The LIBOR Market Model in Practice. Dariusz Gatarek, Przemyslaw Bachert and Robert Maksymiuk.

8 Non-Parametric Calibration Algorithms to Caps and Swaptions

8.1 INTRODUCTION

In the previous chapter we have presented two simple calibration algorithms to cap options. It was shown how to understand the cap and swaption payments mechanism. We have showed step-by-step how to strip caplet volatilities from caps and finally presented the two most popular calibration algorithms to caps. Both were based on the piecewise constant volatility assumption. One of them has presented calibration assuming that volatilities depend only on the time to maturity, whilst the other one has assumed that volatilities depend on the maturity of the underlying forward rate.

Chapter 8 concerns non-parametric calibration algorithms of caps and swaptions. The nature of swaption quotations was presented in the previous chapter in section 7.2. The section contains also market swaption quotations taken from a particular working day, the same day for which caps and interest rate date for EUR were taken previously. Such an approach will enable comparisons between results of various calibration algorithms.

We start with a description of one of the most popular algorithm of calibration to swaptions called the separated approach. The separated approach provides a direct way of calibrating the model to the full set of swaptions. We present in detail all necessary steps allowing an implementation of the algorithm in practice. First we create a matrix of swaption volatilities and after that define the covariance matrix of the forward LIBOR rates. Next we present how to compute the elements of the covariance matrix. Additionally, that part contains intermediate calculations and intermediate results which helps the reader to fully understand the matter. As a result we obtain a variance-covariance matrix of the forward LIBOR rates. We show how to transform the obtained matrices to ensure positivity of the matrices. For this we utilize principal component analysis.

We compute matrices for each of the different variants of the separated approach. The differences arise from different specification of the parameters Λ_i used in the calibration. We see later the exact definition of these parameters. For each variant we present vectors of eigenvalues and additionally the root mean squared error between theoretical and market swaption volatilities. The two variants of calibration are then described in more detail. One can find algorithms that allow the instantaneous volatilities to be derived by a specification as orthogonal vectors. The specification is based first for the assumption of constant volatility through time and next as the piecewise constant case.

Next we develop a previously demonstrated separated approach by adding an optimization algorithm. As a target function we set a root mean squared error for the difference between the theoretical and market swaption volatilities. We minimize that function but under several restrictions for VCV. We postulate that the VCV matrix must be positive definite. For that case we implement a subalgorithm for reducing the VCV matrix by removing eigenvectors

associated with negative eigenvalues. We describe all the necessary steps for that calibration routine. We present the whole calibration algorithm in Matlab code.

Another calibration algorithm presented in the chapter is the locally single factor approach. First we present all necessary assumptions and after that we move into the details of the algorithm. Based on the market reference data we show some results of the computations.

We then move into the calibration to swaptions given the exogenously computed correlations of forward LIBOR rates based on historical market data. In that part of the chapter we present how to compute historical correlations and present the final results.

The last part of the chapter is dedicated to calibration to co-terminal swaptions. We present the nature of co-terminal swaptions and the bootstrap of instantaneous volatility. Finally we present calibration results.

8.2 THE SEPARATED APPROACH

One of the popular approaches of BGM calibration is called the separated approach. This approach provides a direct way of calibrating the model to the full set of swaptions. In many situations there is more worth calibrating the model to swaptions instead of caps. This is especially true when we want to value an exotic instrument which is more dependent on swaption prices than cap volatilities. A good example will be any Bermudan type swaption.

We start our separated approach calibration by creating a matrix of swaption volatilities as below:

$$\Sigma^{SWPT} = \begin{bmatrix} \sigma_{1,2}^{swpt} & \sigma_{1,3}^{swpt} & \sigma_{1,4}^{swpt} & \cdots & \sigma_{1,m+1}^{swpt} \\ \sigma_{2,3}^{swpt} & \sigma_{2,4}^{swpt} & \sigma_{2,5}^{swpt} & \cdots & \sigma_{2,m+2}^{swpt} \\ \sigma_{3,4}^{swpt} & \sigma_{3,5}^{swpt} & \sigma_{3,6}^{swpt} & \cdots & \sigma_{3,m+3}^{swpt} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sigma_{m,m+1}^{swpt} & \sigma_{m,m+2}^{swpt} & \sigma_{m,m+3}^{swpt} & \cdots & \sigma_{m,M}^{swpt} \end{bmatrix}_{m \times m}$$

where in our case m = 10 and M = 20 and $\sigma_{1,2}^{swpt} = \sigma^{swpt}(t, T_1, T_2)$ is the market swaption volatility for a swaption maturing at T_1 with underlying swap period $T_1 \div T_2$. We can define the dependency of the components of Σ^{SWPT} on market swaption volatility symbols in the following way:

$$\sigma_{n,N}^{MKT} = \Sigma^{SWPT}(n, N-n)$$

Let us define the covariance matrix of forward LIBOR rates in the following way:

$$\Phi^{i} = \begin{bmatrix} \varphi_{1,1}^{i} & \varphi_{1,2}^{i} & \varphi_{1,3}^{i} & \cdots & \varphi_{1,m}^{i} \\ \varphi_{2,1}^{i} & \varphi_{2,2}^{i} & \varphi_{2,3}^{i} & \cdots & \varphi_{2,m}^{i} \\ \varphi_{3,1}^{i} & \varphi_{3,2}^{i} & \varphi_{3,3}^{i} & \cdots & \varphi_{3,m}^{i} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_{m,1}^{i} & \varphi_{m,2}^{i} & \varphi_{m,3}^{i} & \cdots & \varphi_{m,m}^{i} \end{bmatrix}_{m \times m}$$

where:

$$\varphi_{kl}^{i} = \int_{0}^{T_i} \sigma^{inst}(t, T_{l-1}, T_l)\, \sigma^{inst}(t, T_{k-1}, T_k)\, dt \text{ for } i < k \text{ and } i < l$$

and $\sigma^{inst}(t, T_{l-1}, T_l)$ is the stochastic instantaneous volatility of the LIBOR rate $L_l(t, T_{l-1}, T_l)$.

We assume that

$$\varphi_{kl}^{i} = \Lambda_i \varphi_{kl}$$

where Λ_i are positive numbers and

$$\Phi = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \varphi_{1,3} & \cdots & \varphi_{1,m} \\ \varphi_{2,1} & \varphi_{2,2} & \varphi_{2,3} & \cdots & \varphi_{2,m} \\ \varphi_{3,1} & \varphi_{3,2} & \varphi_{3,3} & \cdots & \varphi_{3,m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_{m,1} & \varphi_{m,2} & \varphi_{m,3} & \cdots & \varphi_{m,m} \end{bmatrix}_{m \times m}.$$

Parameters on diagonal can be calculated via the closed form formulae

$$\varphi_{kk} = \frac{\delta_{0,k} \sigma^{swpt}(t, T_k, T_{k+1})^2}{\Lambda_k}$$

where k = 1, . . . , m.

