
The Libor Market Model I

Many of the models considered so far describe the evolution of the yield curve in terms of a small set of Markov state variables. While proper calibration procedures allow for successful application of such models to the pricing and hedging of a surprising variety of securities, many exotic derivatives require richer dynamics than what is possible with low-dimensional Markov models. For instance, exotic derivatives may be strongly sensitive to the joint evolution of multiple points of the yield curve, necessitating the usage of several driving Brownian motions. Also, most exotic derivatives may not be related in any obvious way to vanilla European options, making it hard to confidently identify a small, representative set of vanilla securities to which a low-dimensional Markovian model can feasibly be calibrated. What is required in such situations is a model sufficiently rich to capture the full correlation structure across the entire yield curve and to allow for volatility calibration to a large enough set of European options that the volatility characteristics of most exotic securities can be considered “spanned” by the calibration. Candidates for such a model include the multi-factor short-rate models in Chapter 13 and the multi-factor quasi-Gaussian models in Section 14.3. In this chapter, we shall cover an alternative approach to the construction of multi-factor interest rate models, the so-called *Libor market (LM)* model framework. Originally developed in Brace et al. [1996], Jamshidian [1997], and Miltersen et al. [1997], the LM model class enjoys significant popularity with practitioners and is in many ways easier to grasp than, say, the multi-factor quasi-Gaussian models in Chapter 14.

This chapter develops the basic LM model and provides a series of extensions to the original log-normal framework in Brace et al. [1996] and Miltersen et al. [1997] in order to better capture observed volatility smiles. To facilitate calibration of the model, efficient techniques for the pricing of European securities are developed. We provide a detailed discussion of the modeling of forward rate correlations which, along with the pricing formulas for caps and swaptions, serves as the basis for most of the calibration strategies that we proceed to examine. Many of these strategies are generic in nature and apply to

multi-factor models other than the LM class, including the models discussed in Chapters 13 and 14. We wrap up the chapter with a careful discussion of schemes for Monte Carlo simulation of LM models. A number of advanced topics in LM modeling will be covered in Chapter 16.

15.1 Introduction and Setup

15.1.1 Motivation and Historical Notes

Chapter 5 introduced the HJM framework which, in its most general form, involves multiple driving Brownian motions and an infinite set of state variables (namely the set of instantaneous forward rates). As argued earlier, the HJM framework contains any arbitrage-free interest rate model adapted to a finite set of Brownian motions. Working directly with instantaneous forward rates is, however, not particularly attractive in applications, for a variety of reasons. First, instantaneous forward rates are never quoted in the market, nor do they figure directly in the payoff definition of any traded derivative contract. As discussed in Chapter 6, realistic securities (swaps, caps, futures, etc.) instead involve simply compounded (Libor) rates, effectively representing *integrals* of instantaneous forwards. The disconnect between market observables and model primitives often makes development of market-consistent pricing expression for simple derivatives cumbersome. Second, an infinite set of instantaneous forward rates can generally¹ not be represented exactly on a computer, but will require discretization into a finite set. Third, prescribing the form of the volatility function of instantaneous forward rates is subject to a number of technical complications, requiring sub-linear growth to prevent explosions in the forward rate dynamics, which precludes the formulation of a log-normal forward rate model (see Sandmann and Sondermann [1997] and the discussion in Sections 5.5.3 and 12.1.3).

As discovered in Brace et al. [1996], Jamshidian [1997], and Miltersen et al. [1997], the three complications above can all be addressed simultaneously by simply writing the model in terms of a non-overlapping set of simply compounded Libor rates. Not only do we then conveniently work with a finite set of directly observable rates that can be represented on a computer, but, as we shall show, an explosion-free log-normal forward rate model also becomes possible. Despite the change to simply compounded rates, we should emphasize that the Libor market model will still be a special case of an HJM model, albeit one where we only indirectly specify the volatility function of the instantaneous forward rates.

¹As we have seen in earlier chapters, for special choices of the forward rate volatility we can sometimes identify a finite-dimensional Markovian representation of the forward curve that eliminates the need to store the entire curve. This is not possible in general, however.

15.1.2 Tenor Structure

The starting point for our development of the LM model is a fixed *tenor structure*

$$0 = T_0 < T_1 < \dots < T_N. \quad (15.1)$$

The intervals $\tau_n = T_{n+1} - T_n$, $n = 0, \dots, N - 1$, would typically be set to be either 3 or 6 months, corresponding to the accrual period associated with observable Libor rates. Rather than keeping track of an entire yield curve, at any point in time t we are (for now) focused only on a finite set of zero-coupon bonds $P(t, T_n)$ for the set of n 's for which $T_N \geq T_n > t$; notice that this set shrinks as t moves forward, becoming empty when $t > T_N$. To formalize this “roll-off” of zero-coupon bonds in the tenor structure as time progresses, it is often useful to work with an *index function* $q(t)$, defined by the relation

$$T_{q(t)-1} \leq t < T_{q(t)}. \quad (15.2)$$

We think of $q(t)$ as representing the tenor structure index of the shortest-dated discount bond still alive.

On the fixed tenor structure, we proceed to define Libor forward rates according to the relation (see (5.2))

$$L(t, T_n, T_{n+1}) = L_n(t) = \left(\frac{P(t, T_n)}{P(t, T_{n+1})} - 1 \right) \tau_n^{-1}, \quad N - 1 \geq n \geq q(t).$$

We note that when considering a given Libor forward rate $L_n(t)$, we always assume $n \geq q(t)$ unless stated otherwise. For any $T_n > t$,

$$P(t, T_n) = P(t, T_{q(t)}) \prod_{i=q(t)}^{n-1} (1 + L_i(t)\tau_i)^{-1}. \quad (15.3)$$

Notice that knowledge of $L_n(t)$ for all $n \geq q(t)$ is generally *not* sufficient to reconstruct discount bond prices on the entire (remaining) tenor structure; the front “stub” discount bond price $P(t, T_{q(t)})$ must also be known.

15.2 LM Dynamics and Measures

15.2.1 Setting

In the Libor market model, the set of Libor forwards $L_{q(t)}(t), L_{q(t)+1}(t), \dots, L_{N-1}(t)$ constitute the set of state variables for which we wish to specify dynamics. As a first step, we pick a probability measure P and assume that those dynamics originate from an m -dimensional Brownian motion $W(t)$, in the sense that all Libor rates are measurable with respect to the filtration generated by $W(\cdot)$. Further assuming that the Libor

rates are square integrable, it follows from the Martingale Representation Theorem that, for all $n \geq q(t)$,

$$dL_n(t) = \sigma_n(t)^\top (\mu_n(t) dt + dW(t)), \quad (15.4)$$

where μ_n and σ_n are m -dimensional processes, respectively, both adapted to the filtration generated by $W(\cdot)$. From the Diffusion Invariance Principle (see Section 2.5) we know that $\sigma_n(t)$ is measure invariant, whereas $\mu_n(t)$ is not.

As it turns out, for a given choice of σ_n in the specification (15.4), it is quite straightforward to work out explicitly the form of $\mu_n(t)$ in various martingale measures of practical interest. We turn to this shortly, but let us first stress that (15.4) allows us to use a *different* volatility function σ_n for each of the forward rates $L_n(t)$, $n = q(t), \dots, N - 1$, in the tenor structure. This obviously gives us tremendous flexibility in specifying the volatility structure of the forward curve evolution, but in practice will require us to impose quite a bit of additional structure on the model to ensure realism and to avoid an excess of parameters. We shall return to this topic later in this chapter.

15.2.2 Probability Measures

As shown in Lemma 5.2.3, $L_n(t)$ must be a martingale in the T_{n+1} -forward measure $\mathbb{Q}^{T_{n+1}}$, such that, from (15.4),

$$dL_n(t) = \sigma_n(t)^\top dW^{n+1}(t), \quad (15.5)$$

where $W^{n+1}(\cdot) \triangleq W^{T_{n+1}}(\cdot)$ is an m -dimensional Brownian motion in $\mathbb{Q}^{T_{n+1}}$. It is to be emphasized that only *one* specific Libor forward rate — namely L_n — is a martingale in the T_{n+1} -forward measure. To establish dynamics in other probability measures, the following proposition is useful.

Proposition 15.2.1. *Let L_n satisfy (15.5). In measure \mathbb{Q}^{T_n} the process for L_n is*

$$dL_n(t) = \sigma_n(t)^\top \left(\frac{\tau_n \sigma_n(t)}{1 + \tau_n L_n(t)} dt + dW^n(t) \right),$$

where W^n is an m -dimensional Brownian motion in measure \mathbb{Q}^{T_n} .

Proof. From Theorem 2.4.2 we know that the density relating the measures $\mathbb{Q}^{T_{n+1}}$ and \mathbb{Q}^{T_n} is given by

$$\begin{aligned} \varsigma(t) &= \mathbb{E}_t^{T_{n+1}} \left(\frac{d\mathbb{Q}^{T_n}}{d\mathbb{Q}^{T_{n+1}}} \right) \\ &= \frac{P(t, T_n)/P(0, T_n)}{P(t, T_{n+1})/P(0, T_{n+1})} = (1 + \tau_n L_n(t)) \frac{P(0, T_{n+1})}{P(0, T_n)}. \end{aligned}$$

Clearly, then,

$$d\zeta(t) = \frac{P(0, T_{n+1})}{P(0, T_n)} \tau_n dL_n(t) = \frac{P(0, T_{n+1})}{P(0, T_n)} \tau_n \sigma_n(t)^\top dW^{n+1}(t),$$

or

$$d\zeta(t)/\zeta(t) = \frac{\tau_n \sigma_n(t)^\top dW^{n+1}(t)}{1 + \tau_n L_n(t)}.$$

From the Girsanov theorem (Theorem 2.5.1), it follows that the process

$$dW^n(t) = dW^{n+1}(t) - \frac{\tau_n \sigma_n(t)}{1 + \tau_n L_n(t)} dt \quad (15.6)$$

is a Brownian motion in \mathbb{Q}^{T_n} . The proposition then follows directly from (15.5). \square

To gain some further intuition for the important result in Proposition 15.2.1, let us derive it in less formal fashion. For this, consider the forward discount bond $P(t, T_n, T_{n+1}) = P(t, T_{n+1})/P(t, T_n) = (1 + \tau_n L_n(t))^{-1}$. An application of Ito's lemma to (15.5) shows that

$$\begin{aligned} dP(t, T_n, T_{n+1}) &= \tau_n^2 (1 + \tau_n L_n(t))^{-3} \sigma_n(t)^\top \sigma_n(t) dt - \tau_n (1 + \tau_n L_n(t))^{-2} \sigma_n(t)^\top dW^{n+1}(t) \\ &= \tau_n (1 + \tau_n L_n(t))^{-2} \sigma_n(t)^\top \left\{ \tau_n (1 + \tau_n L_n(t))^{-1} \sigma_n(t) dt - dW^{n+1}(t) \right\}. \end{aligned}$$

As $P(t, T_n, T_{n+1})$ must be a martingale in the \mathbb{Q}^{T_n} -measure, it follows that

$$-dW^n(t) = \tau_n (1 + \tau_n L_n(t))^{-1} \sigma_n(t) dt - dW^{n+1}(t)$$

is a Brownian motion in \mathbb{Q}^{T_n} , consistent with the result in Proposition 15.2.1.

While Proposition 15.2.1 only relates “neighboring” measures $\mathbb{Q}^{T_{n+1}}$ and \mathbb{Q}^{T_n} , it is straightforward to use the proposition iteratively to find the drift of $L_n(t)$ in any of the probability measures discussed in Section 5.2. Let us give some examples.

Lemma 15.2.2. *Let L_n satisfy (15.5). Under the terminal measure \mathbb{Q}^{T_N} the process for L_n is*

$$dL_n(t) = \sigma_n(t)^\top \left(- \sum_{j=n+1}^{N-1} \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt + dW^N(t) \right),$$

where W^N is an m -dimensional Brownian motion in measure \mathbb{Q}^{T_N} .

Proof. From (15.6) we know that

$$\begin{aligned} dW^N(t) &= dW^{N-1}(t) + \frac{\tau_{N-1} \sigma_{N-1}(t)}{1 + \tau_{N-1} L_{N-1}(t)} dt \\ &= dW^{N-2}(t) + \frac{\tau_{N-2} \sigma_{N-2}(t)}{1 + \tau_{N-2} L_{N-2}(t)} dt + \frac{\tau_{N-1} \sigma_{N-1}(t)}{1 + \tau_{N-1} L_{N-1}(t)} dt. \end{aligned}$$

Continuing this iteration down to W^{n+1} , we get

$$dW^N(t) = dW^{n+1}(t) + \sum_{j=n+1}^{N-1} \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt.$$

The lemma now follows from (15.5). \square

Lemma 15.2.3. *Let L_n satisfy (15.5). Under the spot measure Q^B (see Section 5.2.3) the process for L_n is*

$$dL_n(t) = \sigma_n(t)^\top \left(\sum_{j=q(t)}^n \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt + dW^B(t) \right), \quad (15.7)$$

where W^B is an m -dimensional Brownian motion in measure Q^B .

Proof. Recall from Section 5.2.3 that the spot measure is characterized by a rolling or “jumping” numeraire

$$B(t) = P(t, T_{q(t)}) \prod_{n=0}^{q(t)-1} (1 + \tau_n L_n(T_n)). \quad (15.8)$$

At any time t , the random part of the numeraire is the discount bond $P(t, T_{q(t)})$, so effectively we need to establish dynamics in the measure $Q^{T_{q(t)}}$. Applying the iteration idea shown in the proof of Lemma 15.2.2, we get

$$dW^{n+1}(t) = dW^{q(t)}(t) + \sum_{j=q(t)}^n \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt,$$

as stated. \square

The spot and terminal measures are, by far, the most commonly used probability measures in practice. Occasionally, however, it may be beneficial to use one of the hybrid measures discussed earlier in this book, for instance if one wishes to enforce that a particular Libor rate $L_n(t)$ be a martingale. As shown in Section 5.2.4, we could pick as a numeraire the asset price process

$$\tilde{P}_{n+1}(t) = \begin{cases} P(t, T_{n+1}), & t \leq T_{n+1}, \\ B(t)/B(T_{n+1}), & t > T_{n+1}, \end{cases} \quad (15.9)$$

where B is the spot numeraire (15.8). Using the same technique as in the proofs of Lemmas 15.2.2 and 15.2.3, it is easily seen that now

$$dL_i(t) = \begin{cases} \sigma_i(t)^\top \left(-\sum_{j=i+1}^n \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt + d\tilde{W}^{n+1}(t) \right), & t \leq T_{n+1}, \\ \sigma_i(t)^\top \left(\sum_{j=q(t)}^i \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} dt + d\tilde{W}^{n+1}(t) \right), & t > T_{n+1}, \end{cases}$$

where $\widetilde{W}^{n+1}(t)$ is a Brownian motion in the measure induced by the numeraire $\widetilde{P}_{n+1}(t)$. Note in particular that $L_n(t)$ is a martingale as desired, and that we have defined a numeraire which — unlike $P(t, T_n)$ — will be alive at any time t .

We should note that an equally valid definition of a hybrid measure will replace (15.9) with the asset process

$$\overline{P}_{n+1}(t) = \begin{cases} B(t), & t \leq T_{n+1}, \\ B(T_{n+1})P(t, T_N)/P(T_{n+1}, T_N), & t > T_{n+1}. \end{cases} \quad (15.10)$$

This type of numeraire process is often useful in discretization of the LM model for simulation purposes; see Section 15.6.1.2 for details.

15.2.3 Link to HJM Analysis

As discussed earlier, the LM model is a special case of the general HJM class of diffusive interest rate models. To explore this relationship a bit further, we recall that HJM models generally has risk-neutral dynamics of the form

$$df(t, T) = \sigma_f(t, T)^\top \int_t^T \sigma_f(t, u) du dt + \sigma_f(t, T)^\top dW(t),$$

where $f(t, T)$ is the time t instantaneous forward rate to time T and $\sigma_f(t, T)$ is the instantaneous forward rate volatility function. From the results in Chapter 5, it follows that dynamics for the forward bond $P(t, T_n, T_{n+1})$ is of the form

$$\frac{dP(t, T_n, T_{n+1})}{P(t, T_n, T_{n+1})} = O(dt) - (\sigma_P(t, T_{n+1})^\top - \sigma_P(t, T_n)^\top) dW(t),$$

where $O(dt)$ is a drift term and

$$\sigma_P(t, T) = \int_t^T \sigma_f(t, u) du.$$

By definition $L_n(t) = \tau_n^{-1}(P(t, T_n, T_{n+1})^{-1} - 1)$, such that

$$dL_n(t) = O(dt) + \tau_n^{-1}(1 + \tau_n L_n(t)) \int_{T_n}^{T_{n+1}} \sigma_f(t, u)^\top du dW(t).$$

By the Diffusion Invariance Principle (see Section 2.5), it follows from (15.5) that the LM model volatility $\sigma_n(t)$ is related to the HJM instantaneous forward volatility function $\sigma_f(t, T)$ by

$$\sigma_n(t) = \tau_n^{-1}(1 + \tau_n L_n(t)) \int_{T_n}^{T_{n+1}} \sigma_f(t, u) du. \quad (15.11)$$

Note that, as expected, $\sigma_n(t) \rightarrow \sigma_f(t, T_n)$ as $\tau_n \rightarrow 0$.

It should be obvious from (15.11) that a complete specification of $\sigma_f(t, T)$ uniquely determines the LM volatility $\sigma_n(t)$ for all t and all n . On the other hand, specification of $\sigma_n(t)$ for all t and all n does *not* allow us to imply a unique HJM forward volatility function $\sigma_f(t, T)$ — all we are specifying is essentially a strip of contiguous integrals of this function in the T -direction. This is hardly surprising, inasmuch as the LM model only concerns itself with a finite set of discrete interest rate forwards and as such cannot be expected to uniquely characterize the behaviors of instantaneous forward rates and their volatilities. Along the same lines, we note that the LM model does not uniquely specify the behavior of the short rate $r(t) = f(t, t)$; as a consequence, the rolling money-market account $\beta(t)$ and the risk-neutral measure are not natural constructions in the LM model². Section 16.3 discusses these issues in more details.

15.2.4 Separable Deterministic Volatility Function

So far, our discussion of the LM model has been generic, with little structure imposed on the $N - 1$ volatility functions $\sigma_n(t)$, $n = 1, 2, \dots, N - 1$. To build a workable model, however, we need to be more specific about our choice of $\sigma_n(t)$. A common prescription of $\sigma_n(t)$ takes the form

$$\sigma_n(t) = \lambda_n(t)\varphi(L_n(t)), \quad (15.12)$$

where $\lambda_n(t)$ is a bounded vector-valued deterministic function and $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ is a time-homogeneous local volatility function. This specification is conceptually very similar to the local volatility models in Chapter 8, although here $\sigma_n(t)$ is vector-valued and the model involves simultaneous modeling of multiple state variables (the $N - 1$ Libor forwards).

At this point, the reader may reasonably ask whether the choice (15.12) in fact leads to a system of SDEs for the various Libor forward rates that is “reasonable”, in the sense of existence and uniqueness of solutions. While we here shall not pay much attention to such technical regularity issues, it should be obvious that not all functions φ can be allowed. One relevant result is given below.

Proposition 15.2.4. *Assume that (15.12) holds with $\varphi(0) = 0$ and that $L_n(0) \geq 0$ for all n . Also assume that φ is locally Lipschitz continuous and satisfies the growth condition*

$$\varphi(x)^2 \leq C(1 + x^2), \quad x > 0,$$

where C is some positive constant. Then non-explosive, pathwise unique solutions of the no-arbitrage SDEs for $L_n(t)$, $q(t) \leq n \leq N - 1$, exist under all measures \mathbb{Q}^{T_i} , $q(t) \leq i \leq N$. If $L_n(0) > 0$, then $L_n(t)$ stays positive at all t .

²In fact, as discussed in Jamshidian [1997], one does not need to assume that a short rate process exist when constructing an LM model.

