6

Implementation and calibration of the LMM

In Chapter 5 we suggested possible parametric forms for both the instantaneous volatility and correlation. Here we use the functional form (5.25) for the instantaneous volatility. We also modelled the correlation between different forward rates by first deriving a sample correlation matrix (see Figure 5.4) based on historical data and then fitting this to a smooth parametric form so as to obtain a full-rank correlation matrix with the desired properties (Figure 5.5). In the present chapter we examine ways of taking this correlation matrix and producing a low-rank approximation convenient for numerical simulation.

In many applications the Monte Carlo method is the only feasible way to simulate the LMM dynamics. It is also the most general and readily applicable numerical method. For this reason, we discuss how to simulate the LMM dynamics in a Monte Carlo framework, starting with the popular Euler scheme, followed by the predictor-corrector method.

Another key issue is calibration. The model should reproduce the vanilla market. In other words, we need to calibrate the model parameters to the prices of traded swaptions and caps/floors. In effect, we use a stochastic interest rate model such as the LMM to infer the price of an exotic option from the market prices of vanilla instruments.

Since market prices are quoted as implied Black volatilities for a given maturity and strike, we also have to incorporate the volatility smile. To

https://doi.org/10.1017/CBO9781139035033.007

keep the presentation simple we do not discuss the volatility smile until Chapter 8.

6.1 Rank reduction

In a typical application of the LMM, for reasons of numerical efficiency, we seldom use the LMM formulation (5.2) whose correlation structure is modelled by a rank-n correlation matrix given by (5.1). Rather, we adopt an LMM formulation similar to (5.18), namely

$$dF_i(t) = \mu_i^j(t)F_i(t)dt + \sigma_i(t)F_i(t)\sum_{k=1}^{m}\eta_{i,k}dW_k^j(t),$$

but with the n forward rates driven by an m-dimensional Brownian motion $(W_1^j(t), \dots, W_m^j(t))$, where $m < n$ (in fact sometimes $m \ll n$). The correlation matrix we use in this approach is a rank-m (rather than n) matrix given by

$$\rho_{i,j} = \sum_{k=1}^{m}\eta_{i,k}\eta_{j,k},$$

where $\eta = (\eta_{i,k})$ is an $n \times m$ matrix.

Denote by $\bar{\rho}$ an exogenously given correlation matrix of rank n, which we could take to be the sample correlation matrix in Figure 5.4, some smooth parametric form as in Figure 5.5 or another correlation structure. What we want is a rank-m correlation matrix ρ that approximates the exogenous full-rank matrix $\bar{\rho}$.

Spectral decomposition

One way of approximating the full-rank $n\times n$ correlation matrix $\bar{\rho}$ by a rank $m < n$ matrix ρ is to use spectral decomposition. Being a real symmetric $n \times n$ matrix, $\bar{\rho}$ can be represented as

$$\bar{\rho} = \bar{U}\bar{\Lambda}\bar{U}^\top,$$

where $\bar{U}$ is an orthonormal matrix whose columns are the eigenvectors of $\bar{\rho}$ and $\bar{\Lambda}$ is a diagonal matrix whose elements are the corresponding eigenvalues. The correlation matrix $\bar{\rho}$ is positive definite if and only if all the eigenvalues are positive. If all the eigenvalues are non-negative, the matrix is positive semidefinite. However, a correlation matrix derived using

https://doi.org/10.1017/CBO9781139035033.007

historical data such as the sample correlation matrix in Figure 5.4 may have some negative eigenvalues.

If the number of positive eigenvalues is greater than m, we can create a valid correlation matrix ρ of rank m as follows. Sort the eigenvalues in descending order $\lambda_1 \geq \cdots \geq \lambda_n$, and use the first m eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$ to form a new $m \times m$ diagonal matrix Λ and a new $n \times m$ matrix U whose columns are the eigenvectors corresponding to eigenvalue entries in Λ. Then an $n \times m$ matrix η can be defined as

$$\eta = DU\Lambda^{1/2},$$

where D is an $n \times n$ diagonal 'scaling' matrix chosen so that the row vectors of η have unit length. The diagonal entries of D are

$$d_i = \left(\sum_{k=1}^{m} U_{ik}^2 \lambda_k \right)^{-1/2}$$

for $i = 1, \ldots, n$. Finally, we put

$$\rho = \eta\eta^\top,$$

which is a real symmetric positive definite matrix of rank m with unit diagonal entries.

