1 Calibrating the Correlation Structure

Before discussing the calibration of the LMM-SABR model to the target correlation function it is important to distinguish clearly between the input and the model correlation matrices.

If N is the number of forward rates in the problem, the input correlation matrix is the full-rank $2N \times 2N$ real symmetric matrix that we can in principle estimate using statistical techniques, or from correlation-dependent market prices. If we could afford a $2N$-factor model, this is the matrix ideally we would like to work with. Then there is the model correlation matrix. This is the matrix, typically of lower rank, that our LMM-SABR model sees. Only rarely will the two be the same.

With the deterministic-volatility LIBOR market model, we do not normally worry too much about the input matrix (this may be a pity, especially for products like CMS spread options, but this is another story). When it comes to the LMM-SABR model, however, matters are different. As we mentioned before, creating a valid (ie, positive definite) input correlation matrix of full dimensionality is not a trivial task. **This is because N of the $N(N-1)/2$ elements are exogenously fixed by the SABR market-given correlations** between the each forward rate and its own volatility. As we show below, this may require some compromises to be made if we want our input matrix to be positive definite (ie, a *bona fide* correlation matrix.)

When we implement the LMM-SABR model, we will probably want to **reduce the number of factors** from $2N$ to a more manageable number. This will introduce a **further distortion** to the correlation matrix, ie, a transformation from the input matrix to the model matrix. This is simply due to the fact

that the fewer factors we retain, the fewer Fourier components we will be able to recover in our model correlation matrix. These problems are well known, and have been discussed at length in Rebonato and Cooper (1995), Rebonato (2002) and Rebonato (2004), among others. **In the deterministic-volatility LIBOR market model this is the only 'compromise' one should worry about. We stress that this is no longer the case with the LMM-SABR model. It is important to keep the different sources of 'corruption' conceptually distinct.**

2 Creating a Valid Model Matrix

For N forward rates, the full correlation matrix, $\boldsymbol{P}$, is a $2N \times 2N$ real symmetric matrix defined as

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{\rho} & \boldsymbol{R} \\ \boldsymbol{R}^{\dagger} & \boldsymbol{r} \end{pmatrix} \tag{1}$$

where $\boldsymbol{\rho}$ is the forward-rate/forward-rate correlation matrix, $\boldsymbol{r}$ is the volatility-volatility correlation matrix and $\boldsymbol{R}$ is the forward-rate/volatility matrix*. The only elements that affect the caplet prices are the diagonal elements of the sub-matrix $\boldsymbol{R}$ - these are the ρ_{SABR} parameters of the SABR model. Any scheme to reduce the number of factors must preserve the diagonal elements, $R_{ii} = \rho^{i}_{\text{SABR}}$, of $\boldsymbol{R}$.

*This is not a standard correlation matrix since the correlation between the i^{th} forward rate and the j^{th} volatility is not necessarily the same as the correlation between the j^{th} forward rate and the i^{th} volatility.

Recall 4 that, neglecting the drifts, the equations of motions can be written for N forward rates and volatilities as:

$$\frac{df_t^i}{\left(f_t^i\right)^{q_i}} = s_t^i \sum_{j=1}^{M} e_{ij} dz_j, \ i = 1 \text{ to } N \tag{2}$$

$$ds_t^i = g(t, T_i) dk_t^i \tag{3}$$

$$\frac{dk_t^i}{k_t^i} = h_t^i \sum_{j=1}^{M} e_{N+i,j} dz_j, \qquad i = 1 \text{ to } N \tag{4}$$

with

$$M = N_F + N_V \tag{5}$$

for N_F forward-rate factors and N_V volatility factors, and

$$\mathbb{E}\left[dz_i dz_j\right] = \rho_{ij} dt \tag{6}$$

$$\sum_{j=1}^{M} e_{ij}^2 = 1 \tag{7}$$

In matrix form this can be rewritten as

$$\boldsymbol{dx} = \boldsymbol{q} \cdot \boldsymbol{E} \cdot \ \boldsymbol{dz} \tag{8}$$

where

- $\boldsymbol{dx}$ is a $[2N, 1]$ vector whose first N elements are given by $\frac{df_t^i}{\left(f_t^i\right)^{q_i}}$, and its last N elements are given by $\frac{dk_t^i}{k_t^i},\ i = 1, 2, .., N$;

- $\boldsymbol{q}$ is the $[2N, 2N]$ diagonal matrix whose first N diagonal elements are given by s_t^i and its last N diagonal elements are given by h_t^i, $i = 1, 2, .., N$;

- $\boldsymbol{E}$ is a $[2N, M]$ matrix of elements e_{ij} and of the stacked form

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{C} \end{bmatrix} \tag{9}$$

- $\boldsymbol{B}$ is the $[N, M]$ matrix with elements b_{ij}, $i = 1, 2, ..., N$ and $j = 1, 2, ..., M$ (and similarly for matrix $\boldsymbol{C}$ of elements c_{ij});

- $\boldsymbol{dz}$ is the $[M, 1]$ vector of elements dz_j

Of course, if $M = 2N$ then we have a full factor model and recover $\boldsymbol{P}$ exactly. This is no longer the case when $M < 2N$. In this case we want to determine

the 'mixing' matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ in such a way that they 'resemble' the $\boldsymbol{P}$ matrix (in some sense to be defined) as closely as possible:

$$\begin{aligned} \boldsymbol{B}\boldsymbol{B}^{\dagger} &\approx \boldsymbol{\rho} \\ \boldsymbol{C}\boldsymbol{C}^{\dagger} &\approx \boldsymbol{r} \\ \boldsymbol{B}\boldsymbol{C}^{\dagger} &\approx \boldsymbol{R} \end{aligned} \tag{10}$$

To do this, we treat the i^{th} row of $\boldsymbol{B}$ as the $[1 \times M]$ vector $\boldsymbol{b}^i$, and the i^{th} row of $\boldsymbol{C}$ as the $[1 \times M]$ vector $\boldsymbol{c}^i$. We have two constraints on these vectors:

- the normalization constraint:

$$\boldsymbol{b}^i \cdot \boldsymbol{b}^i = \sum_{j=1}^{M} b_{ij}^2 = \boldsymbol{c}^i \cdot \boldsymbol{c}^i = \sum_{j=1}^{M} b_{ij}^2 = 1 \quad \forall i \tag{11}$$

- the SABR correlation constraint

$$\boldsymbol{b}^i \cdot \boldsymbol{c}^i = R_{ii} \tag{12}$$

So b^i and c^i are unit vectors in an M-dimensional space.