Having that we can use the simple algorithm below to calculate the diagonal values of the matrix Φ:

Algorithm 8.1 Diagonal elements of matrix Φ

For k = 1 to m//number of rows in market swaption volatility matrix

$$\Phi(k,k) = \delta(0,k)\Sigma(k,1)^2/\Lambda(k)$$

Next k

//where $\delta(0,k)$ is a year fraction for particular day count basis between T_0 and T_k

End of algorithm 8.1

The next step is to compute the parameters $R_{i,j}^{k}(t)$. These parameters will be used later for determining the non-diagonal elements of the matrix Φ. We define $R_{i,j}^{k}(t)$ as:

$$R_{i,j}^{k}(0) = \frac{B(0, T_{k-1}) - B(0, T_k)}{B(0, T_i) - B\left(0, T_j\right)}$$

Now we can compute the whole matrix of parameters **R**. The form of matrix **R** is presented on the next page.

$$
\mathbf{R} = \begin{bmatrix}
\begin{bmatrix} R^2_{1,2}(t) \end{bmatrix} & \begin{bmatrix} R^2_{1,3}(t) \\ R^3_{1,3}(t) \end{bmatrix} & \begin{bmatrix} R^2_{1,4}(t) \\ R^3_{1,4}(t) \\ R^4_{1,4}(t) \end{bmatrix} & \cdots & \begin{bmatrix} R^2_{1,m+1}(t) \\ R^3_{1,m+1}(t) \\ R^4_{1,m+1}(t) \\ \cdots \\ R^{m+1}_{1,m+1}(t) \end{bmatrix} \\
\begin{bmatrix} R^3_{2,3}(t) \end{bmatrix} & \begin{bmatrix} R^3_{2,4}(t) \\ R^4_{2,4}(t) \end{bmatrix} & \begin{bmatrix} R^3_{2,5}(t) \\ R^4_{2,5}(t) \\ R^5_{2,5}(t) \end{bmatrix} & \cdots & \begin{bmatrix} R^3_{2,m+2}(t) \\ R^4_{2,m+2}(t) \\ R^5_{2,m+2}(t) \\ \cdots \\ R^{m+2}_{2,m+2}(t) \end{bmatrix} \\
\begin{bmatrix} R^4_{3,4}(t) \end{bmatrix} & \begin{bmatrix} R^4_{3,5}(t) \\ R^5_{3,5}(t) \end{bmatrix} & \begin{bmatrix} R^4_{3,6}(t) \\ R^5_{3,6}(t) \\ R^6_{3,6}(t) \end{bmatrix} & \cdots & \begin{bmatrix} R^4_{3,m+3}(t) \\ R^5_{3,m+3}(t) \\ R^6_{3,m+3}(t) \\ \cdots \\ R^{m+3}_{3,m+3}(t) \end{bmatrix} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\begin{bmatrix} R^{m+1}_{m,11}(t) \end{bmatrix} & \begin{bmatrix} R^{m+1}_{m,12}(t) \\ R^{m+2}_{m,12}(t) \end{bmatrix} & \begin{bmatrix} R^{m+1}_{m,13}(t) \\ R^{m+2}_{m,13}(t) \\ R^{m+3}_{m,13}(t) \end{bmatrix} & \cdots & \begin{bmatrix} R^{m+1}_{m,M}(t) \\ R^{m+2}_{m,M}(t) \\ R^{m+3}_{m,M}(t) \\ \cdots \\ R^{M}_{m,M}(t) \end{bmatrix}
\end{bmatrix}_{m \times M \times M}
$$

We can treat this matrix as a three dimensional matrix. The first dimension for $R^k_{i,j}(t)$ is index i, the second j and the third k. Having that we can define assignment for the elements of matrix **R**:

$$\mathbf{R}(i, j, k) = R^k_{i,j}(t)$$

And the calculation of the entries for **R** requires the algorithm presented below.

Algorithm 8.2 Elements of matrix R

For $i = 1$ to m // NumberOfSwaptionMaturities
  For j = (i + 1) to (M − m) + i // NumberOfSwaptionUnderlyings + i
    For k = (i + 1) to j
      $\mathbf{R}(i, j, k) = [\mathbf{B}(k-1) - \mathbf{B}(k)] / [\mathbf{B}(i) - \mathbf{B}(j)]$ // **B** is a vector of discount factors
    Next k
  Next j
Next i

End of algorithm 8.2

Vector of discount factors **B** can be presented as:

$$\mathbf{B} = \begin{bmatrix} B(0, T_1) \\ B(0, T_2) \\ \cdots \\ B(0, T_M) \end{bmatrix}$$

Using algorithm 8.2 for our market data taken from 20 January 2005 gives the results presented by Table 8.1.

Table 8.1 Matrix R

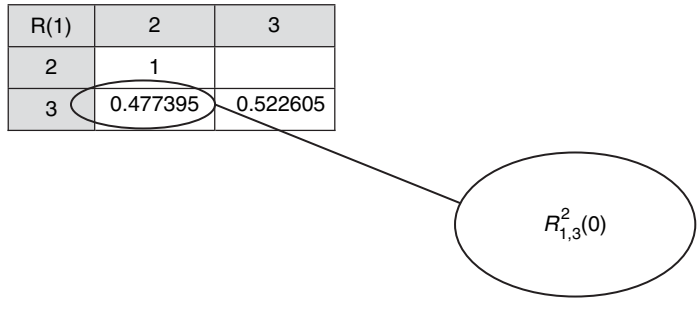
R(1)	2	3	4	5	6	7	8	9	10	11
2	1									
3	0.477395	0.522605								
4	0.306688	0.335732	0.35758							
5	0.223908	0.245112	0.261063	0.269916						
6	0.175379	0.191988	0.204482	0.211416	0.216735					
7	0.143629	0.157231	0.167462	0.173141	0.177497	0.18104				
8	0.121658	0.133179	0.141846	0.146656	0.150345	0.153346	0.152969			
9	0.105449	0.115435	0.122947	0.127116	0.130314	0.132915	0.132588	0.133237		
10	0.093586	0.102449	0.109116	0.112816	0.115654	0.117963	0.117673	0.118249	0.112493	
11	0.084644	0.09266	0.09869	0.102036	0.104603	0.106691	0.106429	0.10695	0.101744	0.095553

Table 8.1 Continued

R(2)	3	4	5	6	**7**	**8**	9	10	11	12
3	1									
4	0.484244	0.515756								
5	0.315829	0.336382	0.347789							
6	0.23282	0.247971	0.25638	0.262829						
7	0.183601	0.195549	0.20218	0.207266	0.211404					
8	0.151626	0.161493	0.166969	0.171169	0.174586	0.174157				
9	0.129042	0.13744	0.1421	0.145675	0.148583	0.148218	0.148943			
10	0.113027	0.120382	0.124464	0.127596	0.130143	0.129823	0.130458	0.124107		
11	0.101228	0.107816	0.111472	0.114276	0.116557	0.116271	0.11684	0.111152	0.104389	
12	0.091758	0.097729	0.101043	0.103585	0.105653	0.105393	0.105909	0.100753	0.094623	0.093556
...										

R(10)	11	12	13	14	15	16	17	18	19	20
11	1									
12	0.502837	0.497163								
13	0.341309	0.337458	0.321233							
14	0.259621	0.256691	0.24435	0.239338						
15	0.210387	0.208013	0.198012	0.193951	0.189637					
16	0.178905	0.176887	0.168382	0.164929	0.16126	0.149636				
17	0.156257	0.154494	0.147066	0.14405	0.140846	0.130693	0.126595			
18	0.139159	0.137589	0.130974	0.128288	0.125434	0.116393	0.112743	0.109419		
19	0.125814	0.124394	0.118414	0.115985	0.113405	0.10523	0.101931	0.098926	0.0959	
20	0.115097	0.113798	0.108327	0.106105	0.103745	0.096267	0.093248	0.090499	0.087731	0.085182

The representation of **R** above is explained with this small submatrix example shown below:



Having computed the matrix of parameters **R** we can determine the off diagonal parameters of our covariance matrix Φ. We will use the following formulae

$$\varphi_{k,N-1} = \frac{\delta_k \sigma_{kN}^2 - \Lambda_k \left(\sum_{l=k+1}^{N} \sum_{i=k+1}^{N} R_{kN}^i(0) \varphi_{i-1l-1} R_{kN}^l(0) - 2R_{kN}^{k+1}(0) \varphi_{k,N-1} R_{kN}^N(0) \right)}{2\Lambda_k R_{kN}^{k+1}(0) R_{kN}^N(0)} \tag{8.1}$$

for k = 1, . . . , m and N = k + 2, ...

