pricing in LIBOR Market Models with SABR Stochastic Volatility Using GPUs. In: Russo, G., Capasso, V., Nicosia, G., Romano, V. (eds) Progress in Industrial Mathematics at ECMI 2014. ECMI 2014. Mathematics in Industry, vol 22. Springer, Cham.
https://doi.org/10.1007/978-3-319-23413-7_11

Link to published version: https://doi.org/10.1007/978-3-319-23413-7_11

General rights:

©2016 This version of the article has been accepted for publication, after peer review and is subject to Springer Nature's AM terms of use, but is not the Version of Record. The Version of Record is available online at:
https://doi.org/10.1007/978-3-319-23413-7_11

Efficient calibration and pricing in LIBOR Market Models with SABR stochastic volatility using GPUs

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

Abstract In order to overcome the drawbacks of assuming deterministic volatility coefficients in the standard LIBOR market models, several extensions of LIBOR models to incorporate stochastic volatilities have been proposed. The efficient calibration to market data of these more complex models becomes a relevant target in practice. The main objective of the present work is to efficiently calibrate some recent SABR/LIBOR market models to real market prices of caplets and swaptions. For the calibration we propose a parallelized version of the simulated annealing algorithm for multi-GPUs. The numerical results clearly illustrate the advantages of using the proposed multi-GPUs tools when applied to real market data and popular SABR/LIBOR models.

1 SABR/LIBOR market models

This work is mainly concerned with three extensions of the LIBOR market model (LMM) that incorporate the volatility smile by means of the SABR stochastic volatility model. The SABR model has become the market standard for interpolating and extrapolating prices of plain vanilla caplets and swaptions [6]. It is widely used because it involves a closed-form formula for the implied volatility which allows an easy calibration of the model. In the more standard LIBOR market model [1] the dynamics of each LIBOR forward rate under the corresponding terminal measure are assumed to be martingales with constant volatility. When adding the SABR stochastic volatility model, the forward rates and volatility processes satisfy the following coupled dynamics

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)

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

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 both dynamics. The main drawback of classical LMM comes from considering constant volatilities. SABR/LIBOR market models combine the advantages of these two models. In this paper we consider the different SABR/LIBOR models proposed by P. Hagan [5], F. Mercurio & M. Morini [8] and R. Rebonato [10]. Hereafter, for sake of brevity we only present the Rebonato model. Interested readers on the other two models are referred to [4].

For each $i = 1,\dots,M$ let F_i and V_i be the i-th forward rate that matures at time T_i and its corresponding stochastic volatility, respectively. Then, under a common measure their dynamics are given by (see [10])

$$dF_i(t) = \mu^{F_i}(t)dt + V_i(t)F_i(t)^{\beta_i}dW_i(t), \quad (1)$$
$$V_i(t) = \kappa_i(t)g_i(t), \quad (2)$$
$$d\kappa_i(t) = \mu^{\kappa_i}(t)dt + \kappa_i(t)h_i(t)dZ_i(t), \quad (3)$$

where

$$g_i(t) = \big(a + b(T_i - t)\big)\exp\big(-c(T_i - t)\big) + d, \quad h_i(t) = \big(\alpha + \beta(T_i - t)\big)\exp\big(-\gamma(T_i - t)\big) + \delta,$$

with the associated correlations denoted by

$$\mathbb{E}[dW_i(t)\cdot dW_j(t)] = \rho_{i,j}dt, \quad \mathbb{E}[dW_i(t)\cdot dZ_j(t)] = \phi_{i,j}dt, \quad \mathbb{E}[dZ_i(t)\cdot dZ_j(t)] = \theta_{i,j}dt,$$

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

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

where the submatrix $\rho = (\rho_{i,j})$ represents the correlations between the forward rates F_i and F_j, the submatrix $\phi = (\phi_{i,j})$ includes the correlations between the forward rates F_i and the instantaneous volatilities V_j, and the submatrix $\theta = (\theta_{i,j})$ contains the correlations between the instantaneous volatilities V_i and V_j.