There are many ways to solve this problem with the above constraints (and many 'optimal' solutions since we have not specified what we mean by 'approximate'). We present two solutions, based on different definitions of distance between two vectors. The first will recover the elements R_{ii} exactly; the second gets arbitrarily close to them, but in general recovers other portions of the $\boldsymbol{P}$ matrix better.

2.1 First Strategy, Stage 1: Diagonalise P

Since $\boldsymbol{P}$ is a real symmetric matrix it is easy and quick to diagonalise. We write

$$\boldsymbol{P} = \boldsymbol{\Lambda} \boldsymbol{D} \boldsymbol{\Lambda}^{\dagger} \tag{13}$$

where $\boldsymbol{D}$ is a diagonal matrix of the (real) eigenvalues of $\boldsymbol{P}$ and $\boldsymbol{\Lambda}$ is a matrix with the eigenvectors of $\boldsymbol{P}$ as its columns.

If all the eigenvalues are positive then $\boldsymbol{P}$ is positive definite (or positive semi-definite if some are zero). If, however, the matrix is not positive semi-definite **we can still proceed provide the number of positive eigenvalues is greater or equal to** M - this is a standard way of dealing with a non-positive semi-definite correlation matrix.

The vectors $\boldsymbol{b}^i$ and $\boldsymbol{c}^i$ are then defined as

$$b^i_j = B_{ij} = \frac{\Lambda_{ij} D_j^{1/2}}{\sum_{k=1}^{M} \Lambda_{ik} D_k^{1/2}} \tag{14}$$

$$c^i_j = C_{ij} = \frac{\Lambda_{i+N,j} D_j^{1/2}}{\sum_{k=1}^{M} \Lambda_{i+N,k} D_k^{1/2}} \tag{15}$$

These relationships preserve $\boldsymbol{b}^i$ and $\boldsymbol{c}^i$ as unit vectors, and satisfy Equation (10). In general, however, the constraint of Equation (12) will not be satisfied. We take care of this as follows.

2.2 First Strategy, Stage 2: Analytic Optimisation of c^i

Recall that the SABR constraint can be written as

$$\boldsymbol{b}^i \cdot \boldsymbol{c}^i = R_{ii} \tag{16}$$

We want to keep the vectors $\boldsymbol{b}^i$ fixed and adjust the vectors $\boldsymbol{c}^i$ – which we call $\widetilde{\boldsymbol{c}}^i$ – in turn in such a way that

1. the SABR constraint is satisfied;

2. the vectors $\widetilde{\boldsymbol{c}}^i$ remain of unit norm; and

3. the distance from the original $\mathbf{c}^i$ is minimized.

First of all note that the vectors $\boldsymbol{b}^i$ and $\widetilde{\boldsymbol{c}}^i$ define a plane (as long as they are not parallel). Think of placing the origin of the vectors $\widetilde{\boldsymbol{c}}^i$ to coincide with the origin of $\boldsymbol{b}^i$. **The SABR constraint (16) prescribes an angle, say R_{ii}, between b^i and c^i.** Let $\phi_i = \boldsymbol{b}^i \cdot \widetilde{\boldsymbol{c}}^i$ then be the (known) angle between $\boldsymbol{b}^i$ and $\widetilde{\boldsymbol{c}}^i$.

The vectors $\widetilde{\boldsymbol{c}}^i$ sweep a cone with vertex at the common origin of $\boldsymbol{b}^i$ and $\widetilde{\boldsymbol{c}}^i$. Therefore, in order to minimize the distance between and $\boldsymbol{c}^i$ and $\widetilde{\boldsymbol{c}}^i$, $\boldsymbol{c}^i$ must lie in the plane defined by $\boldsymbol{b}^i$ and $\widetilde{\boldsymbol{c}}^i$ and is found by the intersection of this plane with the cone described above. The vector $\mathbf{c}^i$ can therefore be written as a linear combination of the two (non-orthogonal) 'basis' vectors $\mathbf{b}^i$ and $\widetilde{\boldsymbol{c}}^i$:

$$\boldsymbol{c}^i = \lambda_1 \boldsymbol{b}^i + \lambda_2 \widetilde{\boldsymbol{c}}^i \tag{17}$$

What we do not know are the coefficients λ_1 and λ_2 in the expansion of the solution vector c^i. To find these quantities we proceed as follows.

Keeping in mind that the required vectors must have unit length we have

$$\boldsymbol{c}^i \cdot \boldsymbol{c}^i = \lambda_1^2 + \lambda_2^2 + 2\lambda_1\lambda_2\phi_i = 1 \tag{18}$$

$$\boldsymbol{b}^i \cdot \boldsymbol{c}^i = \boldsymbol{b}^i \cdot \left(\lambda_1 \boldsymbol{b}^i + \lambda_2 \widetilde{\boldsymbol{c}}^i\right) = \lambda_1 + \lambda_2\phi_i = R_{ii} \tag{19}$$

This gives

$$\lambda_2 = \sqrt{\frac{1 - R_{ii}^2}{1 - \phi_i^2}} \tag{20}$$

The value λ_2 can then be substituted in Equation (17) to solve for λ_1. The simple procedure can be repeated sequentially for each vector $\boldsymbol{c}^i$ to find its 'optimal' value.