Hypersphere decomposition

Another popular method for creating a low-rank correlation matrix is via **hypersphere decomposition**. In this method the row vectors of matrix η are given by points on the surface of a unit hypersphere in m dimensions in terms of angular coordinates. The elements of the $n \times m$ matrix η are expressed in terms of an $n \times (m - 1)$ matrix of angular coordinates θ as

$$\eta_{i,j}(\theta) = \cos\theta_{i,j} \prod_{k=1}^{j-1} \sin\theta_{i,k}, \qquad 1 \leq j < m, \tag{6.1a}$$

$$\eta_{i,m}(\theta) = \prod_{k=1}^{m-1} \sin\theta_{i,k}, \tag{6.1b}$$

where we adopt the convention that $\prod_{k=1}^{0} \sin\theta_{i,k} = 1$. Then we put

$$\rho_{i,j}(\theta) = \sum_{k=1}^{m} \eta_{i,k}(\theta)\eta_{j,k}(\theta) \tag{6.2}$$

for each $i, j = 1, \dots, n$, that is,

$$\rho(\theta) = \eta(\theta)\eta(\theta)^\top.$$

Example 6.1

In the case of a four-factor correlation structure, i.e. when four independent Brownian motions are used, the reduced-rank correlation structure based on angular coordinates is given by an $n \times 4$ matrix η whose ith row is

$$\begin{bmatrix} \cos\theta_{i,1} & \sin\theta_{i,1}\cos\theta_{i,2} & \sin\theta_{i,1}\sin\theta_{i,2}\cos\theta_{i,3} & \sin\theta_{i,1}\sin\theta_{i,2}\sin\theta_{i,3} \end{bmatrix}$$

for $i = 1, \dots, n$.

Exercise 6.1 Show that the Euclidean norm of each row of $\eta(\theta)$ is equal to 1, and therefore all the diagonal elements of $\rho(\theta)$ are equal to 1.

Exercise 6.2 Show that $\rho(\theta)$ is a positive semidefinite matrix.

We use numerical optimisation to calculate the angular parameters for a given exogenous correlation matrix $\bar{\rho}$:

$$\min_{\theta} \sum_{i,j=1}^{n} \left(\bar{\rho}_{i,j} - \rho_{i,j}(\theta)\right)^2. \tag{6.3}$$

Example 6.2

Taking the exogenous correlation matrix $\bar{\rho}$ to be the smooth parametric correlation matrix in Figure 5.5, we can perform the optimisation (6.3) to obtain a rank-4 correlation matrix $\rho(\theta)$ as shown in Figure 6.1.

Parameter reduction

When performing the optimisation (6.3), we are solving for $n\,(m-1)$ angular parameters, which could be more than the number $\frac{n(n-1)}{2}$ of parameters

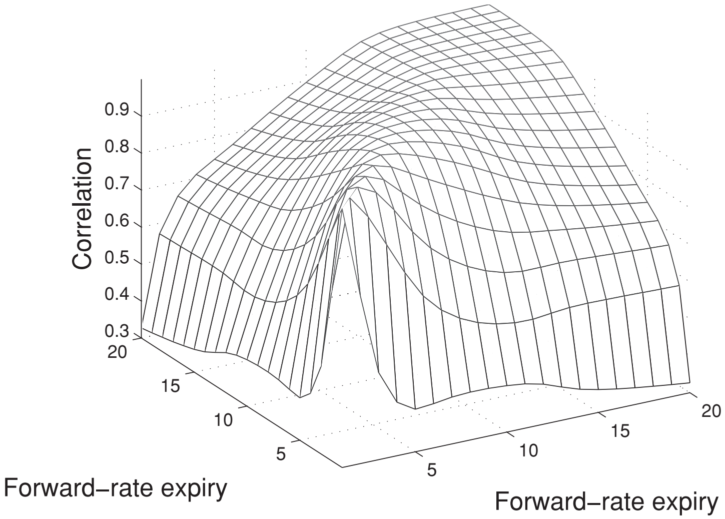

Figure 6.1 The rank-4 correlation matrix $\rho(\theta)$ after calibration to the correlation matrix in Figure 5.5.

needed to characterise the $n \times n$ correlation matrix, a symmetric matrix with all diagonal entries equal to 1. In particular, when $m = n$, the number of angular parameters is $n(n-1)$, double that needed to characterise the correlation matrix.

We can utilise the rotational symmetry of the unit hypersphere in m dimensions to reduce the number of angular parameters. The entries of the correlation matrix ρ are determined by the scalar products of the unit vectors (i.e. points on the hypersphere) corresponding to the rows of the matrix η. Rotations of the hypersphere do not alter these scalar products, which makes it possible to fix some of the angular parameters.

