Interest Rate Volatility

V. Working with SABR-LMM

Andrew Lesniewski
Baruch College and Posnania Inc

First Baruch Volatility Workshop
New York

June 16 - 18, 2015

Outline

MFT approximation

- We consider a multi-factor system whose state variable X is a vector taking values in $\mathbb{R}^N$.
- Suppose that we are faced with the problem of solving a complex system of SDEs involving X:

$$\begin{aligned} dX(t) &= A(t, X(t))dt + B(t, X(t))dW(t), \\ X(0) &= X_0, \end{aligned} \tag{1}$$

  where the drift $A(t, X)$ takes values in $\mathbb{R}^N$, and the diffusion coefficient $B(t, X)$ is a matrix in $\mathrm{Mat}_{N,D}(\mathbb{R})$.
- Closed form solutions for systems of this form are rarely available, and Monte Carlo simulations tend to be computationally expensive. For many purposes, a good approximate solution is sufficient.
- The idea behind the mean field theory (MFT) approximation is to represent the solution to (1) as

$$X(t) \approx \overline{X}(t) + \xi(t), \tag{2}$$

  where $\overline{X}(t)$ is the deterministic component ("mean field") of the evolution, and $\xi(t)$ is the fluctuation around the mean field.

MFT approximation

- Ideally, we would like to have $\overline{X}(t) = \mathrm{E}[X(t)]$. Then,

$$\frac{d}{dt}\overline{X}(t) = \mathrm{E}[A(t, X(t))]. \tag{3}$$

  However, this is *not* an equation for $\overline{X}$ (except if A is linear in X), as the expression on the right hand side is not a function of $\overline{X}$.
- We *define* the mean field $\overline{X}$ as the solution to the system of self-consistent ODEs:

$$\begin{aligned} \frac{d}{dt}\overline{X}(t) &= A(t, \overline{X}(t)), \\ \overline{X}(0) &= X_0. \end{aligned} \tag{4}$$

- In other words, we make an approximation in which we replace X in (3) by its expected value. This idea goes back to the work of Curie and Weiss on the "molecular field" approach to the theory of ferromagnetism in the early 1900s.

MFT approximation

- In order to find the equation for the fluctuation process, we linearize the drift term,

$$\begin{aligned} A(t, X(t)) &= A(t, \overline{X}(t) + \xi(t)) \\ &\approx A(t, \overline{X}(t)) + \nabla A(t, \overline{X}(t))\xi(t). \end{aligned} \tag{5}$$

- Similarly, we approximate the diffusion coefficient by

$$B(t, X(t)) \approx B(t, \overline{X}(t)). \tag{6}$$

- These approximations lead to the following system of SDEs for ξ:

$$\begin{aligned} d\xi(t) &= \nabla A(t, \overline{X}(t))\xi(t)\, dt + B(t, \overline{X}(t))dW(t), \\ \xi(0) &= 0. \end{aligned} \tag{7}$$

- In summary, the mean field approximation consists in replacing the original nonlinear stochastic system (1) with:
  - (i) the system (4) of nonlinear ODEs, and
  - (ii) the system (7) of linear SDEs.

Example

- To illustrate the power of this approach, we consider the following Riccati type 1-factor stochastic dynamics:

$$\begin{aligned} dX(t) &= \mu(1 + X(t)^2)dt + \sigma dW(t), \\ X(0) &= X_0. \end{aligned} \tag{8}$$

- Integrating the equation for the mean field we find readily that

$$\overline{X}(t) = \tan(\mu t + \theta), \tag{9}$$

where $\theta = \tan^{-1}(X_0)$.

- This leads to the following equation for the fluctuation process:

$$d\xi(t) = 2\mu \tan(\mu t + \theta)\xi(t)\, dt + \sigma dW(t), \tag{10}$$

whose solution reads:

$$\xi(t) = \frac{\sigma}{\cos(\mu t + \theta)^2} \int_0^t \cos(\mu s + \theta)^2 dW(s). \tag{11}$$

Example

- Figure 1 shows two sample paths: one generated by the exact dynamics (8) (in black) and one generated by the mean field dynamics (9), (11) (in red). We use $\mu = 0.01$ and $\sigma = 0.15$.