There is some financial justification for the procedure, which treats the matrix $\boldsymbol{B}$ (which is related to the forward-rate/forward-rate portion of the correlation matrix $\boldsymbol{P}$) as more 'privileged' than the matrix $\boldsymbol{C}$. One can in fact reasonably

assume that the forward-rate/forward-rate matrix is better and more reliably known than the volatility/volatility or the volatility/forward-rate blocks. The exact recovery of their elements is therefore a less stringent requirement (apart from the SABR-related diagonal elements R_{ii}).

2.3 Second Strategy: Optimizing Over Angles

The second route to optimizing places different weights on the elements of the matrix $\boldsymbol{P}$ and structures the problem so as to carry out an unconstrained non-linear optimization. It is very similar to the strategy proposed by Rebonato (1999a) to calibrate a deterministic-volatility LMM to an exogenously assigned

correlation matrix. The intuition is very simple: consider the trigonometric relationship

$$\sin^2\theta + \cos^2\theta = 1 \tag{21}$$

which holds for any angle θ. If two quantities, say e_1 and e_2, are set equal to the sine and cosine terms above, the constraint on the sum of their squares, $e_1^2\theta + e_2^2\theta = 1$, (see Equation (11)) is automatically satisfied. The generalization to M variables is immediate: we simply have to use as optimization variables the $(M-1)$ angles that define the surface of a $M-1$-dimensional hypersphere[†] of unit radius (with one such set of angles for each variable – forward rate or volatility):

$$e_{ik} = \cos\theta_{ik}\Pi_{j=1}^{k-1}\sin\theta_{ij} \qquad k = 1, 2, ..., M-1 \tag{22}$$

[†]Mathematicians (and relativists) define such a surface as a S^{k-1} sphere, where k are the dimensions of the space in which the surface is embedded, and $k-1$ the number of *intrinsic* coordinates necessary to specify a point on the surface. So, the perimeter of a circle is S^1 sphere, and the surface of a 3d sphere a S^2 sphere. We do not use this terminology.

$$e_{ik} = \Pi_{j=1}^{k-1} \sin\theta_{ij} \qquad k = M \tag{23}$$

with $i = 1, 2, ..., 2N$.

The optimization is then achieved

- by starting from an initial inspired guess for the angles (or, failing that, from random variates drawn from the uniform $\mathcal{U}[0, 2\pi]$) distribution;

- by obtaining the trial correlation matrix by

$$\boldsymbol{P} = \boldsymbol{e}\boldsymbol{e}^{\dagger} \tag{24}$$

- by calculating the distance χ^2 between the trial ($\boldsymbol{P}$) and target ($\widetilde{\boldsymbol{P}}$) correlation matrices as, for example,

$$\chi^2 = \sum_{ij} w_{ij} \left(P_{ij} - \widetilde{P}_{ij} \right)^2 \tag{25}$$

- by minimizing this distance in an unconstrained manner over the angles.

The requirement that the elements R_{ii} are 'very important' can be translated by giving very high weights to the associated elements of $\widetilde{P}_{ij}$. Similarly, if we believe, as we normally do, that we can trust the forward-rate/forward-rate portion of the correlation matrix more than other parts, we can assign higher weights to these entries.

There is one numerical problem – what we call the Greenland-on-a-world-map effect. Its origin lies in the fact that the mapping from the quantities we want to

optimize over to the angles is very non-linear, and the original surface is locally very strongly deformed (much as Greenland is deformed on a two-dimensional map of the world). In other words, a uniform probability distribution over the angles is very non-uniform in the space of the variables we are interested in. So, minima in the original space can become extremely steep (and easy-to-miss) wells in angle space. The more so, the higher the dimensions of the sphere.

Doust (2007) has suggested an inspired guess to fix this problem. This is presented in the Appendix below. However, we show in the next section the result of a simple optimization using Equations (22) to (23) to a plausible and realistically-complex correlation surface. The outcome, as we shall see, is very satisfactory.

3 A Case Study: Calibration Using the Hypersphere Method

We study the case of ten forward rates and ten volatilities. We assume that the forward-rate/forward-rate block of the correlation matrix is given by a Doust-like correlation, displayed in Table {TestCorrelCalibr I} and Figures {TestCorrelCalibr A} and {TestCorrelCalibr B}. The volatility/forward-rate portion of the correlation is assumed to have market-realistic value of -0.2 along the main diagonal, and a constant value of -0.1 for all the other entries. As for the volatility/volatility block of the overall correlation matrix a Doust-like shape has been assumed, but with a much higher value for the overall level of correlation (between 85% and 100%). This is also market-plausible. Figure {TestCorrelCalibr C} shows the overall correlation matrix. It is useful to pause and

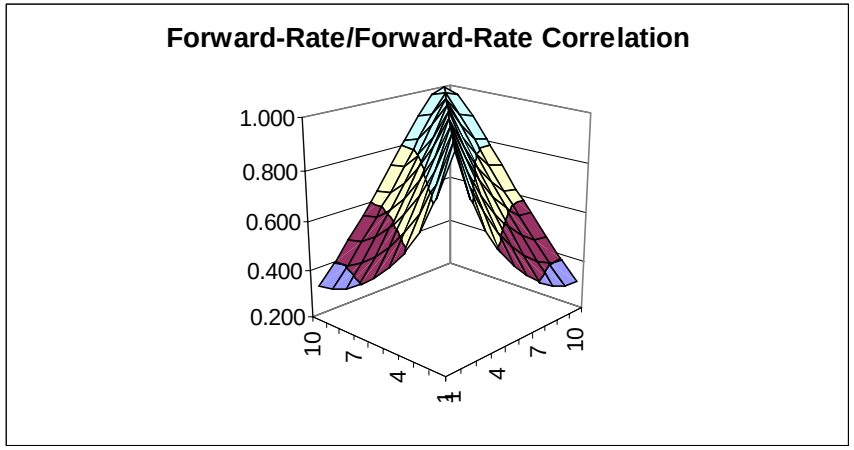
Forward-Rate/Forward-Rate Correlation
1.000
0.800
0.600
0.400
0.200
10
7
4
1
1
4
7
10