Namely, rotating the hypersphere allows us to align the first row in η with the unit vector

$$(1, 0, \dots, 0) \in \mathbb{R}^m,$$

which corresponds to setting

$$\theta_{1,1} = \theta_{1,2} = \cdots = \theta_{1,m-1} = 0.$$

While keeping the first row fixed, we can rotate the hypersphere once again

https://doi.org/10.1017/CBO9781139035033.007

to align the second row of η with the unit vector

$$(\cos\theta_{2,1}, \sin\theta_{2,1}, 0, \ldots, 0) \in \mathbb{R}^m.$$

This corresponds to setting

$$\theta_{2,2} = \cdots = \theta_{2,m-1} = 0.$$

The rotational symmetry of the hypersphere allows us to proceed in this manner up to and including the $(m-1)$st row in η, setting

$$\begin{aligned} \theta_{1,1} = \theta_{1,2} = \cdots = \theta_{1,m-1} &= 0, \\ \theta_{2,2} = \cdots = \theta_{2,m-1} &= 0, \\ &\vdots \\ \theta_{m-1,m-1} &= 0. \end{aligned}$$

This reduces the number of angular parameters by $\frac{m(m-1)}{2}$.

Exercise 6.3 Show that the number of angular parameters remaining after the reduction does not exceed the number of parameters characterising the $n \times n$ correlation matrix, that is,

$$n(m-1) - \frac{m(m-1)}{2} \leq \frac{n(n-1)}{2}$$

for any $m \leq n$.

Example 6.3

When $m = 4$ (four independent Brownian motions), the matrix $\eta(\theta)$ is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \cos\theta_{2,1} & \sin\theta_{2,1} & 0 & 0 \\ \cos\theta_{3,1} & \cos\theta_{3,2}\sin\theta_{3,1} & \sin\theta_{3,1}\sin\theta_{3,2} & 0 \\ \cos\theta_{4,1} & \cos\theta_{4,2}\sin\theta_{4,1} & \cos\theta_{4,3}\sin\theta_{4,1}\sin\theta_{4,2} & \sin\theta_{4,1}\sin\theta_{4,2}\sin\theta_{4,3} \\ \vdots & \vdots & \vdots & \vdots \\ \cos\theta_{n,1} & \cos\theta_{n,2}\sin\theta_{n,1} & \cos\theta_{n,3}\sin\theta_{n,1}\sin\theta_{n,2} & \sin\theta_{n,1}\sin\theta_{n,2}\sin\theta_{n,3} \end{bmatrix}$$

Therefore we need to solve for $3(n-2)$ unknowns in the above as opposed to $3n$ unknowns in the original parameterisation.

6.2 Monte Carlo simulation

Consider a European derivative security with expiry date T_j and payoff of the form $h(F_1(T_j), \ldots, F_n(T_j))$ that depends on the forward rates $F_i(T_j) = F(T_j; T_{i-1}, T_i)$ for $i = j + 1, \ldots, n$ as well as on the LIBOR rates that will have set by time T_j and are denoted here by $F_i(T_j) = L(T_{i-1}, T_i)$ for $i = 1, \ldots, j$. The price at time 0 of this option can be expressed as

$$B(0, T_n)\mathbb{E}_{P_{T_n}}\left(\frac{h(F_1(T_j), \ldots, F_n(T_j))}{B(T_j, T_n)}\right),$$

where we take the bond $B(t, T_n)$ as numeraire and write the option price in terms of the expectation with respect to the terminal measure P_{T_n}.

We want to compute this expectation by using the LMM dynamics of the forward rates under the chosen measure. However, since the LMM dynamics does not give rise to a known distribution, we need to employ some form of numerical approximation. Given the large number of state variables, we will be better off using Monte Carlo simulation rather than finite difference techniques.

To evaluate the expected payoff of an option via Monte Carlo simulation we simulate the evolution through time (called a path) of the set of spanning forward rates $F_1(t), F_2(t), \ldots, F_n(t)$ and compute the payoff discounted by the numeraire. Repeating the simulation, we compute the discounted payoff values for several different paths. Finally, we calculate the mean of these discounted payoffs to obtain an estimate of the expected payoff.