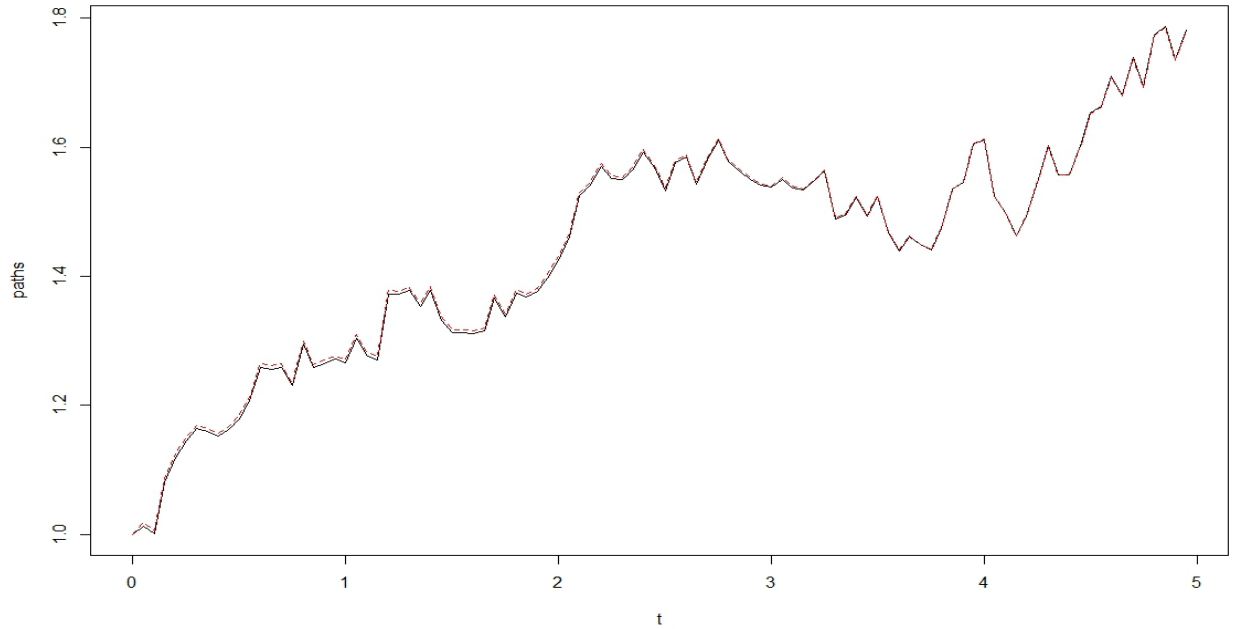

Figure: 1. Exact solution versus mean field solution

MFT approximation

- Suppose now that our objective is to evaluate the expected value of a (nonlinear) function V of X,
$$\mathrm{E}[V(T, X(T))]. \tag{12}$$
In general, this calculation is only possible by means of Monte Carlo simulation.
- Within the mean field approximation, we can compute (12) explicitly. We expand V around the mean field $\overline{X}$ as follows:
$$\begin{aligned} V(T, X(T)) \approx\ & V(T, \overline{X}(T)) + \nabla V(T, \overline{X}(T))^{\mathrm{T}} \xi(T) \\ & + \frac{1}{2} \xi(T)^{\mathrm{T}} \nabla^2 V(T, \overline{X}(T)) \xi(T). \end{aligned} \tag{13}$$
- Its expected value can be now computed explicitly:
$$\mathrm{E}[V(T, X(T))] \approx V(T, \overline{X}(T)) + \frac{1}{2} \operatorname{tr}\left(\mathcal{C}(T) \nabla^2 V(T, \overline{X}(T))\right), \tag{14}$$
where $\mathcal{C}(T)$ is the covariance matrix of the fluctuation process:
$$\mathcal{C}(T) = \mathrm{E}[\xi(T) \xi(T)^{\mathrm{T}}]. \tag{15}$$
- Note that the second term on the right hand side of (14) represents a convexity correction to the mean field value of the expectation.

MFT solution to SABR-LMM

- The MFT equations for $\overline{L}_j$ and $\overline{\sigma}_j$ read:

$$\begin{aligned} \frac{d}{dt}\overline{L}_j(t) &= \Delta_j^{(k)}(t, \overline{L}(t), \overline{\sigma}(t)), \\ \frac{d}{dt}\overline{\sigma}_j(t) &= \Gamma_j^{(k)}(t, \overline{L}(t), \overline{\sigma}(t)), \end{aligned} \tag{16}$$

with

$$\begin{aligned} \overline{L}_j(0) &= L_{j0}, \\ \overline{\sigma_j}(0) &= \sigma_{j0}. \end{aligned} \tag{17}$$

- This is a system of ODEs does not have a closed form solution, and has to be solved numerically. The solution should be cashed.
- The time required to carry out this computation is a fixed overhead which is independent of other resource requirements.

