ACHIEVING DECORRELATION AND SPEED SIMULTANEOUSLY IN THE LIBOR MARKET MODEL

MARK S. JOSHI

Abstract. An algorithm for computing the drift in the LIBOR market model with additional idiosynchratic terms is introduced. This algorithm achieves a computational complexity of order equal to the number of common factors times the number of rates. It is demonstrated that this allows better matching of correlation matrices in reduced-factor models.

1. Introduction

The LIBOR market model is a popular model for pricing exotic interest rate derivatives as it allows easy calibration to market observable prices of vanilla derivatives including caplets and swaptions. There is, however, a perception in some parts of the market that it is too slow, and for this reason it is often not used or used with a small number of driving factors. This then forces rates to be highly correlated with each other. In this note, we see how to achieve an arbitrary amount of decorrelation between rates whilst retaining the speed given by using a low-factor model.

We recall the basics of the LIBOR market model. For more detail see any of [1], [2], [3] or [5]. The basis of the model is that forward interest rates are taken to be the state variables, and are assumed to be log-normal, (or some similar process in extensions.) We fix notation. We are evolving a set of forward rates associated to a set of times $0 < t_0 < t_1 < t_2 < \cdots < t_n$. We let $f_j(s)$ denote the forward rate from time t_j to time t_{j+1}, as observed at time s. We set $\tau_k = t_{k+1} - t_k$. We denote the zero-coupon bond maturing at t_j by $P_j(s)$. The volatility of the rate f_j is denoted by $\sigma_j(s)$.

Martingale pricing is based on picking a measure in which discounted asset price processes are martingales. Discounting is done by picking a fixed zero-coupon bond, P_N, as numeraire. We therefore have that for any asset O, that O/P_N is a martingale and therefore has zero drift. However, forward rates are not assets, nor necessarily the ratio

Date: January 11, 2006.

Key words and phrases. LIBOR market model, low factor, efficiency.

of an asset to the numeraire. This means that their drifts can be, and generally are, non-zero.

The drifts are not just non-zero but state-dependent, and are given by quite complicated expresssions. As these computations are state-dependent, these must be done at every step of a Monte Carlo simulation, and they can easily become a computational bottleneck.

The model supposes that in the pricing measure

$$\frac{df_j}{f_j} = \mu_j dt + \sigma_j(s)dt, \tag{1.1}$$

where μ_j is possibly a function of time and the rates. These rates are then evolved across a number, typically n, of time steps.

If we take the martingale measure associated to the numeraire P_n, then we have the following drifts expressions,

$$\mu_j = -\sum_{k>j} \tilde{C}_{jk} \frac{f_k \tau_k}{1 + f_k \tau_k}, \tag{1.2}$$

where $\tilde{C}$ is the instantaneous covariance between rate j and rate k.

In what follows for simplicity we assume that $N = n$. However, our approach is not restricted to that case. When evolving the rates from time s to time t, we perform a deterministic integral across the time step to obtain the covariance matrix, C, of the logs of the rates, and set

$$C_{jk}(s,t) = \int_s^t \tilde{C}_{jk}(r)dr. \tag{1.3}$$

This yields an approximation to the total drift across the step

$$\mu_j = -\sum_{k>j} C_{jk} \frac{f_k \tau_k}{1 + f_k \tau_k}. \tag{1.4}$$

Note that if $j = n - 1$, the sum is empty, and we take the value to be zero. In practical terms, it is only the quantities C_{jk} that affect the dynamics of the implemented model.

It was shown in [4] that if the covariance matrix is of rank F, all the terms μ_j for a single step could be computed with order nF computations. The other main computation for each step is the effect of Brownian increments. If we write

$$C = AA^t$$

with A an $n \times F$ matrix, then we need to compute

$$\log f_j(t) = \log f_j(s) + \mu_j - 0.5C_{jj} + \sum_{k=1}^{F} a_{jk} Z_k, \tag{1.5}$$

which will require order F computations for each j and is therefore order nF. Various additional book-keeping computations will need to be performed but these vary linearly with the number of rates and will not be affected by the number of factors. This means that the computational order of a single step in the LIBOR market model is nF. (This is, of course, an upper bound but it seems unlikely that one could do better without making additional assumptions.)

