A major Libor fit

by Raoul Pietersz and Patrick Groenen

Interest rate derivatives form a key part of the derivatives industry. Valuation of interest rate derivatives is thus an important topic. European-style options on Libor (caplets and floorlets) and swap rates (swaptions) are valued with Black's formula. Input to this formula is the current level and implied volatility of a single interest rate, be it the forward Libor or swap rates. A more challenging situation arises when derivatives need to be valued that depend on the joint and inter-temporal distribution of several interest rates. Examples of such multi-rate dependent derivatives include ratchet caps, Bermudan swaptions and callable range accrual swaps. The interdependence of interest rates can be modelled, to first-order approximation, by their correlation. Short-rate models feature correlation that depends in a complicated manner on the model parameters. Therefore, calibration to correlation requires a specific indirect approach per short rate model. For a natural and more controllable correlation-calibration, and for other reasons, research has focused on the Libor market model of Brace, Gatarek & Musiela (1997).[1] This model allows the specification of any forward Libor correlation matrix. The hard part of the work, say 80%, of equipping a Libor production model with correlation, consists of estimating this correlation matrix in a stable and meaningful manner from historical or market-implied data. In this article, we address the remaining 20%, which is fitting a low-factor model to the resulting high-rank correlation matrix. The reason for the low number of factors is the gain in computational speed. Moreover, usually a large part of the variance of typical Libor correlation matrices can already be attributed to a low number of factors. Here, we develop a novel algorithm, based on majorisation, to tackle this low-rank approximation problem.

The number of factors driving the model is denoted by d, usually $d \ll n$, with n denoting the number of forward rates. Forward rate l_i satisfies

$$\frac{dl_i}{l_i} = \ldots dt + \sigma_i \left\{ x_{i1} dw_1 + \ldots + x_{id} dw_d \right\}, \qquad \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle = r_{ij} \tag{1}$$

Here, σ_i denotes the volatility of l_i, $\mathbf{X}$ is an $n \times d$ matrix distributing the volatility over the d factors with $\mathbf{x}_i$ denoting row i of $\mathbf{X}$, the w_j denote Brownian motions, and $\mathbf{R}$ is the estimated $n \times n$ forward rates instantaneous correlation matrix. The relation linking $\mathbf{X}$ and $\mathbf{R}$ is $\mathbf{XX}^T = \mathbf{R}$, which implies that $\mathbf{R}$ has rank d or less. For arbitrarily given $\mathbf{R}$, this rank restriction will generally not be satisfied. Therefore, an approximation is required, either of correlation or covariance. When pricing derivatives, we are more certain of volatility, since it is quoted in the market. Therefore, we fit variance exactly and approximate correlation. We are then led to solve the following optimisation problem. Find an $n \times d$ matrix $\mathbf{X}$ to minimise:

$$f(X) := \sum_{i<j} \omega_{ij} \left(r_{ij} - \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \right)^2 \tag{2}$$

subject to $\|\mathbf{x}_i\|_2 = 1$, where $\|\mathbf{x}_i\|_2^2 = \Sigma_{j=1}^d x_{ij}^2$. The ω_{ij} are non-negative weights, given by the user to set the importance of correlation r_{ij}.

A well-known straightforward technique of obtaining a low-rank correlation matrix $\mathbf{XX}^T$, is the modified principal component analysis (PCA) approach, for example, Hull & White (2000).[2] The method takes the d largest eigenvalues from an eigenvalue decomposition, and then rescales (this is the modified part) the resulting matrix $\mathbf{X}_{\text{PCA}}$ to ensure $\mathbf{X}_{\text{PCA}}\mathbf{X}_{\text{PCA}}^T$ has unit diagonal. A strong drawback of modified PCA is its non-optimality. In general, one may find low-rank matrices $\mathbf{XX}^T$ closer to $\mathbf{R}$ than the modified-PCA matrix $\mathbf{X}_{\text{PCA}}\mathbf{X}_{\text{PCA}}^T$, even in the neighbourhood of the latter. Therefore, optimisation methods have been developed to minimise the objective function $f(\mathbf{X})$ of (2).

Majorisation is a general technique from optimisation. The main idea is briefly described here. Let $g(\cdot; \mathbf{Y})$ be an auxiliary function at a given point $\mathbf{Y}$, such that $f(\mathbf{Y}) = g(\mathbf{Y}; \mathbf{Y})$; $f(\mathbf{X}) \le g(\mathbf{X}; \mathbf{Y}), \forall \mathbf{X}$; and $g(\cdot; \mathbf{Y})$ is 'simple', that is, it is straightforward to find the minimum of $g(\cdot; \mathbf{Y})$.