MFT solution to SABR-LMM

- In fact, since the right hand side of (16) is a slowly varying function of t, the following approximation is sufficient for accurate pricing:

$$\begin{aligned} \overline{L}_j(t) &\approx L_{j0} + \int_0^t \Delta_j^{(k)}(s, \overline{L}_0, \overline{\sigma}_0) ds, \\ \overline{\sigma}_j(t) &\approx \sigma_{j0} + \int_0^t \Gamma_j^{(k)}(s, \overline{L}_0, \overline{\sigma}_0) ds. \end{aligned} \tag{18}$$

- The full MFT solution to SABR-LMM reads

$$\begin{aligned} L_j(t) &= \overline{L}_j(t) + \int_0^t \overline{\sigma}_j(s) \overline{L}_j(s)^{\beta_j} dW_j(s) ds, \\ \sigma_j(t) &= \overline{\sigma}_j(t) + \int_0^t \alpha_j(s) \overline{\sigma}_j(s) dZ_j(s), \end{aligned} \tag{19}$$

with $\overline{L}_j$ and $\overline{\sigma}_j$ given by (18).

MFT solution to SABR-LMM

- The covariance matrix (15) is explicitly given by the block matrix

$$\mathcal{C}(T) = \begin{bmatrix} \left\{ \int_0^T \overline{\sigma}_i(t)\, \overline{L}_i(t)^{\beta_i}\, \rho_{ij} \overline{\sigma}_j(t)\, \overline{L}_j(t)^{\beta_j}\, dt \right\} & \left\{ \int_0^T \overline{\sigma}_i(t)\, \overline{L}_i(t)^{\beta_i}\, r_{ij} \alpha_j(t)\, \overline{\sigma}_j(t)\, dt \right\} \\ \left\{ \int_0^T \overline{\sigma}_i(t)\, \overline{L}_i(t)^{\beta_i}\, r_{ij} \alpha_j(t)\, \overline{\sigma}_j(t)\, dt \right\}^{\mathrm{T}} & \left\{ \int_0^T \alpha_i(t)\, \overline{\sigma}_i(t)\, \eta_{ij} \alpha_j(t)\, \overline{\sigma}_j(t)\, dt \right\} \end{bmatrix} \tag{20}$$

- In the following, we will use the abbreviated notation for the matrix of second derivatives:

$$\nabla^2 V = \begin{bmatrix} \left\{ \frac{\partial^2 V}{\partial L_i \partial L_j} \right\} & \left\{ \frac{\partial^2 V}{\partial L_i \partial \sigma_j} \right\} \\ \left\{ \frac{\partial^2 V}{\partial L_i \partial \sigma_j} \right\}^{\mathrm{T}} & \left\{ \frac{\partial^2 V}{\partial \sigma_i \partial \sigma_j} \right\} \end{bmatrix}. \tag{21}$$

Approximate valuation of swaptions

- A key issue is the ability to calibrate SABR-LMM to swaption volatility.
- Remember that a swap rate is a non-linear function of the underlying LIBOR forward rates. The SDE describing the swap rate dynamics that is implied by the SABR-LMM does not have a closed form solution.
- Consequently, pricing swaptions within SABR-LMM requires Monte Carlo simulations. This poses a serious issue for efficient model calibration, as such simulations are noisy and computationally expensive.
- To solve the problem we develop a closed form approximation to the swaption price which can be used to calibrate the model.
- We consider a standard forward starting swap, whose start and end dates are denoted by T_m and T_n, respectively. Recall from Session I that the annuity function of the swap is defined by:

$$A_{mn}(t) = \sum_{m \leq j \leq n-1} \alpha_j P(t, T_{j+1}), \tag{22}$$

where α_j are the day count fractions for fixed rate payments, and where $P(t, T_j)$ are the discount factors.

SABR-LMM dynamics for the forward swap rate

- Using Ito's lemma we see that, under the measure Q_k, the dynamics of the swap rate process can be written in the form:

$$dS_{mn}(t) = \Omega(t, L(t), \sigma(t))dt + \sum_{m \le j \le n-1} \Lambda_j(t, L(t), \sigma(t))dW_j(t), \tag{23}$$

where

$$\Omega = \sum_{m \le j \le n-1} \Delta_j \frac{\partial S}{\partial L_j} + \frac{1}{2} \sum_{m \le i,j \le n-1} \rho_{ij} C_i C_j \frac{\partial^2 S}{\partial L_i \partial L_j}, \tag{24}$$

and

$$\Lambda_j = C_j \frac{\partial S}{\partial L_j}. \tag{25}$$

- Not surprisingly, the stochastic differential equation for S has a drift term: the forward swap rate is not a martingale under a forward measure!
- There are no terms proportional to the dZ_j's in (23), as S does not explicitly depend on the σ_j's.