We will often want to do n steps and our computational complexity will therefore be n^2F. For the LIBOR market, it is normal for a one-factor model to mean that the covariance matrix is of rank 1, and so has a pseudo-square root of size $n \times 1$, whereas in the credit literature, see, e.g., [6], it is taken to mean that there is one *common* factor. It is our purpose here to demonstrate that a model with F common factors can be implemented to run almost as quickly as a model with F total factors. In particular, we show that the computational order is still nF per step. The key observation is that drift arises from covariance between rates and so adding in idiosynchratic factors will not affect the drifts.

We discuss the model's set-up in Section 2. We examine the drift computation in Section 3. We look at how better approximations to correlation matrices can be obtained in Section 4. We conclude in Section 5.

2. The model set-up

We use the LIBOR market model as described in the introduction but assume that each rate is driven by its exposure to F common factors and one idiosynchratic factor. There are a number of ways to express this. The usual way to do so is to write

$$\frac{df_j}{f_j} = \mu_j(f,t)dt + \sum_{k=1}^{F} \sigma_{jk}(t) dW_t^{(k)} + \sigma_j(t) dZ_t^{(j)}, \tag{2.1}$$

with the Brownian motions $W_t^{(k)}$ and $Z_t^{(j)}$ all independent. The term σ_j expresses exposure to the idiosynchratic factor and the σ_{jk} to the common factors.

When evolving in a computer, we will do discrete time-steps and it will therefore generally be more useful to adapt equation (1.5). Our discretized equation is

$$\log f_j(t) = \log f_j(s) + \mu_j - 0.5C_{jj} + \sum_{k=1}^{F} a_{jk}W_k + a_j Z_j, \tag{2.2}$$

with W_k and Z_j all independent standard $N(0,1)$ random variables. Given the drifts μ_j, the increment will clearly still take order F computations per rate, since we have only added on a single extra term.

Thus if we can compute the drifts with order nF computations, our entire step will take order nF computations.

3. Computing the drift

We define an $n \times (n+F)$ matrix, A, by

$$A_{jk} = \begin{cases} a_j & \text{for} \quad j = k, \\ 0 & \text{for} \quad j \neq k, k \leq n, \\ a_{j,k-n} & \text{for} \quad k > n, \end{cases} \tag{3.1}$$

with a_j and a_{jk} as in (1.5). We have that A is a pseudo-square root of the covariance matrix. The drift μ_j is therefore given by

$$\mu_j = -\sum_{k>j} \sum_{r=1}^{n+F} A_{jr} A_{kr} g_k, \tag{3.2}$$

where

$$g_k = \frac{f_k \tau_k}{1 + f_k \tau_k}.$$

Since A_{ij} is zero for $i \neq j$ and $j \leq n$, we have

$$\mu_j = -\sum_{k>j} \sum_{r=n+1}^{n+F} A_{jr} A_{kr} g_k.$$

Changing the order of summation,

$$\mu_j = -\sum_{r=n+1}^{n+F} A_{jr} \sum_{k>j} A_{kr} g_k.$$

Let

$$e_{j,r} = \sum_{k>j} A_{kr} g_k. \tag{3.3}$$

We then have the relations

$$e_{j,r} = e_{j+1,r} + A_{j+1,r} g_{j+1}, \tag{3.4}$$

1.0000	0.9094	0.8352	0.7744	0.7247	0.6839	0.6506	0.6233
0.9094	1.0000	0.9094	0.8352	0.7744	0.7247	0.6839	0.6506
0.8352	0.9094	1.0000	0.9094	0.8352	0.7744	0.7247	0.6839
0.7744	0.8352	0.9094	1.0000	0.9094	0.8352	0.7744	0.7247
0.7247	0.7744	0.8352	0.9094	1.0000	0.9094	0.8352	0.7744
0.6839	0.7247	0.7744	0.8352	0.9094	1.0000	0.9094	0.8352
0.6506	0.6839	0.7247	0.7744	0.8352	0.9094	1.0000	0.9094
0.6233	0.6506	0.6839	0.7247	0.7744	0.8352	0.9094	1.0000

TABLE 1. The correlation matrix before truncation.

and

$$\mu_j = -\sum_{r=n+1}^{n+F} A_{jr} e_{j,r}. \tag{3.5}$$

The number of additional operations for each j is therefore of order F. We conclude that the total number of operations is of order nF.

4. Correlation Matrices

We have seen that we can add an idiosynchratic factor to each rate in the LIBOR market at a small additional cost. What sort of correlation matrices are now possible with small factors? In a one-common-factor model, the simplest example is to take every rate to have exposure to the common factor of magnitude $\sqrt{\rho}$, and obtain a correlation matrix which is constant off the diagonal with correlation ρ. For $\rho < 1$, this matrix will be of rank n and therefore be of order n^2 to compute if traditional methods are used.