Figure 1:

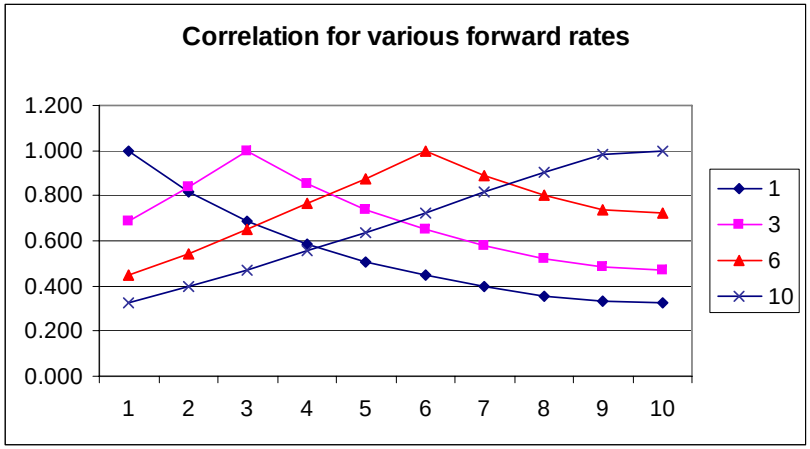

Figure 2:

familiarize oneself with its shape, because it certainly does not resemble any forward-rate/forward-rate correlation matrix the reader may be familiar with.

Figure caption for {TestCorrelCalibr A}: The forward-rate/forward-rate portion of the target correlation matrix

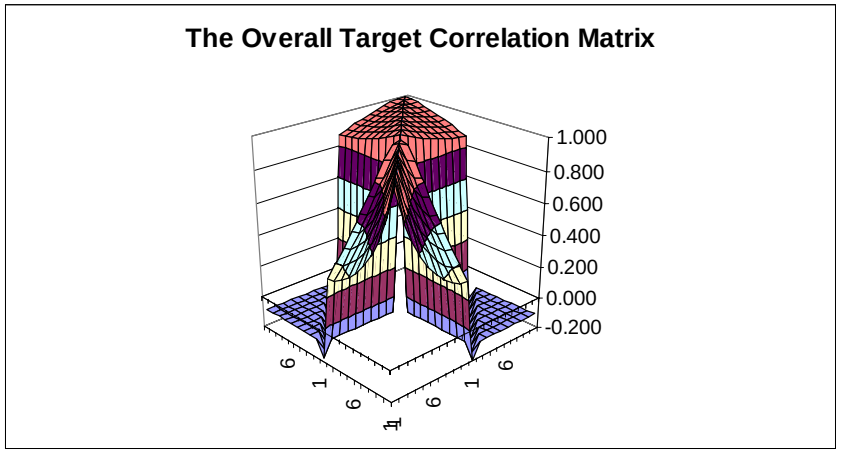

Figure 3:

Figure caption for {TestCorrelCalibr B}: Selected cross-sections of the forward-rate/forward-rate portion of the target correlation matrix

Figure caption for {TestCorrelCalibr C}: The whole target correlation matrix

We stress that this matrix was set up on plausibility grounds, but it was not even checked whether its eigenvalues were all positive definite. (It turned out

that 14 of the 20 eigenvalues were positive.) Warts and all, this is our target matrix.

We then chosed six factors for our LMM-SABR model (we could have chosen up to 14, given the number of positive eigenvalues of the target correlation matrix). We generated a [20 x 5] matrix of random numbers drawn from the $\mathcal{U}$[0 1] uniform distribution. From this matrix we derived the zero-th iteration weights, e_{ik}, $i = 1, 2, .., 20$, $k = 1, 2, ..., 6$, (ie, a [20 x 6] matrix) using Equations (22) to (23). We calculated the zero-th iteration [20 x 20] model correlation matrix by using Equation

$$\widetilde{\boldsymbol{P}} = \boldsymbol{e}\boldsymbol{e}^{\dagger} \tag{26}$$

We calculated the distance between the target and the model correlation matrix using Equation

$$\chi^2 = \sum_{ij} w_{ij} \left(P_{ij} - \widetilde{P}_{ij} \right)^2 \tag{27}$$

All the importance weights were set by by default to 1, with the following exceptions:

- all the entries of the forward-rate/forward-rate block were given an importance weight of 8 – this reflects the fact that we 'trust' the entries of the forward-rate/forward-rate block more;

- the diagonal entries of the forward-rate/volatility block (that correspond to the SABR entries) were given an importance weight of 40 – this reflects our attempt to recover the elements R_{ii} very closely. We do not know at this stage whether a weight of 40 is too 'light' or an overkill.

The unconstrained search procedure took approximately 1 minute of real time on a standard IBM laptop computer. Figures {TestCorrelCalibr D} and {TestCorrelCalibr E} show the excellent results for the overall correlation matrix.