SABR-LMM dynamics for the forward swap rate

- We now shift to the martingale measure Q_{mn} (the swap measure),

$$dW_{mn}(t) = \frac{\Omega(t, L(t), \sigma(t))dt + \sum_{m \leq j \leq n-1} \Lambda_j(t, L(t), \sigma(t))dW_j(t)}{\nu_{mn}(t)}, \quad (26)$$

where

$$\begin{aligned} \nu_{mn}(t)^2 &= \sum_{m \leq i,j \leq n-1} \rho_{ij} \Lambda_i(t, L(t), \sigma(t)) \Lambda_j(t, L(t), \sigma(t)) \\ &= \Lambda(t, L(t), \sigma(t)) \rho \Lambda(t, L(t), \sigma(t)). \end{aligned} \quad (27)$$

- The process $W_{mn}(t)$ is a Brownian motion.
- The process $\nu_{mn}(t)$ is the stochastic volatility process for the swap rate S_{mn} implied by SABR-LMM. The SDE for the swap rate reads

$$dS_{mn}(t) = \nu_{mn}(t) dW(t). \quad (28)$$

- Our next goal to relate (28) to the vanilla swaption market.

SABR-LMM dynamics for the forward swap rate

- To this end, we define

$$\sigma_{mn}(t) = S(t)^{-\beta_{mn}} \nu_{mn}(t), \tag{29}$$

so that

$$dS_{mn}(t) = \sigma_{mn}(t) S(t)^{\beta_{mn}} dW(t). \tag{30}$$

- Let us now derive the SDE for the volatility process $\sigma_{mn}(t)$.
- Using Ito's lemma we see that, under the measure Q_k, we find that:

$$\begin{aligned} d\sigma_{mn}(t) &= \Phi(t)\, dt + \sum_{m \le j \le n-1} \Theta_j(t)\, dW_j(t) + \Psi_j(t)\, dZ_j(t) \\ &= \Phi(t)\, dt + \Theta(t)^{\mathrm{T}}\, dW(t) + \Psi(t)^{\mathrm{T}}\, dZ(t). \end{aligned} \tag{31}$$

SABR-LMM dynamics for the forward swap rate

- Here $\Phi(t) = \Phi(t, L(t), \sigma(t))$ is given by

$$\begin{aligned}\Phi = &\sum_{m \le j \le n-1} \Delta_j \frac{\partial \sigma_{mn}}{\partial L_j} + \Gamma_j \frac{\partial \sigma_{mn}}{\partial \sigma_j} \\ &+ \frac{1}{2} \sum_{m \le i,j, \le n-1} \rho_{ij} C_i C_j \frac{\partial^2 \sigma_{mn}}{\partial L_i \partial L_j} + 2 r_{ij} C_i \Gamma_j \frac{\partial^2 \sigma_{mn}}{\partial L_i \partial \sigma_j} + \eta_{ij} \Gamma_i \Gamma_j \frac{\partial^2 \sigma_{mn}}{\partial \sigma_i \partial \sigma_j},\end{aligned} \tag{32}$$

and

$$\begin{aligned}\Theta_j &= C_j \frac{\partial \sigma_{mn}}{\partial L_j}, \\ \Psi_j &= D_j \frac{\partial \sigma_{mn}}{\partial \sigma_j}.\end{aligned} \tag{33}$$

- We now define a new Brownian motion,

$$dZ_{mn}(t) = \frac{\Phi(t)\, dt + \Theta_j(t)^{\mathrm{T}}\, dW(t) + \Psi(t)^{\mathrm{T}}\, dZ(t)}{\zeta_{mn}(t)}. \tag{34}$$

SABR-LMM dynamics for the forward swap rate

- Here

$$\zeta_{mn}(t)^2 = \Theta(t)^{\mathrm{T}} \rho \Theta(t) + \Theta(t)^{\mathrm{T}} (r + r^{\mathrm{T}}) \Psi(t) + \Psi(t)^{\mathrm{T}} \eta \Psi(t)$$

  is a stochastic vol of vol process, and thus

$$d\sigma_{mn}(t) = \zeta_{mn}(t)\, dZ_{mn}(t).$$

- The correlation ρ_{mn} between dW_{mn} and dZ_{mn},

$$dW_{mn}(t)\, dW_{mn}(t) = \rho_{mn}(t)\, dt$$

  is given by:

$$\rho_{mn}(t) = \frac{\Lambda(t)^{\mathrm{T}} \rho \Theta(t) + \Lambda(t)^{\mathrm{T}} r \Psi(t)}{\nu_{mn}(t)\, \zeta_{mn}(t)}. \tag{35}$$

SABR-LMM dynamics for the forward swap rate

- Since $\zeta_{mn}(t) > 0$ we can write it in the form

$$\zeta_{mn}(t) = \alpha_{mn}(t)\,\sigma_{mn}(t),$$

where $\alpha_{mn}(t)$ is a process, so that

$$d\sigma_{mn}(t) = \alpha_{mn}(t)\,\sigma_{mn}(t)\,dZ_{mn}(t). \tag{36}$$

- The initial value of σ_{mn} is the deterministic constant $\sigma_{mn}(0)$.
- Pricing a swaption using the system (30) - (36) is equivalent to pricing using the original state variables; we have merely switched to the annuity measure.
- The system (30) - (36) looks very much like the SABR dynamics. There is a fly in the ointment: both $\alpha_{mn}(t)$ and $\rho_{mn}(t)$ are stochastic.