The full calculation of matrix Φ requires a detailed recursive algorithm. This is presented below.

Algorithm 8.3 Matrix Φ

```
s = 1
SumTemp = 0
For i = 1 to R // NumberOfRowsInCovarianceMatrix
  For j = i + 1 to R // NumberOfRowsInCovarianceMatrix
    Sum = 0
    For 1 = i + j − 2s + 1 to j + 1
      For k = i + j − 2s + 1 to j + 1
        SumTemp = R(i + j − 2s, j + 1, k)*R(i + j − 2s, j + 1, 1)*Φ(k − 1, 1 − 1)
          Sum = Sum + SumTemp
        Next k
      Next 1
    Φ(i + j − 2s, j) =
    {δ(0, i + j − 2s)*Σ(i + j − 2s, i + 1)^2 − Λ(i + j − 2s)*
    [Sum − 2*R(i + j − 2s, j + 1, i + j − 2s + 1)*Φ(i + j − 2s, j)*R(i + j − 2s, j + 1, j + 1)]}/
    [2*D(i + j − 2s)*R(i + j − 2s, j + 1, i + j − 2s + 1)*R(i + j − 2s, j + 1, j + 1)]
  Φ(j, i + j − 2s) = Φ(i + j − 2s, j)// set up whole matrix
  Next j
  s = s + 1
Next i
```

End of algorithm 8.3

Below we present some of our computed matrices Φ, where elements of the matrices are expressed as $\varphi^i_{kl} = \Lambda_i \varphi_{kl}$ for several arbitrary chosen functions Λ_i. First we have assumed

Table 8.2 VCV matrix for Longstaff-Schwartz-Santa Clara string model

k/l	1	2	3	4	5	6	7	8	9	10
1	5.15 %	5.50 %	4.76 %	3.46 %	2.04 %	2.76 %	1.12 %	1.85 %	0.42 %	1.66 %
2	5.50 %	5.02 %	4.57 %	3.82 %	3.13 %	2.19 %	2.58 %	2.27 %	2.39 %	1.23 %
3	4.76 %	4.57 %	4.37 %	4.00 %	3.09 %	2.61 %	1.75 %	2.33 %	2.08 %	3.15 %
4	3.46 %	3.82 %	4.00 %	3.81 %	3.45 %	2.73 %	2.31 %	1.73 %	2.76 %	1.59 %
5	2.04 %	3.13 %	3.09 %	3.45 %	3.31 %	2.88 %	2.12 %	1.95 %	1.29 %	2.83 %
6	2.76 %	2.19 %	2.61 %	2.73 %	2.88 %	3.05 %	2.68 %	2.00 %	1.82 %	1.37 %
7	1.12 %	2.58 %	1.75 %	2.31 %	2.12 %	2.68 %	2.80 %	2.50 %	1.86 %	1.70 %
8	1.85 %	2.27 %	2.33 %	1.73 %	1.95 %	2.00 %	2.50 %	2.56 %	2.31 %	1.75 %
9	0.42 %	2.39 %	2.08 %	2.76 %	1.29 %	1.82 %	1.86 %	2.31 %	2.33 %	2.14 %
10	1.66 %	1.23 %	3.15 %	1.59 %	2.83 %	1.37 %	1.70 %	1.75 %	2.14 %	2.10 %

Table 8.3 Vector of eigenvalues of VCV matrix for LS Santa Clara string model

i	1	2	3	4	5	6	7	8	9	10
λ_i	26.676 %	5.391 %	2.251 %	1.908 %	1.456 %	0.661 %	−0.028 %	−0.387 %	−0.934 %	−2.501 %

Table 8.4 Matrix of eigenvectors of VCV matrix for LS Santa Clara string model

i	e_{1i}	e_{2i}	e_{3i}	e_{4i}	e_{5i}	e_{6i}	e_{7i}	e_{8i}	e_{9i}	e_{10i}
1	37.39 %	−58.77 %	6.98 %	5.52 %	−28.58 %	−18.30 %	2.07 %	−2.24 %	49.38 %	38.34 %
2	40.99 %	−32.58 %	32.11 %	−20.48 %	13.40 %	45.00 %	0.61 %	41.38 %	−13.76 %	−41.24 %
3	40.29 %	−18.27 %	−35.45 %	−19.65 %	−9.14 %	−17.76 %	18.82 %	−23.41 %	−69.94 %	14.41 %
4	36.23 %	1.46 %	−1.63 %	7.84 %	65.13 %	−14.67 %	−8.01 %	−52.92 %	24.43 %	−26.50 %
5	31.19 %	19.46 %	−37.37 %	44.63 %	11.12 %	46.81 %	−36.76 %	16.85 %	−2.14 %	36.14 %
6	28.49 %	18.80 %	19.77 %	55.79 %	−11.58 %	−56.38 %	−0.95 %	34.45 %	−15.20 %	−24.49 %
7	24.52 %	38.41 %	42.62 %	15.14 %	−15.31 %	30.86 %	59.07 %	−27.35 %	1.82 %	21.22 %
8	24.40 %	28.87 %	23.90 %	−27.40 %	−47.95 %	0.88 %	−61.78 %	−31.10 %	−0.71 %	−12.83 %
9	22.18 %	37.93 %	9.17 %	−52.26 %	32.23 %	−27.86 %	−1.02 %	39.85 %	10.74 %	41.64 %
10	22.63 %	25.87 %	−58.32 %	−16.96 %	−29.13 %	3.22 %	30.29 %	12.65 %	39.10 %	−40.80 %

that $\Lambda_i = \delta_{0,i}$ (the year fraction between time T_0 and T_i). Choosing $\Lambda_i = \delta_{0,i}$ leads to the Longstaff-Schwartz-Santa Clara model. The results are presented below.
As we can see such a matrix Φ has negative eigenvalues. What's more, taking absolute values of eigenvalues we can see that $|\lambda_{10}|$ has third biggest value. One can see that for some market data the model may give significant mispricing of European swaptions. Let us see if this is the case for our market data.

First let us compute a modified matrix Φ^{PCA} created by removing eigenvectors associated with negative eigenvalues. For such modification we have to:

1. Create a new matrix constructed by multiplying eigenvectors by corresponding squared root of eigenvalues. We do that only for positive eigenvalues.

