Two-factor LIBOR Markov-functional model

Mike Bennett and Joanne Kennedy

October 2005

1 Introduction

In this document, we give a brief introduction on the two-factor LIBOR Markov-functional model for which its theoretical construction and numerical implementation hasn't been available in the current literature. The two-factor LIBOR Markov-functional model is similar to the one-factor version but uses the idea of "pre-model" for successful implementation. For the original idea on pre-model, see (Hunt & Kennedy, 2004). In the coming sections, we will first describe the two-factor separable LIBOR market model and the pre-models based on it. We then proceed to the detailed construction and implementation of the two-factor LIBOR Markov-functional model. At each step, we provide sufficient information on the C++ code enclosed with this document.

2 The two-factor LIBOR market model

2.1 Notations and definitions

In this section, we set up the notation that we use throughout the discussion of the LIBOR market model and Markov-functional model in this document. Let D_{tT} denote the time$-t$ value of a zero-coupon discount bond with maturity T. Let $0 = T_0 < T_1 < T_2 < \cdots < T_{n+1}$ be the tenor structure for $i = 1, \ldots, n$. We define the corresponding forward LIBORs as follows

$$L_t^i := \frac{D_{tT_i} - D_{tT_{i+1}}}{\alpha_i D_{tT_{i+1}}},$$

where $\alpha_i := T_{i+1} - T_i$ are the accrual factors.

We develop all models under the terminal measure $\mathbb{F}$, which is the equivalent martingale measure associated with the numeraire $D_{\cdot T_{n+1}}$. It is convenient to define the numeraire-rebased discount bonds

$$\hat{D}_{tT} := \frac{D_{tT}}{D_{tT_{n+1}}},$$

which by definition satisfy the recurrence relationship

$$\hat{D}_{tT_i} = (1 + \alpha_i L_t^i)\hat{D}_{tT_{i+1}}.$$

C++ code: For the construction of the yield curve, see DfCurve.h and DfCurve.cpp. In this class, the yield curve may be set to be increasing, decreasing, flat or the actual market data.

2.2 Standard specification of the two-factor LIBOR market model

Under the two-factor LIBOR market model (Brace, Gatarek, & Musiela, 1997), each of the forward LIBORs L^i solves an SDE of the form

$$\mathrm{d}L_t^i = \mu_i(t)L_t^i\mathrm{d}t + \sigma_i(t)L_t^i.\mathrm{d}W_t, \tag{1}$$

for $i = 1, \ldots, n$ where $\sigma_i : [0, T_i] \to \mathbb{R}^2$ and $W = (W^1, W^2)^T$ is a two-dimensional Brownian motion such that

$$\mathrm{d}W_t^1\mathrm{d}W_t^2 = \rho\mathrm{d}t,$$

for some $\rho \in [-1, 1]$. In the following, we shall assume the volatility structure is separable, that is, σ_i may be written in the form

$$\sigma_i(t) = (\nu_i^1\sigma_t^1, \nu_i^2\sigma_t^2),$$

for some instantaneous volatility functions $\sigma^1, \sigma^2 : [0, T_n] \to \mathbb{R}^+$ and constant vectors $\nu_i := (\nu_i^1, \nu_i^2) \in \mathbb{R}^2, i = 1, \ldots, n$.

If the model is to be arbitrage-free under $\mathbb{F}$, the drift term must have the following form, for $1 \leq i \leq n$

$$\mu_i(L_t^{i+1}, \ldots, L_t^n, t) = -\sum_{j=i+1}^{n}\left(\frac{\alpha_j L_t^j}{1 + \alpha_j L_t^j}\right)s_t^{ij}, \tag{2}$$

where

$$s_t^{ij} := \nu_i^1\nu_j^1(\sigma_t^1)^2 + (\nu_i^1\nu_j^2 + \nu_i^2\nu_j^1)\rho\sigma_t^1\sigma_t^2 + \nu_i^2\nu_j^2(\sigma_t^2)^2. \tag{3}$$

The drift μ^n of the terminal LIBOR L^n is zero because L^n is a martingale.

Let $x = (x^1, x^2)^T$ be defined by

$$x_t^1 := \int_0^t \sigma_s^1\mathrm{d}W_s^1, \qquad x_t^2 := \int_0^t \sigma_s^2\mathrm{d}W_s^2, \tag{4}$$

then

$$\int_0^t \sigma_i(t).\mathrm{d}W_t = \int_0^t \nu_i.\mathrm{d}x_t = \nu_i.x_t. \tag{5}$$

The Markov process x will be the driving process of our two-factor LIBOR Markov-functional model.

For future convenience, let

$$u_t^i := \nu_i^1 x_t^1 + \nu_i^2 x_t^2. \tag{6}$$

The solution to the SDE for the separable two-factor LIBOR market model may now be written in the form

$$L_t^i = L_0^i \exp\left(\int_0^t \mu_i(s)\mathrm{d}s - \frac{1}{2}[u^i]_t + u_t^i\right). \tag{7}$$

In addition, define for $1 \leq j < k \leq n$ the following integrals of the respective instantaneous volatilities of the driving process x:

$$\xi_{j,k}^{11} := \int_{T_j}^{T_k} (\sigma_s^1)^2\mathrm{d}s, \qquad \xi_{j,k}^{22} := \int_{T_j}^{T_k} (\sigma_s^2)^2\mathrm{d}s, \qquad \xi_{j,k}^{12} := \int_{T_j}^{T_k} \rho\sigma_s^1\sigma_s^2\mathrm{d}s. \tag{8}$$

The variance of the process u^i at T_i may now be written as

$$[u^i]_{T_i} = (\nu_i^1)^2\xi_{0,i}^{11} + 2\nu_i^1\nu_i^2\xi_{0,i}^{12} + (\nu_i^2)^2\xi_{0,i}^{22}. \tag{9}$$

We choose to specify our market model such that it is consistent with the Black caplet implied volatility $\widetilde{\sigma}^i$ corresponding to each of the forward LIBORs $L_{T_i}^i$. In this case, $[u^i]_{T_i} = (\widetilde{\sigma}^i)^2 T_i$. This implies some restriction on the specification of the driving process (see section 3.3.3).