There are a number of ways of evolving the entire forward LIBOR curve through time that are consistent with the model dynamics. We assume the LMM dynamics is formulated in terms of a set of $m \leq n$ orthogonal Brownian motions, see Section 6.1. With the T_n-bond $B(t, T_n)$ chosen as numeraire, the dynamics of the forward rates is given as

$$dF_i(t) = \mu_i^n(F(t), t)F_i(t)dt + \sigma_i(t)F_i(t)\sum_{k=1}^{m}\eta_{i,k}dW_k^n(t), \tag{6.4}$$

where the $\sigma_i(t)$ are the instantaneous volatilities and the instantaneous correlations are

$$\rho_{i,j} = \sum_{k=1}^{m}\eta_{i,k}\eta_{j,k},$$

and where we make the dependence of the drift on the vector $F(t) = (F_1(t), \ldots, F_n(t))$ of forward rates explicit. By (5.13), the drift can be ex-

https://doi.org/10.1017/CBO9781139035033.007

pressed as

$$\mu_i^n(F(t),t) = -\sum_{k=i+1}^{n} \frac{\tau_k \rho_{k,i} \sigma_i(t) \sigma_k(t) F_k(t)}{1+\tau_k F_k(t)} \tag{6.5}$$

for any $t \in [0, T_{i-1}]$.

The simplest time-stepping procedure is the **Euler scheme**. Given numerically estimated forward rates $\bar{F}_i(t)$ at time t, we can estimate the rates at time $t + \Delta t$ as

$$\bar{F}_i(t+\Delta t) = \bar{F}_i(t) + \mu_i^n(\bar{F}(t),t)\Delta t + \sigma_i(t)\sqrt{\Delta t}\sum_{k=1}^{m} \eta_{i,k}\varepsilon_k,$$

where $\varepsilon_1, \ldots, \varepsilon_m$ is a vector of m independent random variables, each following the standard normal distribution $N(0,1)$. The step size Δt is often taken to be the forward-rate accrual period, with the simulation times chosen to match the setting dates $T_0, \ldots, T_{n-1}$ of the forward LIBOR rates.

Remark 6.4

A particular drawback with the Euler scheme is its slow rate of convergence. We can improve convergence by working with the logarithms of the forward LIBOR rates rather than the rates themselves. Applying Itô's formula to the LMM dynamics, we have

$$d(\ln F_i(t)) = \left(\mu_i^n(F(t),t) - \frac{1}{2}\sigma_i(t)^2\right)dt + \sigma_i(t)\sum_{k=1}^{m} \eta_{i,k}\, dW_k^n(t).$$

This leads to the Euler scheme

$$\ln \bar{F}_i(t+\Delta t) = \ln \bar{F}_i(t) + \left(\mu_i^n(\bar{F}(t),t) - \frac{1}{2}\sigma_i(t)^2\right)\Delta t + \sigma_i(t)\sqrt{\Delta t}\sum_{k=1}^{m} \eta_{i,k}\varepsilon_k, \tag{6.6}$$

where $\varepsilon_1, \ldots, \varepsilon_m$ are as above.

Predictor-corrector method

A popular way of improving the Euler scheme is to apply a **predictor-corrector method**. Given predictor-corrector estimates $\bar{F}_i(t)$ of the rates at time t, we first calculate estimates $\tilde{F}_i(t+\Delta t)$ of the forward LIBOR rates at time $t + \Delta t$ by using the Euler scheme, namely

$$\ln \tilde{F}_i(t+\Delta t) = \ln \bar{F}_i(t) + \left(\mu_i^n(\bar{F}(t),t) - \frac{1}{2}\sigma_i(t)^2\right)\Delta t + \sigma_i(t)\sqrt{\Delta t}\sum_{k=1}^{m} \eta_{i,k}\varepsilon_k.$$

https://doi.org/10.1017/CBO9781139035033.007

We then recompute the drift term by using the estimated LIBOR rates $\tilde{F}_i(t+\Delta t)$, that is, evaluate $\mu_i^n(\tilde{F}(t+\Delta t), t+\Delta t)$. Finally, we re-estimate the forward LIBOR rates using the same vector of standard normals $\varepsilon_1, \dots, \varepsilon_m$, but with a new 'predictor-corrector' drift. This can be written as

$$\begin{aligned}\ln \bar{F}_i(t+\Delta t) = \ln \bar{F}_i(t) \\ &+ \left(\theta\mu_i^n(\bar{F}(t), t) + (1-\theta)\mu_i^n(\tilde{F}(t+\Delta t), t+\Delta t) - \frac{1}{2}\sigma_i(t)^2\right)\Delta t \\ &+ \sigma_i(t)\sqrt{\Delta t}\sum_{k=1}^{m}\eta_{i,k}\varepsilon_k,\end{aligned}$$

where $\theta \in [0, 1]$.