Table 8.5 Eigenvectors multiplied by squared root of eigenvalues

i	$e_{1i}\sqrt{\lambda_1}$	$e_{2i}\sqrt{\lambda_2}$	$e_{3i}\sqrt{\lambda_3}$	$e_{4i}\sqrt{\lambda_4}$	$e_{5i}\sqrt{\lambda_5}$	$e_{6i}\sqrt{\lambda_6}$
1	19.31 %	−13.64 %	1.05 %	0.76 %	−3.45 %	−1.49 %
2	21.17 %	−7.56 %	4.82 %	−2.83 %	1.62 %	3.66 %
3	20.81 %	−4.24 %	−5.32 %	−2.71 %	−1.10 %	−1.44 %
4	18.71 %	0.34 %	−0.24 %	1.08 %	7.86 %	−1.19 %
5	16.11 %	4.52 %	−5.61 %	6.16 %	1.34 %	3.81 %
6	14.72 %	4.37 %	2.97 %	7.71 %	−1.40 %	−4.58 %
7	12.66 %	8.92 %	6.39 %	2.09 %	−1.85 %	2.51 %
8	12.60 %	6.70 %	3.59 %	−3.78 %	−5.79 %	0.07 %
9	11.46 %	8.81 %	1.38 %	−7.22 %	3.89 %	−2.27 %
10	11.69 %	6.01 %	−8.75 %	−2.34 %	−3.52 %	0.26 %

2. Multiply the matrix created in step 1 by its transposition. In effect we obtain modified matrix Φ^{PCA}.

Table 8.6 Modified covariance matrix

VCV	1	2	3	4	5	6	7	8	9	10
1	5.75 %	5.04 %	4.58 %	3.32 %	2.38 %	2.45 %	1.34 %	1.73 %	0.87 %	1.45 %
2	5.04 %	5.53 %	4.48 %	3.98 %	2.79 %	2.52 %	2.32 %	2.35 %	2.01 %	1.62 %
3	4.58 %	4.48 %	4.90 %	3.79 %	3.22 %	2.59 %	1.84 %	2.31 %	2.12 %	2.74 %
4	3.32 %	3.98 %	3.79 %	4.15 %	3.17 %	2.79 %	2.23 %	1.88 %	2.43 %	1.92 %
5	2.38 %	2.79 %	3.22 %	3.17 %	3.66 %	2.68 %	2.28 %	1.82 %	1.69 %	2.46 %
6	2.45 %	2.52 %	2.59 %	2.79 %	2.68 %	3.27 %	2.51 %	2.04 %	1.60 %	1.58 %
7	1.34 %	2.32 %	1.84 %	2.23 %	2.28 %	2.51 %	2.95 %	2.45 %	2.04 %	1.48 %
8	1.73 %	2.35 %	2.31 %	1.88 %	1.82 %	2.04 %	2.45 %	2.64 %	2.13 %	1.85 %
9	0.87 %	2.01 %	2.12 %	2.43 %	1.69 %	1.60 %	2.04 %	2.13 %	2.83 %	1.77 %
10	1.45 %	1.62 %	2.74 %	1.92 %	2.46 %	1.58 %	1.48 %	1.85 %	1.77 %	2.67 %

Having computed the matrix of modified covariances we can compute the matrix of theoretical swaptions. The matrix will contain theoretical swaption volatilities approximated via principal component modification of the initial covariance matrix. For that purpose we use the equations:

$$\varphi^{PCA^i}{}_{kl} = \Lambda_i \varphi^{PCA}_{kl}$$

and

$$\delta_k \sigma^2_{kN} \cong \Lambda_k \sum_{l=k+1}^{N} \sum_{i=k+1}^{N} R^i_{kN}(0)\, \varphi^{PCA}_{i-1,l-1} R^l_{kN}(0).$$

And because $\delta_k = \Lambda_k$ (for LS Santa Clara string model), we obtain:

$$\sigma^2_{kN} \cong \sum_{l=k+1}^{N} \sum_{i=k+1}^{N} R^i_{kN}(0)\, \varphi^{PCA}_{i-1,l-1} R^l_{kN}(0)$$

Below you can see results of our approximation for theoretical swaption volatilities as well as differences between theoretical and market swaption volatilities:

Table 8.7 Theoretical and market volatilities of swaptions

Theoretical	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	23.98 %	23.10 %	22.15 %	20.93 %	19.62 %	18.61 %	17.61 %	16.91 %	16.32 %	15.90 %
2Y	23.51 %	21.99 %	20.79 %	19.46 %	18.40 %	17.45 %	16.75 %	16.22 %	15.81 %	
3Y	22.14 %	20.38 %	19.14 %	18.06 %	17.06 %	16.35 %	15.86 %	15.50 %		
4Y	20.35 %	18.78 %	17.72 %	16.81 %	16.02 %	15.53 %	15.12 %			
5Y	19.12 %	17.52 %	16.60 %	15.82 %	15.22 %	14.83 %				
6Y	18.07 %	16.76 %	15.94 %	15.27 %	14.64 %					
7Y	17.17 %	16.20 %	15.52 %	14.74 %						
8Y	16.26 %	15.59 %	14.80 %							
9Y	16.82 %	15.05 %								
10Y	16.35 %									

Table 8.7 Continued

Market	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	22.70 %	23.00 %	22.10 %	20.90 %	19.60 %	18.60 %	17.60 %	16.90 %	16.30 %	15.90 %
2Y	22.40 %	21.50 %	20.50 %	19.40 %	18.30 %	17.40 %	16.70 %	16.20 %	15.80 %	
3Y	20.90 %	20.10 %	19.00 %	18.00 %	17.00 %	16.30 %	15.80 %	15.50 %		
4Y	19.50 %	18.70 %	17.70 %	16.80 %	16.00 %	15.50 %	15.10 %			
5Y	18.20 %	17.40 %	16.50 %	15.80 %	15.10 %	14.80 %				
6Y	17.46 %	16.74 %	15.90 %	15.24 %	14.62 %					
7Y	16.72 %	16.08 %	15.30 %	14.68 %						
8Y	15.98 %	15.42 %	14.70 %							
9Y	15.24 %	14.76 %								
10Y	14.50 %									

Difference	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	1.28 %	0.10 %	0.05 %	0.03 %	0.02 %	0.01 %	0.01 %	0.01 %	0.02 %	0.00 %
2Y	1.11 %	0.49 %	0.29 %	0.06 %	0.10 %	0.05 %	0.05 %	0.02 %	0.01 %	
3Y	1.24 %	0.28 %	0.14 %	0.06 %	0.06 %	0.05 %	0.06 %	0.00 %		
4Y	0.85 %	0.08 %	0.02 %	0.01 %	0.02 %	0.03 %	0.02 %			
5Y	0.92 %	0.12 %	0.10 %	0.02 %	0.12 %	0.03 %				
6Y	0.61 %	0.02 %	0.04 %	0.03 %	0.02 %					
7Y	0.45 %	0.12 %	0.22 %	0.06 %						
8Y	0.28 %	0.17 %	0.10 %							
9Y	1.58 %	0.29 %								
10Y	1.85 %									

The root mean squared error between theoretical and market swaptions volatilities is defined as:

$$RMSE = \sum_{i,j=1}^{10} \left(\sigma_{ij}^{THEO} - \sigma_{ij}^{MKT}\right)^2 = 0.0013$$

Having computed our modified matrix Φ^{PCA} we can also determine vectors of instantaneous volatilities for the forward rates. Below we demonstrate how to do this.

Taking into account the equation for elements of covariance matrix Φ^{PCA^i}

$$\varphi^{PCA^i}{}_{kl} = \int_0^{T_i} \gamma_l(t)^T \cdot \gamma_k(t)\, dt$$

for $i \leq k$ and $i \leq l$, we can specify vectors of instantaneous volatility in the following way:

$$\begin{bmatrix} \gamma_1^1 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \dots \\ 0 \end{bmatrix} \begin{bmatrix} \gamma_3^1 \\ \gamma_3^2 \\ \gamma_3^3 \\ \dots \\ 0 \end{bmatrix} \dots \begin{bmatrix} \gamma_N^1 \\ \gamma_N^2 \\ \gamma_N^3 \\ \dots \\ \gamma_N^N \end{bmatrix}$$

for $\gamma_1(t), \gamma_2(t), \gamma_3(t), \dots, \gamma_N(t)$ respectively.