C++ code: For the implementation of the two-factor LIBOR market model, see LiborDriftApprox.h and LiborDriftApprox.cpp. This class contains the interface for both the full implementation of the LIBOR market model (full simulation) and the pre-model for the LIBOR Markov-functional model (described in the next section).

3 The two-factor LIBOR Markov-functional model

3.1 Overview of the construction of the model

In this section, we describe the simpler method of constructing a two-dimensional LIBOR Markov-functional model: a straightforward method that may be used if the pre-model $\tilde{L}_{T_i}^i(u_{T_i}^i, v_{T_i}^i)$ maybe written as a monotonic increasing function of $u_{T_i}^i$ only. As the more general pre-model with a more computationally intensive method is not so different from the simple one, e.g. Brownian Bridge pre-model, we will describe it in the appendix. The simple pre-model suggested by (Hunt & Kennedy, 2004) is the first type and thus admits a straightforward implementation, whereas the two-dimensional drift approximation pre-model does not.

The two-dimensional driving process of the Markov-functional model, x, is assumed to be of the form (4). As in the one-dimensional case, for most pricing applications it is typically sufficient to recover the functional form $D_{T_iT_j}(x_{T_i})$ for $1 \le i \le j \le n$. In fact, it is only necessary to specify the functional forms of the numeraire $D_{T_iT_{n+1}}(x_{T_i})$ because the remaining functional forms may be determined using the martingale property of numeraire-rebased discount factors (as described in subsection 3.2.2). Given the functional forms $\tilde{L}_{T_i}^i(x_{T_i})$ under the pre-model, the functional forms of the numeraire $D_{T_iT_{n+1}}(x_{T_i})$ under the Markov-functional model may be recovered as follows.

We proceed by observing that under the two-factor separable LIBOR market model, the terminal LIBOR can always be written explicitly in terms of u^n under $\mathbb{F}$ (since $\mu^n = 0$):

$$L_t^n = L_0^n \exp\left(u_t^n - \frac{1}{2}[u^n]_t\right). \tag{10}$$

Under this model, $\ln L_{T_n}^n$ has a Normal distribution under $\mathbb{F}$ with mean $(\ln L_0^n - \frac{1}{2}[u^n]_{T_n})$ and variance $[u^n]_{T_n}$. Note that u^n is defined such that the variance $[u^n]_{T_n}$ matches $(\widetilde{\sigma}^n)^2 T_n$, where $\widetilde{\sigma}^n$ is the terminal caplet volatility, so the model is indeed consistent with the Black formula for the price of the terminal caplet. To calibrate the Markov-functional model to the Black formula for the price of the terminal caplet, the distribution of $\ln L_{T_n}^n$ under $\mathbb{F}$ must be the same as under the market model. That is,

$$L_{T_n}^n(u_{T_n}^n) = L_0^n \exp\left(u_{T_n}^n - \frac{1}{2}[u^n]_{T_n}\right).$$

The functional form of the numeraire at T_n may be recovered immediately since

$$D_{T_nT_{n+1}}(u_{T_n}^n) = (1 + \alpha_n L_{T_n}^n(u_{T_n}^n))^{-1}.$$

The functional forms of the numeraire $D_{T_iT_{n+1}}(x_{T_i})$ for $1 \leq i \leq n$ are found by working back recursively from the terminal time T_n through each exercise date. At each exercise date T_i, once the functional form of $L^i_{T_i}(x_{T_i})$ is known it is straightforward to recover the functional form of the numeraire (as in the one-dimensional case).

In these intermediate stages, the form of the pre-model will influence the functional forms of each LIBOR $L^i_{T_i}$ via the relationship $L^i_{T_i} = f^i(\tilde{L}^i_{T_i})$. The monotonic increasing functions f^i are recovered by calibrating to caplet prices. As in the one-dimensional case, calibrating to the prices of the vanilla caplets is equivalent to calibrating to the prices of digital caplets, so for convenience we choose to do the latter.

To find the functional form f^i, in analogy with the one-dimensional case consider the numerically evaluated expectation

$$J^i_0(x^*) := D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_iT_{i+1}}(x_{T_i}) \mathbf{1}_{\{\tilde{L}^i_{T_i}(x_{T_i}) > x^*\}} \right]. \tag{11}$$

The functional form $\hat{D}_{T_iT_{i+1}}(x_{T_i})$ in this expectation may be found from the values of one over the numeraire at T_{i+1}, $\hat{D}_{T_{i+1}T_{i+1}}(x_{T_{i+1}})$ (which will have been determined at the previous iteration), by application of the martingale property (see subsection 3.2.2). Compare (11) with the value of a digital caplet with strike K setting at T_i:

$$\begin{aligned} \tilde{V}^i_0(K) &= D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_iT_{i+1}} \mathbf{1}_{\{L^i_{T_i} > K\}} \right] \\ &= D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_iT_{i+1}} \mathbf{1}_{\{f^i(\tilde{L}^i_{T_i}) > K\}} \right], \end{aligned} \tag{12}$$

which may be determined from the market prices of caplets setting at T_i. If f^i is monotonic then

$$f^i(x^*) = (\tilde{V}^i_0)^{-1}(J^i_0(x^*)).$$

Thus we may obtain the functional form

$$L^i_{T_i}(x_{T_i}) = f^i\left(\tilde{L}^i_{T_i}(x_{T_i})\right).$$

The value of one over the numeraire, $\hat{D}_{T_iT_i}(x_{T_i}) = (D_{T_iT_{n+1}})^{-1}(x_{T_i})$, may now be recovered from $L^i_{T_i}(x_{T_i})$ by observing that

$$\hat{D}_{T_iT_i}(x_{T_i}) = (1 + \alpha_i L^i_{T_i}(x_{T_i}))\hat{D}_{T_iT_{i+1}}(x_{T_i}).$$