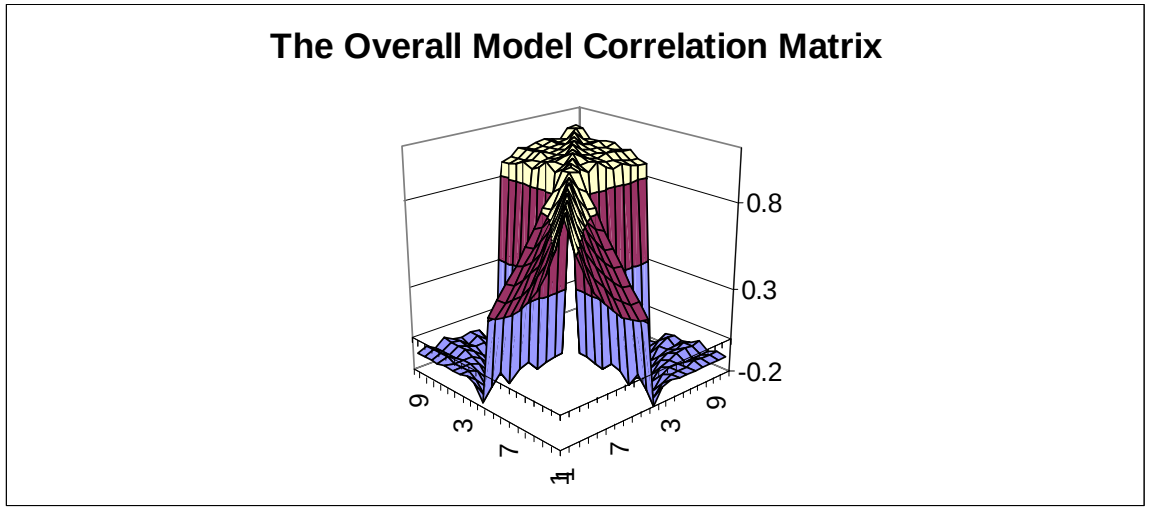

Figure 4:

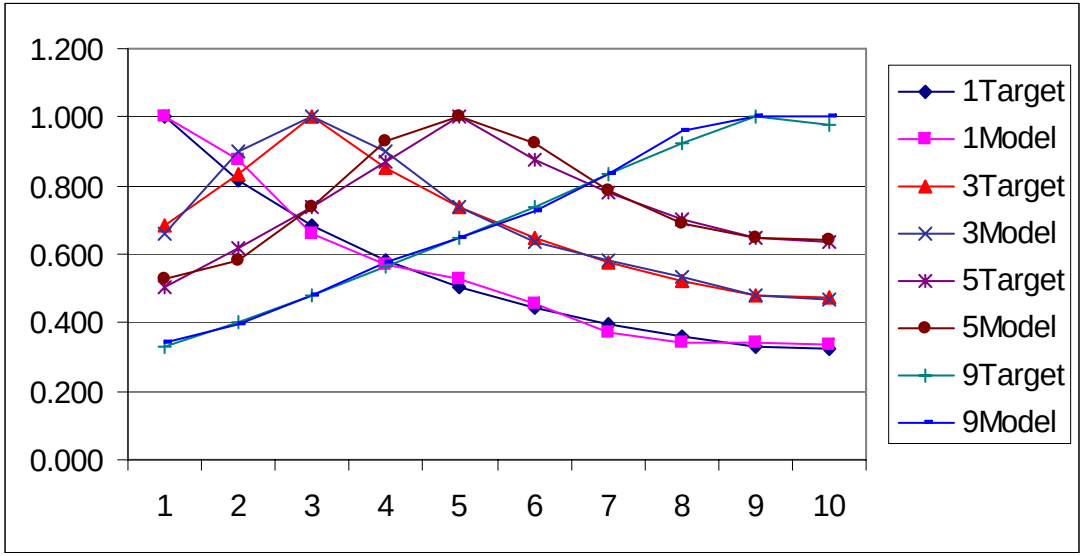

Figure 5:

Caption: The outcome of the best-fit (second method) to the target correlation matrix depicted in Fig {TestCorrelCalibr C} obtained using 6 factors.

Caption: Selected cross sections of the target and fitted portions referring to forward rates of the correlation matrices shown in Figs {TestCorrelCalibr C} and

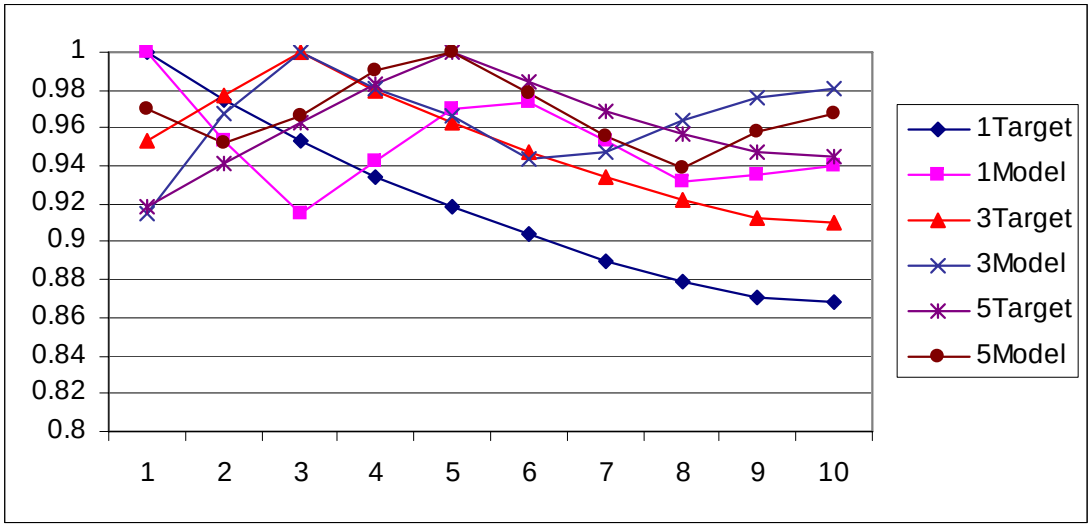

Figure 6:

{TestCorrelCalibr D}, resepctively. A line labelled, say, as 3Target or 3Model, shows the target and fitted correlation between the third forward rate and the others. The results were obtained using 6 factors.