For pricing purposes we can assume that we have constant instantaneous volatilities through a given period of time. Having that assumption in mind we can write:

$$\varphi^{PCA^i}{}_{kl} = \int_0^{T_i} \gamma_l(t)^T \cdot \gamma_k(t)\, dt = \left(\gamma_l(t)^T \cdot \gamma_k(t)\right) \delta_{T_i} = \Lambda_i \varphi_{kl}^{PCA}$$

and

$$\gamma_1(t) = \begin{bmatrix} \gamma_1^1(t) \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \quad \text{and} \quad \gamma_1^1(t) = \gamma_1^1 \quad \text{for } 0 < t \le T_1$$

$$\gamma_2(t) = \begin{bmatrix} \gamma_2^1(t) \\ \gamma_2^2(t) \\ 0 \\ \cdots \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{matrix} \gamma_2^1(t) = \gamma_2^1 \\ \gamma_2^2(t) = \gamma_2^2 \end{matrix} \quad \text{for} \quad 0 < t \le T_2$$

...

$$\gamma_N(t) = \begin{bmatrix} \gamma_N^1(t) \\ \gamma_N^2(t) \\ \gamma_N^3(t) \\ \cdots \\ \gamma_N^N(t) \end{bmatrix} \quad \text{and} \quad \begin{matrix} \gamma_N^1(t) = \gamma_N^1 \\ \gamma_N^2(t) = \gamma_N^2 \\ \gamma_N^3(t) = \gamma_N^2 \\ \cdots \\ \gamma_N^N(t) = \gamma_N^N \end{matrix} \quad \text{for} \quad 0 < t \le T_N$$

We present an example of the determination of vector components $\gamma_i(t)$ for $i = 1, 2, \ldots, 10$

Example 8.1 Components of vectors $\gamma_i(t)$

First we take into consideration the elements of the matrix Φ^{PCA} from upper left corner $\varphi_{11}^{PCA} = 5.75\,\%$ and Λ_1 obtaining:

$$\Lambda_1 \varphi_{11}^{PCA} = \left(\gamma_1(t)^T \cdot \gamma_1(t)\right) \delta_{T_1}$$

Because $\Lambda_1 = \delta_{T_1}$, we can write:

$$\varphi_{11}^{PCA} = \begin{bmatrix} \gamma_1^1 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_1^1 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} = \left(\gamma_1^1\right)^2 \Rightarrow \gamma_1^1 = \sqrt{\varphi_{11}^{PCA}} = \sqrt{5.75\%} = 23.98\,\%$$

For elements $\varphi_{21}^{PCA} = 5.04\,\%$ and Λ_1 we obtain:

$$\Lambda_1 \varphi_{21}^{PCA} = \left(\gamma_1(t)^T \cdot \gamma_2(t)\right)\delta_{T_1} \Leftrightarrow$$
$$\varphi_{21}^{PCA} = \gamma_1(t)^T \cdot \gamma_2(t) \Leftrightarrow$$
$$\varphi_{21}^{PCA} = \begin{bmatrix} \gamma_1^1 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \cdots \\ 0 \end{bmatrix} = \gamma_1^1 \gamma_2^1 \Rightarrow \gamma_2^1 = \frac{\varphi_{21}^{PCA}}{\gamma_1^1} = \frac{5.04\%}{23.98\%} = 21.02\%$$

Similarly for elements φ_{k1}^{PCA} where $k = 3, 4, \ldots, 10$ and Λ_1 we obtain:

$$\Lambda_1 \varphi_{k1}^{PCA} = \left(\gamma_1(t)^T \cdot \gamma_k(t)\right)\delta_{T_1} \Leftrightarrow$$
$$\varphi_{k1}^{PCA} = \gamma_1(t)^T \cdot \gamma_k(t) \Leftrightarrow$$
$$\varphi_{k1}^{PCA} = \begin{bmatrix} \gamma_1^1 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_k^1 \\ \gamma_k^2 \\ \gamma_k^3 \\ \cdots \\ \cdots \end{bmatrix} = \gamma_1^1 \gamma_k^1 \Rightarrow \gamma_k^1 = \frac{\varphi_{k1}^{PCA}}{\gamma_1^1}$$

Taking values for φ_{k1}^{PCA} for $k = 3, 4, \ldots, 10$

K	3	4	5	6	7	8	9	10
φ_{k1}^{PCA}	4.58 %	3.32 %	2.38 %	2.45 %	1.34 %	1.73 %	0.87 %	1.45 %

We obtain:

K	3	4	5	6	7	8	9	10
γ_k^1	19.10 %	13.85 %	9.92 %	10.23 %	5.58 %	7.20 %	3.63 %	6.03 %

Now we choose element $\varphi_{22}^{PCA} = 5.53\,\%$ and Λ_1 or Λ_2 obtaining:

$$\Lambda_1 \varphi_{22}^{PCA} = \left(\gamma_2(t)^T \cdot \gamma_2(t)\right)\delta_{T_1}$$

or

$$\Lambda_2 \varphi_{22}^{PCA} = \left(\gamma_2(t)^T \cdot \gamma_2(t)\right)\delta_{T_2}$$

We assume that the instantaneous volatilities are constant through a given period. Having that in mind we can write:

$$\varphi_{22}^{PCA} = \gamma_2(t)^T \cdot \gamma_2(t) \Leftrightarrow$$

$$\varphi_{22}^{PCA} = \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \dots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \left(\gamma_2^1\right)^2 + \left(\gamma_2^2\right)^2 \Rightarrow \gamma_2^2 = \sqrt{\varphi_{22}^{PCA} - \left(\gamma_2^1\right)^2} \Rightarrow$$

$$\gamma_2 = \sqrt{5.53\% - (21.02)^2} = 10.53\,\%$$

For element $\varphi_{32}^{PCA} = 4.48\,\%$ and Λ_1 or Λ_2 we obtain:

$$\Lambda_1 \varphi_{32}^{PCA} = \left(\gamma_3(t)^T \cdot \gamma_2(t)\right) \delta_{T_1}$$

or

$$\Lambda_2 \varphi_{32}^{PCA} = \left(\gamma_3(t)^T \cdot \gamma_2(t)\right) \delta_{T_2}$$

Once again we have assumed that the instantaneous volatilities are constant through a given period. Having that in mind we can write:

$$\varphi_{32}^{PCA} = \gamma_3(t)^T \cdot \gamma_2(t) \Leftrightarrow$$

$$\varphi_{32}^{PCA} = \begin{bmatrix} \gamma_3^1 \\ \gamma_3^2 \\ \gamma_3^2 \\ \dots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \gamma_3^1\gamma_2^1 + \gamma_3^2\gamma_2^2 \Rightarrow \gamma_3^2 = \frac{1}{\gamma_2^2}\left(\varphi_{32}^{PCA} - \gamma_3^1\gamma_2^1\right) \Rightarrow$$

$$\gamma_3^2 = \frac{1}{10.53\%}\left(4.48\% - 19.10\% \cdot 21.02\%\right) = 4.38\%$$

Similarly for φ_{k2}^{PCA} for $k = 4, 5, \dots, 10$ and Λ_1 or Λ_2 assuming constant instantaneous volatilities we obtain:

$$\varphi_{k2}^{PCA} = \gamma_k(t)^T \cdot \gamma_2(t) \Leftrightarrow$$

$$\varphi_{k2}^{PCA} = \begin{bmatrix} \gamma_k^1 \\ \gamma_k^2 \\ \gamma_k^2 \\ \dots \\ \dots \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_2^1 \\ \gamma_2^2 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \gamma_k^1\gamma_2^1 + \gamma_k^2\gamma_2^2 \Rightarrow \gamma_k^2 = \frac{1}{\gamma_2^2}\left(\varphi_{k2}^{PCA} - \gamma_k^1\gamma_2^1\right)$$