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

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

then, under the associated spot probability measure, the drift terms of the processes defined in (1) and (3) are

$$\mu^{F_i}(t) = V_i(t)F_i(t)^{\beta_i} \sum_{j=h(t)}^{i} \frac{\tau_j \rho_{i,j} V_j(t) F_j(t)^{\beta_j}}{1+\tau_j F_j(t)}, \quad \mu^{\kappa_i}(t) = \kappa_i(t) h_i(t) \sum_{j=h(t)}^{i} \frac{\tau_j \phi_{i,j} V_j(t) F_j(t)^{\beta_j}}{1+\tau_j F_j(t)},$$

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

$$h(t) = j, \text{ if } t \in [T_{j-1}, T_j). \tag{4}$$

The implied volatility for this model can be computed from Hagan second order approximation formula [9]:

$$\sigma\big(K, F_i(0)\big) \approx \frac{\alpha_i}{F_i(0)^{(1-\beta_i)}} \times \Bigg\{ 1 - \frac{1}{2}(1-\beta_i - \phi_{i,i}\sigma_i\omega_i) \cdot \ln\Big(\frac{K}{F_i(0)}\Big) + \frac{1}{12}\Big((1-\beta_i)^2 + (2-3\phi_{i,i}^2)\sigma_i^2\omega_i^2 + 3\big((1-\beta_i) - \phi_{i,i}\sigma_i\omega_i\big)\Big) \cdot \Big[\ln\Big(\frac{K}{F_i(0)}\Big)\Big]^2 \Bigg\}, \tag{5}$$

where $\omega_i = \alpha_i^{-1} F_i(0)^{(1-\beta_i)}$, by using the following parameters denoted with *SABR* superindexes,

$$\beta_i^{SABR} = \beta_i, \quad \phi_{i,i}^{SABR} = \phi_{i,i}, \quad \alpha_i^{SABR} = \kappa_i(0) \left(\frac{1}{T_i}\int_0^{T_i} g_i(t)^2 dt\right)^{\frac{1}{2}},$$
$$\sigma_i^{SABR} = \frac{\kappa_i(0)}{\alpha_i^{SABR} T_i} \left(2\int_0^{T_i} g_i(t)^2 \hat{h}_i(t)^2 t dt\right)^{\frac{1}{2}}, \text{ where } \hat{h}_i(t) = \sqrt{\frac{1}{t}\int_0^t (h_i(s))^2 ds}. \tag{6}$$

For the correlations, we consider the following function parameterizations:

$$\rho_{i,j} = \eta_1 + (1-\eta_1)\exp[-\lambda_1 |T_i - T_j|], \tag{7}$$
$$\theta_{i,j} = \eta_2 + (1-\eta_2)\exp[-\lambda_2 |T_i - T_j|], \tag{8}$$
$$\phi_{i,j} = \operatorname{sign}(\phi_{i,i})\sqrt{|\phi_{i,i}\phi_{j,j}|}\exp\left[-\lambda_3 (T_i - T_j)^+ - \lambda_3 (T_j - T_i)^+\right], \tag{9}$$

where the terms $\phi_{i,i}$ are previously calibrated using Hagan formula (5) for the whole volatilities surface.

In this work we propose an efficient calibration strategy to some market prices for the parameters appearing in the three previous models. More precisely, we consider the market prices of caplets and swaptions and we pose the corresponding global optimization problems to calibrate the model parameters. In order to speed up the optimization algorithm we use an implementation in GPUs.