In the following, the prices of caplets are taken to be given by the Black formula in order to match the two-factor separable LIBOR market model described in section 2.2. In this case, the value of a digital caplet with strike K setting at T_i is given by

$$\tilde{V}^i_0(K) = D_{0T_{n+1}} \Phi\left(\frac{\ln(L^i_0/K))}{\tilde{\sigma}^i\sqrt{T_i}} - \frac{1}{2}\tilde{\sigma}^i\sqrt{T_i} \right),$$

where Φ denotes the standard cumulative Normal distribution function. Therefore,

$$L^i_{T_i}(x_{T_i}) = L^i_0 \exp\left(-\frac{1}{2}(\tilde{\sigma}^i)^2 T_i - \tilde{\sigma}^i\sqrt{T_i}\Phi^{-1}\left(\frac{J^i_0(\tilde{L}^i_{T_i}(x_{T_i}))}{D_{0T_{i+1}}} \right) \right).$$

3.2 Simplified implementation with restricted pre-model

In the special case where the pre-model $\tilde{L}^i_{T_i}$ may be written as a monotonic increasing function of $u^i_{T_i}$ only, it is possible to simplify the general implementation methods described in Appendix B. This relatively efficient implementation is described using the example of the pre-model originally suggested in (Hunt & Kennedy, 2004), although these methods apply whenever the pre-model is an increasing function of the form $\tilde{L}^i_{T_i}(u^i_{T_i})$.

Under the simple pre-model,

$$\begin{aligned}\tilde{L}^i_{T_i} &= L^i_0 \exp(\mu^i + \nu_i.x_{T_i}) \\ &= L^i_0 \exp(\mu^i + u^i_{T_i}),\end{aligned} \tag{13}$$

where μ^i assumed constant and ν_i and x are as defined above. Note that $\tilde{L}^i_{T_i}$ is trivially a monotonic increasing function of $u^i_{T_i}$. If we assume that all LIBORs under this model have zero drift under $\mathbb{F}$, we obtain a suitable value for μ^i (note this approximate model is clearly not arbitrage-free). Now, $u^i_{T_i}$ is Normally distributed under $\mathbb{F}$ with zero mean and variance $[u^i]_{T_i}$. Therefore,

$$\begin{aligned}\mathbb{E}_{\mathbb{F}}[\tilde{L}^i_{T_i}] &= L^i_0 \exp(\mu^i)\mathbb{E}_{\mathbb{F}}[\exp(u^i_{T_i})] \\ &= L^i_0 \exp(\mu^i) \exp\left(\frac{1}{2}[u^i]_{T_i}\right) \\ &= L^i_0\end{aligned}$$

under the assumption of zero drift, hence

$$\mu^i = -\frac{1}{2}[u^i]_{T_i}.$$

C++ code: this pre-model is known as the "quickApprox" type described in LiborDriftApprox.cpp.

The following subsections describe in detail how a two-dimensional Markov-functional model based on this particular pre-model may be constructed numerically on a grid.

3.2.1 Numerical evaluation of value of J^i_0

Working back from the terminal time T_n recursively through each exercise date T_i, it is necessary to evaluate expectations of the form (11). Assume for the moment the functional form of $\hat{D}_{T_iT_{i+1}}$ has already been determined. For the simple pre-model (13), equation (11) may be rewritten in the form

$$I^i_0(u^*) := D_{0T_{n+1}}\mathbb{E}_{\mathbb{F}}\left[\hat{D}_{T_iT_{i+1}}(u^i_{T_i}, v^i_{T_i})\mathbf{1}_{\{u^i_{T_i} > u^*\}}\right], \tag{14}$$

where $u^* = \ln(x^*/L^i_0) - \mu^i$. Recall from (6) that $u^i_{T_i}$ satisfies

$$u^i_{T_i} = \nu^1_i x^1_{T_i} + \nu^2_i x^2_{T_i}.$$

Suppose we choose

$$v^i_{T_i} := x^1_{T_i} + \gamma_i x^2_{T_i},$$

such that $v^i_{T_i}$ and $u^i_{T_i}$ are independent. Now $u^i_{T_i}$ and $v^i_{T_i}$ are both Normal and $\mathbb{E}_{\mathbb{F}}[u^i_{T_i}] = \mathbb{E}_{\mathbb{F}}[v^i_{T_i}] = 0$, so for independence we must have

$$\begin{aligned}
\mathbb{E}_{\mathbb{F}}[u^i_{T_i} v^i_{T_i}] &= \mathbb{E}_{\mathbb{F}}[(\nu^1_i x^1_{T_i} + \nu^2_i x^2_{T_i})(x^1_{T_i} + \gamma_i x^2_{T_i})] \\
&= \mathbb{E}_{\mathbb{F}}[\nu^1_i (x^1_{T_i})^2 + (\nu^1_i \gamma_i + \nu^2_i) x^1_{T_i} x^2_{T_i} + \nu^2_i \gamma_i (x^2_{T_i})^2] \\
&= \nu^1_i \mathrm{var}(x^1_{T_i}) + (\nu^1_i \gamma_i + \nu^2_i)\mathrm{cov}(x^1_{T_i}, x^2_{T_i}) + \nu^2_i \gamma_i \mathrm{var}(x^2_{T_i}) \\
&= \nu^1_i \xi^{11}_{0,i} + (\nu^1_i \gamma_i + \nu^2_i)\xi^{12}_{0,i} + \nu^2_i \gamma_i \xi^{22}_{0,i} \\
&= 0,
\end{aligned}$$

hence

$$\gamma_i = -\frac{\nu^1_i \xi^{11}_{0,i} + \nu^2_i \xi^{12}_{0,i}}{\nu^1_i \xi^{12}_{0,i} + \nu^2_i \xi^{22}_{0,i}}.$$