Proof. (Sketch) Due to the recursive relationship between measures, it suffices to consider the system of SDEs (15.7) under the spot measure Q^B :

$$dL_n(t) = \varphi(L_n(t)) \lambda_n(t)^\top (\mu_n(t) dt + dW^B(t)), \tag{15.13}$$

$$\mu_n(t) = \sum_{j=q(t)}^n \frac{\tau_j \varphi(L_j(t)) \lambda_j(t)}{1 + \tau_j L_j(t)}. \tag{15.14}$$

Under our assumptions, it is easy to see that each term in the sum for μ_n is locally Lipschitz continuous and bounded. The growth condition on φ in turn ensures that the product $\varphi(L_n(t)) \lambda_n(t)^\top \mu_n(t)$ is also locally Lipschitz continuous and, due to the boundedness of μ_n , satisfies a linear growth condition. Existence and uniqueness now follows from Theorem 2.6.1. The result that 0 is a non-accessible boundary for the forward rates if started above 0 follows from standard speed-scale boundary classification results; see Andersen and Andreasen [2000b] for the details. \square

Some standard parameterizations of φ are shown in Table 15.1. Of those, only the log-normal specification and the LCEV specification directly satisfy the criteria in Proposition 15.2.4. The CEV specification violates Lipschitz continuity at $x = 0$, and as a result uniqueness of the SDE fails. As shown in Andersen and Andreasen [2000b], we restore uniqueness by specifying that forward rates are *absorbed* at the origin (see also Section 8.2.3). As for the displaced diffusion specification $\varphi(x) = ax + b$, we here violate the assumption that $\varphi(0) = 0$, and as a result we cannot always guarantee that forward rates stay positive. Also, to prevent explosion of the forward rate drifts, we need to impose additional restrictions to prevent terms of the form $1 + \tau_n L_n(t)$ from becoming zero. As displaced diffusions are of considerable practical importance, we list the relevant restrictions in Lemma 15.2.5 below.

Name	$\varphi(x)$
Log-normal	x
CEV	$x^p, \quad 0 < p < 1$
LCEV	$x \min(\varepsilon^{p-1}, x^{p-1}), \quad 0 < p < 1, \varepsilon > 0$
Displaced log-normal	$bx + a, \quad b > 0, a \neq 0$

Table 15.1. Common DVF Specifications

Lemma 15.2.5. *Consider a local volatility Libor market model with local volatility function $\varphi(x) = bx + a$, where $b > 0$ and $a \neq 0$. Assume that $bL_n(0) + a > 0$ and $a/b < \tau_n^{-1}$ for all $n = 1, 2, \dots, N - 1$. Then non-explosive, pathwise unique solutions of the no-arbitrage SDEs for $L_n(t)$, $q(t) \leq n \leq N - 1$, exist under all measures Q^{T_i} , $q(t) \leq i \leq N$. All $L_n(t)$ are bounded from below by $-a/b$.*

Proof. Define $H_n(t) = bL_n(t) + a$. By Ito's lemma, we have

$$dH_n(t) = b dL_n(t) = bH_n(t)\lambda_n(t)^\top (\mu_n(t) dt + dW^B(t)),$$

$$\mu_n(t) = \sum_{j=q(t)}^n \frac{\tau_j H_j(t)\lambda_j(t)}{1 + \tau_j (H_j(t) - a)/b}.$$

From the assumptions of the lemma, we have $H_n(0) > 0$ for all n , allowing us to apply the result of Proposition 15.2.4 to $H_n(t)$, provided that we can guarantee that $\mu_n(t)$ is bounded for all positive H_j , $j = q(t), \dots, n$. This follows from $1 - \tau_j a/b > 0$ or $a/b < \tau_j^{-1}$. \square

We emphasize that the requirement $a/b < \tau_n^{-1}$ implies that only in the limit of $\tau_j \rightarrow 0$ — where the discrete Libor forward rates become instantaneous forward rates — will a pure Gaussian LM model specification ($b = 0$) be meaningful; such a model was outlined in Section 5.5.1. On the flip-side, according to Proposition 15.2.4, a finite-sized value of τ_j ensures that a well-behaved log-normal forward rate model exists, something that we saw earlier (Section 12.1.3) was *not* the case for models based on instantaneous forward rates. The existence of log-normal forward rate dynamics in the LM setting was, in fact, a major driving force behind the development and popularization of the LM framework, and all early examples of LM models (see Brace et al. [1996], Jamshidian [1997], and Miltersen et al. [1997]) were exclusively log-normal.

We recall from earlier chapters that it is often convenient to specify displaced diffusion models as $\varphi(L_n(t)) = (1 - b)L_n(0) + bL_n(t)$, in which case the constant a in Lemma 15.2.5 is different from one Libor rate to the next. In this case, we must require

$$(1 - b)/b < (L_n(0)\tau_n)^{-1}, \quad n = 1, \dots, N - 1.$$

As $L_n(0)\tau_n$ is typically in the magnitude of a few percent, the regularity requirement on b in (15.2.4) is not particularly restrictive.

15.2.5 Stochastic Volatility

As discussed earlier in this book, to ensure that the evolution of the volatility smile is reasonably stationary, it is best if the skew function φ in (15.14) is (close to) monotonic in its argument. Typically we are interested in specifications where $\varphi(x)/x$ is downward-sloping, to establish the standard behavior of interest rate implied volatilities tending to increase as interest rates decline. In reality, however, markets often exhibit non-monotonic volatility smiles or “smirks” with high-struck options trading at implied volatilities above the at-the-money levels. An increasingly popular mechanism to capture such behavior in LM models is through the introduction of stochastic volatility. We have already encountered stochastic volatility models in Chapters 9, 10 and, in the context of term structure models, in Sections 14.2 and 14.3; we now discuss

how to extend the notion of stochastic volatility models to the simultaneous modeling of multiple Libor forward rates.

As our starting point, we take the process (15.14), preferably equipped with a φ that generates either a flat or monotonically downward-sloping volatility skew, but allow the term on the Brownian motion to be scaled by a stochastic process. Specifically, we introduce a mean-reverting scalar process z , with dynamics of the form

$$dz(t) = \theta(z_0 - z(t)) dt + \eta\psi(z(t)) dZ(t), \quad z(0) = z_0, \quad (15.15)$$

where θ , z_0 , and η are positive constants, Z is a Brownian motion under the spot measure \mathbb{Q}^B , and $\psi: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a well-behaved function. We impose that (15.15) will not generate negative values of z , which requires $\psi(0) = 0$. We will interpret the process in (15.15) as the (scaled) variance process for our forward rate diffusions, in the sense that the square root of z will be used as a stochastic, multiplicative shift of the diffusion term in (15.14). That is, our forward rate processes in \mathbb{Q}^B are, for all $n \geq q(t)$,

$$dL_n(t) = \sqrt{z(t)}\varphi(L_n(t))\lambda_n(t)^\top \left(\sqrt{z(t)}\mu_n(t) dt + dW^B(t) \right), \quad (15.16)$$

$$\mu_n(t) = \sum_{j=q(t)}^n \frac{\tau_j\varphi(L_j(t))\lambda_j(t)}{1 + \tau_j L_j(t)},$$

where $z(t)$ satisfies (15.15). This construction naturally follows the specification of vanilla stochastic volatility models in Chapter 9, and the specification of stochastic volatility quasi-Gaussian models in Chapter 14. As we discussed previously, it is often natural to scale the process for z such that $z(0) = z_0 = 1$.

Let us make two important comments about (15.16). First, we emphasize that a single common factor \sqrt{z} simultaneously scales all forward rate volatilities; movements in volatilities are therefore perfectly correlated across the various forward rates. In effect, our model corresponds only to the first principal component of the movements of the instantaneous forward rate volatilities. This is a common assumption that provides good balance between realism and parsimony, and we concentrate mostly on this case — although we do relax it later in the book, in Chapter 16. Second, we note that the clean form of the z -process (15.15) in the measure \mathbb{Q}^B generally does not carry over to other probability measures, as we would expect from Proposition 9.3.9. To state the relevant result, let $\langle Z(t), W(t) \rangle$ denote the quadratic covariation between $Z(t)$ and the m components of $W(t)$ (recall the definition of covariation in Remark 2.1.7). We then have

Lemma 15.2.6. *Let dynamics for $z(t)$ in the measure \mathbb{Q}^B be as in (15.15). Then the SDE for $z(t)$ in measure $\mathbb{Q}^{T^{n+1}}$, $n \geq q(t) - 1$, is*

$$dz(t) = \theta(z_0 - z(t)) dt + \eta\psi(z(t)) \times \left(-\sqrt{z(t)}\mu_n(t)^\top \langle dZ(t), dW^B(t) \rangle + dZ^{n+1}(t) \right), \quad (15.17)$$

where $\mu_n(t)$ is given in (15.16) and Z^{n+1} is a Brownian motion in measure $\mathbb{Q}^{T_{n+1}}$.

Proof. From earlier results, we have

$$dW^{n+1}(t) = \sqrt{z(t)}\mu_n(t) dt + dW^B(t).$$

Let us denote

$$a(t) = \langle dZ(t), dW^B(t) \rangle / dt,$$

so that we can write

$$dZ(t) = a(t)^\top dW^B(t) + \sqrt{1 - \|a(t)\|^2} d\tilde{W}(t),$$

where \tilde{W} is a scalar Brownian motion independent of W^B . In the measure $\mathbb{Q}^{T_{n+1}}$, we then have

$$\begin{aligned} dZ(t) &= a(t)^\top \left(dW^{n+1}(t) - \sqrt{z(t)}\mu_n(t) dt \right) + \sqrt{1 - \|a(t)\|^2} d\tilde{W}(t) \\ &= dZ^{n+1}(t) - a(t)^\top \sqrt{z(t)}\mu_n(t) dt, \end{aligned}$$

and the result follows. \square

The process (15.17) is awkward to deal with, due to presence of the drift term $\mu_n(t)^\top \langle dZ(t), dW^B(t) \rangle$ which will, in general, depend on the state of the Libor forward rates at time t . For tractability, on the other hand, we would like for the z -process to only depend on z itself. To achieve this, and to generally simplify measure shifts in the model, we make the following assumption³ about (15.15)–(15.16):

Assumption 15.2.7. *The Brownian motion $Z(t)$ of the variance process $z(t)$ is independent of the vector-valued Brownian motion $W^B(t)$.*

We have already encountered the same assumption in the context of stochastic volatility quasi-Gaussian models, see Section 14.2.1, where we also have a discussion of the implications of such a restriction.

The diffusion coefficient of the variance process, the function ψ , is traditionally chosen to be of power form, $\psi(x) = x^\alpha$, $\alpha > 0$. While it probably makes sense to keep the function monotone, the power specification is probably a nod to tradition rather than anything else. Nevertheless, some particular choices lead to analytically tractable specifications, as we saw in Chapter 9; for that reason, $\alpha = 1/2$ (the Heston model) is popular.

Remark 15.2.8. Going forward we shall often use the stochastic volatility (SV) model in this section as a benchmark for theoretical and numerical work. As the stochastic volatility model reduces to the local volatility model in Section 15.2.4 when $z(t)$ is constant, all results for the SV model will carry over to the DVF setting.

³We briefly return to the general case in Section 16.6.

15.2.6 Time-Dependence in Model Parameters

In the models we outlined in Sections 15.2.4 and 15.2.5, the main role of the vector-valued function of time $\lambda_n(t)$ was to establish a term structure “spine” of at-the-money option volatilities. To build volatility smiles around this spine, we further introduced a universal skew-function φ , possibly combined with a stochastic volatility scale $z(t)$ with time-independent process parameters. In practice, this typically gives us a handful of free parameters with which we can attempt to match the market-observed term structures of volatility smiles for various cap and swaption tenors. As it turns out, a surprisingly good fit to market skew data can, in fact, often be achieved with the models of Sections 15.2.4 and 15.2.5. For a truly precise fit to volatility skews across all maturities and swaption tenors, it may, however, be necessary to allow for time-dependence in both the dynamics of $z(t)$ and, more importantly, the skew function φ . The resulting model is conceptually similar to the model in Section 15.2.5, but involves a number of technical intricacies that draw heavily on the material presented in Chapter 10. To avoid cluttering this first chapter on LM model with technical detail, we relegate the treatment of the time-dependent case to the next chapter on advanced topics in LM modeling.

15.3 Correlation

In a one-factor model for interest rates — such as the ones presented in Chapters 11 and 12 — all points on the forward curve always move in the same direction. While this type of forward curve move indeed is the most commonly observed type of shift to the curve, “rotational steepenings” and the formation of “humps” may also take place, as may other more complex types of curve perturbations. The empirical presence of such non-trivial curve movements is an indication of the fact that various points on the forward curve do not move co-monotonically with each other, i.e. they are imperfectly correlated. A key characteristic of the LM model is the consistent use of vector-valued Brownian motion drivers, of dimension m , which gives us control over the instantaneous correlation between various points on the forward curve.

Proposition 15.3.1. *The correlation between forward rate increments $dL_k(t)$ and $dL_j(t)$ in the SV-model (15.16) is*

$$\text{Corr}(dL_k(t), dL_j(t)) = \frac{\lambda_k(t)^\top \lambda_j(t)}{\|\lambda_k(t)\| \|\lambda_j(t)\|}.$$

Proof. Using the covariance notation of Remark 2.1.7, we have, for any j and k ,

$$d\langle L_k(t), L_j(t) \rangle = z(t) \varphi(L_k(t)) \varphi(L_j(t)) \lambda_k(t)^\top \lambda_j(t) dt.$$

Using this in the definition of the correlation,

$$\text{Corr}(dL_k(t), dL_j(t)) = \frac{\langle dL_k(t), dL_j(t) \rangle}{\sqrt{\langle dL_k(t) \rangle \langle dL_j(t) \rangle}},$$

gives the result of the proposition. \square

A trivial corollary of Proposition 15.3.1 is the fact that $\text{Corr}(dL_k(t), dL_j(t)) = 1$ always when $m = 1$, i.e. when we only have one Brownian motion. As we add more Brownian motions, our ability to capture increasingly complicated correlation structures progressively improves (in a sense that we shall examine further shortly), but at a cost of increasing the model complexity and, ultimately, computational effort. To make rational decisions about the choice of model dimension m , let us turn to the empirical data.

15.3.1 Empirical Principal Components Analysis

For some fixed value of τ (e.g. 0.25 or 0.5) define “sliding” forward rates⁴ l with tenor x as

$$l(t, x) = L(t, t + x, t + x + \tau).$$

For a given set of tenors x_1, \dots, x_{N_x} and a given set of calendar times t_0, t_1, \dots, t_{N_t} , we use market observations⁵ to set up the $N_x \times N_t$ observation matrix O with elements

$$O_{i,j} = \frac{l(t_j, x_i) - l(t_{j-1}, x_i)}{\sqrt{t_j - t_{j-1}}}, \quad i = 1, \dots, N_x, \quad j = 1, \dots, N_t.$$

Notice the normalization with $\sqrt{t_j - t_{j-1}}$ which annualizes the variance of the observed forward rate increments. Also note that we use absolute increments in forward rates here. This is arbitrary — we could have used, say, relative increases as well, if we felt that rates were more log-normal than Gaussian. For small sampling periods, the precise choice is of little importance.

Assuming time-homogeneity and ignoring small drift terms, the data collected above will imply a sample $N_x \times N_x$ variance-covariance matrix equal to

$$C = \frac{OO^\top}{N_t}.$$

For our LM model to conform to empirical data, we need to use a sufficiently high number m of Brownian motions to closely replicate this variance-covariance matrix. A formal analysis of what value of m will suffice can proceed with the tools of principal components analysis (PCA), as established in Section 4.1.3.

⁴The use of sliding forward rates, i.e. forward rates with a fixed time to maturity rather than a fixed time of maturity, is often known as the *Musiela parameterization*.

⁵For each date in the time grid t_j we thus construct the forward curve from market observable swaps, futures, and deposits, using techniques such as those in Chapter 7.

15.3.1.1 Example: USD Forward Rates

To give a concrete example of a PCA run, we set $N_x = 9$ and use tenors of $(x_1, \dots, x_9) = (0.5, 1, 2, 3, 5, 7, 10, 15, 20)$ years. We fix $\tau = 0.5$ (i.e., all forwards are 6-months discrete rates) and use 4 years of weekly data from the USD market, spanning January 2003 to January 2007, for a total of $N_t = 203$ curve observations. The eigenvalues of the matrix C are listed in Table 15.2, along with the percentage of variance that is explained by using only the first m principal components.

m	1	2	3	4	5	6	7	8	9
Eigenvalue	7.0	0.94	0.29	0.064	0.053	0.029	0.016	0.0091	0.0070
% Variance	83.3	94.5	97.9	98.7	99.3	99.6	99.8	99.9	100

Table 15.2. PCA for USD Rates. All eigenvalues are scaled up by 10^4 .

As we see from the table, the first principal component explains about 83% of the observed variance, and the first three together explain nearly 98%. This pattern carries over to most major currencies, and in many applications we would consequently expect that using $m = 3$ or $m = 4$ Brownian motions in a LM model would adequately capture the empirical covariation of the points on the forward curve. An exception to this rule-of-thumb occurs when a particular derivative security depends strongly on the correlation between forward rates with tenors that are close to each other; in this case, as we shall see in Section 15.3.4, a high number of principal components is required to provide for sufficient decoupling of nearby forwards.

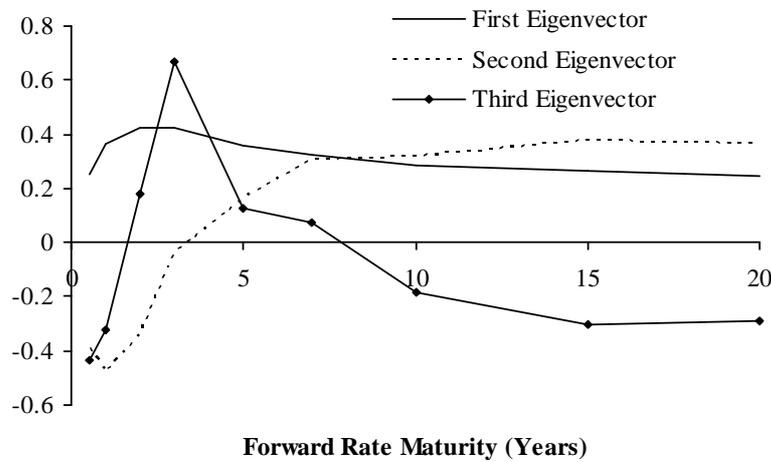
The eigenvectors corresponding to the largest three eigenvectors in Table 15.2 are shown in the Figure 15.1; the figure gives us a reasonable idea about what the (suitably scaled) first three elements of the $\lambda_k(t)$ vectors should look like as functions of $T_k - t$. Loosely speaking, the first principal component can be interpreted as a near-parallel shift of the forward curve, whereas the second and third principal components correspond to forward curve twists and bends, respectively.

15.3.2 Correlation Estimation and Smoothing

Empirical estimates for forward rate correlations can proceed along the lines of Section 15.3.1. Specifically, if we introduce the diagonal matrix

$$c \triangleq \begin{pmatrix} \sqrt{C_{1,1}} & 0 & \cdots & 0 \\ 0 & \sqrt{C_{2,2}} & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \cdots & \sqrt{C_{N_x, N_x}} \end{pmatrix},$$

Fig. 15.1. Eigenvectors



Notes: Eigenvectors for the largest three eigenvalues in Table 15.2.

then the empirical $N_x \times N_x$ forward rate correlation matrix R becomes

$$R = c^{-1} C c^{-1}.$$

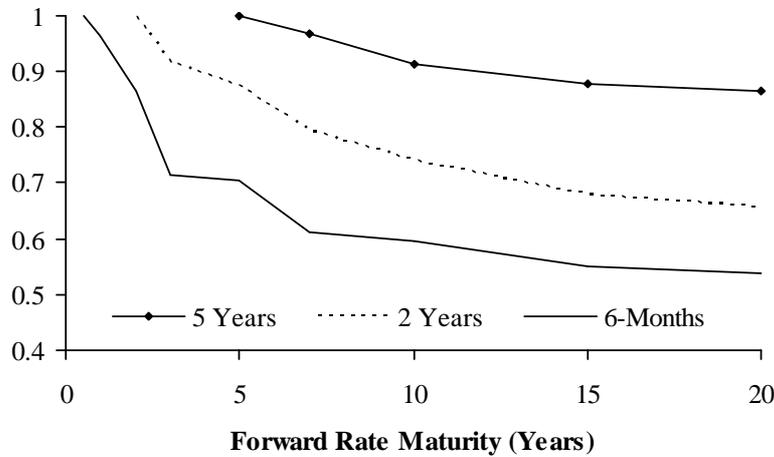
Element $R_{i,j}$ of R provides a sample estimate of the instantaneous correlation between increments in $l(t, x_i)$ and $l(t, x_j)$, under the assumption that this correlation is time-homogeneous.