2 Model calibration

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 $\boldsymbol{x} = (\phi_{ii}, \kappa_i,$ parameters of the volatility functions g and $h)$ and the correlation ones $\boldsymbol{y} = (\eta_1, \lambda_1, \eta_2, \lambda_2, \lambda_3)$. 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 (5) with the parameters (6), 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.
- Function to calibrate the market prices of swaptions:
$$f_s(\boldsymbol{y}) = \sum_{i=1}^{numSws} \left(S_{\text{Black}}(swaption_i) - S_{MC}(swaption_i)\right)^2(\boldsymbol{y}),$$
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.

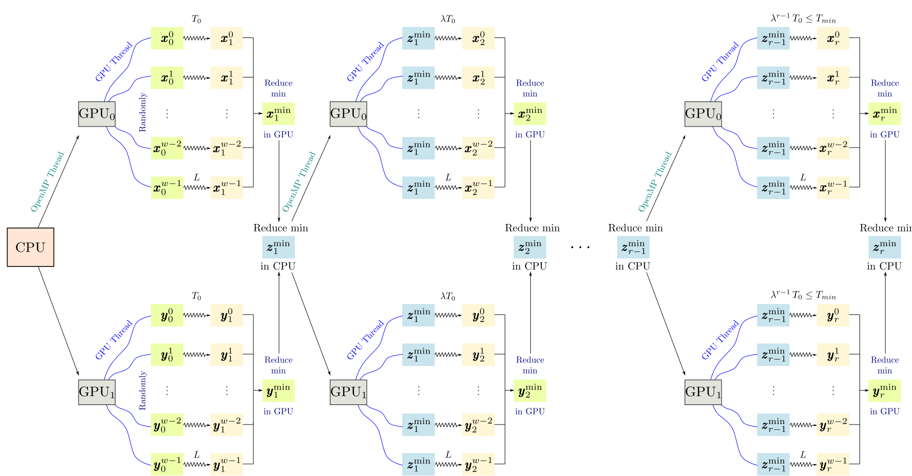

Fig. 1 Sketch of the parallel SA algorithm using two GPUs and OpenMP.

In this work, the calibration of the parameters has been performed with a Simulated Annealing (SA) global optimization algorithm [7]. 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 [3], the parallelization of the SA algorithm has been performed with GPUs. 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). This multi-GPU algorithm was presented in [2], where it was used to calibrate some SABR models to a volatility surface.

In order to calibrate models with many parameters, as the Rebonato one, the multi-GPU version becomes more suitable, since the minimization process is very costly.

In the SABR/LIBOR market models, for the calibration to swaption market prices there is not an explicit formula to price swaptions. Therefore, we use a Monte Carlo simulation technique to price swaptions, thus leading to two nested Monte Carlo loops: one for the SA and the other one for the swaption pricer. So, as the Monte Carlo swaption pricer 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. So, each OpenMP SA thread uses a GPU to evaluate the Monte Carlo objective function.

3 Numerical results

Market data correspond to the 6 month EURIBOR rate. In this section, for sake of brevity, we only present the results of the calibration of the model to the smiles of the swap rates shown in Table 2. The results of the previous calibration to the smiles of the forward rates presented in Table 1 are detailed in the article [4].

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%

The calibrated correlation parameters are $\eta_1 = 0.650997$, $\lambda_1 = 3.617546$, $\eta_2 = 0.999000$, $\lambda_2 = 0.380984$ and $\lambda_3 = 0.001000$. Using two GPUs the execution time was approximately 2 hours (by using a cluster of GPUs time could be substantially reduced). In Table 3, some market vs. model swaption prices are shown. The mean absolute error considering all market swaptions is 6.30×10^{-2}. Figure 2 shows the model fitting to the first four swaption market prices.

Table 3 Calibration to swaptions, S_{Black} vs. S_{MC}, prices in %.