Then

$$I^i_0(u^*) = D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}}[g(u^i_{T_i}) \mathbf{1}_{\{u^i_{T_i} > u^*\}}],$$

where

$$\begin{aligned}
g(u^i_{T_i}) &= \mathbb{E}_{\mathbb{F}}[\hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v^i_{T_i}) | u^i_{T_i}] \\
&= \int_{-\infty}^{\infty} \hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v) \phi_{v^i_{T_i}}(v) \mathrm{d}v, \qquad (15)
\end{aligned}$$

and $\phi_{v^i_{T_i}}(v)$ is the density of $v^i_{T_i}$, which is Normal with mean zero and variance

$$[v]^i_{T_i} = \xi^{11}_{0,i} + 2\gamma_i \xi^{12}_{0,i} + (\gamma_i)^2 \xi^{22}_{0,i}.$$

The integral (15) is of similar form to those integrals encountered in the evaluation of $J^i_0(x^*)$ under the one-dimensional Markov-functional model and the same methods may be applied. Once g has been computed on a grid of values of $u^i_{T_i}$, the same integration methods may be used to compute

$$I^i_0(u^*) = D_{0T_{n+1}} \int_{u^*}^{\infty} g(u) \phi_{u^i_{T_i}}(u) \mathrm{d}u,$$

where $\phi_{u^i_{T_i}}$ is the density of $u^i_{T_i}$, which is Normal with mean zero and variance $[u^i]_{T_i}$.

C++ code: the whole evaluation process is described in the member functions "calcI0()" and "j0()" of LMFGrid.cpp.

3.2.2 Application of martingale property of numeraire-rebased discount factors

The evaluation of I^i_0 described above (14) presupposes that $\hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v^i_{T_i})$ is known on a grid of values of $u^i_{T_i}, v^i_{T_i}$. The values of $\hat{D}_{T_i T_j}$ for all $1 \le i \le j \le n$ may be computed from the functional form of $\hat{D}_{T_j T_j}$ (once this has been evaluated), since $\hat{D}_{\cdot T_j}$ is a martingale under $\mathbb{F}$ so

$$\hat{D}_{T_i T_j} = \mathbb{E}_{\mathbb{F}}\left[\hat{D}_{T_j T_j} | \mathcal{F}_{T_i}\right].$$

To compute the functional form of

$$\hat{D}_{T_i T_j}(u^i_{T_i}, v^i_{T_i}) = \mathbb{E}_{\mathbb{F}}\left[\hat{D}_{T_j T_j}(u^j_{T_j}, v^j_{T_j}) | u^i_{T_i}, v^i_{T_i}\right],$$

assume that the functional form of $\hat{D}_{T_jT_j} = 1/D_{T_jT_{n+1}}$ will have been determined over a grid of values of $u^j_{T_j}, v^j_{T_j}$ at the previous iteration. For any given coordinates $u^i_{T_i}, v^i_{T_i}$ it is trivial to recover $x^1_{T_i}, x^2_{T_i}$ using

$$x^2_{T_i} = \frac{u^i_{T_i} - \nu^1_i v^i_{T_i}}{\nu^2_i - \nu^1_i \gamma_i}, \qquad x^1_{T_i} = v^i_{T_i} - \gamma_i x^2_{T_i}.$$

Now, the distribution of $u^j_{T_j}$ and $v^j_{T_j}$ given $u^i_{T_i}, v^i_{T_i}$ is a bivariate Normal distribution. The covariance is given by

$$\begin{aligned}
\hat{\sigma}_{u^i,v^i} &:= \operatorname{cov}\left(u^j_{T_j}, v^j_{T_j} | x^1_{T_i}, x^2_{T_i}\right) \\
&= \operatorname{cov}\left(\nu^1_j x^1_{T_j} + \nu^2_j x^2_{T_j}, x^1_{T_j} + \gamma_j x^2_{T_j} | x^1_{T_i}, x^2_{T_i}\right) \\
&= \nu^1_j \xi^{11}_{i,j} + (\nu^1_j \gamma_j + \nu^2_j)\xi^{12}_{i,j} + \nu^2_j \gamma_j \xi^{22}_{i,j}.
\end{aligned}$$

Therefore, the correlation between $u^j_{T_j}$ and $v^j_{T_j}$ given $u^i_{T_i}, v^i_{T_i}$ is given by

$$\hat{\rho}_{u^i,v^i} = \frac{\hat{\sigma}_{u^i,v^i}}{\hat{\sigma}_{u^i}\hat{\sigma}_{v^i}},$$

where

$$\begin{aligned}
(\hat{\sigma}_{u^i})^2 &:= \operatorname{var}(u^j_{T_j} | x^1_{T_i}, x^2_{T_i}) = (\nu^1_j)^2 \xi^{11}_{i,j} + 2\nu^1_j \nu^2_j \xi^{12}_{i,j} + (\nu^2_j)^2 \xi^{22}_{i,j}, \\
(\hat{\sigma}_{v^i})^2 &:= \operatorname{var}(v^j_{T_j} | x^1_{T_i}, x^2_{T_i}) = \xi^{11}_{i,j} + 2\gamma^{12}_{i,j} + (\gamma_j)^2 \xi^{22}_{i,j}.
\end{aligned}$$