Taking values for φ_{k2}^{PCA} for $k = 4, 5, \ldots, 10$.

k	4	5	6	7	8	9	10
φ_{k2}^{PCA}	3.98 %	2.79 %	2.52 %	2.32 %	2.35 %	2.01 %	1.62 %

We obtain:

k	4	5	6	7	8	9	10
γ_k^2	10.13 %	6.64 %	3.52 %	10.87 %	7.95 %	11.85 %	3.33 %

Going further and generalizing the algorithm for elements φ_{kk}^{PCA}, for $k = 2, 3, 4, \ldots, 10$ for each Λ_i where $i \leq k$ instantaneous volatilities will be obtained via:

$$\varphi_{kk}^{PCA} = \gamma_k(t)^T \cdot \gamma_k(t) \Leftrightarrow$$

$$\varphi_{kk}^{PCA} = \begin{bmatrix} \gamma_k^1 \\ \gamma_k^2 \\ \gamma_k^3 \\ \ldots \\ \gamma_k^k \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_k^1 \\ \gamma_k^2 \\ \gamma_k^3 \\ \ldots \\ \gamma_k^k \end{bmatrix} = \left(\gamma_k^1\right)^2 + \left(\gamma_k^2\right)^2 + \ldots + \left(\gamma_k^k\right)^2 \Rightarrow$$

$$\gamma_k^k = \sqrt{\varphi_{kk}^{PCA} - \left(\gamma_k^1\right)^2 - \left(\gamma_k^2\right)^2 + \ldots - \left(\gamma_k^{k-1}\right)^2}$$

Similarly, for elements φ_{kl}^{PCA}, for $l = 2, 3, 4, \ldots, 10$ and $l < k < 10$ for each Λ_i where $i \leq k$ and $i \leq l$ instantaneous volatilities will be obtained via:

$$\varphi_{kl}^{PCA} = \gamma_k(t)^T \cdot \gamma_l(t) \Leftrightarrow$$

$$\varphi_{kl}^{PCA} = \begin{bmatrix} \gamma_k^1 \\ \gamma_k^2 \\ \gamma_k^2 \\ \ldots \\ \gamma_k^2 \end{bmatrix}^T \cdot \begin{bmatrix} \gamma_l^1 \\ \gamma_l^2 \\ \gamma_l^2 \\ \ldots \\ \gamma_l^2 \end{bmatrix} = \gamma_k^1 \gamma_l^1 + \gamma_k^2 \gamma_l^2 + \ldots + \gamma_k^l \gamma_l^l \Rightarrow$$

$$\gamma_k^l = \frac{1}{\gamma_l^l}\left(\varphi_{kl}^{PCA} - \gamma_k^1 \gamma_l^1 - \gamma_k^2 \gamma_l^2 + \ldots - \gamma_k^{l-1} \gamma_l^{l-1}\right)$$

End of example 8.1

Having that we can present the full results of computations for the instantaneous volatility vectors:

Table 8.8 Vectors of instantaneous volatilities

Index	$\gamma_1(t)$	$\gamma_2(t)$	$\gamma_3(t)$	$\gamma_4(t)$	$\gamma_5(t)$	$\gamma_6(t)$	$\gamma_7(t)$	$\gamma_8(t)$	$\gamma_9(t)$	$\gamma_{10}(t)$
1	23.98 %	21.02 %	19.10 %	13.85 %	9.92 %	10.23 %	5.58 %	7.20 %	3.63 %	6.03 %
2		10.53 %	4.38 %	10.13 %	6.64 %	3.52 %	10.87 %	7.95 %	11.85 %	3.33 %
3			10.29 %	6.85 %	10.06 %	4.70 %	2.94 %	5.73 %	8.86 %	14.04 %
4				8.57 %	5.07 %	8.11 %	1.83 %	−3.72 %	1.35 %	−2.45 %
5					9.81 %	5.61 %	6.33 %	1.98 %	−4.27 %	3.63 %
6						9.51 %	9.68 %	9.94 %	5.58 %	1.89 %
7							0.00 %	0.00 %	0.00 %	0.00 %
8								0.00 %	0.00 %	0.00 %
9									0.00 %	0.00 %
10										0.00 %

The structure of the matrix confirms that our computations were done properly. This is because during PCA modification of the VCV matrix we have removed four eigenvectors with associated negative eigenvalues.

One additional remark is very important. The presented algorithm is very sensitive to the precision of the computed elements of the VCV matrix. In the case of too much approximation in the values in the matrix the algorithm may fail.

We can do the same computations for VCV matrix reduced to two factors. Figure below shows two eigenvectors with the biggest eigenvalues:

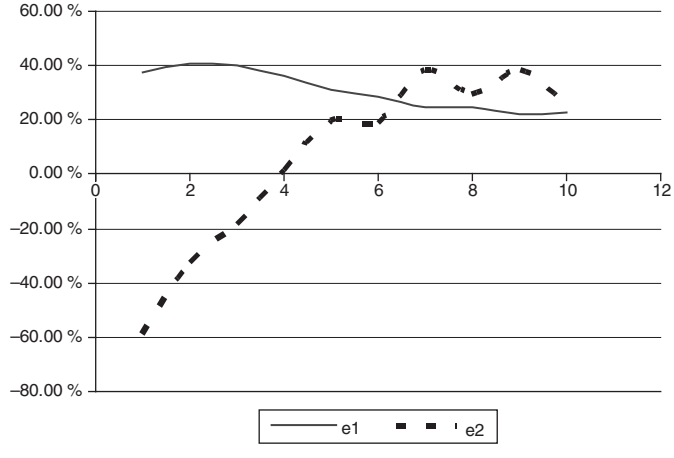


Figure 8.1 Eigenvectors with the biggest eigenvalues for $\Lambda_i = \delta_i$.

Using only two factors we obtain the following results for the theoretical swaptions:

Reducing number of factors to only the two biggest gives slightly worse results. The root mean squared error equals:

$$RMSE = \sum_{i,j=1}^{10} \left(\sigma_{ij}^{THEO} - \sigma_{ij}^{MKT}\right)^2 = 0.0030$$