Moneyness	0.5×1 swaptions			1×1 swaptions		
	S_{Black}	S_{MC}	$\|S_{Black} - S_{MC}\|$	S_{Black}	S_{MC}	$\|S_{Black} - S_{MC}\|$
−40%	0.4866	0.4870	4.00×10^{-4}	0.5917	0.5839	7.80×10^{-3}
−20%	0.3562	0.3669	1.07×10^{-2}	0.4661	0.4693	3.20×10^{-3}
0%	0.2356	0.2477	1.21×10^{-2}	0.3467	0.3546	7.90×10^{-3}
20%	0.1363	0.1441	7.80×10^{-3}	0.2394	0.2488	9.40×10^{-3}
40%	0.0680	0.0699	1.90×10^{-3}	0.1517	0.1606	8.90×10^{-3}
Moneyness	1.5×1 swaptions			2×1 swaptions		
	S_{Black}	S_{MC}	$\|S_{Black} - S_{MC}\|$	S_{Black}	S_{MC}	$\|S_{Black} - S_{MC}\|$
−40%	0.7357	0.6902	4.55×10^{-2}	0.8184	0.7465	7.19×10^{-2}
−20%	0.5908	0.5612	2.96×10^{-2}	0.6603	0.6028	5.75×10^{-2}
0%	0.4536	0.4339	1.97×10^{-2}	0.5118	0.4620	4.98×10^{-2}
20%	0.3277	0.3171	1.06×10^{-2}	0.3754	0.3354	4.00×10^{-2}
40%	0.2213	0.2188	2.50×10^{-3}	0.2587	0.2308	2.79×10^{-2}

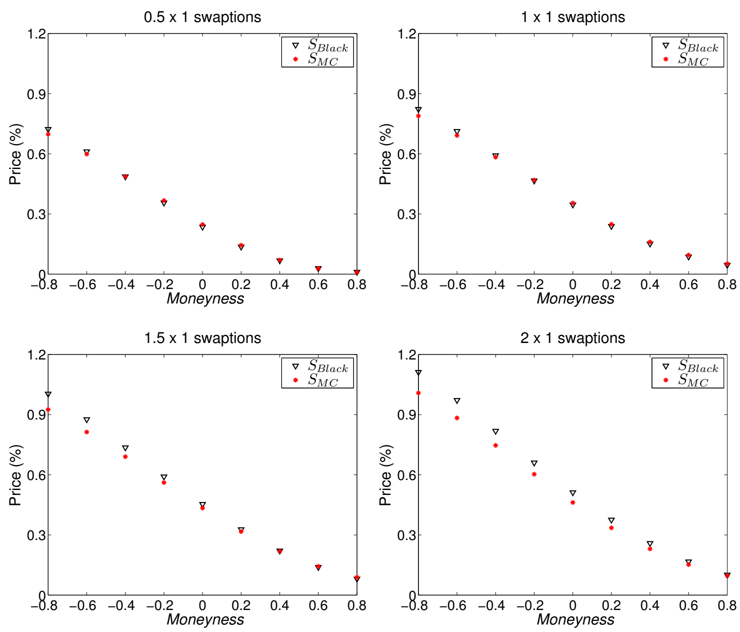

Fig. 2 S_{Black} vs. S_{MC}, $\{0.5, \ldots, 2\} \times 1$ swaptions.

References

1. Brace, A., Gatarek, D., Musiela, M.: The Market model of interest rate dynamics. Mathematical Finance **7**(2), 127–155 (1997)
2. 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. Mathe-

matics and Computers in Simulation **94**, 55–75 (2013)
3. 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)
4. 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, http://dx.doi.org/10.1016/j.amc.2014.05.017 (2014)
5. Hagan, P., Lesniewski, A.: LIBOR market model with SABR style stochastic volatility. Working paper, available at http://lesniewski.us/papers/working/SABRLMM.pdf (2008)
6. Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E.: Managing Smile Risk. Wilmott Magazine (2002)
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
8. 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)
9. Oblój, J.: Fine-tune your smile: Correction to Hagan *et al.* Wilmott Magazine (2008) (2008)
10. Rebonato, R.: A time-homogeneous SABR-consistent extension of the LMM. Risk (2008)