The matrix R is often relatively noisy, partially as a reflection of the fact that correlations are well-known to be quite variable over time, and partially as a reflection of the fact that the empirical correlation estimator has rather poor sample properties with large confidence bounds (see Johnson et al. [1995] for details). Nevertheless, a few stylistic facts can be gleaned from the data. In Figure 15.2, we have graphed a few slices of the correlation matrix for the USD data in Section 15.3.1.1.

To make a few qualitative observations about Figure 15.2, we notice that correlations between forward rates $l(\cdot, x_k)$ and $l(\cdot, x_j)$ generally decline in $|x_k - x_j|$; this decline appears near-exponential for x_k and x_j close to each other, but with a near-flat asymptote for large $|x_k - x_j|$. It appears that the rate of the correlation decay and the level of the asymptote depend not only $|x_k - x_j|$, but also on $\min(x_k, x_j)$. Specifically, the decay rate decreases with $\min(x_k, x_j)$, and the asymptote level increases with $\min(x_k, x_j)$.

In practice, unaltered empirical correlation matrices are typically too noisy for comfort, and might contain non-intuitive entries (e.g., correlation between a 10-year forward and a 2-year forward might come out higher than between a 10-year forward and a 5-year forward). As such, it is common practice in

Fig. 15.2. Forward Rate Correlations



Notes: For each of three fixed forward rate maturities (6 months, 2 years, and 5 years), the figure shows the correlation between the fixed forward rate and forward rates with other maturities (as indicated on the x-axis of the graph).

multi-factor yield curve modeling to work with simple parametric forms; this not only smoothes the correlation matrix, but also conveniently reduces the effective parameter dimension of the correlation distinct matrix object, from $N_x(N_x - 1)/2$ matrix elements to the number of parameters in the parametric form.

Several candidate parametric forms for the correlation have been proposed in the literature, see Rebonato [2002], Jong et al. [2001], and Schoenmakers and Coffey [2000], among many others. Rather than list all of these, we instead focus on a few reasonable forms that we have designed to encompass most or all of the empirical facts listed above. Our first parametric form is as follows:

$$\text{Corr}(dL_k(t), dL_j(t)) = q_1(T_k - t, T_j - t),$$

where

$$q_1(x, y) = \rho_\infty + (1 - \rho_\infty) \exp(-a(\min(x, y)) |y - x|), \tag{15.18}$$

$$a(z) = a_\infty + (a_0 - a_\infty)e^{-\kappa z},$$

subject to $0 \leq \rho_\infty \leq 1$, $a_0, a_\infty, \kappa \geq 0$. Fundamentally, $q_1(x, y)$ exhibits correlation decay at a rate of a as $|y - x|$ is increased, with the decay rate a itself being an exponential function of $\min(x, y)$. We would always expect to have $a_0 \geq a_\infty$, in which case

$$\frac{\partial q_1(x, y)}{\partial x} = (1 - \rho_\infty)e^{-a(x)(y-x)} [a(x) + (y - x)\kappa(a_0 - a_\infty)e^{-\kappa x}], \quad x < y,$$

is non-negative, as one would expect.

Variations on (15.18) are abundant in the literature — the case $a_0 = a_\infty$ is particularly popular — and q_1 generally has sufficient degrees of freedom to provide a reasonable fit to empirical data. One immediate issue, however, is a lack of control of the asymptotic correlation level at $|x - y| \rightarrow \infty$ which, as we argued above, is typically not independent of x and y . As the empirical data suggests that ρ_∞ tends to increase with $\min(x, y)$, we introduce yet another decaying function

$$\rho_\infty(z) = b_\infty + (b_0 - b_\infty)e^{-\alpha z}, \quad (15.19)$$

and extend q_1 to the “triple-decaying” form

$$q_2(x, y) = \rho_\infty(\min(x, y)) + (1 - \rho_\infty(\min(x, y))) \exp(-a(\min(x, y))|y - x|) \quad (15.20)$$

with $a(z)$ is given in (15.18), and where $0 \leq b_0, b_\infty \leq 1, \alpha \geq 0$. Empirical data suggests that normally $b_0 \leq b_\infty$, in which case we have

$$\begin{aligned} \frac{\partial q_2(x, y)}{\partial x} &= -\alpha(b_0 - b_\infty)e^{-\alpha x} \left(1 - e^{-a(x)(y-x)}\right) \\ &+ (1 - \rho_\infty(x)) e^{-a(x)(y-x)} \times [a(x) + (y - x)\kappa(a_0 - a_\infty)e^{-\kappa x}], \quad x < y, \end{aligned}$$

which remains non-negative if $b_0 \leq b_\infty$ and $a_0 \geq a_\infty$.

In a typical application, the four parameters of q_1 and the six parameters of q_2 are found by least-squares optimization against an empirical correlation matrix. Any standard optimization algorithm, such as the Levenberg-Marquardt algorithm in Press et al. [1992], can be used for this purpose. Some parameters are here subject to simple box-style constraints (e.g. $\rho_\infty \in [0, 1]$) which poses no particular problems for most commercial optimizers. In any case, we can always use functional mappings to rewrite our optimization problem in terms of variables with unbounded domains. For instance, for form q_1 , we can set

$$\rho_\infty = \frac{1}{2} + \frac{\arctan(u)}{\pi}, \quad u \in (-\infty, \infty),$$

and optimize on the variable u instead of ρ_∞ . Occasionally, we may sometimes wish to optimize correlation parameters against more market-driven targets than empirical correlation matrices; see Section 15.5.9 for details on this.

15.3.2.1 Example: Fit to USD data

Let R be the 10×10 empirical correlation matrix generated from the data in Section 15.3.1.1, and let $R_2(\xi)$, $\xi \equiv (a_0, a_\infty, \kappa, b_0, b_\infty, \alpha)^\top$, be the 10×10 correlation matrix generated from the form q_2 , when using the 10 specific forward tenors in 15.3.1.1. To determine the optimal parameter vector ξ^* , we minimize an unweighted Frobenius (least-squares) matrix norm, subject to a non-negativity constraint

$$\xi^* = \operatorname{argmin}_{\xi} \left(\operatorname{tr} \left((R - R_2(\xi))(R - R_2(\xi))^{\top} \right) \right), \text{ subject to } \xi \geq 0.$$

The resulting fit is summarized in Table 15.3; Figure 15.3 in Section 15.3.4.1 contains a 3D plot of the correlation matrix $R_2(\xi^*)$.

a_0	a_{∞}	κ	b_0	b_{∞}	α
0.312	0.157	0.264	0.490	0.946	0.325

Table 15.3. Best-Fit Parameters for q_2 in USD Market

The value of the Frobenius norm at ξ^* was 0.070, which translates into an average absolute correlation error (excluding diagonal elements) of around 2%. If we use the four parameter form q_1 instead of q_2 in the optimization exercise, the Frobenius norm at the optimum increases to 0.164. As we would expect from Figure 15.2, allowing correlation asymptotes to increase in tenors thus adds significant explanatory power to the parametric form.

15.3.3 Negative Eigenvalues

While some functional forms are designed to always return valid correlation matrices (the function in Schoenmakers and Coffey [2000] being one such example), many popular forms — including our q_1 and q_2 above — can, when stressed, generate matrices R that fail to be positive definite. While this rarely happens in real applications, it is not inconceivable that on occasion one or more eigenvalues of R may turn out to be negative, requiring us to somehow “repair” the matrix. A similar problem can also arise due to rounding errors when working with large empirical correlation matrices.

Formally, when faced with an R matrix that is not positive definite, we would ideally like to replace it with a modified matrix R^* which i) is a valid correlation matrix; and ii) is as close as possible to R , in the sense of some matrix norm. The problem of locating R^* then involves computing the norm

$$\{\|R - X\| : X \text{ is a correlation matrix}\}$$

and setting R^* equal to the matrix X that minimizes this distance. If $\|\cdot\|$ is a weighted Frobenius norm, numerical algorithms for the computation of R^* have recently emerged, see Higham [2002] for a review and a clean approach.

If the negative eigenvalues are small in absolute magnitude (which is often the case in practice), it is often reasonable to abandon a full-blown optimization algorithm in favor of a more heuristic approach where we simply raise all offending negative eigenvalues to some positive cut-off value; we present one obvious algorithm below.

As a starting point, we write

$$R = E\Lambda E^\top,$$

where Λ is a diagonal matrix of eigenvalues, and E is a matrix with the eigenvectors of R in its column. Let Λ^* be the diagonal matrix with all-positive entries

$$\Lambda_{ii}^* = \max(\epsilon, \Lambda_{ii}), \quad i = 1, \dots, N_x,$$

for some small cut-off value $\epsilon > 0$. Then set

$$C^* = E\Lambda^*E^\top,$$

which we interpret as a covariance matrix, i.e. of the form

$$C^* = c^*R^*c^*$$

where c^* is a diagonal matrix with elements $c_{ii}^* = \sqrt{C_{ii}^*}$ and R^* is the valid, positive definite correlation matrix we seek. R^* is then computed as

$$R^* = (c^*)^{-1}C^*(c^*)^{-1}. \quad (15.21)$$

We emphasize that R^* as defined in (15.21) will have 1's in its diagonal, whereas C^* will not. Both C^* and R^* are, by construction, positive definite.

15.3.4 Correlation PCA

We now turn to a problem that arises in certain important applications, such as the calibration procedure we shall discuss in Section 15.5. Consider a p -dimensional Gaussian variable Y , where all elements of Y have zero mean and unit variance. Let Y have a positive definite correlation matrix R , given by

$$R = E(YY^\top).$$

Consider now writing, as an approximation,

$$Y \approx DX \quad (15.22)$$

where X is an m -dimensional vector of independent standard Gaussian variables, $m < p$, and D is a $p \times m$ -dimensional matrix. We wish to strictly enforce that DX remains a vector of variables with zero means and unit variances, thereby ensuring that the matrix DD^\top has the interpretation of a valid correlation matrix. In particular, we require that the diagonal of DD^\top is one everywhere.

Let $v(D)$ be the p -dimensional vector of the diagonal elements of DD^\top , i.e. $v_i = (DD^\top)_{ii}$, $i = 1, \dots, p$. Working as before with an unweighted⁶ Frobenius norm, we set

⁶The introduction of user-specified weights into this norm is a straightforward extension.

$$h(D; R) = \text{tr} \left((R - DD^\top) (R - DD^\top)^\top \right), \quad (15.23)$$

and define the optimal choice of D , denoted D^* , as

$$D^* = \text{argmin}_D h(D; R), \quad \text{subject to } v(D) = \mathbf{1}, \quad (15.24)$$

where $\mathbf{1}$ is a p -dimensional vector of 1's.

Proposition 15.3.2. *Let μ be a p -dimensional vector, and let D_μ be given as the unconstrained optimum*

$$D_\mu = \text{argmin}_D h(D; R + \text{diag}(\mu)),$$

with h given in (15.23). Define D^* as in (15.24) and let μ^* be the solution to

$$v(D_\mu) - \mathbf{1} = \mathbf{0}.$$

Then $D^* = D_{\mu^*}$.

Proof. We only provide a sketch of the proof; for more details, see Zhang and Wu [2003]. First, we introduce the Lagrangian

$$\mathfrak{S}(D, \mu) = h(D; R) - 2\mu^\top (v(D) - \mathbf{1}).$$

(The factor 2 on μ^\top simplifies results). Standard matrix calculus shows that

$$\frac{d\mathfrak{S}}{dD} = \left\{ \frac{d\mathfrak{S}}{dD_{i,j}} \right\} = -4RD + 4DD^\top D.$$

Setting the derivatives of the Lagrangian \mathfrak{S} with respect to D and μ to zero yields, after a little rearrangement,

$$-(R + \text{diag}(\mu))D + DD^\top D = \mathbf{0}, \quad v(D) = \mathbf{1}.$$

The first of these conditions identifies the optimum as minimizing the (unconstrained) optimization norm $h(D; R + \text{diag}(\mu))$. \square

Remark 15.3.3. For any fixed value of μ , D_μ can be computed easily by standard PCA methods provided we interpret $R + \text{diag}(\mu)$ as the target covariance matrix.

With Proposition 15.3.2, determination of D^* is reduced to solving the p -dimensional root-search problem $v(D_\mu) - \mathbf{1} = \mathbf{0}$ for μ . Many standard methods will suffice; for instance, one can use straightforward secant search methods such as the Broyden algorithm in [Press et al., 1992, p. 389].

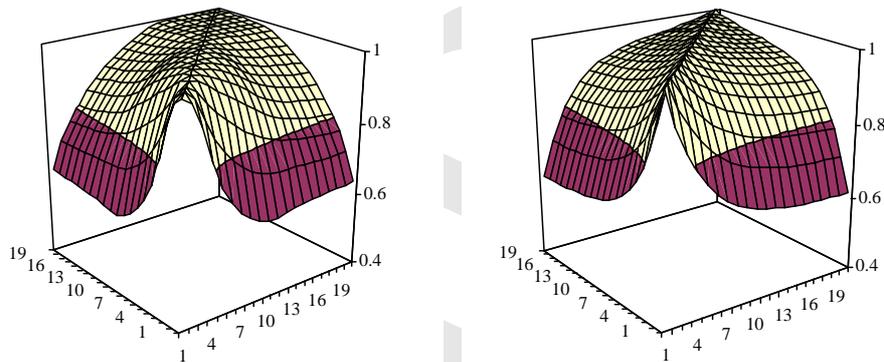
As is the case for ordinary PCA approximations of covariance matrices, the ‘‘correlation PCA’’ algorithm outlined so far will return a correlation matrix approximation $D^*(D^*)^\top$ that has reduced rank (from p down to m), a consequence of the PCA steps taken in estimating D_μ .

Computation of optimal rank-reduced correlation approximations is a relatively well-understood problems, and we should note the existence of several recent alternatives to the basic algorithm we outline here. A survey can be found in [Pietersz and Groenen \[2004\]](#) where an algorithm based on *majorization* is developed⁷. We should also note that certain heuristic (and non-optimal) methods have appeared in the literature, some of which are closely related to the simple algorithm we outlined in Section 15.3.3 for repair of correlation matrices. We briefly outline one such approach below (in Section 15.3.4.2), but first we consider a numerical example.

15.3.4.1 Example: USD Data

We here consider performing a correlation PCA analysis the correlation matrix R generated from our best-fit form q_2 in Section 15.3.2.1. The 3D plots in Figure 15.3 below show the correlation fit we get with a rank 3 correlation matrix.

Fig. 15.3. Forward Rate Correlation Matrix in USD



Notes: The right-hand panel shows the correlation matrix R for form q_2 calibrated to USD data. The left-hand panel shows the best-fitting rank-3 correlation matrix, computed by the algorithm in Proposition 15.3.2. In both graphs, the x - and y -axes represent the Libor forward rate maturity in years.

Looking at Figure 15.3, the effect of rank-reduction is, loosely, that the exponential decay of our original matrix R has been replaced with a “sigmoid” shape (to paraphrase Riccardo Rebonato) that is substantially too high close

⁷In our experience, the majorization method in [Pietersz and Groenen \[2004\]](#) is faster than the method in Proposition 15.3.2 but, contrary to claims in [Pietersz and Groenen \[2004\]](#), less robust, particularly for very large and irregular correlation matrices.

to the matrix diagonal. As the rank of the approximating correlation matrix is increased, the sigmoid shape is — often rather slowly — pulled towards exponential shape of the full-rank data. Intuitively, we should not be surprised at this result: with the rank m being a low number, we effectively only incorporate smooth, large-scale curve movements (e.g. parallel shifts and twists) into our statistical model, and there is no mechanism to “pull apart” forward rates with maturities close to each other.

Analysis of this difference — rather than the simple PCA considerations of Section 15.3.1 — often forms the basis for deciding how many factors m to use in the model, especially for pricing derivatives with strong correlation dependence. For the reader’s guidance, we find that $m = 5$ to 10 suffices to recover the full-rank correlation shape in most cases.

15.3.4.2 Poor Man’s Correlation PCA

For the case where the $p \times p$ correlation matrix R is well-represented by a rank- m representation of the form (15.22), it may sometimes be sufficiently accurate to compute the loading matrix D by a simpler algorithm based on *standard* PCA applied directly to the correlation matrix. Specifically, suppose that we as a first step compute

$$R_m = E_m \Lambda_m E_m^\top$$

where Λ_m is an $m \times m$ diagonal matrix of the m largest eigenvalues of R , and E_m is a $p \times m$ matrix of eigenvectors corresponding to these eigenvalues. While the error $R_m - R$ minimizes a least-squares norm, R_m itself is obviously not a valid approximation to the correlation matrix R as no steps were taken to ensure that R_m has a unit diagonal. A simple way to accomplish this borrows the ideas of Section 15.3.3 and writes

$$R \approx r_m^{-1} R_m r_m^{-1} \tag{15.25}$$

where r_m is a diagonal matrix with elements $(r_m)_{ii} = \sqrt{(R_m)_{ii}}$, $i = 1, \dots, p$. We note that this approximation sets the matrix D in (15.22) to

$$D = r_m^{-1} E_m \sqrt{\Lambda_m}.$$

It is clear that the difference between the “poor man’s” PCA result (15.25) and the optimal result in Proposition 15.3.2 will generally be small if R_m is close to having a unit diagonal, as the heuristic step taken in (15.25) will then have little effect. For large, complex correlation matrices, however, unless m is quite large, the optimal approximation in Proposition 15.3.2 will often be quite different from (15.25).

15.4 Pricing of European Options

The previous section laid the foundation for calibration of an LM model to empirical forward curve correlation data, a topic that we shall return to in more detail in Section 15.5. Besides correlation calibration, however, we need to ensure that the forward rate *variances* implied by the LM model are in line with market data. In most applications — and certainly in all those that involve pricing and hedging of traded derivatives — this translates into a requirement that the vectors $\lambda_k(t)$ of the model are such that it will successfully reproduce the prices of liquid plain-vanilla derivatives, i.e. swaptions and caps. A condition for practical uses of the LM model is thus that we can find pricing formulas for vanilla options that are fast enough to be embedded into an iterative calibration algorithm.

15.4.1 Caplets

Deriving formulas for caplets is generally straightforward in the LM model, a consequence of the fact that Libor rates — which figure directly in the payout formulas for caps — are the main primitives of the LM model itself. Indeed, the word “market” in the term “Libor market model” originates from the ease with which the model can accommodate market-pricing of caplets by the Black formula.

As our starting point here, we use the generalized version of the LM model with skews and stochastic volatility; see (15.15) and (15.16). Other, simpler models, are special cases of this framework, and the fundamental caplet pricing methodology will carry over to these cases in a transparent manner. We consider a c -strike caplet $V_{\text{caplet}}(\cdot)$ maturing at time T_n and settling at time T_{n+1} . That is,

$$V_{\text{caplet}}(T_{n+1}) = \tau_n (L_n(T_n) - c)^+.$$

For the purpose of pricing the caplet, the m -dimensional Brownian motion $W^{n+1}(t)$ can here be reduced to one dimension, as shown in the following result.

Proposition 15.4.1. *Assume that the forward rate dynamics in the spot measure are as in (15.15)–(15.16), and that Assumption 15.2.7 holds. Then*

$$V_{\text{caplet}}(0) = P(0, T_{n+1}) \tau_n \mathbb{E}^{T_{n+1}} \left((L_n(T_n) - c)^+ \right),$$

where

$$\begin{aligned} dL_n(t) &= \sqrt{z(t)} \varphi(L_n(t)) \|\lambda_n(t)\| dY^{n+1}(t), \\ dz(t) &= \theta(z_0 - z(t)) dt + \eta \psi(z(t)) dZ(t), \end{aligned} \quad (15.26)$$

and $Y^{n+1}(t)$ and $Z(t)$ are independent scalar Brownian motions in measure $\mathbb{Q}^{T_{n+1}}$. Specifically, $Y^{n+1}(t)$ is given by

$$Y^{n+1}(t) = \int_0^t \frac{\lambda_n(s)^\top}{\|\lambda_n(s)\|} dW^{n+1}(s).$$

Proof. $Y^{n+1}(t)$ is clearly Gaussian, with mean 0 and variance \sqrt{t} , identifying it as a Brownian motion such that $\|\lambda_n(t)\| dY^{n+1}(t) = \lambda_n(t)^\top dW^{n+1}(t)$. The remainder of the Proposition follows from the martingale property of L_n in $Q^{T_{n+1}}$, combined with the assumed independence of the forward rates and the z -process. \square

While rather obvious, Proposition 15.4.1 is useful as it demonstrates that caplet pricing reduces to evaluation of an expectation of $(L_n(T_n) - c)^+$, where the process for L_n is now identical to the types of scalar stochastic volatility diffusions covered in detail in Chapters 9 and 10; the pricing of caplets can therefore be accomplished with the formulas listed in these chapters. In the same way, when dealing with LM models of the simpler local volatility type, we compute caplet prices directly from formulas in Chapter 8.