Thus the conditional distribution of $v^j_{T_j}$ given $u^j_{T_j}, u^i_{T_i}, v^i_{T_i}$ is Normal with mean

$$v^j_{T_i} + \left(\hat{\rho}_{u^i,v^i} \frac{\hat{\sigma}_{v^i}}{\hat{\sigma}_{u^i}}\right)(u^j_{T_j} - u^j_{T_i}),$$

and variance

$$(\hat{\sigma}_{v^i})^2 (1 - (\hat{\rho}_{u^i,v^i})^2).$$

Denoting this density by $\hat{\phi}_{v^i}$, the usual integration methods may be used to compute

$$\begin{aligned}
h(u^j_{T_j}) &:= \mathbb{E}_{\mathbb{F}}\left[\hat{D}_{T_jT_j}(u^j_{T_j}, v^j_{T_j}) | u^j_{T_j}, u^i_{T_i}, v^i_{T_i}\right] \\
&= \int_{-\infty}^{\infty} \hat{D}_{T_jT_j}(u^j_{T_j}, v)\hat{\phi}_{v^i}(v)\mathrm{d}v.
\end{aligned}$$

The density $\hat{\phi}_{u^i}$ of $u^j_{T_j}$ given $u^i_{T_i}, v^i_{T_i}$ is also Normal with mean $u^j_{T_i}$ and variance $(\hat{\sigma}_{u^i})^2$, hence the same integration methods may be used to evaluate

$$\hat{D}_{T_iT_j}(u^j_{T_i}, v^j_{T_i}) = \int_{-\infty}^{\infty} h(u)\hat{\phi}_{u^i}(u)\mathrm{d}u.$$

After the functional forms of one over the numeraire, $\hat{D}_{T_iT_i}(u^i_{T_i}, v^i_{T_i})$, have been recovered for all exercise dates T_i, it will usually be necessary to compute $\hat{D}_{T_iT_j}(u^i_{T_i}, v^i_{T_i})$ for all $1 \le i < j \le n$ to compute the specification of the model. However, as in the one-factor case, this is typically all that is required in practice.

C++ code: the application of martingale property of numeraire-rebased discount bonds is described in "populateTerminalSlice()" and "populateIntermediateSlice()" of LMFGrid.cpp.

3.2.3 Computation of intermediate functional forms

To find the functional form f^i, in analogy with the one-dimensional case we compare the numerically evaluated expectation (11) and (12). Comparing the form of these equations, it follows that as f^i is a monotonic increasing function (by assumption),

$$f^i(x^*) = \left(\tilde{V}_0^i\right)^{-1}(J_0^i(x^*)).$$

Therefore, setting $x^* = \tilde{L}^i_{T_i}(u^i_{T_i})$ for a given value of $u^i_{T_i}$,

$$\begin{aligned} L^i_{T_i}(u^i_{T_i}) &= f^i\left(\tilde{L}^i_{T_i}(u^i_{T_i})\right) \\ &= f^i(L_0^i \exp(\mu^i + u^i_{T_i})) \\ &= \left(\tilde{V}_0^i\right)^{-1}(J_0^i(L_0^i \exp(\mu^i + u^i_{T_i}))) \\ &= \left(\tilde{V}_0^i\right)^{-1}(I_0^i(u^i_{T_i})) \qquad (16) \\ &= L_0^i \exp\left(-\frac{1}{2}(\widetilde{\sigma}^i)^2 T_i - \widetilde{\sigma}^i \sqrt{T_i}\Phi^{-1}\left(\frac{I_0^i(u^i_{T_i})}{D_{0T_{i+1}}}\right)\right), \qquad (17) \end{aligned}$$

where Φ is the standard Normal cumulative density function, assuming digital caplet prices are given by the Black formula with implied volatility $\widetilde{\sigma}^i$. Note that for this particular pre-model, it is clear that $L^i_{T_i}$ is independent of $v^i_{T_i}$.

C++ code: the computation of intermediate functional forms is described in "populateIntermediateSlice()" of LMFGrid.cpp.

3.3 Specification of the instantaneous volatility structure

The specification of the instantaneous volatility structure used in this model is motivated by a "principal components analysis" type representation of movements in the term structure. Consider the SDE describing the separable two-factor LIBOR market model (1). A suitable generic specification of a separable volatility structure is given by the following:

$$\mathrm{d}u_t^i = \sigma_t \sum_{j=1}^{2} f_j(T_i - t)\mathrm{d}W_t^j, \qquad (18)$$

where σ_t is a deterministic function and $\mathrm{d}W_t^1 \mathrm{d}W_t^2 = \rho \mathrm{d}t$ for some constant ρ. If $f_j(t) := c_j e^{-\eta_j t}$ for some constants c_j and η_j, then the volatility structure will be of the desired separable form. That is,

$$\mathrm{d}u_t^i = \nu^i.\mathrm{d}x_t,$$

where

$$\nu^i = (c_1 e^{-\eta_1 T_i}, c_2 e^{-\eta_2 T_i}),$$

and

$$\mathrm{d}x_t = (e^{\eta_1 t}\sigma_t \mathrm{d}W_t^1, e^{\eta_2 t}\sigma_t \mathrm{d}W_t^2)^T.$$

If σ_t is a piecewise constant function this enables calibration to the relevant caplet prices, as described in section 3.3.3.