We could also take a one-factor model where the exposure to the common factor varies with the index. Rate i would have exposure a_i, and the correlation between rates i and j will be $a_i a_j$ if $i \neq j$. This could allow us to have rates with greater time to maturity to be more highly correlated than those close to expiry.

More generally, given a correlation matrix we can perform factor reduction on it and then add use the idiosynchratic terms to reduce the correlations to make the new correlation matrix closer to the original one. We do a worked example. Suppose rate j runs from time $2j$ to $2(j+1)$, with $j = 1, \dots, 7$, and the correlation between rates i and j is

$$(1 - L)e^{-\beta|t_i - t_j|} + L$$

where $L = 0.5$ and $\beta = 0.1$. (We are taking forward rates with larger than usual tenor to stop the example becoming unwieldy.) The correlation matrix is then as in Table 1. We work with a two-factor model

1.0000	0.9993	0.9887	0.9471	0.8632	0.7611	0.6826	0.6554
0.9993	1.0000	0.9935	0.9582	0.8810	0.7843	0.7088	0.6826
0.9887	0.9935	1.0000	0.9845	0.9290	0.8496	0.7843	0.7611
0.9471	0.9582	0.9845	1.0000	0.9796	0.9290	0.8810	0.8632
0.8632	0.8810	0.9290	0.9796	1.0000	0.9845	0.9582	0.9471
0.7611	0.7843	0.8496	0.9290	0.9845	1.0000	0.9935	0.9887
0.6826	0.7088	0.7843	0.8810	0.9582	0.9935	1.0000	0.9993
0.6554	0.6826	0.7611	0.8632	0.9471	0.9887	0.9993	1.0000

TABLE 2. The correlation matrix after two-factor truncation.

1.0000	0.8920	0.8816	0.8427	0.7680	0.6786	0.6093	0.5831
0.8920	1.0000	0.8888	0.8554	0.7865	0.7016	0.6348	0.6093
0.8816	0.8888	1.0000	0.8779	0.8285	0.7592	0.7016	0.6786
0.8427	0.8554	0.8779	1.0000	0.8717	0.8285	0.7865	0.7680
0.7680	0.7865	0.8285	0.8717	1.0000	0.8779	0.8554	0.8427
0.6786	0.7016	0.7592	0.8285	0.8779	1.0000	0.8888	0.8816
0.6093	0.6348	0.7016	0.7865	0.8554	0.8888	1.0000	0.8920
0.5831	0.6093	0.6786	0.7680	0.8427	0.8816	0.8920	1.0000

TABLE 3. The correlation matrix after two-factor truncation with idiosynchratic factors.

and obtain a correlation matrix by taking the two largest eigenvalues. We then obtain the correlation matrix in Table 2. Not surprisingly, the correlation numbers have increased and the maximum difference is 0.1723. The sum of squares error is 0.5815.

We can reduce the difference by introducing idiosynchratic factors. A quick numerical search shows that a weight of 0.1098 for all rates minimizes the sum of squares error. The new correlation matrix is shown in Table 3. Our errors now range from -0.0683 to 0.0492, and the sum of squares error is 0.0645. We have achieved a correlation matrix much closer to the original with little extra computational cost.

5. Conclusion

We have shown that by introducing idiosynchratic rates in the LIBOR market model, greater decorrelation can be achieved between rates without sacrificing speed. In particular, we have achieved an algorithm of order nF where n is number of rates and F is the number of common factors.

References

[1] A. Brace, D. Gatarek, M. Musiela, The market model of interest-rate dynamics, *Mathematical Finance* **7**, 127–155, 1997
[2] D. Brigo, F. Mercurio, *Interest Rate Models – Theory and Practice*, Springer Verlag, 2001
[3] M. Joshi, The concepts and practice of mathematical finance, Cambridge University Press 2003
[4] M. Joshi, Rapid Drift Computations in the LIBOR market model, *Wilmott*, May 2003
[5] M. Musiela, M. Rutowski, *Martingale Methods in Financial Modelling*, Springer Verlag, 1997.
[6] P.J. Schönbucher, Credit derivatives pricing models, Wiley 2003

Centre for Actuarial Studies, Department of Economics, University of Melbourne, Victoria 3010, Australia
E-mail address: `mark@markjoshi.com`