15.4.2 Swaptions

Whereas pricing of caplets is, by design, convenient in LM models, swaption pricing requires a bit more work and generally will involve some amount of approximation if a quick algorithm is required. In this section, we will outline one such approximation which normally has sufficient accuracy for calibration applications. A more accurate (but also more complicated) approach can be found in Chapter 16.

First, let us recall some notations. Let $V_{\text{swaption}}(t)$ denote the time t value of a payer swaption that matures at time $T_j \geq t$, with the underlying security being a fixed-for-floating swap making payments at times T_{j+1}, \dots, T_k , where $j < k \leq N$. We define an annuity factor for this swap as (see (5.8))

$$A(t) \triangleq A_{j,k-j}(t) = \sum_{n=j}^{k-1} P(t, T_{n+1}) \tau_n, \quad \tau_n = T_{n+1} - T_n. \quad (15.27)$$

Assuming that the swap underlying the swaption pays a fixed coupon of c against Libor flat, the payout of V_{swaption} at time T_j is (see Section 5.1.3)

$$V_{\text{swaption}}(T_j) = A(T_j) (S(T_j) - c)^+.$$

where we have defined a par forward swap rate (see (5.10))

$$S(t) \triangleq S_{j,k-j}(t) = \frac{P(t, T_j) - P(t, T_k)}{A(t)}.$$

Assume, as in Section 15.4.1, that we are working in the setting of a stochastic volatility LM model, of the type defined in Section 15.2.5; the procedure we shall now outline will carry over to simpler models unchanged.

Proposition 15.4.2. *Assume that the forward rate dynamics in the spot measure are as in (15.15)–(15.16). Let \mathbb{Q}^A be the measure induced by using $A(t)$ in (15.27) as a numeraire, and let W^A be an m -dimensional Brownian motion in \mathbb{Q}^A . Then, in measure \mathbb{Q}^A ,*

$$dS(t) = \sqrt{z(t)}\varphi(S(t)) \sum_{n=j}^{k-1} w_n(t)\lambda_n(t)^\top dW^A(t), \quad (15.28)$$

where the stochastic weights are

$$w_n(t) = \frac{\varphi(L_n(t))}{\varphi(S(t))} \times \frac{\partial S(t)}{\partial L_n(t)} = \frac{\varphi(L_n(t))}{\varphi(S(t))} \times \frac{S(t)\tau_n}{1 + \tau_n L_n(t)} \times \left[\frac{P(t, T_k)}{P(t, T_j) - P(t, T_k)} + \frac{\sum_{i=n}^{k-1} \tau_i P(t, T_{i+1})}{A(t)} \right]. \quad (15.29)$$

Proof. It follows from Lemma 5.2.4 that $S(t)$ is a martingale in measure \mathbb{Q}^A , hence we know that the drift of the process for $S(t)$ must be zero in this measure. From its definition, $S(t)$ is a function of $L_j(t), L_{j+1}(t), \dots, L_{k-1}(t)$, and an application of Ito's lemma shows that

$$dS(t) = \sum_{n=j}^{k-1} \sqrt{z(t)}\varphi(L_n(t)) \frac{\partial S(t)}{\partial L_n(t)} \lambda_n(t)^\top dW^A(t).$$

Evaluating the partial derivative proves the proposition. \square

It should be immediately obvious that the dynamics of the par rate in (15.28) are too complicated to allow for analytical treatment. The main culprit are the random weights $w_n(t)$ in (15.29) which depend on the entire forward curve in a complex manner. All is not lost, however, as one would intuitively expect that $S(t)$ is well-approximated by a weighted sum of its “component” forward rates $L_j(t), L_{j+1}(t), \dots, L_{k-1}(t)$, with weights varying little over time. In other words, we expect that, for each n , $\partial S(t)/\partial L_n(t)$ is a near-constant quantity.

Consider now the ratio $\varphi(L_n(t))/\varphi(S(t))$ which multiplies $\partial S(t)/\partial L_n(t)$ in (15.29). For forward curves that are reasonably flat and forward curve movements that are predominantly parallel (which is consistent with our earlier discussion in Section 15.3.1.1), it is often reasonable to assume that the ratio is close to constant. This assumption obviously hinges on the precise form of φ , but holds well for many of the functions that we would consider using in practice. To provide some loose motivation for this statement, consider first the extreme case where $\varphi(x) = \text{Const}$ (i.e. the model is Gaussian) in which case the ratio $\varphi(L_n(t))/\varphi(S(t))$ is constant, by definition. Second, let us consider the log-normal case where $\varphi(x) = x$. In this case, a parallel shift h of the forward curve at time t would move the ratio to

$$\frac{L_n(t) + h}{S(t) + h} = \frac{L_n(t)}{S(t)} + h \frac{S(t) - L_n(t)}{S(t)^2} + O(h^2),$$

which is small if the forward curve slope (and thereby $S(t) - L_n(t)$) is small. As the φ 's that we use in practical applications are mostly meant to produce skews that lie somewhere between log-normal and Gaussian ones, assuming that $\varphi(L_n(t))/\varphi(S(t))$ is constant thus appears reasonable.

The discussion above leads to the following approximation, where we “freeze” the weights $w_n(t)$ at their time 0 values.

Proposition 15.4.3. *The time 0 price of the swaption is given by*

$$V_{\text{swaption}}(0) = A(0)E^A((S(T_j) - c)^+). \quad (15.30)$$

Let $w_n(t)$ be as in Proposition 15.4.2 and set

$$\lambda_S(t) = \sum_{n=j}^{k-1} w_n(0)\lambda_n(t),$$

The swap rate dynamics in Proposition 15.4.2 can then be approximated as

$$\begin{aligned} dS(t) &\approx \sqrt{z(t)}\varphi(S(t))\|\lambda_S(t)\| dY^A(t), \\ dz(t) &= \theta(z_0 - z(t))dt + \eta\psi(z(t))dZ(t), \end{aligned} \quad (15.31)$$

where $Y^A(t)$ and $Z(t)$ are independent scalar Brownian motions in measure Q^A , and

$$\|\lambda_S(t)\| dY^A(t) = \sum_{n=j}^{k-1} w_n(0)\lambda_n(t)^\top dW^A(t).$$

Proof. The equation (15.30) follows from standard properties of Q^A . The remainder of the proposition is proven the same way as Proposition 15.4.1, after approximating $w_n(t) \approx w_n(0)$. \square

We emphasize that the scalar term $\|\lambda_S(\cdot)\|$ is purely deterministic, whereby the dynamics of $S(\cdot)$ in the annuity measure have precisely the same form as the Libor rate SDE in Proposition 15.4.1. Therefore, computation of the Q^A -expectation in (15.30) can lean directly on the analytical results we established for scalar stochastic volatility processes in Chapter 9 and, for simpler DVF-type LM models, in Chapter 8. We review relevant results and apply them to LM models in Chapter 16; here, to give an example, we list a representative result for a displaced log-normal local volatility LM model.

Proposition 15.4.4. *Let each rate $L_n(\cdot)$ follow a displaced log-normal process in its own forward measure,*

$$dL_n(t) = (bL_n(t) + (1-b)L_n(0))\lambda(t)^\top dW^{n+1}, \quad n = 1, \dots, N-1.$$

Then the time 0 price of the swaption is given by

$$V_{\text{swaption}}(0) = A(0)c_B(0, S(0)/b; T_j, c - S(0) + S(0)/b, b\bar{\lambda}_S),$$

where $c_B(\dots, \sigma)$ is the Black formula with volatility σ , see Remark 8.2.8, and the term swap rate volatility $\bar{\lambda}_S$ is given by

$$\bar{\lambda}_S = \left(\frac{1}{T_j} \int_0^{T_j} \|\lambda_S(t)\|^2 dt \right)^{1/2},$$

with $\|\lambda_S(t)\|$ defined in Proposition 15.4.3.

Proof. By Proposition 15.4.3, the approximate dynamics of $S(\cdot)$ are given by

$$dS(t) \approx (bS(t) + (1-b)S(0)) \|\lambda_S(t)\| dY^A(t).$$

The result then follows by Proposition 8.2.12. \square

While we do not document the performance of the approximation (15.31) in detail here, many tests are available in the literature; see e.g. Andersen and Andreasen [2000b], Rebonato [2002], and Glasserman and Merener [2001]. Suffice to say that the approximation above is virtually always accurate enough for the calibration purposes for which it was designed, particularly if we restrict ourselves to pricing swaptions with strikes close to the forward swap rate. As mentioned above, should further precision be desired, one can turn to the more sophisticated swaption pricing approximations that we discuss in Chapter 16. Finally, we should note the existence of models where no approximations are required to price swaptions; these so-called *swap market models* are reviewed in Section 16.4.

15.4.3 Spread Options

When calibrating LM models to market data, the standard approach is to fix the correlation structure in the model to match empirical forward rate correlations. It is, however, tempting to consider whether one alternatively could imply the correlation structure directly from traded market data, thereby avoiding the need for “backward-looking” empirical data altogether. As it turns out, the dependence of swaption and caps to the correlation structure is, not surprisingly, typically too indirect to allow one to simultaneously back out correlations and volatilities from the prices of these types of instruments alone. To overcome this, one can consider amending the set of calibration instruments with securities that have stronger sensitivity to forward rate correlations. A good choice would here be to use *yield curve spread options*, a type of security that we encountered earlier in Section 6.13.3. Spread options are natural candidates, not only because their prices are highly sensitive to correlation, but also because they are relatively liquid and not too difficult to device approximate prices for in an LM model setting.

15.4.3.1 Term Correlation

Let $S_1(t) = S_{j_1, k_1 - j_1}(t)$ and $S_2(t) = S_{j_2, k_2 - j_2}(t)$ be two forward swap rates, and assume that we work with a stochastic volatility LM model of type

(15.15)–(15.16). Following the result of Proposition 15.4.3, for $i = 1, 2$ we have, to good approximation,

$$dS_i(t) \equiv O(dt) + \sqrt{z(t)}\varphi(S_i(t))\lambda_{S_i}(t)^\top dW^B(t), \quad \lambda_{S_i}(t) \triangleq \sum_{n=j_i}^{k_i-1} w_{S_i,n}(0)\lambda_n(t),$$

where W^B is a vector-valued Brownian motion in the spot measure, and we use an extended notation $w_{S_i,n}$ to emphasize which swap rate a given weight relates to. Notice the presence of drift terms, of order $O(dt)$. The quadratic variation and covariation of $S_1(t)$ and $S_2(t)$ satisfy

$$\begin{aligned} d\langle S_1(t), S_2(t) \rangle &= z(t)\varphi(S_1(t))\varphi(S_2(t))\lambda_{S_1}(t)^\top\lambda_{S_2}(t) dt, \\ d\langle S_i(t) \rangle &= z(t)\varphi(S_i(t))^2\|\lambda_{S_i}(t)\|^2 dt, \quad i = 1, 2, \end{aligned}$$

and the instantaneous correlation is

$$\text{Corr}(dS_1(t), dS_2(t)) = \frac{\lambda_{S_1}(t)^\top\lambda_{S_2}(t)}{\|\lambda_{S_1}(t)\|\|\lambda_{S_2}(t)\|}. \quad (15.32)$$

Instead of the instantaneous correlation, in many applications we are normally more interested in an estimate for *term correlation* $\rho_{\text{term}}(\cdot, \cdot)$ of S_1 and S_2 on some finite interval $[T', T]$. Formally, we define this time 0 measurable quantity as

$$\rho_{\text{term}}(T', T) \triangleq \text{Corr}(S_1(T) - S_1(T'), S_2(T) - S_2(T')).$$

Ignoring drift terms and freezing the swap rates at their time 0 forward levels, to decent approximation we can write

$$\begin{aligned} \rho_{\text{term}}(T', T) &\approx \frac{\varphi(S_1(0))\varphi(S_2(0))\int_{T'}^T \mathbb{E}^B(z(t))\lambda_{S_1}(t)^\top\lambda_{S_2}(t) dt}{\varphi(S_1(0))\varphi(S_2(0))\prod_{i=1}^2\sqrt{\int_{T'}^T \mathbb{E}^B(z(t))\|\lambda_{S_i}(t)\|^2 dt}} \\ &= \frac{\int_{T'}^T \lambda_{S_1}(t)^\top\lambda_{S_2}(t) dt}{\sqrt{\int_{T'}^T \|\lambda_{S_1}(t)\|^2 dt}\sqrt{\int_{T'}^T \|\lambda_{S_2}(t)\|^2 dt}}, \end{aligned} \quad (15.33)$$

where we in the second equality have used that fact that the parametrization (15.15) implies that, for all $t \geq 0$,

$$\mathbb{E}^B(z(t)) = z_0.$$

15.4.3.2 Spread Option Pricing

Consider a spread option paying at time $T \leq \min(T_{j_1}, T_{j_2})$

$$V_{\text{spread}}(T) = (S_1(T) - S_2(T) - K)^+,$$

such that

$$V_{\text{spread}}(0) = P(0, T) E^T (S_1(T) - S_2(T) - K)^+,$$

where, as always, E^T denotes expectations in measure Q^T . An accurate (analytic) evaluation of this expected value is somewhat involved, and we postpone it till Chapter 18. Here, as a preview, we consider a cruder approach which may, in fact, be adequate for calibration purposes. We assume that the spread

$$\varepsilon(T) = S_1(T) - S_2(T)$$

is a Gaussian variable with mean

$$E^T (\varepsilon(T)) = E^T (S_1(T)) - E^T (S_2(T)).$$

In a pinch, the mean of $\varepsilon(T)$ can be approximated as $S_1(0) - S_2(0)$, which assumes that the drift terms of $S_1(\cdot)$ and $S_2(\cdot)$ in the T -forward measure are approximately identical. For a better approximation, see Chapter 17. As for the variance of $\varepsilon(T)$, it can be approximated in several different ways, but one approach simply writes

$$\begin{aligned} \text{Var}^T (\varepsilon(T)) &\approx \sum_{i=1}^2 \varphi(S_i(0))^2 z_0 \int_0^T \|\lambda_{S_i}(t)\|^2 dt \\ &\quad - 2\rho_{\text{term}}(0, T) z_0 \prod_{i=1}^2 \varphi(S_i(0)) \left(\int_0^T \|\lambda_{S_i}(t)\|^2 dt \right)^{1/2}. \end{aligned} \quad (15.34)$$

With these approximations, the Bachelier formula (8.15) yields

$$V_{\text{spread}}(0) = P(0, T) \sqrt{\text{Var}^T (\varepsilon(T))} (d\Phi(d) + \phi(d)), \quad d = \frac{E^T (\varepsilon(T)) - K}{\sqrt{\text{Var}^T (\varepsilon(T))}}. \quad (15.35)$$

15.5 Calibration

15.5.1 Basic Principles

Suppose that we have fixed the tenor structure, have decided upon the number of factors m to be used, and have selected the basic form (e.g. DVF or SV) of the LM model that we are interested in deploying. Suppose also, for now, that any skew functions and stochastic volatility dynamics have been exogenously specified by the user. To complete our model specification, what then remains is the fundamental question of how to establish the set of m -dimensional deterministic volatility vectors $\lambda_k(t)$, $k = 1, 2, \dots, N - 1$, that together determine the overall correlation and volatility structure of forward rates in the model.

As evidenced by the large number of different calibration approaches proposed in the literature, there are no precise rules for calibration of LM models. Still, certain common steps are nearly always invoked:

- Prescribe the basic form of $\|\lambda_k(t)\|$, either through direct parametric assumptions, or by introduction of discrete time- and tenor-grids.
- Use correlation information to establish a map from $\|\lambda_k(t)\|$ to $\lambda_k(t)$.
- Choose the set of observable securities against which to calibrate the model.
- Establish the norm to be used for calibration.
- Recover $\lambda_k(t)$ by norm optimization.

In the next few sections, we will discuss these steps in sequence. In doing so, our primary aim is to expose a particular calibration methodology that we personally prefer for most applications, rather than give equal mention to all possible approaches that have appeared in the literature. We note up front that our discussion is tilted towards applications that ultimately involve pricing and hedging of exotic Libor securities (see e.g. Chapters 19 and 20).

15.5.2 Parameterization of $\|\lambda_k(t)\|$

For convenience, let us write

$$\lambda_k(t) = h(t, T_k - t), \quad \|\lambda_k(t)\| = g(t, T_k - t), \quad (15.36)$$

for some functions $h : \mathbb{R}_+^2 \rightarrow \mathbb{R}^m$ and $g : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$ to be determined. We focus on g in this section, and start by noting that many ad-hoc parametric forms for this function have been proposed in the literature. A representative example is the following 4-parameter specification, due to [Rebonato \[1998\]](#):

$$g(t, x) = g(x) = (a + bx)e^{-cx} + d, \quad a, b, c, d \in \mathbb{R}_+. \quad (15.37)$$

We notice that this specification is *time-stationary* in the sense that $\|\lambda_k(t)\|$ does not depend on calendar-time t , but only on the remaining time to maturity ($T_k - t$) of the forward rate in question. While attractive from a perspective of smoothness of model volatilities, assumptions of perfect time-stationarity will generally not allow for a sufficiently accurate fit to market prices. To address this, some authors have proposed “separable” extensions of the type

$$g(t, x) = g_1(t)g_2(x), \quad (15.38)$$

where g_1 and g_2 are to be specified separately. See [Brace et al. \[1996\]](#) for an early approach along these lines.

For the applications we have in mind, relying on separability or parametric forms is ultimately too inflexible, and we seek a more general approach. For this, let us introduce a rectangular grid of times and tenors $\{t_i\} \times \{x_j\}$, $i =$

$1, \dots, N_t, j = 1, \dots, N_x$; and an $N_t \times N_x$ -dimensional matrix G . The elements $G_{i,j}$ will be interpreted as

$$g(t_i, x_j) = G_{i,j}. \quad (15.39)$$

When dimensioning the grid $\{t_i\} \times \{x_j\}$, we would normally⁸ require that $t_1 + x_{N_x} \geq T_N$, to ensure that all forwards in the Libor forward curve are covered by the table; beyond this, there need not be any particular relationship between the grid and the chosen tenor structure, although we find it convenient to ensure that $t_i + x_j \in \{T_n\}$ as long as $t_i + x_j \leq T_N$ — a convention we adopt from now on. Note that the bottom right-hand corner of the grid contains Libor maturities beyond that of our tenor structure and would effectively be redundant.

A few further comments on the grid-based approach above are in order. First, we notice that both time-stationary and separable specifications along the lines of (15.37) and (15.38) can be emulated closely as special cases of the grid-based approach. For instance, the parametric specification (15.37) would give rise to a matrix G where

$$G_{i,j} = (a + bx_j)e^{-cx_j} + d,$$

i.e. all rows would be perfectly identical. We also point out that free parameters to be determined here equate all non-superfluous elements in G . In practice N_t and N_x would often both be around 10-15, so even after accounting for the fact that the bottom-right corner of G is redundant, the total number of free parameters to be determined is potentially quite large. To avoid overfitting, additional regularity conditions must be imposed — an important point to which we return in Section 15.5.6.

15.5.3 Interpolation on the Whole Grid

Suppose that we have somehow managed to construct the matrix G in (15.39), i.e. we have uncovered $\|\lambda_k(t)\| = g(t, T_k - t)$ for the values of t and $x = T_k - t$ on the grid $\{t_i\} \times \{x_j\}$. The next step is to construct $\|\lambda_k(t)\|$ for *all* values of t and k , $k = 1, \dots, N - 1$.

It is common⁹ to assume that for each k , the function $\|\lambda_k(t)\|$ is piece-wise constant in t , with discontinuities at T_n , $n = 1, \dots, k - 1$,

$$\|\lambda_k(t)\| = \sum_{n=1}^k 1_{\{T_{n-1} \leq t < T_n\}} \|\lambda_{n,k}\| = \sum_{n=1}^k 1_{\{q(t)=n\}} \|\lambda_{n,k}\|. \quad (15.40)$$

In this case, we are left with constructing the matrix $\|\lambda_{n,k}\|$ from G , for all $1 \leq n \leq k \leq N - 1$. This is essentially a problem of two-dimensional interpolation

⁸An alternative would be to rely on extrapolation.