The specific parametrization described in the remainder of this section is of this form and is designed to capture the first two principal component of the movements in the forward curve over time.

C++ code: the specification of the instantaneous volatility structure is described as the "correlationStructure" in LiborDriftApprox.cpp. The PCA-type parametrization described below is implemented as the "meanReversion" type in LiborDriftApprox.cpp. The other type known as "meanReversionSpecifyingRho" is not mentioned in this document. This type fixes the instantaneous correlation of the two driving Brownian motions.

3.3.1 PCA-type parametrization

Suppose

$$\mathrm{d}u_t^i = \sigma_t \sum_{j=1}^{2} A_j g_j(T_i - t)\mathrm{d}Z_t^j,$$

where (Z^1, Z^2) is a standard two-dimensional Brownian motion, for some constants $A_1, A_2 \in [0,1]$ such that

$$|A_1| + |A_2| = 1.$$

Consider the following parametrization of g_1, g_2:

$$\begin{aligned} g_1(t) &:= B_1 e^{-\eta_1 t} \\ g_2(t) &:= B_2 e^{-\eta_2 t} - B_3 e^{-\eta_1 t}. \end{aligned}$$

Here, g_1 reflects a "parallel" shift of the forward curve if $\eta_1 > 0$ is small, in which case g_2 reflects the tilt of the forward curve (where η_2 is between 20% and 40%, say). When A_1 is close to 1, this two-dimensional model is effectively a one-factor model. For suitable values of the constants $\eta_1, \eta_2 > 0$, the values of B_1, B_2, B_3 may be chosen such that g_1, g_2 are orthonormal over the tenor structure under consideration, that is

$$\int_0^{T_{n+1}} g_i(u) g_j(u) \mathrm{d}u = \delta_{ij}, \tag{19}$$

for $i, j = 1, 2$ (as described in section 3.3.2).

To see this representation is of the separable form (18), observe that

$$\sigma_i(t).\mathrm{d}W_t = \sigma_t[e^{-\eta_1(T_i - t)}(A_1 B_1 \mathrm{d}Z_t^1 - A_2 B_3 \mathrm{d}Z_t^2) + e^{-\eta_2(T_i - t)} A_2 B_2 \mathrm{d}Z_t^2].$$

Setting

$$\begin{aligned} K_1 \mathrm{d}W_t^1 &= A_1 B_1 \mathrm{d}Z_t^1 - A_2 B_3 \mathrm{d}Z_t^2 \\ K_2 \mathrm{d}W_t^2 &= A_2 B_2 \mathrm{d}Z_t^2. \end{aligned}$$

It is clear that

$$\begin{aligned} K_1 &= \sqrt{(A_1 B_1)^2 + (A_2 B_3)^2}, \\ K_2 &= A_2 B_2, \end{aligned}$$

and also

$$\rho = \mathrm{d}W_t^1 \mathrm{d}W_t^2 = -\frac{-A_2^2 B_2 B_3}{K_1 K_2} = -\frac{A_2 B_3}{K_1},$$

(thus the correlation is fixed given the other parameters). Therefore,

$$\mathrm{d}u_t^i = \sigma_t (K_1 e^{-\eta_1 (T_i - t)} \mathrm{d}W_t^1 + K_2 e^{-\eta_2 (T_i - t)} \mathrm{d}W_t^2). \tag{20}$$

3.3.2 Computation of constants to ensure orthonormal basis

To compute B_1, B_2, B_3 to satisfy (19), for convenience define

$$\begin{aligned} P_1 &:= 1 - e^{-2\eta_1 T_{n+1}} \\ P_2 &:= 1 - e^{-2\eta_2 T_{n+1}} \\ P_{12} &:= 1 - e^{-(\eta_1 + \eta_2) T_{n+1}}. \end{aligned}$$

These quantities are all positive for $\eta_j > 0$. By evaluating (19) for $i = j = 1$,

$$\left(\frac{1 - e^{-2\eta_1 T_{n+1}}}{2\eta_1} \right) B_1^2 = 1,$$

thus

$$B_1 = \sqrt{\frac{2\eta_1}{P_1}}.$$

For $i = 1$ and $j = 2$, (19) gives

$$\frac{B_1 B_2 P_{12}}{\eta_1 + \eta_2} - \frac{B_1 B_3 P_1}{2\eta_1} = 0,$$

hence

$$B_3 = \frac{2\eta_1 P_{12}}{(\eta_1 + \eta_2) P_1} B_2. \tag{21}$$

Finally, for $i = j = 2$,

$$\frac{B_2^2 P_2}{2\eta_2} - \frac{2 B_2 B_3 P_{12}}{\eta_1 + \eta_2} + \frac{B_3^2 P_1}{2\eta_1} = 1,$$

and substituting B_3 using (21) yields

$$B_2^2 = \left(\frac{P_2}{2\eta_2} - \frac{2\eta_1 P_{12}^2}{(\eta_1 + \eta_2)^2 P_1} \right)^{-1}.$$

Therefore the model is well defined if the bracket on the right hand side is strictly positive, in which case $B_2 > 0$ is immediate and from (21) so is $B_3 > 0$ (clearly $B_1 > 0$).

3.3.3 Calibration to caplet prices

We choose to specify the instantaneous volatility of the two-factor separable LIBOR market model such that vanilla caplets are priced correctly under our model (we shall assume these are given by the Black formula). This is achieved by allowing σ_t to be piecewise constant.