SABR-LMM dynamics for the forward swap rate

- We *approximate* this system with the system below where $\alpha_{mn}(t)$ and $\alpha_{mn}(t)$ are replaced with their MFT approximations given by (14):

$$\begin{aligned}\overline{\alpha}_{mn}(t) &\approx \alpha(t,\overline{L},\overline{\sigma}) + \frac{1}{2}\operatorname{tr}\left(\mathcal{C}(t)\nabla^2\alpha(t,\overline{L},\overline{\sigma})\right),\\ \overline{\rho}_{mn}(t) &\approx \rho(t,\overline{L},\overline{\sigma}) + \frac{1}{2}\operatorname{tr}\left(\mathcal{C}(t)\nabla^2\rho(t,\overline{L},\overline{\sigma})\right),\end{aligned} \tag{37}$$

  where $\mathcal{C}(t)$ is the covariance matrix defined by (11).
- This leads to the following values of the swaption α and ρ parameters:

$$\begin{aligned}\overline{\alpha}_{mn} &= \sqrt{\frac{1}{T_m}\int_0^{T_m}\overline{\alpha}_{mn}(t)^2\,dt},\\ \overline{\rho}_{mn} &= \frac{1}{T_m\,\overline{\alpha}_{mn}}\int_0^{T_m}\overline{\alpha}_{mn}(t)\,\overline{\rho}_{mn}(t)\,dt.\end{aligned} \tag{38}$$

Optimization

- Optimization consists in determining the SABR-LMM model parameters, so that its prices of the benchmark caps / floors and swaptions match closely the market prices.
- We have already observed that SABR-LMM can be exactly calibrated to the caps. For simplicity of exposition, we assume that this has been done; in practice one usually calibrates simultaneously to caps and swaptions.
- We select the set of swaptions for calibration: since we wish to calibrate the model globally, we may choose the entire swaption matrix. Each benchmark swaption is represented by the market observed SABR parameters $\sigma_{mn,0}$, α_{mn}, β_{mn}, and ρ_{mn}.
- To this end, for each benchmark swaption we consider the objective function:

$$\begin{aligned}\mathcal{L}_{mn} = \frac{1}{2}\Big(&(\sigma_{mn}(0) - \sigma_{mn,0})^2 + (\overline{\alpha}_{mn} - \alpha_{mn})^2 \\ &+ (a_{mn}^{\mathrm{T}}\beta + b_{mn} - \beta_{mn})^2 + (\overline{\rho}_{mn} - \rho_{mn})^2\Big),\end{aligned} \tag{39}$$

where $\sigma_{mn}(0)$ is the $t = 0$ value of the process (29), $\overline{\alpha}_{mn}$ and ρ_{mn} are given by (38), and $a_{mn}^{\mathrm{T}}\beta + b_{mn}$ is defined in equation (32) of Presentation IV.

Optimization

- The total objective function is given to the sum of $\mathcal{L}_{mn}$ over all benchmark swaptions,

$$\mathcal{L} = \sum_{m,n} \mathcal{L}_{mn}. \tag{40}$$

- Depending on the choice of parameterization, $\mathcal{L}$ is a function of moderate number of variables. It can be minimized using an efficient optimization algorithm such the Levenberg-Marquardt algorithm.
- As noticed above, it is advantageous to optimize jointly to caps and swaptions (rather than setting the cap volatility parameters), in which case (40) is supplemented by addition terms coming from the benchmark caps.
- Also, a Tikhonov style penalty term may added to (40) to assure a smooth structure of the calibrated model parameters.

Monte Carlo method for SABR-LMM

- SABR-LMM does not allow for a natural implementation based on recombining trees, and thus all valuations have to be performed via Monte Carlo simulations.
- We shall describe two numerical schemes for generating Monte Carlo paths for SABR-LMM: Euler's scheme and Milstein's scheme. They both rely on replacing continuous time stochastic differential equations by suitable finite difference schemes.
- First, we perform a reduction in the number of stochastic drivers of the system.

Factor reduction

- In a market where the forward curve spans 30 years, there are 120 quarterly LIBOR forwards and 120 corresponding stochastic volatilities. Such a process is driven by 240 stochastic factors.
- Having a large number of factors is neither desirable nor necessary for the model implementation. On the performance side, the "curse of dimensionality" kicks in, leading to unacceptably slow performance. On the financial side, overabundance of parameters may result in their instability.
- It is known that the dimensionality of the system can be reduced without substantial loss of information: PCA shows that three factors capture virtually all of the curve and volatility matrix dynamics.
- We are thus led to the idea that only a small number d of independent Brownian motions $w_a(t)$, $a = 1, \ldots, d$ (d =1, 2, 3, or 4), with

$$dw_a(t)\, dw_b(t) = \delta_{ab} dt\,, \tag{41}$$

should drive the rates process, while the independent Brownian motions $z_a(t)$, $a = 1, \ldots, d$,

$$dz_a(t)\, dz_b(t) = \delta_{ab} dt \tag{42}$$

drive the volatilities process.