⁹A more refined approach, especially for low values of time-to-maturity, is advisable for some applications where the fine structure of short-term volatilities is important. See the brief discussion in Remark 16.1.1.

(and, perhaps, extrapolation if the $\{t_i\} \times \{x_j\}$ does not cover the whole tenor structure). Simple, robust schemes such as separate t - and x -interpolation of low order seem to perform well, whereas high-order interpolation (cubic or beyond) may lead to undesirable effects during risk calculations.

Hence, the main decision that need to be made is the choice of the order of interpolation, either 0 for piecewise constant or 1 for piecewise linear, for each of the dimensions t and x . Suppose, for concreteness, that linear interpolation in both dimensions is chosen. Then for each n, k ($1 \leq n \leq k \leq N-1$) we have the following scheme

$$\|\lambda_{n,k}\| = w_{++}G_{i,j} + w_{+-}G_{i,j-1} + w_{-+}G_{i-1,j} + w_{--}G_{i-1,j-1}, \quad (15.41)$$

where, denoting $\tau_{n,k} = T_k - T_{n-1}$, we have

$$\begin{aligned} i &= \min \{a : t_a \geq T_{n-1}\}, & j &= \min \{b : x_b \geq \tau_{n,k}\}, \\ w_{++} &= \frac{(T_{n-1} - t_{i-1})(\tau_{n,k} - x_{j-1})}{(t_i - t_{i-1})(x_j - x_{j-1})}, & w_{+-} &= \frac{(T_{n-1} - t_{i-1})(x_j - \tau_{n,k})}{(t_i - t_{i-1})(x_j - x_{j-1})}, \\ w_{-+} &= \frac{(t_i - T_{n-1})(\tau_{n,k} - x_{j-1})}{(t_i - t_{i-1})(x_j - x_{j-1})}, & w_{--} &= \frac{(t_i - T_{n-1})(x_j - \tau_{n,k})}{(t_i - t_{i-1})(x_j - x_{j-1})}. \end{aligned}$$

Apart from the order of interpolation, we can also choose which volatilities we actually want to interpolate. To explain, let us recall from Chapters 8 and 9 that we often normalize the local volatility function in such a way that $\phi(L_n(0)) \approx L_n(0)$. Then, $\|\lambda_k(\cdot)\|$'s have the dimensionality of log-normal, or percentage, volatilities, and (15.41) defines interpolation in *log-normal* Libor volatilities. This is not the only choice, and using volatilities that are scaled differently in the interpolation could sometimes lead to smoother results. To demonstrate the basic idea, let us fix p , $0 \leq p \leq 1$. Then we can replace (15.41) with

$$\begin{aligned} L_k(0)^{1-p} \|\lambda_{n,k}\| &= w_{++}L_{n(i,j)}(0)^{1-p}G_{i,j} + w_{+-}L_{n(i,j-1)}(0)^{1-p}G_{i,j-1} \\ &+ w_{-+}L_{n(i-1,j)}(0)^{1-p}G_{i-1,j} + w_{--}L_{n(i-1,j-1)}(0)^{1-p}G_{i-1,j-1}, \quad (15.42) \end{aligned}$$

where the indexing function $n(i, j)$ is defined by $T_{n(i,j)} = t_i + x_j$. For $p = 0$, this can be interpreted as interpolation in Gaussian¹⁰ volatilities (see Remark 8.2.9). For arbitrary p , the formula (15.42) specifies interpolation in ‘‘CEV’’ volatilities.

Finally, note that even if we use linear interpolation between the knot points (either in t or x or both), it is normally better to use *constant* extrapolation before the initial t_1 and x_1 and after the final t_{N_t} and x_{N_x} .

¹⁰When applied to interest rates, Gaussian volatilities are often called *basis-point*, or *bp*, volatilities.

15.5.4 Construction of $\lambda_k(t)$ from $\|\lambda_k(t)\|$

Suppose the values of volatility norm $\|\lambda_{n,k}\|$ are known on the full grid $1 \leq n \leq k \leq N-1$. In our method, for each T_n , the components of the m -dimensional $\lambda_k(T_n)$ vectors are obtained from instantaneous Libor rate volatilities $\|\lambda_{n,k}\|$ for $k \geq n$, and instantaneous correlations of Libor rates fixing on or after T_n . The procedure is similar in spirit to the one we employed previously for parameterizing multi-factor Gaussian short rate models in Section 13.1.7. So, with the calendar time fixed at some value T_n , we introduce an $(N-n) \times (N-n)$ instantaneous correlation matrix $R(T_n)$, with elements

$$(R(T_n))_{i,j} = \text{Corr}(dL_i(T_n-), dL_j(T_n-)), \quad i, j = n, \dots, N-1.$$

The correlation matrix would, in many applications, be computed from an estimated parametric form, such as those covered in Section 15.3.2. Furthermore, we define a diagonal volatility matrix $c(T_n)$ with elements $\|\lambda_{n,n}\|, \|\lambda_{n,n+1}\|, \dots, \|\lambda_{n,N-1}\|$ along its diagonal. That is,

$$(c(T_n))_{j,j} = \|\lambda_{n,n+j-1}\|, \quad j = 1, \dots, N-n,$$

with all other elements set to zero. Given $R(T_n)$ and $c(T_n)$, an instantaneous covariance matrix¹¹ $C(T_n)$ for forward rates on the grid can now be computed as

$$C(T_n) = c(T_n) R(T_n) c(T_n). \quad (15.43)$$

Let us define $H(T_n)$ to be an $(N-n) \times m$ matrix composed by stacking each dimension of $h(T_n, T_{n+j-1})$ (see 15.36) side by side, with j running on the grid:

$$(H(T_n))_{j,i} = h_i(T_n, T_{n+j-1}), \quad j = 1, \dots, N-n, \quad i = 1, \dots, m.$$

Then, it follows that we should have

$$C(T_n) = H(T_n) H(T_n)^\top. \quad (15.44)$$

Equations (15.43) and (15.44) specify two different representations of the covariance matrix, and we want them to be identical, i.e.

$$H(T_n) H(T_n)^\top = c(T_n) R(T_n) c(T_n), \quad (15.45)$$

which gives us a way to construct the $H(T_n)$ matrix, and thereby the vectors $h(T_n, T_{n+j-1})$ for all values of n, j on the full grid $1 \leq n \leq N-1, 1 \leq j \leq N-n$. Assuming, as before, piecewise constant interpolation of $\lambda_k(t)$ for t between knot dates $\{T_i\}$, the full set of factor volatilities $\lambda_k(t)$ can be constructed for all t and T_k .

¹¹Earlier results show that the true instantaneous covariance matrix for forward rates may involve DVF- or SV-type scales on the elements of c . For the purposes of calibration of λ_k , we omit these scales.

As written, equation (15.45) will normally *not* have a solution as the left-hand side is rank-deficient, whereas the right-hand side will typically have full rank. To get around this, we can proceed to apply PCA methodology, in several different ways. We discuss two methods below.

15.5.4.1 Covariance PCA

In this approach, we apply PCA decomposition to the entire right-hand side of (15.45), writing

$$c(T_n) R(T_n) c(T_n) \approx e_m(T_n) \Lambda_m(T_n) e_m(T_n)^\top,$$

where $\Lambda_m(T_n)$ is an $m \times m$ diagonal matrix of the m largest eigenvalues of $c(T_n) R(T_n) c(T_n)$, and $e_m(T_n)$ is an $(N - n) \times m$ matrix of eigenvectors corresponding to these eigenvalues. Inserting this result into (15.45) leads to

$$H(T_n) = e_m(T_n) \sqrt{\Lambda_m(T_n)}. \quad (15.46)$$

As discussed in Chapter 4, this approximation is optimal in the sense of minimizing the Frobenius norm on the covariance matrix errors.

15.5.4.2 Correlation PCA

An attractive alternative to the approach in Section 15.5.4.1 uses the correlation PCA decomposition discussed in Section 15.3.4. Here we write

$$R(T_n) = D(T_n) D(T_n)^\top, \quad (15.47)$$

for an $(N - n) \times m$ matrix D found by the techniques in Section 15.3.4. Inserting this into (15.45) yields

$$H(T_n) = c(T_n) D(T_n). \quad (15.48)$$

In computing the matrix D , we would normally use the result from Proposition 15.3.2, which would minimize the Frobenius norm on correlation matrix errors.

15.5.4.3 Discussion and Recommendation

Many papers in the literature focus on the method in Section 15.5.4.1 (e.g. Sidenius [2000], and Pedersen [1998]), but we nevertheless strongly prefer the approach in Section 15.5.4.2 for calibration applications. Although performing the PCA decomposition (as in Proposition 15.3.2) of a correlation matrix is technically more difficult than the same operation on a covariance matrix, the correlation PCA is independent of the c matrix and as such will not have to be updated when we update guesses for the G matrix (on which c depends) in a calibration search loop. When the correlation matrix $R(T_n)$ originates

from a parametric form independent of calendar time (which we recommend), the matrix D in (15.47) will, in fact, need estimation only once, at a minimal computational overhead cost. In comparison, the covariance PCA operation will have to be computed $(N - 1)$ times — once for each time T_n in the time grid — every time G gets updated. We also notice that the fact that $D(T_n)D(T_n)^\top$ has a unit diagonal will automatically ensure that the total forward rate volatility will be preserved if m is changed; this is *not* the case for covariance PCA, where the total volatility of forward rates will normally increase as m is increased, *ceteris paribus*.

If the complexity of the optimal PCA algorithm in Proposition 15.3.2 of Section 15.3.4 is deemed too egregious, the simplified approach of Section 15.3.4.2 could be used instead. It shares the performance advantages of the “true” correlation PCA as it only needs to be run once outside the calibration loop, but its theoretical deficiencies suggest that its use should, in most circumstances, be limited to the case where the correlations are themselves calibrated, rather than exogenously specified by the user. We return to the concept of correlation calibration in Section 15.5.9.

15.5.5 Choice of Calibration Instruments

In a standard LM model calibration, we choose a set of swaptions and caps (and perhaps Eurodollar options) with market-observable prices; these prices serve as calibration targets for our model. The problem of determining precisely which caps and swaptions should be included in the calibration is a difficult and contentious one, with several opposing schools of thought represented in the literature. We shall spend this section¹² outlining the major arguments offered in the literature as well as our own opinion on the subject. Before commencing on this, we emphasize that the calibration algorithm we develop in this book accommodates arbitrary sets of calibration instruments and as such will work with any selection philosophy.

One school of thought — the *fully calibrated approach* — advocates calibrating an LM model to a large set of available interest options, including both caps and swaptions in the calibration set. When using grid-based calibration, this camp would typically recommend using at-the-money swaptions with maturities and tenors chosen to coincide with each point in the grid. That is, if T_s is the maturity of a swaption and T_e is the end-date of its underlying swap, then we would let T_s take on all values in the time grid $\{t_i\}$, while at the same time letting $T_e - T_s$ progress through all values¹³ of the tenor grid $\{x_j\}$. On top of this, one would often add at-the-money caps at expiries ranging from $T = t_1$ to $T = t_{N_t}$.

¹²We also revisit the subject in the context of callable Libor exotics in Section 19.1.

¹³One would here limit T_s to be no larger than T_N , so the total number of swaptions would be less than $N_t \cdot N_x$. See our discussion of redundant grid entries in Section 15.5.2.

The primary advantage of the fully calibrated approach is that a large number of liquid volatility instruments are consistently priced within the model. This, in turn, gives us some confidence that the vanilla option market is appropriately “spanned” and that the calibrated model can be used on a diverse set of exotic securities. In vega-hedging an exotic derivative, one will undoubtedly turn to swaptions and caps, so mispricing these securities in the model would be highly problematic.

Another school of thought — *the parsimonious approach* — judiciously chooses a small subset of caps and swaptions in the market, and puts significant emphasis on specification of smooth and realistic term structures of forward rate volatilities. Typically this will involve imposing strong time-homogeneity assumptions, or observed statistical relationships, on the $\lambda_k(t)$ vectors. The driving philosophy behind the parsimonious approach is the observation that, fundamentally, the price of a security in a model is equal to the model-predicted cost of hedging the security over its lifetime. Hedging profits in the future as specified by the model are, in turn, directly related to the forward rate volatility structures that the model predicts for the future. For these model-predicted hedging profits to have any semblance to the actual realized hedging profits, the dynamics of the volatility structure in the model should be a reasonable estimate of the actual dynamics. In many cases, however, our best estimate of future volatility structures might be today’s volatility structures (or those we have estimated historically), suggesting that the evolution of volatility should be as close to being time-homogeneous as possible. This can be accomplished, for instance, by using time-homogeneous mappings such as (15.37) or similar.

The strong points of the parsimonious approach are, of course, weak ones of the fully calibrated approach. Forward rate volatilities produced by the fully calibrated model can easily exhibit excessively non-stationary behavior, impairing the performance of dynamic hedging. On the other hand, the inevitable mis-pricings of certain swaptions and/or caps in the parsimonious approach are troublesome. In a pragmatic view of a model as a (sophisticated, hopefully) interpolator that computes prices of complex instruments from prices of simple ones, mis-pricing of simple instruments obviously does not inspire confidence in the prices returned for complex instruments. As discussed, the parsimonious approach involves an attempt to identify a small enough set of “relevant” swaptions and caps that even a time-homogeneous model with a low number of free parameters can fit reasonably well, but it can often be very hard to judge which swaption and cap volatilities are important for a particular exotic security. In that sense, a fully calibrated model is more universally applicable, as the need to perform trade-specific identification of a calibration set is greatly reduced.

It is easy to imagine taking both approaches to the extremes to generate results that would convincingly demonstrate the perils of using either of them. To avoid such pitfalls we recommend looking for an equilibrium between the two. While we favor the fully calibrated approach, it is clear to us that, at the

very least, it should be supplemented by an explicit mechanism to balance price precision versus regularity (e.g. smoothness and time-homogeneity) of the forward rate volatility functions. In addition, one should always perform rigorous checks of the effects of calibration assumptions on pricing and hedging results produced by the model. These checks should cover, as a minimum, result variations due to changes in

- Number of factors used (m).
- Relative importance of recovering all cap/swaption prices vs. time-homogeneity of the resulting volatility structure.
- Correlation assumptions.

A final question deserves a brief mention: should one calibrate to either swaptions or caps, or should one calibrate to both simultaneously? Followers of the parsimonious approach will typically argue that there is a persistent basis between cap and swaption markets, and any attempt to calibrate to both markets simultaneously is bound to distort the model dynamics. Instead, it is argued, one should only calibrate to one of the two markets, based on an analysis of whether the security to be priced is more cap- or swaption-like. Presumably this analysis would involve judging whether either caps or swaptions will provide better vega hedges for the security in question. The drawback of this approach is obvious: many complicated interest rates securities depend on the evolution of both Libor rates as well as swap rates and will simultaneously embed “cap-like” and “swaption-like” features.

To avoid discarding potentially valuable information from either swaptions or cap markets, we generally recommend that both markets be considered in the calibration of the LM model. However, we do not necessarily advocate that both types of instruments receive equal weighting in the calibration objective function; rather, the user should be allowed some mechanism to affect the relative importance of the two markets. We return to this idea in the next section.

15.5.6 Calibration Objective Function

As discussed above, several issues should be considered in the choice of a calibration norm, including the smoothness and time-stationarity of the $\lambda_k(t)$ functions; the precision to which the model can replicate the chosen set of calibration instruments; and the relative weighting of caps and swaptions. To formally state a calibration norm that will properly encompass these requirements, assume that we have chosen calibration targets that include N_S swaptions, $V_{\text{swpt},1}, V_{\text{swpt},2}, \dots, V_{\text{swpt},N_S}$, and N_C caps, $V_{\text{cap},1}, V_{\text{cap},2}, \dots, V_{\text{cap},N_C}$. Strategies for selecting these instruments were discussed in the previous section. We let \hat{V} denote their quoted market prices and, adopting the grid-based framework from Section 15.5.2, we let $\bar{V}(G)$ denote their model-generated prices as functions of the volatility grid G as defined as in Section 15.5.2. We introduce a calibration objective function \mathcal{I} as

$$\begin{aligned}
\mathcal{I}(G) = & \frac{w_S}{N_S} \sum_{i=1}^{N_S} \left(\bar{V}_{\text{swpt},i}(G) - \hat{V}_{\text{swpt},i} \right)^2 + \frac{w_C}{N_C} \sum_{i=1}^{N_C} \left(\bar{V}_{\text{cap},i}(G) - \hat{V}_{\text{cap},i} \right)^2 \\
& + \frac{w_{\partial t}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_x} \left(\frac{\partial G_{i,j}}{\partial t_i} \right)^2 + \frac{w_{\partial x}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_x} \left(\frac{\partial G_{i,j}}{\partial x_j} \right)^2 \\
& + \frac{w_{\partial t^2}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_x} \left(\frac{\partial^2 G_{i,j}}{\partial t_i^2} \right)^2 + \frac{w_{\partial x^2}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_x} \left(\frac{\partial^2 G_{i,j}}{\partial x_j^2} \right)^2, \quad (15.49)
\end{aligned}$$

where $w_S, w_C, w_{\partial t}, w_{\partial x}, w_{\partial t^2}, w_{\partial x^2} \in \mathbb{R}_+$ are exogenously specified weights. In (15.49) the various derivatives of the elements in the table G are, in practice, to be interpreted as discrete difference coefficients on neighboring table elements — see (15.50) below for an example definition¹⁴.

As we have defined it, $\mathcal{I}(G)$ is a weighted sum of i) the mean-squared swaption price error; ii) the mean-squared cap price error; iii) the mean-squared average of the derivatives of G with respect to calendar time; iv) the mean-squared average of the second derivatives of G with respect to calendar time; v) the mean-squared average of the derivatives of G with respect to forward rate tenor; and vi) the mean-squared average of the second derivatives of G with respect to forward rate tenor. The terms in i) and ii) obviously measure how well the model is capable of reproducing the supplied market prices, whereas the remaining four terms are all related to regularity. The term iii) measures the degree of volatility term structure time-homogeneity and penalizes volatility functions that vary too much over calendar time. The term iv) measures the smoothness of the calendar time evolution of volatilities and penalizes deviations from linear evolution (a straight line being perfectly smooth). Terms v) and vi) are similar to iii) and iv) and measure constancy and smoothness in the tenor direction. In (15.49), the six weights $w_S, w_C, w_{\partial t}, w_{\partial x}, w_{\partial t^2}, w_{\partial x^2}$ determine the trade-off between volatility smoothness and price accuracy, and are normally to be supplied by the user based on his or her preferences. In typical applications, the most important regularity terms are those scaled by the weights $w_{\partial t}$ and $w_{\partial x^2}$ which together determine the degree of time-homogeneity and tenor-smoothness in the resulting model.

We should note that there are multiple ways to specify smoothness criteria, with (15.49) being one of many. For example, as we generalized the basic log-normal interpolation scheme (15.41) to allow for interpolation in “CEV” volatilities in (15.42), we can adjust the definition of smoothness to be in terms of compatible quantities. In particular, instead of using

$$\frac{w_{\partial x}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=2}^{N_x} \left(\frac{G_{i,j} - G_{i,j-1}}{x_j - x_{j-1}} \right)^2 \quad (15.50)$$

as implicit in (15.49) for the tenor-smoothness term, we could use

¹⁴Depending on how table boundary elements are treated, notice that the range for i and j may not always be as stated in (15.49).

$$\frac{w_{\partial x}}{N_x N_t} \sum_{i=1}^{N_t} \sum_{j=2}^{N_x} \left(\frac{L_{n(i,j)}(0)^{1-p} G_{i,j} - L_{n(i,j-1)}(0)^{1-p} G_{i,j-1}}{x_j - x_{j-1}} \right)^2 \quad (15.51)$$

for some p , $0 \leq p \leq 1$. The case of $p = 0$ would then correspond to smoothing basis-point Libor volatilities rather than log-normal Libor volatilities.