To begin, observe that (20) is of separable form (6) with

$$\begin{aligned}\nu_j^1 &= K_1 e^{-\eta_1 T_j} \\ \nu_j^2 &= K_2 e^{-\eta_2 T_j},\end{aligned}$$

for $j = 1, \ldots, n$. Define

$$\begin{aligned}\psi_j^{11} &:= \int_{T_{j-1}}^{T_j} e^{2\eta_1 u} \mathrm{d}u = \frac{1}{2\eta_1}(e^{2\eta_1 T_j} - e^{2\eta_1 T_{j-1}}) \\ \psi_j^{12} &:= \int_{T_{j-1}}^{T_j} \rho e^{(\eta_1+\eta_2)u} \mathrm{d}u = \frac{\rho}{\eta_1+\eta_2}(e^{(\eta_1+\eta_2)T_j} - e^{(\eta_1+\eta_2)T_j - 1}) \\ \psi_j^{22} &:= \int_{T_{j-1}}^{T_j} e^{2\eta_2 u} \mathrm{d}u = \frac{1}{2\eta_2}(e^{2\eta_2 T_j} - e^{2\eta_2 T_{j-1}}).\end{aligned}$$

and suppose that for $j = 1, \ldots, n$

$$\sigma_t := \zeta_j,$$

on each interval $(T_{j-1}, T_j]$, for some constants ζ_j. As our volatility specification (20) is separable, the quadratic variation of u^i at T_i may be written in the form (9)

$$[u^i]_{T_i} = (\nu_i^1)^2 \xi_{0,i}^{11} + 2\nu_i^1 \nu_i^2 \xi_{0,i}^{12} + (\nu_i^2)^2 \xi_{0,i}^{22}, \tag{22}$$

where

$$\begin{aligned}\xi_{0,i}^{11} &= \sum_{j=1}^{i} \zeta_j^2 \psi_j^{11} \\ \xi_{0,i}^{12} &= \sum_{j=1}^{i} \zeta_j^2 \psi_j^{12} \\ \xi_{0,i}^{22} &= \sum_{j=1}^{i} \zeta_j^2 \psi_j^{22}.\end{aligned}$$

The constants ζ_i may be determined iteratively for $i = 1, \ldots, n$ by ensuring that $[u^i]_{T_i} = (\widetilde{\sigma}^i)^2 T_i$ to match the i^{th} caplet implied volatility $\widetilde{\sigma}^i$. For $i = 1$, it is clear from (22) that

$$\zeta_1 = \sqrt{\frac{(\widetilde{\sigma}^1)^2 T_1}{(\nu_1^1)^2 \psi_1^{11} + 2\nu_1^1 \nu_1^2 \psi_1^{12} + (\nu_1^2)^2 \psi_1^{22}}}.$$

For a given $i > 1$, suppose the constants ζ_j have already been determined for all $1 \leq j \leq i-1$ to be consistent with the first $(i-1)^{\mathrm{th}}$ caplet implied volatilities. Then $\xi_{0,i-1}^{11}, \xi_{0,i-1}^{12}$ and $\xi_{0,i-1}^{22}$ may be computed, and by rearranging (22),

$$\zeta_i = \sqrt{\frac{(\widetilde{\sigma}^i)^2 T_i - ((\nu_i^1)^2 \xi_{0,i-1}^{11} + 2\nu_i^1 \nu_i^2 \xi_{0,i-1}^{12} + (\nu_i^2)^2 \xi_{0,i-1}^{22})}{(\nu_i^1)^2 \psi_i^{11} + 2\nu_i^1 \nu_i^2 \psi_i^{12} + (\nu_i^2)^2 \psi_i^{22}}}.$$

Acknowledgement: I would like to thank Michael Bennett for his original work on the two-factor LIBOR Markov-functional model. A major part of this document is based on Mike's PhD thesis at the University of Warwick.

References

Brace, A., Gatarek, D., & Musiela, M. (1997). The market model of interest rate dynamics. *Mathematical Finance*, *7*(2), 127-155.

Hunt, P. J., & Kennedy, J. E. (2004). *Financial derivatives in theory and practice.* Revised edition, Wiley.

Pietersz, R., Pelsser, A., & Regenmortel, M. (2004). Fast drift approximated pricing in the bgm model. *Journal of Computational Finance*, *8*(1), 93-124.

A Brownian Bridge pre-model

In this appendix, we describe the Brownian bridge drift approximation (Pietersz, Pelsser, & Regenmortel, 2004) that may be used to construct a tractable two-dimensional "drift approximation" model that has similar behaviour to the two-factor separable LIBOR market model described in section 2.2. This is a simple extension to the one-factor case. Note that this approximation can then be used as a pre-model for the two-factor LIBOR Markov-functional model. As in the one-factor case, (Pietersz et al., 2004) show that the approximated LIBORs can be written in terms of a two-dimensional driving Markov process if the market model has a separable volatility structure. This is the case we consider here.

To begin, a method based on a Brownian bridge may be used to approximate the drift of each of the still-alive LIBORs from time zero to a given exercise date T_k. As the drift of the n$^{\text{th}}$ LIBOR is zero, $L^n_{T_k}$ is immediate given the value of $u^n_{T_k}$. Working back recursively from the n$^{\text{th}}$ LIBOR down to the first, suppose that for a given $i < n$ we already have approximations for $L^{i+1}_{T_k}, \dots, L^n_{T_k}$ and we wish to estimate $L^i_{T_k}$. Rewriting equation (7) using (2),

$$L^i_{T_k} = L^i_0 \exp\left(H^i_{T_k} - \frac{1}{2}[u^i]_{T_k} + u^i_{T_k} \right), \tag{23}$$

where

$$H^i_{T_k} := -\int_0^{T_k} \sum_{j=i+1}^{n} \frac{\alpha_j L^j_s}{1+\alpha_j L^j_s} s^{ij}_s \,\mathrm{d}s, \tag{24}$$

and s^{ij}_s is given by (3). It is clear that $L^i_{T_k}$ may be estimated given the value of $u^i_{T_k}$ if we have an approximation for $H^i_{T_k}$. If each $L^j_{T_k} (j > i)$ has already been estimated, then the value of L^j_s for any $s \in (0, T_k)$ may be approximated by the mean at time s of the generalized geometric Brownian bridge that joins L^j_0 and $L^j_{T_k}$. This is given by

$$m_j(s) = L^j_0 \left(\frac{L^j_{T_k}}{L^j_0} \right)^{\frac{[u^j]_s}{[u^j]_{T_k}}} \exp\left\{ \left(\frac{[u^j]_s}{2[u^j]_{T_k}} \right) \left([u^j]_{T_k} - [u^j]_s \right) \right\}. \tag{25}$$