Caption: Selected cross sections of the target and fitted portions referring to

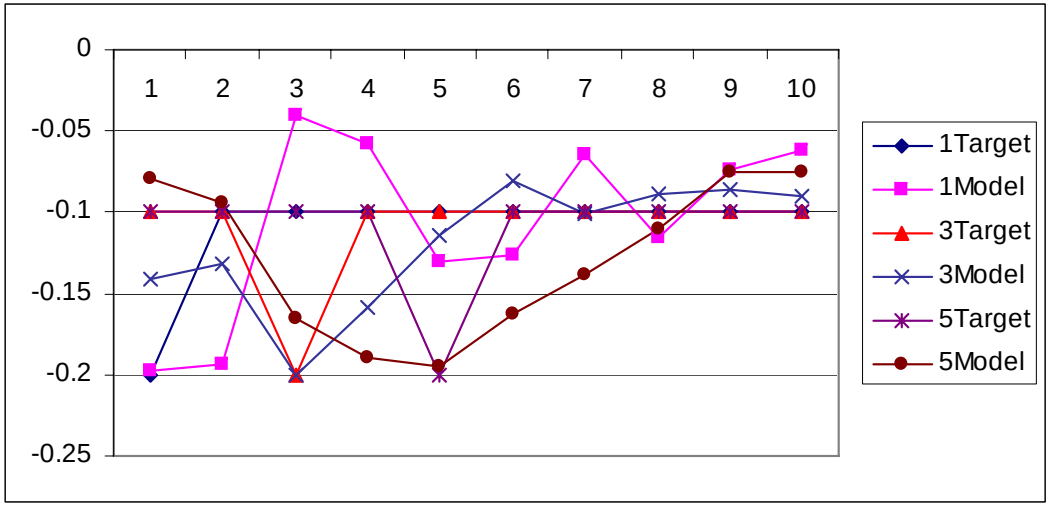

Figure 7:

volatilities of the correlation matrices shown in Figs {TestCorrelCalibr C} and {TestCorrelCalibr D}, resepctively. A line labelled, say, as 3Target or 3Model, shows the target and fitted correlation between the third volatility and the others. The results were obtained using 6 factors.

Caption: Selected cross sections of the target and fitted portions referring to

forward rates and volatilities of the correlation matrices shown in Figs {TestCorrelCalibr C} and {TestCorrelCalibr D}, resepctively. A line labelled, say, as 3Target or 3Model, shows the target and fitted correlation between the third volatility and the other forward rates. Note that for all the forward rates shown in these cross sections the element R_{ii} are recovered almost exactly – recall that their common target value was -0.2. The results were obtained using 6 factors.

Since 3d graphs are dazzling, but difficult to read, Figure {TestCorrelCalibr E} shows selected sections of the forward-rate/forward-rate portion of the correlation matrix for the target and the optimized model. Figures {TestCorrelCalibr F} and {TestCorrelCalibr G} do the same for the forward-rate/volatility and volatility/volatility sub-matrices. Note from figure {TestCorrelCalibr G} that for the forward-rate/volatility sub-matrix the model recovers very accurately the all-important diagonal elements. See Table {TestCorrelCalibr K}. It does

not do a great job at recovering the correlation elements adjacent to the main diagonal – but, for the stylized example that we have presented, this may not be a bad thing, as the discontinuity we assumed for the target matrix (all diagonal elements at -0.2 and all other elements at -0.1) may, after all, not be very 'physical'. Had we used the functional form suggested in Equation (6.14) we would have probably obtained better overall results.

In sum, the important components of the overall target correlation matrix (ie, the forward-rate/forward-rate and the diagonal of the forward-rate/volatility sub matrices) have been recovered really well. The remaining portions are not perfectly recovered, but, given the much lower number of factors (6) than variables (20), **we are hitting against the intrinsic limitations of low-dimensionality models explored at length in Rebonato and Cooper (1995) and Rebonato (2002)**. We have no reason to believe that the models shortcomings are due to numerical deficiencies of the chosen approach.

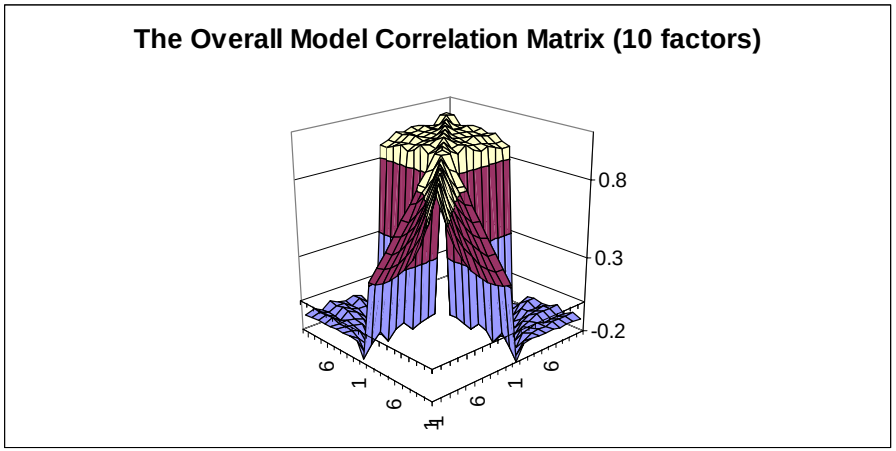

Figure 8:

Indeed as we increase the number of factors from six to ten we observe a steady improvement of the solution, as shown in Figures {P1} to {P4}. The improvement is particularly marked for the off-diagonal elements of the R matrix.