Factor reduction

- We set

$$\begin{aligned} dW_j(t) &= \sum_{1\le a\le d} U_{ja}\, dw_a(t)\,, \\ dZ_j(t) &= \sum_{1\le a\le d} V_{ja}\, dz_a(t)\,, \end{aligned} \tag{43}$$

  where U and V are $N \times d$ matrices such that $UU^{\mathrm{T}} \approx \rho$ and $VV^{\mathrm{T}} \approx \eta$, respectively. U and V can be determined from PCA.
- We can easily rewrite the dynamics of the model in terms of the independent Brownian motions:

$$\begin{aligned} dL_j(t) &= \Delta_j(t)\, dt + \sum_{1\le a\le d} C_j(t)\, U_{ja}\, dw_a(t)\,, \\ d\sigma_j(t) &= \Gamma_j(t)\, dt + \sum_{1\le a\le d} D_j(t)\, V_{ja}\, dz_a(t)\,. \end{aligned} \tag{44}$$

- This system is the *factor reduced* SABR-LMM dynamics.
- For notational ease, we continue using the Brownian motions W_j and Z_j, with the understanding that they are defined in terms of the independent Brownian motion by means of (43).

One factor Brownian motion

- Next, we discuss briefly an efficient, low simulation noise methodology for generation of Brownian paths.
- There exist several more of less refined methods for simulating a Brownian motion; here we compare two of them.
- The *random walk method* is easy to implement at the expense of being rather noisy. It represents a Brownian motion as a random walk sampled at a finite set of event dates $t_0 < t_1 < \ldots < t_m$:

$$\begin{aligned} Z(t_{-1}) &= 0, \\ Z(t_n) &= Z(t_{n-1}) + \sqrt{t_n - t_{n-1}}\, \xi_n, \quad n = 0, \ldots, m, \end{aligned} \tag{45}$$

  where $t_{-1} = 0$, and where ξ_n are i.i.d. random variables with $\xi_n \sim N(0, 1)$.
- A good method of generating the ξ_n's is to first generate a sequence of uniform pseudorandom numbers u_n (using, say, the Mersenne twister algorithm), and then set

$$\xi_n = N^{-1}(u_n), \tag{46}$$

  where $N^{-1}(x)$ is the inverse cumulative normal function. $N^{-1}(x)$ can be efficiently and accurately computed using e.g. the Beasley-Springer-Moro algorithm, see [1].

One factor Brownian motion

- The *spectral decomposition method* generally leads to much better performance than the random walk method. It assures that the simulated process has the same covariance matrix $\mathcal{C}$ as the Brownian motion $Z(t)$ sampled at $t_0, t_1, \ldots, t_m$.
- The covariance matrix is explicitly given by:

$$\begin{aligned} \mathcal{C}_{ij} &= \mathrm{E}\left[Z(t_i)Z(t_j)\right] \\ &= \min(t_i,\ t_j). \end{aligned} \tag{47}$$

- Consider the eigenvalue problem for $\mathcal{C}$:

$$\mathcal{C}E_j = \lambda_j E_j, \qquad j = 0, \ldots, m, \tag{48}$$

with orthonormal E_j's.

- Since the covariance matrix $\mathcal{C}$ is positive definite, all of its eigenvalues λ_j are nonnegative, and we will assume that

$$\lambda_0 \geq \ldots \geq \lambda_m \geq 0. \tag{49}$$

One factor Brownian motion

- We will denote the n-th component of the vector E_j by $E_j(t_n)$, and consider the random variable

$$Z(t_n) = \sum_{0 \le j \le m} \sqrt{\lambda_j}\, E_j(t_n)\xi_j, \tag{50}$$

  where ξ_j are, again, i.i.d. random variables with $\xi_j \sim N(0,\, 1)$.
- These numbers are best calculated by applying the inverse cumulative normal function to a sequence of Sobol numbers. Alternatively, one could use a sequence of uniform pseudorandom numbers; this, however, leads to a higher sampling variance.
- Then, for each $n = 0, \ldots, m$, $Z(t_n) \sim N(0,\, t_n)$, and

$$\begin{aligned} \mathrm{E}\left[Z(t_i)Z(t_j)\right] &= \sum_{0 \le k \le 0} \lambda_k\, E_k(t_i)E_k(t_j) \\ &= \mathcal{C}_{ij}. \end{aligned} \tag{51}$$

- We can thus regard $Z(t_n)$ a realization of the discretized Wiener process[1]. For computational efficiency, we may want to truncate (50) at some $p < m$. This eliminates the *high frequencies* from $Z(t_n)$, and lowers the variance. The price for this may be systematically lower accuracy.