Table 8.9 Theoretical and market volatilities of swaptions

Theoretical	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	23.65 %	22.83 %	22.07 %	20.86 %	19.50 %	18.48 %	17.49 %	16.84 %	16.29 %	15.90 %
2Y	22.48 %	21.79 %	20.53 %	19.18 %	18.19 %	17.28 %	16.67 %	16.18 %	15.81 %	
3Y	21.24 %	19.84 %	18.50 %	17.57 %	16.77 %	16.24 %	15.83 %	15.49 %		
4Y	18.70 %	17.55 %	16.76 %	16.16 %	15.73 %	15.41 %	15.11 %			
5Y	16.72 %	16.03 %	15.65 %	15.29 %	15.06 %	14.77 %				
6Y	15.34 %	15.21 %	14.89 %	14.74 %	14.44 %					
7Y	15.48 %	14.85 %	14.70 %	14.33 %						
8Y	14.27 %	14.31 %	13.93 %							
9Y	14.44 %	13.76 %								
10Y	13.14 %									
Market	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	22.70 %	23.00 %	22.10 %	20.90 %	19.60 %	18.60 %	17.60 %	16.90 %	16.30 %	15.90 %
2Y	22.40 %	21.50 %	20.50 %	19.40 %	18.30 %	17.40 %	16.70 %	16.20 %	15.80 %	
3Y	20.90 %	20.10 %	19.00 %	18.00 %	17.00 %	16.30 %	15.80 %	15.50 %		
4Y	19.50 %	18.70 %	17.70 %	16.80 %	16.00 %	15.50 %	15.10 %			
5Y	18.20 %	17.40 %	16.50 %	15.80 %	15.10 %	14.80 %				
6Y	17.46 %	16.74 %	15.90 %	15.24 %	14.62 %					
7Y	16.72 %	16.08 %	15.30 %	14.68 %						
8Y	15.98 %	15.42 %	14.70 %							
9Y	15.24 %	14.76 %								
10Y	14.50 %									
Difference	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	0.95 %	−0.17 %	−0.03 %	−0.04 %	−0.10 %	−0.12 %	−0.11 %	−0.06 %	−0.01 %	0.00 %
2Y	0.08 %	0.29 %	0.03 %	−0.22 %	−0.11 %	−0.12 %	−0.03 %	−0.02 %	0.01 %	
3Y	0.34 %	−0.26 %	−0.50 %	−0.43 %	−0.23 %	−0.06 %	0.03 %	−0.01 %		
4Y	−0.80 %	−1.15 %	−0.94 %	−0.64 %	−0.27 %	−0.09 %	0.01 %			
5Y	−1.48 %	−1.37 %	−0.85 %	−0.51 %	−0.04 %	−0.03 %				
6Y	−2.12 %	−1.53 %	−1.01 %	−0.50 %	−0.18 %					
7Y	−1.24 %	−1.23 %	−0.60 %	−0.35 %						
8Y	−1.71 %	−1.11 %	−0.77 %							
9Y	−0.80 %	−1.00 %								
10Y	−1.36 %									

Let us move to analyse further the specifications for the functions Λ_i. A popular choice of function is $\Lambda_i = \sqrt{\delta_i}$. For such a specification we obtain the following eigenvalues and eigenvectors:

Table 8.10 Vector of eigenvalues of VCV matrix

i	1	2	3	4	5	6	7	8	9	10
λ_i	37.82 %	21.19 %	6.90 %	6.07 %	3.06 %	1.39 %	0.50 %	0.16 %	−2.49 %	−4.03 %

Comparing the results to the LS Santa Clara string model we can see that we have obtained only two negative eigenvalues with modulus smaller than the fourth biggest positive eigenvalue. Creating a modified VCV matrix by removing the two eigenvectors associated with negative eigenvalues gives result presented below.

Table 8.11 Matrix of eigenvectors of VCV matrix

i	e_{1i}	e_{2i}	e_{3i}	e_{4i}	e_{5i}	e_{6i}	e_{7i}	e_{8i}	e_{9i}	e_{10i}
1	−35.25 %	−30.68 %	−11.64 %	−6.32 %	−14.40 %	23.03 %	10.27 %	−4.63 %	77.71 %	−27.16 %
2	−4.81 %	−51.76 %	4.66 %	−45.55 %	−13.49 %	−44.29 %	−15.86 %	44.99 %	−0.44 %	27.97 %
3	15.41 %	−51.06 %	39.68 %	−6.17 %	−28.87 %	46.23 %	−17.41 %	−21.68 %	−34.42 %	−24.77 %
4	27.22 %	−43.54 %	8.69 %	10.22 %	62.04 %	−5.41 %	16.66 %	−39.52 %	17.69 %	33.94 %
5	34.04 %	−26.60 %	−6.61 %	53.98 %	−19.46 %	−41.73 %	31.98 %	18.34 %	−0.43 %	−41.19 %
6	35.23 %	−13.56 %	−53.23 %	17.16 %	5.30 %	51.86 %	−16.68 %	44.86 %	1.83 %	20.87 %
7	37.87 %	3.33 %	−45.32 %	−26.49 %	−15.69 %	−26.26 %	−45.24 %	−47.74 %	10.30 %	−20.78 %
8	37.00 %	12.38 %	−5.79 %	−44.98 %	−32.17 %	13.51 %	69.67 %	−8.63 %	3.16 %	16.30 %
9	37.16 %	19.07 %	31.92 %	−34.13 %	44.56 %	5.62 %	−8.06 %	34.84 %	18.95 %	−49.16 %
10	34.23 %	21.38 %	47.00 %	25.39 %	−35.42 %	−2.55 %	−28.20 %	2.64 %	44.55 %	38.34 %

Table 8.12 Modified covariance matrix

VCV	1	2	3	4	5	6	7	8	9	10
1	6.95 %	4.06 %	1.25 %	−1.20 %	−3.01 %	−3.31 %	−4.81 %	−5.33 %	−6.50 %	−6.28 %
2	4.06 %	7.37 %	5.45 %	3.80 %	1.12 %	−0.14 %	−0.24 %	−0.76 %	−1.94 %	−3.36 %
3	1.25 %	5.45 %	8.09 %	5.91 %	4.38 %	2.28 %	0.67 %	1.20 %	0.75 %	1.17 %
4	−1.20 %	3.80 %	5.91 %	8.12 %	5.92 %	4.73 %	2.88 %	1.73 %	2.89 %	1.32 %
5	−3.01 %	1.12 %	4.38 %	5.92 %	8.04 %	5.77 %	4.27 %	2.73 %	2.15 %	4.04 %
6	−3.31 %	−0.14 %	2.28 %	4.73 %	5.77 %	7.60 %	6.13 %	4.36 %	2.99 %	2.41 %
7	−4.81 %	−0.24 %	0.67 %	2.88 %	4.27 %	6.13 %	7.46 %	6.40 %	4.77 %	3.35 %
8	−5.33 %	−0.76 %	1.20 %	1.73 %	2.73 %	4.36 %	6.40 %	7.09 %	6.08 %	4.81 %
9	−6.50 %	−1.94 %	0.75 %	2.89 %	2.15 %	2.99 %	4.77 %	6.08 %	8.02 %	5.70 %
10	−6.28 %	−3.36 %	1.17 %	1.32 %	4.04 %	2.41 %	3.35 %	4.81 %	5.70 %	7.70 %

Having that we can compute the matrix of theoretical swaption volatilities and compare it to observed market swaption volatilities:

Table 8.13 Theoretical and market volatilities of swaptions

Theoretical	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	23.68 %	23.73 %	22.23 %	21.01 %	19.71 %	18.65 %	17.68 %	16.95 %	16.43 %	16.04 %
2Y	22.89 %	21.63 %	20.62 %	19.41 %	18.31 %	17.40 %	16.70 %	16.22 %	15.83 %	
3Y	21.64 %	20.14 %	19.18 %	18.01 %	17.03 %	16.30 %	15.86 %	15.54 %		
4Y	20.18 %	18.73 %	17.73 %	16.82 %	16.02 %	15.56 %	15.17 %			
5Y	19.02 %	17.45 %	16.60 %	15.83 %	15.24 %	14.88 %				
6Y	17.67 %	16.75 %	15.92 %	15.30 %	14.70 %					
7Y	16.94 %	16.10 %	15.48 %	14.80 %						
8Y	16.10 %	15.57 %	14.84 %							
9Y	16.36 %	15.06 %								
10Y	15.64 %									
Market	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	22.70 %	23.00 %	22.10 %	20.90 %	19.60 %	18.60 %	17.60 %	16.90 %	16.30 %	15.90 %
2Y	22.40 %	21.50 %	20.50 %	19.40 %	18.30 %	17.40 %	16.70 %	16.20 %	15.80 %	
3Y	20.90 %	20.10 %	19.00 %	18.00 %	17.00 %	16.30 %	15.80 %	15.50 %		
4Y	19.50 %	18.70 %	17.70 %	16.80 %	16.00 %	15.50 %	15.10 %			
5Y	18.20 %	17.40 %	16.50 %	15.80 %	15.10 %	14.80 %				
6Y	17.46 %	16.74 %	15.90 %	15.24 %	14.62 %					
7Y	16.72 %	16.08 %	15.30 %	14.68 %						
8Y	15.98 %	15.42 %	14.70 %							
9Y	15.24 %	14.76 %								
10Y	14.50 %									