FIGURE {P1}: Same as Fig {TestCorrelCalibr D} with ten factors

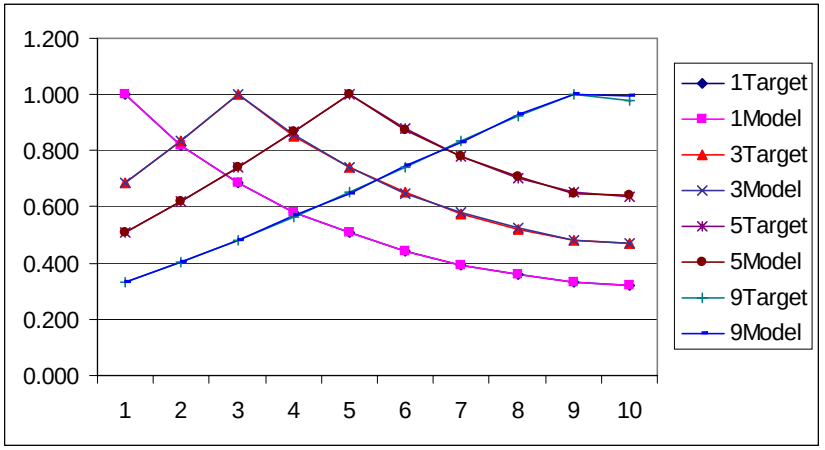
1.200
1.000
0.800
0.600
0.400
0.200
0.000
1
2
3
4
5
6
7
8
9
10
1Target
1Model
3Target
3Model
5Target
5Model
9Target
9Model

Figure 9:

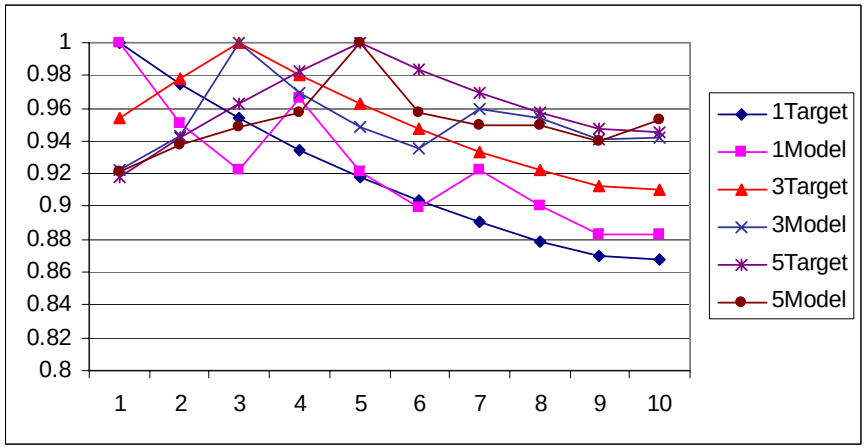

Figure 10:

FIGURE {P2}: Same as Fig {TestCorrelCalibr E} with ten factors

FIGURE {P3}: Same as Fig {TestCorrelCalibr F} with ten factors

FIGURE {P4}: Same as Fig {TestCorrelCalibr G} with ten factors

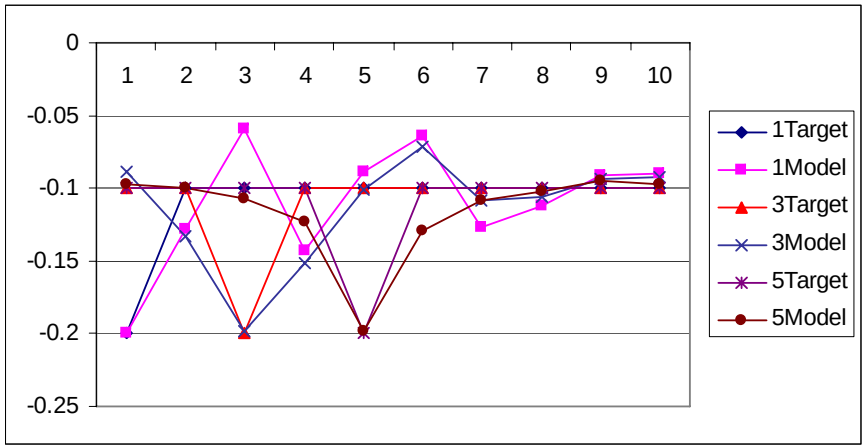
0
-0.05
-0.1
-0.15
-0.2
-0.25
1
2
3
4
5
6
7
8
9
10
1Target
1Model
3Target
3Model
5Target
5Model

Figure 11:

4 Which Method Should One Choose?

We have presented two methods, one analytic and one numeric to calibrate the LMM-SABR model to a target correlation matrix. In general analytic methods are thought to be superior (and faster). Why should one consider the numerical method at all? There are two possible reasons.

Let's start from the target correlation matrix. Since this matrix contains a number of exogenously-assigned (and negative[‡]) entries, it is likely not to be a feasible correlation matrix (ie, not to have all positive eigenvalues.) The analytic procedure suggests to perform an orthogonalization first, and to retain at most as many factors as the number of positive eigenvectors. This is perfectly

[‡]Negative entries in a correlation matrix reduce considerably the set of possible solutions with positive eigenvalues.

correct, but **the number of factors one is left to work with is outside the control of the modeller**. If, say, only two eigenvalues are positive, no model implementation with four, six or ten factors will be possible.

The second, and closely related, potential problem (or advantage, depending on the point of view) is that the analytic solution provides an *exact* recovery of the SABR correlation R_{ii}. Why is getting an *exact* recovery a bad thing? Because **there is no way to introduce a trade-off between a very small error in recovering some of the R_{ii} elements, and increasing the number of positive eigenvalues**. Perhaps if we accepted, say, $R_{22} = -0.1975$ instead of $R_{22} = -0.200$ we could have six instead of four positive eigenvalues, but the analytic procedure has no concept of the fact that an overall more desirable solution may be lurking just 'nearby' (in solution space).

This is where the numerical (and imperfect) method can provide an advantage. If the target correlation matrix has all positive eigenvalues, then

the analytic method should certainly be used. The same applies if it has at least as many positive eigenvalues as factors the trader wants to retain. But if the target correlation matrix happens to have fewer positive eigenvalues than the desired number of factors, then the numerical approach *can* provide a better solution. It can do so by finding the positive-definite correlation matrix (of whatever rank up to $2N$) that is closest to a perhaps-poorly-specified target matrix. The more factors in the model, the closer one can get to the 'wrong' target matrix, while remaining within the set of positive definite real symmetric matrices.