[1] This realization of the discretized Brownian motion is related to the well known Karhounen-Loeve expansion of the (continuous time) Brownian motion.

Multi factor Brownian motion

- We now consider the case of a multi-factor Brownian motion $Z_a(t)$, with

$$\mathrm{E}[dZ_a(t)\, dZ_b(t)] = \rho_{ab} dt.$$

- The Cholesky decomposition of ρ yields

$$\rho = LL^{\mathrm{T}}, \tag{52}$$

where L is a $d \times d$ dimensional, lower triangular matrix.

- For example, if

$$\rho = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix}, \tag{53}$$

then

$$L = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho_{12}^2} \end{bmatrix}, \tag{54}$$

Multi factor Brownian motion

- Now, if $X \in \mathbb{R}^d$ is a vector of independent standard normal variables, then LX is a multivariate normal variable with correlation matrix ρ.
- Indeed,

$$\begin{aligned}
\mathrm{E}[(LX)_a(LX)_b] &= \sum_{0\leq k,l\leq d} L_{ak}L_{bl}\mathrm{E}[X_kX_l] \\
&= \sum_{0\leq k,l\leq d} L_{ak}L_{bl}\delta_{kl} \\
&= \sum_{0\leq k\leq d} L_{ak}L_{bk} \\
&= \sum_{0\leq k\leq d} L_{ak}(L^{\mathrm{T}})_{kb} \\
&= (LL^{\mathrm{T}})_{kl} \\
&= \rho_{kl}.
\end{aligned}$$

Discretizing SABR-LMM

- Finally, we discretize the system of SDEs defining the SABR-LMM process.
- We choose a sequence of event dates $t_0, t_1, \ldots, t_m$, and denote by $L_{j,k} \simeq L_j(t_k)$ and $\sigma_{j,k} \simeq \sigma_j(t_k)$ the approximate solution. We also set

$$\begin{aligned} \Delta_{j,k} &= \Delta_j(t_k, L_k), \\ C_{j,k} &= C_j(t_k, L_{j,k}), \\ &\ldots. \end{aligned} \tag{55}$$

- We let $\delta_k = t_{k+1} - t_k$ denote the k-th time step, and we let

$$\begin{aligned} \Delta W_{j,k} &= W_j(t_{k+1}) - Z_j(t_k), \\ \Delta Z_{j,k} &= Z_j(t_{k+1}) - Z_j(t_k) \end{aligned} \tag{56}$$

denote the increments of the Brownian motions.

Discretizing SABR-LMM: the log-Euler scheme

- To assure that the simulated values of the volatility parameters are positive, we introduce the state variables ζ by

$$\sigma_j(t) = \exp\left(\int_0^t \alpha_j(s)\, d\zeta_j(s) - \frac{1}{2}\int_0^t \alpha_j(s)^2\, ds\right). \tag{57}$$

- The dynamics of ζ_j reads:

$$d\zeta_j(t) = \gamma_j(t)\, dt + dZ_j(t), \tag{58}$$

where the drift terms $\gamma_j(t)$ are given by

$$\gamma_j(t) = \begin{cases} -\sum_{j+1\le i\le k} \dfrac{r_{ji}\delta_i\sigma_i(t) L_i(t)^{\beta_i}}{1+\delta_i F_i(t)}, & \text{if } j<k, \\ 0, & \text{if } j=k, \\ \sum_{k+1\le i\le j} \dfrac{r_{ji}\delta_i\sigma_i(t) L_i(t)^{\beta_i}}{1+\delta_i F_i(t)}, & \text{if } j>k. \end{cases} \tag{59}$$

under Q_k, and

$$\gamma_j(t) = \sum_{\gamma(t)\le i\le j} \frac{r_{ji}\delta_i\sigma_i(t) L_i(t)^{\beta_i}}{1+\delta_i F_i(t)}, \tag{60}$$

under the spot measure.

Discretizing SABR-LMM: the log-Euler scheme

- Discretization of the SDE for L_j is straightforward:

$$\begin{aligned} L_{j,k+1} &= \left(L_{j,k} + \Delta_{j,k}\delta_k + \sigma_{j,k} L_{j,k}^{\beta_j} \Delta W_{j,k}\right)^+, \\ L_{j,0} &= L_{j0}. \end{aligned} \tag{61}$$

- Equation (58) leads to the following discretization of (57):

$$\begin{aligned} \sigma_{j,k+1} &= \sigma_{j,k} \exp\left(\alpha_{j,k}\Delta Z_{j,k} + (\zeta_{j,k} - \alpha_{j,k}^2/2)\delta_k\right) \\ \sigma_{j,0} &= \sigma_{j0}. \end{aligned} \tag{62}$$

- We refer to the numerical scheme (61) - (62) as the log-Euler scheme for SABR-LMM.