As written, the terms of the calibration norm that measure precision in cap and swaption pricing involve mean-squared errors directly on prices. In practice, however, the error function is often applied to some transform of out-right prices, e.g. implied volatilities. For an SV-type LM model, for instance, we could institute a pre-processing step where the market price of each swaption $\widehat{V}_{\text{swpt},i}$ would be converted into a constant implied “skew-volatility” $\widehat{\lambda}_{S_i}$, in such a way that the scalar SDE for the swap rate S_i underlying the swaption $V_{\text{swpt},i}$

$$dS_i(t) = \sqrt{z(t)} \widehat{\lambda}_{S_i} \varphi(S_i(t)) dY_i(t),$$

would reproduce the observed swaption market price. Denoting by $\bar{\lambda}_{S_i}(G)$ the corresponding model volatility of the swap rate S_i (as given by, for example, Proposition 15.4.4) and repeating this exercise for all caps and swaptions in the calibration set, we obtain an alternative calibration norm definition where the cap and swaption terms in (15.49) are modified as follows:

$$\mathcal{I}(G) = \frac{w_S}{N_S} \sum_{i=1}^{N_S} \left(\bar{\lambda}_{S_i}(G) - \widehat{\lambda}_{S_i} \right)^2 + \frac{w_C}{N_C} \sum_{i=1}^{N_C} \left(\bar{\lambda}_{C_i}(G) - \widehat{\lambda}_{C_i} \right)^2 + \dots \quad (15.52)$$

The advantage of working with implied volatilities in the precision norm is two-fold. First, the relative scaling of individual swaptions and caps is more natural; when working directly with prices, high-value (long-dated) trades would tend to be overweighted relative to low-value (short-dated) trades¹⁵. Second, in many models computation of the implied volatility terms $\bar{\lambda}_{S_i}$ and $\bar{\lambda}_{C_i}$ can often be done by simple time-integration of (combinations of) $\lambda_k(t)$ (see e.g. Proposition 15.4.4) avoiding the need to apply a possibly time-consuming option pricing formula to compute the prices $\bar{V}_{\text{swpt},i}$ and $\bar{V}_{\text{cap},i}$. Considerable attention to this particular issue was paid in Section 10.3 (for SV models) and Section 8.6.2 (for DVF models), and we review relevant results and apply them to LM models in Chapter 16.

The quality-of-fit objective can be expressed in terms of *scaled* volatilities, which improves performance sometimes. Following the ideas developed for interpolation (15.42) and smoothing (15.51), we could express the fit objective as

$$\mathcal{I}(G) = \frac{w_S}{N_S} \sum_{i=1}^{N_S} \left(S_i(0)^{1-p} \left(\bar{\lambda}_{S_i}(G) - \widehat{\lambda}_{S_i} \right) \right)^2 + \dots,$$

¹⁵ Another approach to producing more equitable scaling involves using relative (=percentage) price errors, rather than absolute price errors.

for a given p , $0 \leq p \leq 1$. Taking this idea further we note that a more refined structure of mean-squared weights in the definition of calibration norm is possible. For instance, rather than weighting all swaptions equally with the term w_S , one could use different weights for each swaption in the calibration set. Similarly, by using node-specific weights on the derivatives of the entries in G one may, say, express the view that time-homogeneity is more important for large t than for small t .

15.5.7 Sample Calibration Algorithm

At this point, we are ready to state our full grid-based calibration algorithm. We assume that a tenor structure and a time/tenor grid $\{t_i\} \times \{x_j\}$ have been selected, as have the number of Brownian motions (m), a correlation matrix R , and the set of calibration swaptions and caps. In addition, the user must select the weights in the calibration norm \mathcal{I} in (15.49) or (15.52). Starting from some guess for G , we then run the following iterative algorithm:

1. Given G , interpolate using (15.41) or (15.42) to obtain the full norm volatility grid $\|\lambda_{n,k}\|$ for all Libor indices $k = 1, \dots, N - 1$ and all expiry indices $n = 1, \dots, k$.
2. For each $n = 1, \dots, N - 1$, compute the matrix $H(T_n)$, and ultimately volatility loadings $\lambda_k(T_n)$, from $\|\lambda_{n,k}\|$, $k \geq n$, by PCA methodology, using either (15.46) or (15.48).
3. Given $\lambda_k(\cdot)$ for all $k = 1, \dots, N - 1$, use the formulas in Sections 15.4.1 and 15.4.2 to compute model prices for all swaptions and caps in the calibration set.
4. Establish the value of $\mathcal{I}(G)$ by direct computation of either (15.49) or (15.52).
5. Update G and repeat Steps 1–4 until $\mathcal{I}(G)$ is minimized.

Step 5 in the above algorithm calls for the use of a robust high-dimensional numerical optimizer. Good results can, in our experience, be achieved with several algorithms, including the Spellucci algorithm¹⁶, the Levenberg-Marquardt algorithm, and the downhill simplex method (the last two can be found in Press et al. [1992]). These, and many alternative algorithms, are available in standard numerical packages, such as IMSL¹⁷ and NAG¹⁸. On a standard PC, a well-implemented calibration algorithm should generally complete in well under a minute from a cold start (i.e. where we do not have a good initial guess for G) for, say, a 40 year model with quarterly Libor rolls.

¹⁶donlp2 SQP/ECQP algorithm, available on http://www.mathematik.tu-darmstadt.de:8080/ags/ag8/Mitglieder/spellucci_de.html.

¹⁷<http://www.imsl.com/>.

¹⁸<http://www.nag.com/>.

15.5.8 Speed-Up Through Sub-Problem Splitting

An LM model calibration problem involves a substantial number of free input variables to optimize over, namely all elements of the matrix G . In a typical setup, the number of such variables may range from a few dozen to a few hundred. As the number of terms, or “targets”, in the calibration norm is of the same order of magnitude, we are dealing with a fairly sizable optimization problem. While modern optimization algorithms implemented on modern hardware can successfully handle the full-blown problem, it is still of interest to examine whether there are ways of to improve computational efficiency. For instance, if we could split the optimization problem into a sequence of smaller sub-problems solved separately and sequentially, the performance of the algorithm would typically improve. Indeed, imagine for illustrative purposes that we have an optimization problem with $m = m_1 m_2$ variables and computational complexity of the order¹⁹ $O(m^2) = O(m_1^2 m_2^2)$. However, if we could find the solution by sequentially solving m_1 problems of m_2 variables each, then the computational cost would be $m_1 O(m_2^2)$, yielding savings of the order $O(m_1)$.

Our ability to split the problem into sub-problems typically relies on exploring its particular structure, i.e. the relationship between input variables and targets. If, for example, target 1 depends on variable 1 but not — or only mildly — on other variables, then it makes sense to find the optimal value for the variable 1 by optimizing for target 1 while keeping other variables constant, and so on. Fortunately, the LM model optimization problem presents good opportunities for this type of analysis. First, recall that the main calibration targets for the problem are the differences in market and model prices (or implied volatilities) of caps and swaptions. Let us consider a swaption with expiry T_j and maturity T_n ; let i be such that $T_j = t_i$. Then, as follows from the swaption approximation formula (15.31), the model volatility for this swaption depends on $\lambda_k(t)$'s for $t \in [0, t_i]$ and for $k = j, \dots, n-1$. Hence, the part of the calibration norm associated with the price fit of the swaption will depend on the first i rows of the matrix G *only*. This observation suggests splitting the calibration problem into a collection of “row by row” calibration problems.

To simplify notations, we assume that the set of fit targets consists of *all* swaptions with expiries t_i and tenors x_l , $i = 1, \dots, N_t$, $l = 1, \dots, N_x$ (a “full” calibration set). In a row-by-row calibration algorithm, the first row of the matrix G is calibrated to all N_x swaptions with expiry t_1 , then the second row of G is calibrated to the swaptions with expiry t_2 , and so on.

As we emphasized earlier, having regularity terms in the calibration norm is important to ensure a smooth solution. Fortunately, regularity terms can

¹⁹As many of the algorithms we have in mind compute, at the very least, the sensitivity of each calibration target to each input variable, the computational complexity is at least $O(m^2)$; if the order of complexity is higher, the case for problem splitting is even more compelling.

generally be organized in the same row-by-row format as the precision terms. For instance, the regularity terms in the tenor direction naturally group into row-specific collections. As for the terms controlling the regularity of the matrix G in calendar time t , when optimizing on time slice t_i , we would only include in the norm the terms that involve rows of G with index less than or equal to i . We trust that the reader can see how to arrange this, and omit straightforward details.

Computational savings from the row by row scheme could be substantial — for a 40 year model with quarterly Libor rolls, a well-tuned algorithm should converge in less than a second or two. There are, however, certain drawbacks associated with problem splitting. In particular, as the calibration proceeds from one row to the next, the optimizer does not have the flexibility to adjust previous rows of the matrix G to the current row of swaption volatilities. This may result in a tell-tale “ringing” pattern of the Libor volatilities in the time direction, as the optimizer attempts to match each row of price targets through excessively large moves in the elements of G , in alternating up and down directions. Judicious application of regularity terms in the optimization norm can, however, help control this behavior, and overall the row-by-row scheme performs well. We recommend it as the default method for most applications, but note that sometimes a combination of full-blown and row-by-row calibration is the best choice. For instance, one could use full-blown optimization to fundamentally calibrate G , and use some version of row-by-row optimization when examining the effect of making small perturbations to input prices (e.g. when computing vegas). We discuss this idea in Chapter ??.

Returning to the row-by-row calibration idea, one can try to take it further and split the calibration to an ever-finer level, eventually fitting each individual price target — a given caplet or swaption volatility, say — separately, by moving just a *single* element of the matrix G . This should seemingly work because the (t_i, x_{l+1}) -swaption volatility depends on the same elements of matrix G as the (t_i, x_l) -swaption volatility *plus* $G_{i,l+1}$. (This is not entirely true due to some grid interpolation effects, but the general idea is correct). So, in principle, $G_{i,l+1}$ can be found by just solving a quadratic equation, i.e. in closed form. For full details we refer the reader to [Brigo and Mercurio \[2001\]](#) where this *bootstrap*, or *cascade* algorithm is described in detail. While this may appear to be a strong contender for practical LM calibration — full calibration is performed by just solving $N_t N_x$ quadratic equations — the scheme generally does not work for practically-sized problems. The cascade calibration suffers strongly from the ringing problem discussed above, and the quadratic equations typically fail to have a solution for swaption targets with just a few years of total maturity. While it is possible to include regularity terms that preserve the closed-form nature of the solution, the ringing problem is difficult to remedy and calibration to long-dated options is rarely feasible. We find this to be true, even if one applies ad-hoc remediation methods proposed by various authors (see e.g. [Brigo and Morini \[2006\]](#)).

15.5.9 Correlation Calibration to Spread Options

In the calibration algorithm in Section 15.5.7 the matrix R is specified exogenously and would typically originate from an empirical analysis similar to that in Section 15.3.2. As we discussed earlier in Section 15.4.3, an alternative approach attempts to imply R directly from market data for spread options. Less is known about the robustness of calibrations based on this approach, but this shall not stop us from listing a possible algorithm.

First, to make the problem tractable, we assume that the matrix R is time-homogeneous and specified as some parametric function of a low-dimension parameter-vector ξ ,

$$R = R(\xi).$$

Possible parameterizations include those listed in Section 15.3.2. We treat ξ as an unknown vector, to be determined in the calibration procedure along with the elements of the volatility matrix G . For this, we introduce a set of market-observable spread option prices $\widehat{V}_{\text{spread},1}, \widehat{V}_{\text{spread},2}, \dots, \widehat{V}_{\text{spread},N_{SP}}$, their corresponding model-based prices $\overline{V}_{\text{spread},1}(G, \xi), \overline{V}_{\text{spread},2}(G, \xi), \dots, \overline{V}_{\text{spread},N_{SP}}(G, \xi)$, and update the norm \mathcal{I} in (15.49) (or (15.52)) to the norm $\mathcal{I}^*(G, \xi)$, where²⁰

$$\mathcal{I}^*(G, \xi) = \mathcal{I}(G, \xi) + \frac{w_{SP}}{N_{SP}} \sum_{i=1}^{N_{SP}} \left(\overline{V}_{\text{spread},i}(G, \xi) - \widehat{V}_{\text{spread},i} \right)^2. \quad (15.53)$$

The algorithm in Section 15.5.7 proceeds as before with a few obvious changes; we list the full algorithm here for completeness.

1. Given G , interpolate using (15.41) or (15.42) to obtain the full norm volatility grid $\|\lambda_{n,k}\|$ for all Libor indices $k = 1, \dots, N - 1$ and all expiry indices $n = 1, \dots, k$.
2. Given ξ , compute $R = R(\xi)$.
3. For each $n = 1, \dots, N - 1$ and using $R(\xi)$, compute the matrix H , and ultimately volatility loadings $\lambda_k(T_n)$, from $\|\lambda_{n,k}\|$, $k \geq n$, by PCA methodology, using either (15.46) or (15.48).
4. Given $\lambda_k(\cdot)$ for all $k = 1, \dots, N - 1$, use the formulas in Sections 15.4.1, 15.4.2 and 15.4.3 to compute model prices for all swaptions, caps and spread options in the calibration set.
5. Establish the value of $\mathcal{I}^*(G, \xi)$ by direct computation of (15.53).
6. Update G and repeat Steps 1–5 until $\mathcal{I}(G, \xi)$ is minimized.

When using a correlation PCA algorithm in Step 3, in practice one may find that it is most efficient to use the “poor man’s” approach in Section 15.3.4.2, rather than the slower expression listed in Proposition 15.3.2. Indeed, as long as the spread option prices ultimately are well-matched, we can be

²⁰Note that our original norm \mathcal{I} now also is a function of ξ , since cap and swaption prices depend on the correlation matrix R .

confident that our model has a reasonable correlation structure, irrespective of which PCA technique was used.

As was the case for our basic algorithm, let us note that it may be useful to transform spread option prices into implied volatilities or, even better, into implied term correlations²¹ when evaluating the mean-squared error. For spread options, a definition of implied term correlation can be extracted from the simple Gaussian spread approach in Section 15.4.3, equations (15.34) and (15.35) or, for more accurate formulas, using the results of Chapter 18 and in particular Sections 18.4.2 and 18.9.1.

Finally, we should note that the optimization problem embedded in the algorithm above can be quite challenging to solve in practice. To stabilize the numerical solution, it may be beneficial to employ a split calibration approach, where we first freeze correlation parameters ξ and then optimize G over the parts of the calibration norm that do not involve spread options. Then we freeze G at its optimum and optimize ξ over the parts of the calibration norm that do not involve caps and swaptions. This alternating volatility- and correlation-calibration is then repeated iteratively until (hopefully) convergence. A similar idea can be employed when calibrating models to a volatility smile; see Section 16.2.3 for LM model applications and Section 17.2.3 for applications to vanilla models.

15.5.10 Volatility Skew Calibration

The calibration algorithm we have discussed so far will normally take at-the-money options as calibration targets when establishing the $\lambda_k(t)$ functions. Establishing the volatility smile away from at-the-money strikes must be done in a separate step, through specification of a DVF skew function φ and, possibly, a stochastic volatility process $z(t)$. For the time-stationary specifications of these two mechanisms that we considered in Section 15.2.5, best-fitting to the volatility skew can be done relatively easily — in fact, it is probably best to leave the parameters²² of the skew function φ as a free parameter for trader's input. We study the problem of volatility smile calibration for LM models in more detail in Chapter 16.

²¹By representing spread options through implied term correlations, the information extracted from spread options in the correlation is more “orthogonal” to that extracted from caps and swaptions, something that can help improve the numerical properties of the calibration algorithm, particularly if split calibration approach, an approach we shall consider in a moment, is used.

²²Assuming that there are only a few parameters that define the shape of the function. We strongly recommend using simple skew functions that could be described by a single-parameter family, such as linear or power functions.

15.6 Monte Carlo Simulation

Once the LM model has been calibrated to market data, we can proceed to use the parameterized model for the pricing and risk management of non-vanilla options. In virtually all cases, pricing of such options will involve numerical methods. As the LM model involves a very large number of Markov state variables — namely the full number of Libor forward rates on the yield curve plus any additional variables used to model stochastic volatility — finite difference methods are rarely applicable (but see the brief discussion in Section 16.3 for a special case), and we nearly always have to rely on Monte Carlo methods. As we discussed in Chapter 4, the main idea of Monte Carlo pricing is straightforward: i) simulate independent paths of the collection of Libor rates through time; ii) for each path, sum the numeraire-deflated values of all cash flows generated by the specific interest rate dependent security at hand; iii) repeat i)-ii) many times and form the average. Proper execution of step i) is obviously key to this algorithm, and begs an answer to the following question: given a probability measure and the state of the Libor forward curve at time t , how do we move the entire Libor curve (and the numeraire) forward to time $t + \Delta$, $\Delta > 0$, in a manner that is consistent with the LM model dynamics? We address this question here.

15.6.1 Euler-Type Schemes

Assume that we stand at time t , and have knowledge of forward Libor rates maturing at all dates in the tenor structure after time t . We wish to devise a scheme to advance time to $t + \Delta$ and construct a sample of $L_{q(t+\Delta)}(t + \Delta), \dots, L_{N-1}(t + \Delta)$. Notice that $q(t + \Delta)$ may or may not exceed $q(t)$; if it does, some of the front-end forward rates expire and “drop off” the curve as we move to $t + \Delta$.

For concreteness, assume for now that we work in the spot measure Q^B in which case Lemma 15.2.3 tells us that general LM model dynamics are of the form

$$dL_n(t) = \sigma_n(t)^\top (\mu_n(t) dt + dW^B(t)), \quad \mu_n(t) = \sum_{j=q(t)}^n \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)}, \quad (15.54)$$

where the $\sigma_n(t)$ are adapted vector-valued volatility functions and W^B is an m -dimensional Brownian motion in measure Q^B . The simplest way of drawing an approximate sample $\widehat{L}_n(t + \Delta)$ for $L_n(t + \Delta)$ would be to apply a first-order Euler-type scheme. Drawing on results in Section 4.2.3, Euler (15.55) and log-Euler (15.56) schemes for (15.54) are, for $n = q(t + \Delta), \dots, N - 1$,

$$\widehat{L}_n(t + \Delta) = \widehat{L}_n(t) + \sigma_n(t)^\top \left(\mu_n(t)\Delta + \sqrt{\Delta}Z \right), \quad (15.55)$$

$$\widehat{L}_n(t + \Delta) = \widehat{L}_n(t) \exp \left\{ \frac{\sigma_n(t)^\top}{\widehat{L}_n(t)} \left(\left(\mu_n(t) - \frac{1}{2} \frac{\sigma_n(t)}{\widehat{L}_n(t)} \right) \Delta + \sqrt{\Delta}Z \right) \right\}, \quad (15.56)$$

where Z is a vector of m independent $\mathcal{N}(0, 1)$ Gaussian draws²³. For specifications of $\sigma_n(t)^\top$ that are close to proportional in $L_n(t)$ (e.g. the log-normal LM model), we would expect the log-Euler scheme (15.56) to produce lower biases than the Euler scheme (15.55). As discussed in Chapter 4, the log-Euler scheme will keep forward rates positive, whereas the Euler scheme will not.

Both schemes (15.55), (15.56) as shown advance time only by a single time-step, but creation of a full path of forward curve evolution through time is merely a matter of repeated application²⁴ of the single-period stepping schemes on a (possibly non-equidistant) time line t_0, t_1, \dots . When working in the spot measure, it is preferable to have the tenor structure dates T_1, T_2, \dots, T_{N-1} among the simulation dates, in order to keep track of the spot numeraire $B(t)$ without having to resort to extrapolations. In fact, it is common in practice to set $t_i = T_i$, which, unless accrual periods τ_i are unusually long or volatilities unusually high, will normally produce an acceptable discretization error for many types of LM models. See e.g. Andersen and Andreasen [2000b] and Glasserman and Zhao [2000] for some numerical investigations of the Euler bias.

Remark 15.6.1. When t coincides with a date in the tenor structure, $t = T_k$, say, $q(t)$ will equal T_{k+1} due to our definition of q being right-continuous. As a result, when stepping forward from time $t = T_k$, $\widehat{L}_k(T_k)$ will *not* be included in the computation of the drifts μ_n , $n \geq k+1$. As it turns out, this convention reduces discretization bias, a result that makes sense when we consider that the contribution from $L_k(t)$ to the drifts drops to zero at time $T_k + dt$ in a continuous-time setting.