We should mention a last possible advantage of the numerical procedure. We pointed out in Chapter 4, Section 4 that, when we use the more symmetric matrix of loadings $\boldsymbol{e}$, which treats all the variables (forward rates and volatilities) on the same footing, there is no exact way to impose that, say, four of our six factors should be devoted to recovering the forward-rate/forward-rate

portion of the correlation matrix and only two to the poorly-known volatility part. (We may want to do so perhaps because we trust forward-rate portion more, or because we consider it more important for the pricing of the product at hand.) This remains true if we use the analytic calibration described above. However, if we use the numerical procedure with non-constant importance weights we are implicitly specifying how we want to use our silver bullets (ie, the degrees of freedom afforded by the number of factors that we have decided to retain). Assigning importance weights, say, ten times as large for the forward-rate/forward-rate portion of the correlation matrix than for the volatility/volatility portion does something very similar to assigning far more factors to the description of the forward rates than of the volatilities. The relationship between weights and number of factors assigned to a given portion is clearly complex, and certainly non-linear. Nonetheless the choice of the importance weights performs a very similar role to the choice of the number of factors for the various drivers.

5 Appendix[§]

Various problems involve a constrained maximization or minimization over quantities whose squares must sum to 1. Denoting by p_i these n squared quantities, this means that the constraints to be satisfied are

$$\sum_{i=1}^{n} p_i = 1 \quad , \qquad p_i \geqslant 0 \ \text{ for all } i. \tag{28}$$

Without loss of generality we can think of the squared quantities as probabilities. One way to satisfy these conditions automatically (See Section 7.2.3) is to use

[§]The following is taken from Doust (2008) with thanks.

internal variables θ_j $(j = 1\ to\ n-1)$ where

$$p_i = p_i(\theta_j) = \begin{cases} \cos^2\theta_1 & i = 1 \\ \sin^2\theta_1 \cos^2\theta_2 & i = 2 \\ \sin^2\theta_1 \sin^2\theta_2 \cos^2\theta_3 & i = 3 \\ \ldots & \\ \sin^2\theta_1 \ \ldots \ \sin^2\theta_{n-2} \cos^2\theta_{n-1} & i = n-1 \\ \sin^2\theta_1 \ \ldots \ \sin^2\theta_{n-2} \sin^2\theta_{n-1} & i = n \end{cases} . \quad (29)$$

and where $0 \leqslant \theta_j < \pi/2$.

However, assuming that numerical methods are being used to solve the problem at hand, Doust (2007, 2008) points out that Equation (29) is sub-optimal because a uniform probability distribution for each θ_i implies $E(p_i) \sim 2^{-i}$ which is highly non-uniform.

A better framework is to use the following algorithm:

$$
\begin{aligned}
n = 2 \quad & \begin{cases} p_1 = \cos^2\theta_1 \\ p_2 = \sin^2\theta_1 \end{cases} \\
n = 3 \quad & \begin{cases} p_1 = \cos^2\theta_1 \cos^2\theta_2 \\ p_2 = \sin^2\theta_1 \\ p_3 = \cos^2\theta_1 \sin^2\theta_2 \end{cases} \\
n = 4 \quad & \begin{cases} p_1 = \cos^2\theta_1 \cos^2\theta_2 \\ p_2 = \sin^2\theta_1 \cos^2\theta_3 \\ p_3 = \cos^2\theta_1 \sin^2\theta_2 \\ p_4 = \sin^2\theta_1 \sin^2\theta_3 \end{cases} \\
n = 5 \quad & \begin{cases} p_1 = \cos^2\theta_1 \cos^2\theta_2 \cos^2\theta_4 \\ p_2 = \sin^2\theta_1 \cos^2\theta_3 \\ p_3 = \cos^2\theta_1 \sin^2\theta_2 \\ p_4 = \sin^2\theta_1 \sin^2\theta_3 \\ p_5 = \cos^2\theta_1 \cos^2\theta_2 \sin^2\theta_4 \end{cases} \\
n = 6 \quad & \begin{cases} p_1 = \cos^2\theta_1 \cos^2\theta_2 \cos^2\theta_4 \\ p_2 = \sin^2\theta_1 \cos^2\theta_3 \cos^2\theta_5 \\ p_3 = \cos^2\theta_1 \sin^2\theta_2 \end{cases}
\end{aligned}
\tag{30}
$$

and so on.

Although it is hard to write down a closed-form formula for this framework, the rule to generate the formulae is easy to specify. To generate the formulae for $n = m$ given the formulae for $n = m - 1$, work down the formulae for $n = m-1$ starting at p_1 and locate the first p_i with fewer $\sin / \cos$ terms than p_1. Then multiply this p_i by $\cos^2 \theta_{m-1}$ and define p_m by using the original p_i but multiplied by $\sin^2 \theta_{m-1}$. If $m - 1$ is a power of 2 then all the p_i have the same number of terms, so modify p_1 to create p_m.

For (30), if the θ_i are uniformly distributed then $E(p_i)$ just depends on how many $\sin / \cos$ terms the formula for p_i contains. If n is such that $2^{m-1} < n \leqslant 2^m$, all p_i will have either $m - 1$ or m $\sin / \cos$ terms and $E(p_i) = 2^{-m-1}$ or 2^{-m} accordingly. Furthermore if $n = 2^m$ then $E(p_i) = 2^{-m}$ for all i and in these special cases the distribution will be uniform across the p_i.

For both (29) and (30) the derivatives $\frac{\partial p_i}{\partial \theta_j}$ are easy to calculate if needed. Again this is more easily specified by a rule rather than a formula as follows

$$\frac{\partial p_i}{\partial \theta_j} = \begin{cases} \frac{2p_i}{\tan(\theta_j)} & \text{if } p_i \text{ contains a term in } \sin^2\theta_j \\ -2p_i \tan(\theta_j) & \text{if } p_i \text{ contains a term in } \cos^2\theta_j \\ 0 & \text{otherwise} \end{cases} \quad . \tag{31}$$

Note that (28) guarantees that

$$\sum_i \frac{\partial p_i}{\partial \theta_j} = 0 \quad . \tag{32}$$