Then, starting at a point $\mathbf{Y} = \mathbf{X}_0$, we find the minimum $\mathbf{X}_1$ of the auxiliary function $g(\mathbf{X}; \mathbf{X}_0)$. Then, we set $\mathbf{Y} = \mathbf{X}_1$, find the minimum $\mathbf{X}_2$ of the auxiliary function $g(\mathbf{X}; \mathbf{X}_1)$, and so on. This procedure guarantees that the objective function value does not increase over the points produced by the algorithm, since:

$$f\left(\mathbf{X}_{k+1}\right) \le g\left(\mathbf{X}_{k+1}; \mathbf{X}_k\right) \overset{(*)}{\le} g\left(\mathbf{X}_k; \mathbf{X}_k\right) = f\left(\mathbf{X}_k\right) \tag{3}$$

For low-rank approximation majorisation, Pietersz & Groenen (2004)[3] show that inequality (*) is strict whenever $\mathbf{X}_k$ is not a local minimum. This fact guarantees that majorisation is globally convergent to a local minimum. The majorisation algorithm for (2) can be derived as follows. Essentially, quadratic problem (2) is tackled by splitting it up into a sequence of linear problems. We majorise the objective function $f(\cdot)$ per row, that is, as a function of the ith row of $\mathbf{X}$. If we denote by $\mathbf{x}_i$ and $\mathbf{y}$ the running and current ith row of $\mathbf{X}$, respectively, then it is shown in Pietersz & Groenen (2004), that:

$$f\left(\mathbf{x}_i; \mathbf{X}\right) \le -2\mathbf{x}_i^T \left[\lambda \mathbf{y} - \mathbf{B}\mathbf{y} + \sum_{j: j \ne i} \omega_{ij} r_{ij} \mathbf{x}_j \right] + \left(const\ in\ \mathbf{x}_i \right) =: g\left(\mathbf{x}_i; \mathbf{X}\right) \tag{4}$$

with the $d \times d$ matrix:

$$\mathbf{B} = \sum_{j: j \ne i} \omega_{ij} \mathbf{x}_j \mathbf{x}_j^T$$

and with λ the largest eigenvalue of $\mathbf{B}$. The minimum of $g(\cdot; \mathbf{X})$ is $\mathbf{x}_i^*$, defined by:

$$\mathbf{z} = \lambda \mathbf{y} - \mathbf{B}\mathbf{y} + \sum_{j: j \ne i} \omega_{ij} r_{ij} \mathbf{x}_j, \qquad \mathbf{x}_i^* = \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \tag{5}$$

If $\mathbf{z} = 0$, then it may be shown that $\mathbf{X}$ is already a local minimum with respect to the ith row, and then $\mathbf{x}_i^* = \mathbf{y}$. The resulting majorisation algorithm is displayed below.

■ Input $\mathbf{R}$, Ω, d, ε. Take $\mathbf{X}$ as the modified-PCA solution $\mathbf{X}_{\text{PCA}}$. For $k = 1, 2, \ldots$. For $i = 1, \ldots, n$:

$$\mathbf{B} := \sum_{j \ne i} \omega_{ij} \mathbf{x}_j \mathbf{x}_j^T$$

Calculate λ as the largest eigenvalue of $\mathbf{B}$. Calculate $\mathbf{z}$ as:

$$\mathbf{z} := \lambda \mathbf{x}_i - \mathbf{B}\mathbf{x}_i + \sum_{j \ne i} \omega_{ij} r_{ij} \mathbf{x}_j$$

If $\mathbf{z} \ne \mathbf{0}$, set $\mathbf{x}_i$ equal to $\mathbf{z}/\|\mathbf{z}\|_2$. End for. Stop if the function-value improvement $f_{k-1}/f_k - 1$ is less than ε. End for. Output: $\mathbf{XX}^T$ is a rank-d approximation of $\mathbf{R}$.

The majorisation algorithm has been implemented in a Matlab package called major, downloadable from *www.few.eur.nl/few/people/pietersz*.

Introducing a novel algorithm is only justified if it is more efficient than existing algorithms. Pietersz & Groenen (2004) compare majorisation numerically with the algorithms available in the literature, with a benchmark of the accuracy of fit given a computational time constraint. The limit on computational time is close to financial practice, since decisions based on derivatives pricing calculations need to be made often within seconds. The results show that majorisation compares favourably with the other algorithms. In fact, in all cases considered, majorisation is the most efficient algorithm. ■

Raoul Pietersz is a PhD student at Erasmus University Rotterdam and a senior derivatives researcher at ABN Amro in Amsterdam. Patrick Groenen is a professor in statistics at Erasmus University Rotterdam. The views expressed in this article do not necessarily reflect those of ABN Amro

[1] *A Brace, D Gatarek & M Musiela, 1997, The market model of interest rate dynamics, Mathematical Finance 7, pages 127–155*

[2] *J Hull & A White, 2000, Forward rate volatilities, swap rate volatilities, and implementation of the Libor market model, Journal of Fixed Income 3, pages 46–62*

[3] *R Pietersz & P Groenen, 2004, Rank reduction of correlation matrices by majorization, forthcoming in Quantitative Finance*