While Euler-type schemes such as (15.55) and (15.56) are not very sophisticated and, as we recall from Chapter 4, result in rather slow convergence of the discretization bias ($O(\Delta)$), these schemes are appealing in their straightforwardness and universal applicability. Further, they serve to highlight the basic structure of an LM simulation and the computational effort in advancing the forward curve.

²³In addition to these time-stepping schemes for the forward rates, it may be necessary to simultaneously evolve stochastic volatility variables if one works with models such as those in Section 15.2.5.

²⁴A different Z is needed at each time-step, needless to say.

15.6.1.1 Analysis of Computational Effort

Focusing on the straight Euler scheme (15.55), a bit of contemplation reveals that the computational effort involved in advancing L_n is dominated by the computation of $\mu_n(t)$ which, in a direct implementation of (15.54), involves

$$m \cdot (n - q(t) + 1) = O(mn)$$

operations for a given value of n . To advance all $N - q(t + \Delta)$ forwards, it follows that the computational effort is $O(mN^2)$ for a single time-step. Assuming that our simulation time line coincides with the tenor structure dates, generation of a full path of forward curve scenarios from time 0 to time T_{N-1} will thus require a total computational effort of $O(mN^3)$. As N is often big (e.g. a 25-year curve of quarterly forward rates will have $N = 100$), a naive application of the Euler scheme will often require considerable computing resources.

As should be rather obvious, however, the computational order of $O(mN^3)$ is easy to improve on, as there is no need to spend $O(mN)$ operations on the computation of each μ_n . Instead, we can invoke the recursive relationship

$$\mu_n(t) = \mu_{n-1}(t) + \frac{\tau_n \sigma_n(t)}{1 + \tau_n \widehat{L}_n(t)}, \quad (15.57)$$

which allows us to compute all μ_n , $n = q(t + \Delta), \dots, N - 1$, by an $O(mN)$ -step iteration starting from

$$\mu_{q(t+\Delta)}(t) = \sum_{j=q(t)}^{q(t+\Delta)} \frac{\tau_j \sigma_j(t)}{1 + \tau_j \widehat{L}_j(t)}.$$

In total, the computational effort of advancing the full curve one time-step will be $O(mN)$, and the cost of taking N such time steps will be $O(mN^2)$ — and not $O(mN^3)$.

We summarize this result in a lemma.

Lemma 15.6.2. *Assume that we wish to simulate the entire Libor forward curve on a time line that contains the dates in the tenor structure and has $O(N)$ points. The computational effort of Euler-type schemes — such as (15.55) and (15.56) — is $O(mN^2)$.*

Remark 15.6.3. The results of the lemma can be verified to hold for any of probability measures we examined in Section 15.2.2.

We note that when simulating in other measures, the starting point of the iteration for μ_n will be measure-dependent. For instance, in the terminal measure,

$$\mu_n(t) = - \sum_{j=n+1}^{N-1} \frac{\tau_j \sigma_j(t)}{1 + \tau_j \widehat{L}_j(t)}$$

and the equation (15.57) still holds. Now, however, the iteration starts at

$$\mu_{N-1}(t) = 0,$$

and proceeds *backwards* through $\mu_{N-2}, \mu_{N-3}, \dots, \mu_{q(t+\Delta)}$. We leave it to the reader to carry out the analysis for other probability measures.

15.6.1.2 Long Time-Steps

Most exotic interest rate derivatives involve revolving cash flows paid on a tightly spaced schedule (e.g. quarterly). As our simulation time line should always include dates on which cash flows take place, the average time spacing used in path generation will thus normally, by necessity, be quite small. In certain cases, however, there may be large gaps between cash flow dates, e.g. when a security is forward-starting or has an initial lock-out period. When simulating across large gaps, we may always choose to sub-divide the gap into smaller time-steps, thereby retaining a tightly spaced simulation time line. To save computational time, however, it is often tempting to cover large gaps in a small number of coarse time-steps, in order to lower overall computation effort. Whether such coarse stepping is possible is, in large part, a question of how well we can keep the discretization bias under control as we increase the time-step, something that is quite dependent on the magnitude of volatility and the particular formulation of the LM model under consideration. Section 15.6.2 below deals with this question and offers strategies to improve on the basic Euler scheme. Here, we instead consider the pure mechanics of taking large time-steps, i.e. steps that skip past several dates in the tenor structure.

Assume that we stand at the j -th date in the tenor structure, $t = T_j$, and wish to simulate the forward curve to time T_k , $k > j + 1$, in a single step. As noted earlier, the mere notion of skipping over dates in the tenor structure makes usage of the spot measure Q^B inconvenient, as the numeraire $B(T_k)$ cannot be constructed without knowledge of the realizations of $L_{j+1}(T_{j+1}), L_{j+2}(T_{j+2}), \dots, L_{k-1}(T_{k-1})$; in turn, numeraire-deflation of cash flows is not possible and derivatives cannot be priced. Circumventing this issue, however, is merely a matter of changing the numeraire from B to an asset that involves no roll-over in the interval $[T_j, T_k]$. One such asset is $P(t, T_N)$, the choice of which corresponds to running our simulated paths in the terminal measure. In particular, we recognize that

$$P(T_k, T_N) = \prod_{n=k}^{N-1} \frac{1}{1 + \tau_n L_n(T_k)}, \quad (15.58)$$

which depends only on the state of the forward curve at time T_k . Another valid numeraire asset would be $\bar{P}_j(t)$, as defined in Section 15.2.2:

$$\bar{P}_j(t) = \begin{cases} B(t), & t \leq T_j, \\ B(T_j)P(t, T_N)/P(T_j, T_N), & t > T_j. \end{cases}$$

The numeraire $\bar{P}_j(T_k)$ can always be computed without knowledge of $L_{j+1}(T_{j+1}), \dots, L_{k-1}(T_{k-1})$, as long as $B(T_j)$ is known²⁵. In the measure induced by this asset, the LM model dynamics are

$$dL_n(t) = \begin{cases} \sigma_n(t)^\top \left(-\sum_{l=n+1}^{N-1} \frac{\tau_l \sigma_l(t)}{1+\tau_l L_l(t)} dt + d\bar{W}^j(t) \right), & t > T_j, \\ \sigma_n(t)^\top \left(\sum_{l=q(t)}^n \frac{\tau_l \sigma_l(t)}{1+\tau_l L_l(t)} dt + d\bar{W}^j(t) \right), & t \leq T_j. \end{cases}$$

15.6.1.3 Notes on the Choice of Numeraire

Given our discussion above, the terminal measure may strike the reader as an obvious first choice for simulating the LM model — after all, simulations in the terminal measure will never fail to be meaningful, irrespective of the coarseness of the simulation time line. Other issues, however, come in play here as well. For instance, updating the numeraire $P(t, T_N)$ from one time-step to the next is generally a more elaborate operation than updating the spot numeraire $B(t)$: the former requires multiplying together $O(N)$ terms (see (15.58)), whereas the latter only involves multiplying B at the previous time-step with a single discount bond price. Also, the statistical sample properties of price estimators in the terminal measure may be inferior to those in the spot measure, in the sense that the Monte Carlo noise is larger in the terminal measure. Glasserman and Zhao [2000] list empirical results indicating that this is, indeed, often the case for many common interest rate derivatives. A formal analysis of this observation is complex, but we can justify it by considering the pricing of a very simple derivative security, namely a discount bond maturing at some arbitrary time T_k in the tenor structure. In the spot measure, we would estimate the price of this security by forming the sample average of random variables

$$P(T_k, T_k)/B(T_k) = B(T_k)^{-1} = \frac{1}{\prod_{n=0}^{k-1} (1 + \tau_n L_n(T_n))}, \quad (15.59)$$

whereas in the terminal measure we would form the sample average of random variables

$$P(T_k, T_k)/P(T_k, T_N) = P(T_k, T_N)^{-1} = \prod_{n=k}^{N-1} (1 + \tau_n L_n(T_k)). \quad (15.60)$$

Assuming that Libor rates stay positive, the important thing to notice is that the right-hand side of (15.59) is bounded from above by 1, whereas the right-hand side of (15.60) can grow arbitrarily large. For moderate to high Libor

²⁵This precludes the existence of other large gaps in the simulation time line prior to time T_j . When using a hybrid measure such as \bar{P}_j , we would need to position T_j at the start of the first simulation time step that spans multiple dates in the tenor structure.

rate volatilities, we would thus intuitively expect price estimators based on (15.60) to typically have higher sample error.

As discussed in Section 15.6.1.2, sometimes it is mechanically inconvenient to simulate in the spot measure, due to a desire to take large time-steps. In these cases, usage of a hybrid numeraire \bar{P} that switches from $B(t)$ to $P(t, T_N)$ at the latest possible date may be a useful strategy.

15.6.2 Other Simulation Schemes

When simulating on a reasonably tight time schedule, the accuracy of the Euler or log-Euler schemes is adequate for most applications. However, as discussed above, we may occasionally be interested in using coarse time-steps in some parts of the path generation algorithm, requiring us to pay more attention to the discretization scheme. Generic techniques for these purposes were introduced in detail in Chapter 4; we proceed to discuss a few of these in the context of LM models. We also consider the case where special-purpose schemes happen to exist for the discretization of the stochastic integral in the forward rate dynamics.

15.6.2.1 Special-Purpose Schemes with Drift Predictor-Corrector

In integrated form, the general LM dynamics in (15.54) become

$$\begin{aligned} L_n(t + \Delta) &= L_n(t) + \int_t^{t+\Delta} \sigma_n(u)^\top \mu_n(u) du + \int_t^{t+\Delta} \sigma_n(u)^\top dW^B(u) \\ &\triangleq L_n(t) + D_n(t, t + \Delta) + M_n(t, t + \Delta), \end{aligned}$$

where $M_n(t, t + \Delta)$ is a zero-mean martingale increment and $D_n(t, t + \Delta)$ is the increment of a predictable process. In many cases of practical interest, high-performance special-purpose schemes exist for simulation of $M_n(t, t + \Delta)$. This, for instance, is the case for the SV-LM model specification (Section 15.2.5), as discussed in detail in Section 10.5. In such cases, we obviously will choose to generate $M_n(t, t + \Delta)$ from the special-purpose scheme, and it thus suffices to focus on the term $D_n(t, t + \Delta)$. A simple approach is to use Euler stepping:

$$\widehat{L}_n(t + \Delta) = \widehat{L}_n(t) + \sigma_n(t)^\top \mu_n(t) \Delta + \widehat{M}_n(t, t + \Delta), \quad (15.61)$$

where $\widehat{M}_n(t, t + \Delta)$ is generated by a special-purpose scheme.

The drift adjustments in (15.61) are explicit in nature, as they are based only on the forward curve at time t . To incorporate information from time $t + \Delta$, we can use the *predictor-corrector* scheme from Section 4.2.5, which for (15.61) will take the two-step form

$$\bar{L}_n(t + \Delta) = \hat{L}_n(t) + \sigma_n \left(t, \hat{\mathbf{L}}(t) \right)^\top \mu_n \left(t, \hat{\mathbf{L}}(t) \right) \Delta + \widehat{M}_n(t, t + \Delta), \quad (15.62)$$

$$\begin{aligned} \hat{L}_n(t + \Delta) &= \hat{L}_n(t) + \theta_{\text{PC}} \sigma_n \left(t, \hat{\mathbf{L}}(t) \right)^\top \mu_n \left(t, \hat{\mathbf{L}}(t) \right) \Delta \\ &\quad + (1 - \theta_{\text{PC}}) \sigma_n \left(t + \Delta, \bar{\mathbf{L}}(t + \Delta) \right)^\top \mu_n \left(t + \Delta, \bar{\mathbf{L}}(t + \Delta) \right) \Delta \\ &\quad + \widehat{M}_n(t, t + \Delta), \end{aligned} \quad (15.63)$$

where θ_{PC} is a parameter in $[0, 1]$ that determines the amount of implicitness we want in our scheme ($\theta_{\text{PC}} = 1$: fully explicit; $\theta_{\text{PC}} = 0$: fully implicit). In practice, we would nearly always go for the balanced choice of $\theta_{\text{PC}} = 1/2$. In (15.62)–(15.63), \mathbf{L} denotes the vector of all Libor rates, $\mathbf{L}(t) = (L_1(t), \dots, L_{N-1}(t))^\top$ (with the convention that $L_i(t) \equiv L_i(T_i)$ for $i < q(t)$), and $\hat{\mathbf{L}}, \bar{\mathbf{L}}$ defined accordingly. In particular, the short-hand notation

$$\mu_n \left(t, \hat{\mathbf{L}}(t) \right)$$

is used to indicate that μ_n (and σ_n) may depend on the state of the entire forward curve at time t .

The technique above was based on a standard (additive) Euler scheme. If one is more inclined to use a multiplicative scheme in the vein of (15.56), we may replace the explicit scheme (15.61) with

$$\hat{L}_n(t + \Delta) = \hat{L}_n(t) \exp \left\{ \frac{\sigma_n(t)^\top}{\hat{L}_n(t)} \mu_n(t) \Delta \right\} \widehat{M}_n(t, t + \Delta), \quad (15.64)$$

where $\widehat{M}_n(t, t + \Delta)$ now has been re-defined to be a unit-mean positive random variable, often a discretized multiplicative increment of an exponential martingale. The construction of a predictor-corrector extension of (15.64) follows closely the steps above, and is left for the reader.

While the weak convergence order of simulation schemes may not be affected by predictor-corrector schemes (Section 4.2.5), experiments show that (15.62)–(15.63) often will reduce the bias significantly relative to a fully explicit Euler scheme. Some results for the simple log-normal LM model can be found in Hunter et al. [2001] and Rebonato [2002]. As the computational effort of computing the predictor-step is not insignificant, the speed-accuracy trade-off must be evaluated on a case-by-case basis. Section 15.6.2.3 below discusses a possible modification of the predictor-corrector scheme to improve efficiency.

15.6.2.2 Euler Scheme with Predictor-Corrector

In simulating the term $M_n(t, t + \Delta)$ in the predictor-corrector scheme above, we can always use an Euler scheme, i.e. in (15.61) we set

$$\widehat{M}_n(t, t + \Delta) = \sigma_n(t)^\top \sqrt{\Delta} Z,$$

where Z is an m -dimensional vector of standard Gaussian draws. As we recall from Chapter 4, however, it may also be useful to apply the predictor-corrector principle to the martingale part of the forward rate evolution itself, although this would involve the evaluation of derivatives of the LM volatility term with respect to the forward Libor rates; see Chapter 4 for details.

15.6.2.3 Lagging Predictor-Corrector Scheme

Drift calculations, as was pointed out earlier, are the most computationally expensive part of any Monte Carlo scheme for a Libor market model. The predictor-corrector scheme of (15.62)–(15.63) requires *two* calculations of the drift and is thus considerably more expensive than the standard Euler scheme. We often prefer to use a “lagging” modified predictor-corrector scheme which, as it turns out, allows us to realize most of the benefits if the predictor-corrector scheme, while keeping computational costs comparable to the standard Euler scheme.

Recall the definition of the drift of the n -th Libor rate under the spot measure,

$$\mu_n(t) = \sum_{j=q(t)}^n \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)}.$$

Note that the drift depends on the values of the Libor rates of indices less than or equal to n . Let us split the contributions coming from Libor rates with an index strictly less than n , and the n -th Libor rate,

$$\mu_n(t) = \sum_{j=q(t)}^{n-1} \frac{\tau_j \sigma_j(t)}{1 + \tau_j L_j(t)} + \frac{\tau_n \sigma_n(t)}{1 + \tau_n L_n(t)}.$$

Denoting $t' = t + \Delta$, we observe that if we simulate the Libor rates in the order of increasing index, then by the time we need to simulate $L_n(t')$, we have already simulated $L_j(t')$, $j = q(t), \dots, n-1$. Hence, it is natural to use the predictor-corrector technique for the part of the drift that depends on Libor rates maturing strictly before T_n , while treating the part of the drift depending on the n -th Libor rate explicitly. This idea leads to the following scheme (compare to (15.61) or (15.62)–(15.63) with $\theta_{PC} = 1/2$),

$$\begin{aligned} \widehat{L}_n(t') &= \widehat{L}_n(t) + \sigma_n(t)^\top \\ &\times \left(\frac{1}{2} \sum_{j=q(t)}^{n-1} \left(\frac{\tau_j \sigma_j(t)}{1 + \tau_j \widehat{L}_j(t)} + \frac{\tau_j \sigma_j(t')}{1 + \tau_j \widehat{L}_j(t')} \right) + \frac{\tau_n \sigma_n(t)}{1 + \tau_n \widehat{L}_n(t)} \right) \Delta + \widehat{M}_n(t, t'). \end{aligned} \quad (15.65)$$

Importantly, the drifts required for this scheme also satisfy a recursive relationship, allowing for an efficient update. Defining

$$\hat{\alpha}_n(t') = \sum_{j=q(t)}^n \left(\frac{\tau_j \sigma_j(t)}{1 + \tau_j \hat{L}_j(t)} + \frac{\tau_j \sigma_j(t')}{1 + \tau_j \hat{L}_j(t')} \right),$$

we see that, clearly,

$$\hat{\alpha}_n(t') = \hat{\alpha}_{n-1}(t') + \frac{\tau_n \sigma_n(t)}{1 + \tau_n \hat{L}_n(t)} + \frac{\tau_n \sigma_n(t')}{1 + \tau_n \hat{L}_n(t')},$$

and (15.65) can be rewritten as

$$\hat{L}_n(t') = \hat{L}_n(t) + \sigma_n(t)^\top \left(\frac{1}{2} \hat{\alpha}_{n-1}(t') + \frac{\tau_n \sigma_n(t)}{1 + \tau_n \hat{L}_n(t)} \right) \Delta + \hat{M}_n(t, t'). \quad (15.66)$$

The scheme above can easily be applied to other probability measures. In fact, since in the terminal measure the drift

$$\mu_n(t) = - \sum_{j=n+1}^{N-1} \frac{\tau_j \sigma_j(t)}{1 + \tau_j \hat{L}_j(t)},$$

does not depend on $L_n(t)$ in the first place, no “lag” is required in this measure. Indeed, we simply re-define

$$\hat{\alpha}_n(t') = - \sum_{j=n+1}^{N-1} \left(\frac{\tau_j \sigma_j(t)}{1 + \tau_j \hat{L}_j(t)} + \frac{\tau_j \sigma_j(t')}{1 + \tau_j \hat{L}_j(t')} \right)$$

and, starting from $n = N - 1$ and working backwards, use the scheme

$$\hat{L}_n(t') = \hat{L}_n(t) + \sigma_n(t)^\top \frac{1}{2} \hat{\alpha}_n(t') \Delta + \hat{M}_n(t, t'). \quad (15.67)$$

Notice that $\hat{\alpha}_n$ now satisfies the recursion

$$\hat{\alpha}_{n-1}(t') = \hat{\alpha}_n(t') - \frac{\tau_n \sigma_n(t)}{1 + \tau_n \hat{L}_n(t)} - \frac{\tau_n \sigma_n(t')}{1 + \tau_n \hat{L}_n(t')},$$

to be started at $\hat{\alpha}_{N-1}(t) = 0$.

The modifications of (15.66) and (15.67) to accommodate log-Euler stepping are trivial and left to the reader to explore. The lagging predictor-corrector scheme in the spot Libor measure has, as far as we know, never appeared in the literature, and its theoretical properties are not well-known (although the terminal measure version was studied in [Joshi and Stacey \[2008\]](#)). Still, its practical performance is very good and we do not hesitate recommending it as the default choice for many applications.

15.6.2.4 Further Refinements of Drift Estimation

For large time-steps, it may be useful to explicitly integrate the time-dependent parts of the drift, rather than rely on pure Euler-type approximations. Focusing on, say, (15.61), assume that we can write

$$\sigma_n(u)^\top \mu_n(u) \approx g(u, \mathbf{L}(t)), \quad u \geq t, \quad (15.68)$$

for a function g that depends on time as well as the state of the forwards frozen at time t . Then,

$$D_n(t, t + \Delta) = \int_t^{t+\Delta} \sigma_n(u)^\top \mu_n(u) du \approx \int_t^{t+\Delta} g(u, \mathbf{L}(t)) du. \quad (15.69)$$

As g evolves deterministically for $u > t$, the integral on the right-hand side can be evaluated either analytically (if g is simple enough) or by numerical quadrature. If doing the integral numerically, a decision must be made on the spacing of the integration grid. For volatility functions that are piecewise flat on the tenor-structure — which is a common assumption in model calibration — it is natural to align the grid with dates in the tenor structure.