See Appendix A of (Pietersz et al., 2004) for details. The approximation for $H^i_{T_k}$ is computed by substituting this approximation for all terms $L^j_s (j > i)$ appearing in the integrand of (24) and evaluating the integral numerically.

It is clear that any LIBOR $L^i_{T_k}$ may be written in terms of the two-dimensional Markov process x_{T_k} since it is a function of $H^i_{T_k}$ and $u^i_{T_k}$, and $H^i_{T_k}$ is in turn a function of all LIBORs $L^j_{T_k} (j > i)$, that is, it is a function of $u^j_{T_k}, (j > i)$. As in the one-dimensional case, the application of drift approximations to a two-factor separable LIBOR market model leads to efficient implementation, though of course the use of approximations introduces arbitrage.

B Implementation using a general pre-model

When the pre-model $\tilde{L}^i_{T_i}$ cannot be written as a function of $u^i_{T_i}$, the numerical evaluation of $J^i_0(x^*)$ (11) is more involved since it is no longer straightforward to rewrite the two-dimensional integral such that the integration region is rectangular (recall the integration region corresponds to the area where $\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i}) > x^*$). One possible solution is to fix $v^i_{T_i}$ and integrate over $u^i_{T_i}$ first, in which case the second integral over $v^i_{T_i}$ will be against a smooth function.

Explicitly, it is possible to evaluate

$$J^i_0(x^*) = D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}} \left[\mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v^i_{T_i}) \mathbf{1}_{\{\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i}) > x^*\}} | v^i_{T_i} \right] \right], \tag{26}$$

as follows. In evaluating the inner expectation for each $v^i_{T_i}$, write

$$\mathbf{1}_{\{\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i}) > x^*\}} = \mathbf{1}_{\{u^i_{T_i} > u^*\}},$$

where $u^* = u^*(x^*, v^i_{T_i})$ solves

$$\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i}) = x^*,$$

for this particular value of $v^i_{T_i}$. In solving for u^*, $\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i})$ may be approximated (for fixed $v^i_{T_i}$) as a polynomial in $u^i_{T_i}$ and a standard root-finding algorithm may be applied.

The inner expectation of (26) is given by

$$\begin{aligned} \tilde{g}(v^i_{T_i}) &= \mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v^i_{T_i}) \mathbf{1}_{\{\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i}) > x^*\}} | v^i_{T_i} \right] \\ &= \mathbb{E}_{\mathbb{F}} \left[\hat{D}_{T_i T_{i+1}}(u^i_{T_i}, v^i_{T_i}) \mathbf{1}_{\{u^i_{T_i} > u^*(x^*, v^i_{T_i})\}} | v^i_{T_i} \right] \\ &= \int_{u^*(x^*, v^i_{T_i})}^{\infty} \hat{D}_{T_i T_{i+1}}(u, v^i_{T_i}) \phi_{u^i_{T_i}}(u) \mathrm{d}u, \end{aligned}$$

where as before $\phi_{u^i_{T_i}}$ is the density of $u^i_{T_i}$. In practice, $\tilde{g}$ will be a smooth function of $v^i_{T_i}$ and it is therefore straightforward to evaluate

$$\begin{aligned} J^i_0(x^*) &= D_{0T_{n+1}} \mathbb{E}_{\mathbb{F}}[\tilde{g}(v^i_{T_i})] \\ &= D_{0T_{n+1}} \int_{-\infty}^{\infty} \tilde{g}(v) \phi_{v^i_{T_i}}(v) \mathrm{d}v. \end{aligned}$$

To improve the efficiency of the implementation, notice that it is only necessary to evaluate J^i_0 on a grid of values of x^*, since J^i_0 may be approximated at other points using a suitable interpolation (notice that $J^i_0(.)$ will be a smooth function, assuming the pre-model $\tilde{L}^i_{T_i}(u^i_{T_i}, v^i_{T_i})$ is also smooth).

C Other C++ files and their descriptions

In this appendix, we give a brief description of the main C++ files involved in the model.

- ImpliedVol: this class implements the implied volatility input for the model. We can set the caplet implied volatilities to be flat, increasing, decreasing in expiry or the actual market data.
- BermudanSwaption: this class implements the Bermudan swaption product which can be priced by the LIBOR Markov-functional model (with different pre-models), LIBOR Market model (full simulation or Brownian Bridge drift approximation).
- Interpolator: this class implements the interpolation of grid points to calculate the functional forms. There are two types of interpolation methods described here: Lagrange polynomial and cubic spline. It is recommended to use cubic spline as it gives better accuracy.
- NormalExpectation: this class implements the calculation of all different (conditional) expectations in this document.
- Schedule: this class implements the set up of the tenor structure for the construction of yield curve, implied volatilities, and Bermudan swaptions as well as the models.
- SimpsonIntegral: this class implements the calculation of deterministic integrals by the Simpson method (e.g. the (co)variances of the two components of the driving process at different exercise dates).