For $\theta = 1$ the scheme is said to be fully explicit. In this case it is identical to the simple Euler scheme. For $\theta = 0$ the scheme is fully implicit. Market practitioners typically choose $\theta = 0.5$.

Remark 6.5

A well-known weakness of Monte Carlo simulation is its slow rate of convergence. For N simulated paths the error is proportional to $1/\sqrt{N}$. This means that to get an accurate estimate of the option price we need to perform a prohibitively large number of simulations. Fortunately, there are several so-called variance-reduction techniques that can be applied to improve convergence. Examples are the control variate technique and the antithetic variable technique. Error reduction via a control variate is covered in [NMFC].

Monte Carlo pricing of Bermudan swaptions

Above we discussed Monte Carlo simulation as applied to pricing interest rate derivatives of European type. However, the method is just as useful for pricing other exotic derivatives. The Bermudan swaption, which was discussed in Section 3.6, is a good example of the type of instrument for which Monte Carlo simulation is often the only viable numerical approach when we are working in the LMM framework.

We recall that, for a set of dates $0 < T_0 < T_1 < \cdots < T_n$, the holder of a payer (or receiver) Bermudan swaption with strike K has the right to enter a payer (or receiver) interest rate swap with swap rate K at any time T_k, where $k = 0, \dots, l$. The value of the option at time T_i is given by (3.30), that is, it is the maximum of the exercise value $\mathrm{E}(T_i)$ and the continuation value $\mathrm{C}(T_i)$.

In Section 3.6 the value of the Bermudan at time 0 was found recursively, starting at the last exercise date and working backwards in time. The difficulty is that, in a Monte Carlo simulation, the paths of the underlying forward LIBOR rates are generated forward in time, so at time T_i the future values of the rates $F_j(t)$ with $j > i+1$ remain unknown. As a consequence, the continuation value $\mathrm{C}(T_i)$ is also unknown. This makes it difficult to apply (3.30).

One way of overcoming this difficulty is to use an approximation to the continuation value. A popular approach is the method of Longstaff and Schwartz, where an approximation is found by regressing estimates of the continuation value on an appropriate set of basis functions. However, the details of this important application of the Monte Carlo method are beyond the scope of this book.

6.3 Calibration

There are several different (though closely related) approaches we can take when it comes to calibrating the model parameters. The deciding factor in choosing one method over another will depend on the derivative we need to value.

Calibrating to caplets

The following procedure is suitable for instruments whose price depends mainly on caplet volatility. We use the time-homogeneous parametric form (5.25) for the instantaneous volatility, which we write as

$$\sigma_i(t;\alpha) = (a + b(T_{i-1} - t))\mathrm{e}^{-c(T_{i-1}-t)} + d,$$

with α denoting the set of parameters a, b, c, d. As shown in Section 5.2, the time 0 price of the ith caplet in the LMM is given by Black's formula (2.24) with

$$v_i(\alpha) = \sqrt{\frac{1}{T_{i-1}} \int_0^{T_{i-1}} \sigma_i(t;\alpha)^2 dt}$$

substituted for the volatility σ. Taking the volatilities $\hat{\sigma}_i^{\text{caplet}}$ implied by the market prices of caplets via (2.25), we can perform the least squares optimisation

$$\min_{\alpha} \sum_{i=1}^{n} \left((\hat{\sigma}_i^{\text{caplet}})^2 - v_i(\alpha)^2 \right)^2 \tag{6.7}$$

to calibrate the parameters.

With just four parameters, the time-homogeneous form (5.25) for the instantaneous volatility may not be flexible enough to provide a good fit to the observed data. To obtain a perfect fit we can introduce a constant parameter κ_i for each forward LIBOR rate $F_i(t)$ and write the instantaneous volatility as

$$\sigma_i(t) = \kappa_i \sigma_i(t; \alpha). \tag{6.8}$$

Since the parameters κ_i are unique to each forward LIBOR rate, this new form for the instantaneous volatility is no longer time homogeneous. To calibrate to the set of observed caplet volatilities we can use the parameters α found by the least squares optimisation (6.7) and choose the κ_i so that

$$\hat{\sigma}_i^{\text{caplet}} = \kappa_i v_i(\alpha). \tag{6.9}$$

In practice this gives values of κ_i that are typically close to 1, so we can think of (6.8) as approximately time homogeneous.

Calibrating to swaptions

In general, we need to calibrate to the implied swaption volatilities in addition to implied caplet volatilities. For example, when valuing a Bermudan swaption we should aim for the exact recovery of the set of co-terminal swaptions that underlie the Bermudan.