Discretizing SABR-LMM: the quasi Milstein scheme

- The structure of the SABR-LMM model is incompatible with the Milstein scheme: second order (in Brownian motion increments) discretization contains Levy area terms of the form $\int_t^{t+\delta} dW_j(s)\, dZ_j(s)$.
- Remarkably, no Levy area terms

$$\begin{aligned} &\int_t^{t+\delta} dW_i(s)\, dW_j(s), \\ &\int_t^{t+\delta} dZ_i(s)\, dZ_j(s), \\ &\int_t^{t+\delta} dW_i(s)\, dZ_j(s), \end{aligned} \tag{63}$$

for $i \neq j$, are present in the second order dicretization.

Discretizing SABR-LMM: the quasi Milstein scheme

- This observation leads to the following scheme:

$$
\begin{aligned}
L_{j,k+1} &= \left(L_{j,k} + \Delta_{j,k}\delta_k + \sigma_{j,k} L_{j,k}^{\beta_j} \Delta W_{j,k} + \frac{\beta_j}{2}\sigma_{j,k}^2 L_{j,k}^{2\beta_j - 1}(\Delta W_{j,k}^2 - \delta_k)\right)^+, \\
\sigma_{j,k+1} &= \sigma_{j,k} \exp\left(\alpha_{j,k}\Delta Z_{j,k} + (\zeta_{j,k} - \alpha_{j,k}^2/2)\delta_k\right).
\end{aligned}
\tag{64}
$$

- We refer to this discretization scemen as the quasi Milstein scheme for SABR-LMM. It is more accurate than the Euler scheme.

Efficient computation of drift coefficients

- Note that these deceivingly simply looking recursions involve the drift terms $\Delta_{j,k}$ and $\gamma_{j,k}$, which get updated at each time step. Each of the drift terms is a sum of a large number of terms, and their evaluation is computationally expensive.
- Evaluating the drift terms along each Monte Carlo path accounts for well over 50% of total simulation time.
- On the other hand, the drift term contributions are relatively small as compared to the initial values of the LIBOR forwards and volatilities.
- It is desirable to develop an efficient methodology for accurate approximate evaluation of the drift terms.
- We describe such an efficient methodology based on the mean field solution to the model.

Efficient computation of drift coefficients

- The simplest and most computationally efficient approach is the frozen curve approximation. We simply evaluate the drifts Δ_j and γ_j at today's values of L_j and σ_j,

$$\begin{aligned} \Delta_j(t, L(t), \sigma(t)) &\approx \Delta_j(t, L_0, \sigma_0), \\ \gamma_j(t, L(t), \sigma(t)) &\approx \gamma_j(t, L_0, \sigma_0), \end{aligned} \tag{65}$$

  and store the result. This is a fixed overhead cost, as the same drifts can be reused for all Monte Carlo paths.
- The frozen curve approximation is insufficient for instruments with optionality going beyond two years.

Efficient drift calculation

- A far superior methodology is based on the MFT solution $\overline{L}_j$ and $\overline{\sigma}_j$ to the LMM-SABR model discussed in Presentation IV.
- Specifically, we make the approximation:

$$\begin{aligned}\Delta_j(t, L(t), \sigma(t)) &\approx \Delta_j(t, \overline{L}(t), \overline{\sigma}(t)) + \frac{1}{2}\operatorname{tr}\left(\mathcal{C}(t)\nabla^2\Delta_j(t, \overline{L}(t), \overline{\sigma}(t))\right), \\ \gamma_j(t, L(t), \sigma(t)) &\approx \gamma_j(t, \overline{L(t)}, \overline{\sigma}(t)) + \frac{1}{2}\operatorname{tr}\left(\mathcal{C}(t)\nabla^2\gamma_j(t, \overline{L}(t), \overline{\sigma}(t))\right),\end{aligned} \tag{66}$$

  where the covariance matrix $\mathcal{C}(t)$ is given by equation (11).
- This computation is a fixed overhead cost, and the drift values can be reused for all Monte Carlo paths.
- The accuracy of the MFT approximation for instruments of all maturities is basically indistinguishable from simulations involving full evaluation of the drift terms.

References


Glasserman, P.: *Monte Carlo Methods in Financial Engineering*, Springer Verlag (2003).


Kloeden, P. E., and Platen, E.: *Numerical Solution of Stochastic Differential Equationse*, Springer Verlag (1992).