To give an example, consider the DVF LM model, where we get (in the terminal measure, for a change)

$$\begin{aligned} \sigma_n(u)^\top \mu_n(u) &= -\varphi(L_n(u)) \lambda_n(u)^\top \sum_{j=n+1}^{N-1} \frac{\tau_j \lambda_j(u) \varphi(L_j(u))}{1 + \tau_j L_j(u)} \\ &\approx -\varphi(L_n(t)) \lambda_n(u)^\top \sum_{j=n+1}^{N-1} \frac{\tau_j \lambda_j(u) \varphi(L_j(t))}{1 + \tau_j L_j(t)}, \quad u \geq t, \end{aligned}$$

which is of the form (15.68). For stochastic volatility models we might, say, additionally assume that the process $z(t)$ would stay on its forward path, i.e. $z(u) \approx E_t^B(z(u))$ which can often be computed in closed form for models of interest. For instance, for the SV model in (15.15) we have

$$E_t^B(z(u)) = z_0 + (z(t) - z_0)e^{-\theta(u-t)}.$$

The approach in (15.69) easily combines with predictor-corrector logic, i.e. we could write

$$\begin{aligned} D_n(t, t + \Delta) &\approx \theta_{\text{PC}} \int_t^{t+\Delta} g(u, \mathbf{L}(t)) du \\ &\quad + (1 - \theta_{\text{PC}}) \int_t^{t+\Delta} g(u, \bar{\mathbf{L}}(t + \Delta)) du, \quad (15.70) \end{aligned}$$

where $\bar{L}_i(t + \Delta)$ has been found in a predictor-step using (15.69) in (15.61). The “lagged” schemes in Section 15.6.2.3 work equally well. Formula (15.69) also applies to exponential-type schemes such as (15.64), with or without predictor-corrector adjustment; we leave details to the reader.

15.6.2.5 Brownian-Bridge Schemes and Other Ideas

As a variation on the predictor-corrector scheme, we could attempt a further refinement to take into account variance of the Libor curve between the sampling dates t and $t + \Delta$. Schemes attempting to do so by application of *Brownian bridge* techniques²⁶ were proposed in Andersen [2000b] and Pietersz et al. [2004], among others. While performance of these schemes is mixed — tests in Joshi and Stacey [2008] show rather unimpressive results in comparison to simpler predictor-corrector schemes — the basic idea is sufficiently simple and instructive to merit a brief mention. In a nutshell, the Brownian bridge approach aims to replace in (15.69) all forward rates $\mathbf{L}(t)$ with expectation of $\mathbf{L}(u)$, conditional upon the forward rates ending up at $\bar{\mathbf{L}}(t + \Delta)$, with $\bar{\mathbf{L}}(t + \Delta)$ generated in a predictor-step. Under simplifying assumptions on the dynamics of $L_n(t)$, a closed-form expression is possible for this expectation.

Proposition 15.6.4. *Assume that*

$$dL_n(t) \approx \sigma_n(t)^\top dW(t),$$

where $\sigma_n(t)$ is deterministic and $W(t)$ is an m -dimensional Brownian motion in some probability measure \mathbb{P} . Let

$$v_n(t, T) = \int_t^T \|\sigma_n(s)\|^2 ds, \quad T \geq t.$$

Then, for $u \in [t, t + \Delta]$,

$$\mathbb{E}(L_n(u) | L_n(t), L_n(t + \Delta)) = L_n(t) + \frac{v_n(t, u)}{v_n(t, t + \Delta)} (L_n(t + \Delta) - L_n(t)).$$

Proof. We first state a very useful general result for multi-variate Gaussian variables.

Lemma 15.6.5. *Let $X = (X_1, X_2)^\top$ be a partitioned vector of Gaussian variables, where X_1 and X_2 are themselves vectors. Assume that the covariance matrix between X_i and X_j is $\Sigma_{i,j}$ such that the total covariance matrix of X is*

$$\Sigma = \begin{pmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{pmatrix}$$

(where, of course, $\Sigma_{2,1} = \Sigma_{1,2}^\top$). Let the vector means of X_i be μ_i , $i = 1, 2$, and assume that $\Sigma_{2,2}$ is invertible. Then $X_1 | X_2 = x$ is Gaussian:

$$(X_1 | X_2 = x) \sim \mathcal{N}(\mu_1 + \Sigma_{1,2} \Sigma_{2,2}^{-1} (x - \mu_2), \Sigma_{1,1} - \Sigma_{1,2} \Sigma_{2,2}^{-1} \Sigma_{2,1}).$$

²⁶See Section 4.2.9 for an introduction to the Brownian bridge, albeit for a somewhat different application.

In Lemma 15.6.5, set $X_1 = L_n(u) - L_n(t)$ and $X_2 = L_n(t + \Delta) - L_n(t)$. Note that $\mu_1 = \mu_2 = 0$ and

$$\Sigma_{1,2} = \Sigma_{2,1} = \Sigma_{1,1} = v_n(t, u), \quad \Sigma_{2,2} = v_n(t, t + \Delta).$$

The result of Proposition 15.6.4 follows. \square

We can use the result of Proposition 15.6.4 in place of the ordinary corrector step. For instance, in (15.70) we write

$$D_n(t, t + \Delta) \approx \int_t^{t+\Delta} g(u, \mathbf{m}(u)) du,$$

where, for $\mathbf{m}(u) = (m_1(u), \dots, m_{N-1}(u))$,

$$m_i(u) = E(L_i(u) | L_i(t), \bar{L}_i(t + \Delta))$$

is computed according to Proposition 15.6.4 once $\bar{L}_i(t + \Delta)$ has been sampled in a predictor step.

In some cases, it may be more appropriate to assume that L_n is roughly log-normal, in which case Proposition 15.6.4 must be altered slightly.

Lemma 15.6.6. *Assume that*

$$dL_n(t)/L_n(t) \approx \sigma_n(t)^\top dW(t),$$

where σ_n is deterministic and $W(t)$ is an m -dimensional Brownian motion in some probability measure P . Then, for $u \in [t, t + \Delta]$,

$$E(L_n(u) | L_n(t), L_n(t + \Delta)) = L_n(t) \left(\frac{L_n(t + \Delta)}{L_n(t)} \right)^{v_n(t, u)/v_n(t, t + \Delta)} \\ \times \exp \left(\frac{v_n(t, u)(v_n(t, t + \Delta) - v_n(t, u))}{2v_n(t, t + \Delta)} \right),$$

where $v_n(t, T)$ is given in Proposition 15.6.4.

Proof. Apply Lemma 15.6.5 to $X_1 = \ln L_n(u) - \ln L_n(t)$ and $X_2 = \ln L_n(t + \Delta) - \ln L_n(t)$. To translate back to find the conditional mean of e^{X_1} , one may use the fact that $E(e^{a+bY}) = e^{a+b^2/2}$ if Y is Gaussian $\mathcal{N}(0, 1)$. \square

Joshi and Stacey [2008] investigate a number of other possible discretization schemes for the drift term in the LM model, including ones that attempt to incorporate information about the correlation between various forward rates. In general, many of these schemes will result in some improvement of the discretization error, but at the cost of more computational complexity and effort. All things considered, we hesitate to recommend any of these methods (and this goes for the Brownian bridge scheme above) for general-purpose use, as the bias produced by simpler methods is often adequate. If not, it may, in fact, often be the case that we can insert a few extra simulation dates

inside large gaps to bring down the bias, yet still spend less computational time than we would if using a more complex method of bridging the gap in a single step. Finally, we should note that most authors (including [Joshi and Stacey \[2008\]](#)) exclusively examine simple log-normal models where the martingale component (M_n in the notation of [Section 15.6.2.1](#)) can be simulated completely bias-free. When using more realistic models, this will not always be the case, in which case high-precision simulation of the drift term D_n will likely be a waste of time.

15.6.2.6 High-Order Schemes

Even with predictor-corrector adjustment, all Euler-type discretization schemes are limited to a convergence order of Δ . To raise this, one possibility is to consider higher-order schemes, such as the Milstein scheme and similar Taylor-based approaches; see [Section 4.2.6](#) for details. Many high-order schemes unfortunately become quite cumbersome to deal with for the type of high-dimensional vector-SDE that arises in the context of LM models and, possibly as a consequence of this, there are currently very few empirical results in the literature to lean on. One exception is Brotherton-Ratcliffe ([Brotherton-Ratcliffe \[1997\]](#)) where a Milstein scheme has been developed for the basic log-normal LM model with piecewise flat volatilities. The efficacy of this, and similar high-order schemes, in the context of the generalized LM model would obviously depend strongly on the particular choice of model formulation.

A simple alternative to classical Taylor-based high-order schemes involves Richardson extrapolation based on prices found by simulating on two separate time lines, one coarser than the other (see [Section 4.2.7](#) for details). [Andersen and Andreasen \[2000b\]](#) list some results for Richardson extrapolation, the effect of which seems to be rather modest.

15.6.3 Martingale Discretization

Consider again the hybrid measure induced by the numeraire \tilde{P}_{n+1} , defined in [Section 15.2.2](#). As discussed, one effect of using this measure is to render the process for the n -th forward Libor rate, $L_n(t)$, a martingale. When time-discretizing the LM model using, say, an Euler scheme, the martingale property of $L_n(t)$ is automatically preserved, ensuring that the expectation of the discretized approximation to $L_n(\cdot)$, $\hat{L}_n(\cdot)$, will have expectation $L_n(0)$, with no discretization bias. Also, when using Monte Carlo to estimate the price of the zero-coupon bond maturing at time T_{n+1} , we get

$$P(0, T_{n+1}) = \tilde{P}_{n+1}(0) E^{n+1}(1),$$

which will (obviously) be estimated bias-free as well.

As the discussion above highlights, it is possible to select a measure such that a particular zero-coupon bond and a particular FRA will be priced bias-free²⁷ by Monte Carlo simulation, even when using a simple Euler scheme. While we are obviously rarely interested in pricing zero-coupon bonds by Monte Carlo methods, this observation can nevertheless occasionally help guide the choice of simulation measure, particularly if, say, a security can be argued to depend primarily on a single forward rate (e.g. caplet-like securities). In practice, matters are rarely this clear-cut, and one wonders whether perhaps simulation schemes exist that will simultaneously price all zero-coupon bonds $P(\cdot, T_1), P(\cdot, T_2), \dots, P(\cdot, T_N)$ bias-free. It should be obvious that this cannot be accomplished by a simple measure-shift, but will require a more fundamental change in simulation strategy.

15.6.3.1 Deflated Bond Price Discretization

Fundamentally, we are interested in a simulation scheme that by construction will ensure that all numeraire-deflated bond prices are martingales. The easiest way to accomplish this is to follow a suggestion offered by [Glasserman and Zhao \[2000\]](#): instead of discretizing the dynamics for Libor rates directly, simply discretize the deflated bond prices themselves. To demonstrate, let us consider the spot measure, and define

$$U(t, T_{n+1}) = \frac{P(t, T_{n+1})}{B(t)}. \quad (15.71)$$

Lemma 15.6.7. *Let dynamics in the spot measure Q^B be as in Lemma 15.2.3. The dynamics for deflated zero-coupon bond prices (15.71) are given by*

$$\frac{dU(t, T_{n+1})}{U(t, T_{n+1})} = - \sum_{j=q(t)}^n \tau_j \frac{U(t, T_{j+1})}{U(t, T_j)} \sigma_j(t)^\top dW^B(t), \quad n = q(t), \dots, N-1. \quad (15.72)$$

Proof. We note that, by definition,

$$U(t, T_{n+1}) = \frac{P(t, T_{q(t)}) \prod_{j=q(t)}^n \frac{1}{1+\tau_j L_j}}{P(t, T_{q(t)}) B(T_{q(t)-1})} = \frac{\prod_{j=q(t)}^n \frac{1}{1+\tau_j L_j}}{B(T_{q(t)-1})},$$

where $B(T_{q(t)-1})$ is non-random at time t . It follows that

$$\frac{U(t, T_{j+1})}{U(t, T_j)} = \frac{1}{1 + \tau_j L_j(t)}.$$

$U(t, T_{n+1})$ must, by construction, be a martingale. An application of Ito's lemma to the diffusion term of U gives

²⁷But not error-free, obviously — there will still be a statistical mean-zero error on the simulation results. See Section 15.6.4 below.

$$dU(t, T_{n+1}) = -U(t, T_{n+1}) \sum_{j=q(t)}^n \frac{\tau_j \sigma_j(t)^\top}{1 + \tau_j L_j(t)} dW^B(t),$$

and the lemma follows. \square

Discretization schemes for (15.72) that preserve the martingale property are easy to construct. For instance, we could use the log-Euler scheme

$$\widehat{U}(t + \Delta, T_{n+1}) = \widehat{U}(t, T_{n+1}) \exp \left(-\frac{1}{2} \|\gamma_{n+1}(t)\|^2 \Delta + \gamma_{n+1}(t)^\top Z \sqrt{\Delta} \right), \quad (15.73)$$

where, as before, Z is an m -dimensional standard Gaussian draw, and

$$\gamma_{n+1}(t) \triangleq - \sum_{j=q(t)}^n \tau_j \frac{\widehat{U}(t, T_{j+1})}{\widehat{U}(t, T_j)} \sigma_j(t). \quad (15.74)$$

We have several remarks to the log-Euler scheme (15.73). First, for models where interest rates cannot become negative, $U(t, T_{n+1})/U(t, T_n) = P(t, T_{n+1})/P(t, T_n)$ could not exceed 1 in a continuous-time model, so it might be advantageous to replace (15.74) with

$$\gamma_{n+1}(t) \triangleq - \sum_{j=q(t)}^n \tau_j \max \left(\frac{\widehat{U}(t, T_{j+1})}{\widehat{U}(t, T_j)}, 1 \right) \sigma_j(t),$$

as recommended in Glasserman and Zhao [2000]. Second, for computational efficiency we should rely on iterative updating,

$$\gamma_{n+1}(t) = \gamma_n(t) - \tau_n \max \left(\frac{\widehat{U}(t, T_{n+1})}{\widehat{U}(t, T_n)}, 1 \right) \sigma_n(t),$$

using the same arguments as those presented in Section 15.6.1.1. Third, once $\widehat{U}(t + \Delta, T_n)$ has been drawn for all possible n , we can reconstitute the Libor curve from the relation

$$\widehat{L}_n(t + \Delta) = \frac{\widehat{U}(t + \Delta, T_n) - \widehat{U}(t + \Delta, T_{n+1})}{\tau_n \widehat{U}(t + \Delta, T_{n+1})}, \quad n = q(t + \Delta), \dots, N - 1. \quad (15.75)$$

For completeness, we note that dynamics of the deflated bond prices in the terminal measure Q^{T_N} can easily be derived to be

$$\frac{dU(t, T_{n+1})}{U(t, T_{n+1})} = \sum_{j=n+1}^{N-1} \tau_j \frac{U(t, T_{j+1})}{U(t, T_j)} \sigma_j(t)^\top dW^N(t), \quad (15.76)$$

where we must now (re-)define $U(t, T_n)$ as a forward bond price

$$U(t, T_n) = P(t, T_n, T_N) = P(t, T_n)/P(t, T_N).$$

Equation (15.76) can form the basis of a discretization scheme in much the same manner as above.

15.6.3.2 Comments and Alternatives

The discretization scheme presented above will preserve the martingale property of all deflated bonds maturing in the tenor structure, and in this sense can be considered arbitrage-free. The resulting lack of bias on bond prices, however, does not necessarily translate into a lack of bias on any other derivative security price, e.g. a caplet or a swaption. In particular, we notice that nothing in the scheme above will ensure that bond price moments of any order other than one will be simulated accurately.

The extent of the bias induced by the scheme in Section 15.6.3.1 is specific to the security and model under consideration. For instance, using a log-Euler scheme for deflated bonds might work well in an LM model with rates that are approximately Gaussian, but might work less well in a model where rates are approximately log-normal. If results are disappointing, we can replace (15.73) with another discretization of (15.72) (see Chapter 4 for many examples), or we can try to discretize a quantity other than the deflated bonds $U(t, T_n)$. The latter idea is pursued in Glasserman and Zhao [2000], where several suggestions for discretization variables are considered. For instance, one can consider the differences

$$U(t, T_n) - U(t, T_{n+1}) \quad (15.77)$$

which are martingales since the U 's are. As follows from (15.75), discretizing $U(t, T_n) - U(t, T_{n+1})$ is, in a sense, close to discretizing L_n itself which may be advantageous. Joshi and Stacey [2008] contains some tests of discretization schemes based on (15.77), but, again, only in a log-normal setting.

15.6.4 Variance Reduction

We recall from the discussion in Chapter 4 that the errors involved in Monte Carlo pricing of derivatives can be split into two sources: the statistical Monte Carlo error (the standard error); and a bias unique to the discretization scheme employed. So far, our discussion has centered exclusively on the latter of these two types of errors and we now wish to provide some observations about the former. We should note, however, that it is difficult to provide generic prescription for variance reduction techniques in the LM model, as most truly efficient schemes tend to be quite specific to the product being priced. We shall offer several such product-specific variance reduction schemes in later chapters, and here limit ourselves to rather brief suggestions.

We recall that Chapter 4 discussed three types of variance reduction techniques: i) antithetic sampling; ii) control variates; and iii) importance sampling. All have potential uses in simulation of LM models.

15.6.4.1 Antithetic Sampling

Application of antithetic sampling to LM modeling is straightforward. Using the Euler scheme as an example, each forward rate sample path generated

from the relation

$$\widehat{L}_n(t + \Delta) = \widehat{L}_n(t) + \sigma_n(t)^\top \left(\mu_n(t)\Delta + \sqrt{\Delta}Z \right)$$

is simply accompanied by a “reflected” sample path computed by flipping the vector-valued Gaussian variable Z around the origin, i.e.

$$\widehat{L}_n^{(a)}(t + \Delta) = \widehat{L}_n^{(a)}(t) + \sigma_n(t)^\top \left(\mu_n^{(a)}(t)\Delta - \sqrt{\Delta}Z \right).$$

The reflection of Z is performed at each time-step, with both paths having identical starting points, $\widehat{L}_n^{(a)}(0) = \widehat{L}_n(0) = L_n(0)$. Using antithetic variates thus doubles the number of sample paths that will be generated from a fixed budget of random number draws. In practice, the variance reduction associated with antithetic variates is often relatively modest.

15.6.4.2 Control Variates

As discussed in Chapter 4, the basic (product-based) control variate method involves determining a set of securities (control variates) that i) have payouts close to that of the instrument we are trying to price; and ii) have known means in the probability measure in which we simulate. Obvious control variates in the LM model include (portfolios of) zero-coupon bonds and caplets. Due to discretization errors in generation of sample paths, we should note, however, that the sample means of zero-coupon bonds and caplets will deviate from their true continuous-time means with amounts that depend on the time-step and the discretization scheme employed. This error will nominally cause a violation of condition ii) — we are generally able only to compute in closed-form the continuous-time means — but the effect is often benign and will theoretically²⁸ be of the same order as the weak convergence order of the discretization scheme employed. Swaptions can also be included in the control variate set, although additional care must be taken here due to the presence of hard-to-quantify approximation errors in the formulas in Section 15.4.2. See Jensen and Svenstrup [2003] for an example of using swaptions as control variates for Bermudan swaptions.

An alternative interpretation of the control variate idea involves pricing a particular instrument using, in effect, two different LM models, one of which allows for an efficient computation of the instrument price, and one of which is the true model we are interested in applying. We shall return to it in Chapter 25.

Finally, the dynamic control variate method, based on the idea that an (approximate) self-financed hedging strategy could be a good proxy for the value of a security, is available for LMM models as well. The method was developed in Section 4.4.3.2.

²⁸Suppose that we estimate $E(X) \approx E(X' + Y' - \mu_Y)$, where $\mu_Y = E(Y') + O(\Delta^p)$ and $E(X') = E(X) + O(\Delta^p)$. Then clearly also $E(X' + Y' - \mu_Y) = E(X) + O(\Delta^p)$.

15.6.4.3 Importance Sampling

Importance sampling techniques have so far seen few applications in the simulation of LM models, with one exception: the simulation of securities with a knock-out barrier. The basic idea is here that sample paths are generated conditional on a barrier not being breached, ensuring that all paths survive to maturity; this conditioning step induces a change of measure. We will expose the details of this technique in Chapter 21, where we discuss the pricing of the TARN product introduced in Section 6.15.2.

SAMPLE