Since calibrating to a large matrix of swaption prices (equivalently, implied Black swaption volatilities) is computationally intensive, the standard approach is to use an analytic approximation to Black's swaption volatility, which makes it possible to quickly calculate the volatility given a set of model parameters. A popular analytic approximation for Black's swaption volatility is Rebonato's formula (5.40). To adapt the formula for a T_l into $T_m - T_l$ swaption, we need to replace the indices $0, n$ by l, m. This gives the following approximation for the volatility (5.34) in Black's formula for such a swaption:

$$v_{l,m}^2 = \frac{1}{T_l} \sum_{i,j=l+1}^{m} \frac{w_i(0) w_j(0) F_i(0) F_j(0)}{S_{l,m}(0)^2} \int_0^{T_l} \sigma_i(t) \sigma_j(t) \rho_{i,j} dt, \tag{6.10}$$

where

$$w_i(t) = \frac{\tau_i B(t, T_i)}{\sum_{j=l+1}^{m} \tau_j B(t, T_j)} = \frac{\tau_i \prod_{k=l+1}^{i} \frac{1}{1 + \tau_k F_k(t)}}{\sum_{j=l+1}^{m} \tau_j \prod_{k=l+1}^{j} \frac{1}{1 + \tau_k F_k(t)}}.$$

Having chosen the functional form (6.8) for the instantaneous volatility, and (6.1) and (6.2) with parameter reduction for the instantaneous correlation, we denote the model parameters κ_i, $i = 1, \ldots, n$ and a, b, c, d for the volatility by κ and α, and those for the correlation by θ. Thus the volatility in Black's swaption formula approximated by using (6.10) can be denoted by $v_{l,m}(\kappa, \alpha, \theta)$.

One approach to calibration would be to take the market-implied swaption volatilities $\hat{\sigma}_{l,m}^{\text{swpt}}$ and the corresponding Black swaption volatilities $v_{l,m}(\kappa, \alpha, \theta)$ calculated by using (6.10), and to perform the least squares optimisation

$$\min_{\kappa,\alpha,\theta} \sum_{l,m} a_{l,m} \left((\hat{\sigma}_{l,m}^{\text{swpt}})^2 - v_{l,m}(\kappa, \alpha, \theta)^2 \right)^2 \tag{6.11}$$

with some weights $a_{l,m}$. The choice of these weights depends on the instrument to be valued. In particular, for a Bermudan swaption we need to recover as accurately as possible the implied volatilities of the co-terminal swaptions, therefore we could make the weights associated with the co-terminal set equal to 1 and all the remaining weights equal to 0.

A major drawback of the above optimisation is that fitting the correlation parameters to the implied volatilities by means of (6.11) will often result in a 'noisy' correlation structure with features that contradict those observed empirically. The implied Black volatility is only weakly dependent on the instantaneous correlation. This makes it difficult to infer the 'correct' correlation structure from the implied volatilities. For this reason, practitioners will often estimate the correlation parameters from a sample correlation matrix calculated using historical data, such as the one in Figure 5.4. This approach is outlined in Section 5.8. As a result, we can drop θ in the optimisation problem (6.11) and calibrate only to the instantaneous volatility parameters κ and α. Moreover, we can also drop κ by defining the κ_i by means of (6.9) as functions of the parameters $\alpha = (a, b, c, d)$, thereby achieving a perfect fit to the caplet volatilities $\hat{\sigma}_i^{\text{caplet}}$ given by the market.

In general, the chosen parameterisation may not be rich enough to ensure

https://doi.org/10.1017/CBO9781139035033.007

an accurate fit to all caplets and the entire swaption matrix. As is often the case for calibration problems, we need to make a compromise between the accuracy of the fit and ensuring that the behaviour of the model as implied by the choice of parameters is realistic.

6.4 Numerical example

We present an example of European option pricing via Monte Carlo simulation. Consider a set of dates $0 = T_0 < T_1 < \cdots < T_n$ with accrual periods $\tau_i = T_i - T_{i-1}$ for $i = 1, \ldots, n$, and suppose that the initial term structure of forward LIBOR rates $F_i(T_0)$ for $i = 1, \ldots, n$ is given along with the implied Black caplet volatilities $\hat{\sigma}_i^{\text{caplet}}$ for $i = 2, \ldots, n$. Note that the first forward rate $F_1(T_0) = L(T_0, T_1)$ will have set at time T_0, therefore we shall be simulating the evolution in time of a set of $n - 1$ forward rates. For simplicity we consider the one-factor case, with $m = 1$ in (6.4).

To begin, we need to calibrate the instantaneous volatility $\sigma_i(t)$ to the implied volatilities $\hat{\sigma}_i^{\text{caplet}}$. We assume the time-homogeneous piecewise constant form (5.24) for the instantaneous volatility, which allows us to apply a bootstrapping method rather than the more involved least squares optimisation (6.7). For each $i = 2, \ldots, n$ the implied volatility $\hat{\sigma}_i^{\text{caplet}}$ quoted in the market is set equal to the LMM implied volatility (5.9), namely

$$\left(\hat{\sigma}_i^{\text{caplet}}\right)^2 = v_i^2 = \frac{1}{T_{i-1}} \int_0^{T_{i-1}} \sigma_i(s)^2 ds = \frac{1}{T_{i-1}} \sum_{j=1}^{i-1} \sigma_{i-j}^2 (T_j - T_{j-1}).$$

In particular, $\sigma_1 = \hat{\sigma}_2^{\text{caplet}}$. Moreover, if we have already computed the piecewise constant volatilities $\sigma_1, \ldots, \sigma_{i-2}$ for some i, then we can obtain σ_{i-1} by rewriting the above as

$$\sigma_{i-1}^2 = \frac{1}{T_1} \left(T_{i-1} \left(\hat{\sigma}_i^{\text{caplet}}\right)^2 - \sum_{j=2}^{i-1} \sigma_{i-j}^2 (T_j - T_{j-1}) \right).$$

Example 6.6

The initial term structure of forward LIBOR rates $F_i(T_0)$ for $i = 1, \ldots, n$ is given in Table 6.1 along with the implied Black caplet volatilities $\hat{\sigma}_i^{\text{caplet}}$ for $i = 2, \ldots, n$, with $n = 12$ and accrual periods $\tau_i = T_i - T_{i-1} = 0.25$ years for $i = 1, \ldots, n$. Calibrating as above to the implied volatilities $\hat{\sigma}_i^{\text{caplet}}$, we

https://doi.org/10.1017/CBO9781139035033.007

Table 6.1 *Quarterly forward LIBOR rates $F_i(T_0)$ and implied caplet volatilities $\hat{\sigma}_i^{\text{caplet}}$ at time $T_0 = 0$.*

i	τ_i	$F_i(T_0)$	$\hat{\sigma}_i^{\text{caplet}}$
1	0.25	2.000%	
2	0.25	2.051%	40.000%
3	0.25	2.103%	39.012%
4	0.25	2.156%	38.049%
5	0.25	2.210%	37.110%
6	0.25	2.266%	36.193%
7	0.25	2.324%	35.300%
8	0.25	2.382%	34.428%
9	0.25	2.443%	33.578%
10	0.25	2.505%	32.749%
11	0.25	2.568%	31.941%
12	0.25	2.633%	31.152%

compute the piecewise constant volatilities σ_{i-1} for $i = 2, \ldots, 12$ listed in Table 6.2.

Once the model is calibrated, we are ready to evolve the set of forward rates though time. We use the Euler scheme (6.6) with time steps equal to the forward-rate accrual periods. Given the estimated forward rates $\bar{F}_i(T_j)$ at time T_j for some $j = 0, \ldots, n-2$, we estimate the rates at time T_{j+1} as

$$\bar{F}_i(T_{j+1}) = \bar{F}_i(T_j) \exp\left(\left(\mu_i^n(\bar{F}(T_j), T_j) - \frac{1}{2}\sigma_i(T_j)^2\right)\tau_{j+1} + \sigma_i(T_j)\sqrt{\tau_{j+1}}\varepsilon\right)$$

for $i = j+2, \ldots, n$, where ε is a random variable with the standard normal distribution $N(0,1)$ and where the drift term is given by (6.5), which becomes

$$\mu_i^n(\bar{F}(T_j), T_j) = -\sum_{k=i+1}^{n} \frac{\tau_k \rho_{k,i} \sigma_i(T_j) \sigma_k(T_j) F_k(T_j)}{1 + \tau_k F_k(T_j)}.$$

Having estimated the forward rates at time T_j, we can compute the value at time T_j of the zero-coupon bond maturing at time T_i for $i = j+1, \ldots, n$ as

$$B(T_j, T_i) = \prod_{k=j+1}^{i} \frac{1}{(1 + \tau_k \bar{F}_k(T_j))}.$$

Table 6.2 *Piecewise constant volatilities σ_{i-1} and LMM drift terms $\mu_i^n(\bar{F}(0), 0)$ used in the time T_1 estimate of the forward rate $\bar{F}_i(T_1)$. The zero-coupon bond prices $B(T_1, T_i)$ are also given.*

i	σ_{i-1}	$\mu_i^n(\bar{F}(0), 0)$	$\bar{F}_i(T_1)$	$B(T_1, T_i)$
2	40.000%	0.006 873	2.494%	0.993 80
3	37.999%	0.005 774	2.535%	0.987 54
4	36.046%	0.004 781	2.577%	0.981 22
5	34.137%	0.003 887	2.620%	0.974 84
6	32.269%	0.003 088	2.664%	0.968 39
7	30.441%	0.002 378	2.709%	0.961 87
8	28.647%	0.001 752	2.754%	0.955 30
9	26.886%	0.001 205	2.801%	0.948 65
10	25.152%	0.000 734	2.849%	0.941 95
11	23.440%	0.000 333	2.897%	0.935 17
12	21.746%	0.000 000	2.946%	0.928 34

Example 6.7

Continuing the numerical example, in Table 6.3 we show the evolution in time of the forward rates for a single simulated path. To keep the table concise, we list only the rates computed up to time T_6. In Table 6.2 we show the discount bonds $B(T_1, T_i)$, together with the drift terms $\mu_i^n(\bar{F}(0), 0)$, for the path whose fragment up to time T_6 is shown in Table 6.3.

For a derivative security that makes a payment $h(F_1(T_j), \ldots, F_n(T_j))$ at time T_j depending on the forward rates $F_i(T_j) = F(T_j; T_{i-1}, T_i)$ for $i = j + 1, \ldots, n$ as well as on the LIBOR rates $F_i(T_j) = L(T_{i-1}, T_i)$ for $i = 1, \ldots, j$ (a simple example being a caplet payment with payoff $(T_j - T_{j-1})(L(T_{j-1}, T_j) - K)^+$ at time T_j), the numeraire-discounted payoff expressed as a function of the Monte Carlo estimated rates $\bar{F}_1(T_j), \ldots, \bar{F}_n(T_j)$ is

$$\frac{h(\bar{F}_1(T_j), \ldots, \bar{F}_n(T_j))}{B(T_j, T_n)}.$$

We can compute the discounted payoff for a number of different paths by repeating the above simulation and then calculate the mean and multiply it by $B(0, T_n)$ to obtain a Monte Carlo estimate of the value of the derivative security at time $T_0 = 0$.

https://doi.org/10.1017/CBO9781139035033.007

Table 6.3 *Estimated forward rates $\bar{F}_i(t)$ for $t = T_0 \dots, T_6$ along a simulated path.*

t	T_0	T_1	T_2	T_3	T_4	T_5	T_6
ε		1.087 40	−1.166 18	0.213 40	−0.990 00	0.885 75	−2.022 10
$\bar{F}_1(t)$	2.000%						
$\bar{F}_2(t)$	2.051%	2.494%					
$\bar{F}_3(t)$	2.103%	2.535%	1.964%				
$\bar{F}_4(t)$	2.156%	2.577%	2.025%	2.068%			
$\bar{F}_5(t)$	2.210%	2.620%	2.087%	2.132%	1.712%		
$\bar{F}_6(t)$	2.266%	2.664%	2.149%	2.196%	1.785%	2.086%	
$\bar{F}_7(t)$	2.324%	2.709%	2.213%	2.261%	1.859%	2.159%	1.411%
$\bar{F}_8(t)$	2.382%	2.754%	2.279%	2.327%	1.935%	2.232%	1.492%
$\bar{F}_9(t)$	2.443%	2.801%	2.345%	2.394%	2.013%	2.307%	1.576%
$\bar{F}_{10}(t)$	2.505%	2.849%	2.413%	2.462%	2.092%	2.382%	1.662%
$\bar{F}_{11}(t)$	2.568%	2.897%	2.482%	2.531%	2.174%	2.458%	1.751%
$\bar{F}_{12}(t)$	2.633%	2.946%	2.552%	2.601%	2.256%	2.535%	1.842%

Example 6.8

Consider a cap with payment dates $T_2, \dots, T_n$, a 1 million USD notional and strike of 1.75%. This simple example is a good test of Monte Carlo simulation in the LMM. We should be able to price caps very well given that the model has been calibrated to the implied caplet volatilities. Monte Carlo simulation with 100 000 paths does indeed produce a price within 0.05% of that given by Black's formula, which is \$18 358.

In Section 7.4 we are going to use the same approach to price a ratchet floater.