Table 8.13 Continued

Difference	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	0.98 %	0.73 %	0.13 %	0.11 %	0.11 %	0.05 %	0.08 %	0.05 %	0.13 %	0.14 %
2Y	0.49 %	0.13 %	0.12 %	0.01 %	0.01 %	0.00 %	0.00 %	0.02 %	0.03 %	
3Y	0.74 %	0.04 %	0.18 %	0.01 %	0.03 %	0.00 %	0.06 %	0.04 %		
4Y	0.68 %	0.03 %	0.03 %	0.02 %	0.02 %	0.06 %	0.07 %			
5Y	0.82 %	0.05 %	0.10 %	0.03 %	0.14 %	0.08 %				
6Y	0.21 %	0.01 %	0.02 %	0.06 %	0.08 %					
7Y	0.22 %	0.02 %	0.18 %	0.12 %						
8Y	0.12 %	0.15 %	0.14 %							
9Y	1.12 %	0.30 %								
10Y	1.14 %									

The root mean squared error equals:

$$RMSE = \sum_{i,j=1}^{10} \left(\sigma_{ij}^{THEO} - \sigma_{ij}^{MKT}\right)^2 = 0.0019$$

The instantaneous volatilities cannot be specified as a constant through time for this specification of the parameters Λ_i. Instead we deal with piecewise constant instantaneous volatilities. We describe this is more detail below.

Having the components of instantaneous volatility vectors as

$$\begin{bmatrix} \gamma_1^1(t) \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \begin{bmatrix} \gamma_2^1(t) \\ \gamma_2^2(t) \\ 0 \\ \dots \\ 0 \end{bmatrix} \begin{bmatrix} \gamma_3^1(t) \\ \gamma_3^2(t) \\ \gamma_3^3(t) \\ \dots \\ 0 \end{bmatrix} \dots \begin{bmatrix} \gamma_N^1(t) \\ \gamma_N^2(t) \\ \gamma_N^3(t) \\ \dots \\ \gamma_N^N(t) \end{bmatrix}$$

Then, respectively for $\gamma_1(t), \gamma_2(t), \gamma_3(t), \dots, \gamma_N(t)$, we can write:

$$\gamma_1(t) = \begin{bmatrix} \gamma_1^1(t) \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad \text{and} \quad \gamma_1^1(t) = \gamma_{1,0\div T_1}^1 \quad \text{for } 0 < t \le T_1$$

$$\gamma_2(t) = \begin{bmatrix} \gamma_2^1(t) \\ \gamma_2^2(t) \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{aligned} \gamma_2^2(t) &= \begin{cases} \gamma_{2,0\div T_1}^1 & \text{for } 0 < t \le T_1 \\ \gamma_{2,T_1\div T_2}^1 & \text{for } T_1 < t \le T_2 \end{cases} \\ \gamma_2^2(t) &= \begin{cases} \gamma_{2,0\div T_1}^2 & \text{for } 0 < t \le T_1 \\ \gamma_{2,T_1\div T_2}^2 & \text{for } T_1 < t \le T_2 \end{cases} \end{aligned}$$

...

$$\gamma_N(t) = \begin{bmatrix} \gamma_N^1(t) \\ \gamma_N^2(t) \\ \gamma_N^3(t) \\ \dots \\ \gamma_N^N(t) \end{bmatrix} \quad \text{and}$$

$$\gamma_N^1(t) = \begin{cases} \gamma^1_{N,0\div T_1} & \text{for} \quad 0 < t \le T_1 \\ \gamma^1_{N,T_1\div T_2} & \text{for} \quad T_1 < t \le T_2 \\ \gamma^1_{N,T_2\div T_3} & \text{for} \quad T_2 < t \le T_3 \\ & \dots \\ \gamma^1_{N,T_{N-1}\div T_N} & \text{for} \quad T_{N-1} < t \le T_N \end{cases}$$

$$\gamma_N^2(t) = \begin{cases} \gamma^2_{N,0\div T_1} & \text{for} \quad 0 < t \le T_1 \\ \gamma^2_{N,T_1\div T_2} & \text{for} \quad T_1 < t \le T_2 \\ \gamma^2_{N,T_2\div T_3} & \text{for} \quad T_2 < t \le T_3 \\ & \dots \\ \gamma^2_{N,T_{N-1}\div T_N} & \text{for} \quad T_{N-1} < t \le T_N \end{cases}$$

$$\gamma_N^3(t) = \begin{cases} \gamma^3_{N,0\div T_1} & \text{for} \quad 0 < t \le T_1 \\ \gamma^3_{N,T_1\div T_2} & \text{for} \quad T_1 < t \le T_2 \\ \gamma^3_{N,T_2\div T_3} & \text{for} \quad T_2 < t \le T_3 \\ & \dots \\ \gamma^3_{N,T_{N-1}\div T_N} & \text{for} \quad T_{N-1} < t \le T_N \end{cases}$$

$$\dots$$

$$\gamma_N^N(t) = \begin{cases} \gamma^N_{N,0\div T_1} & \text{for} \quad 0 < t \le T_1 \\ \gamma^N_{N,T_1\div T_2} & \text{for} \quad T_1 < t \le T_2 \\ \gamma^N_{N,T_2\div T_3} & \text{for} \quad T_2 < t \le T_3 \\ & \dots \\ \gamma^N_{N,T_{N-1}\div T_N} & \text{for} \quad T_{N-1} < t \le T_N \end{cases}$$

Therefore we can see that the generation of the instantaneous volatilities is quite similar to the previous example and thus we do not repeat that example again.

On the other hand we may reduce VCV only to three factors. The figure below shows the three eigenvectors with the biggest eigenvalues:

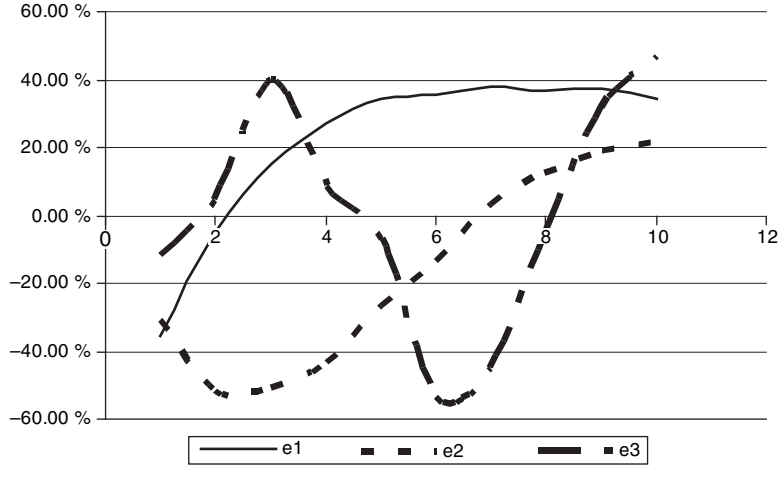


Figure 8.2 Eigenvectors with the biggest eigenvalues for $\Lambda_i = \sqrt{\delta_i}$.

Using only three factors we obtain the following results for the theoretical swaptions:

Table 8.14 Theoretical and market volatilities of swaptions

Theoretical	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y
1Y	26.06 %	22.60 %	21.41 %	20.81 %	19.68 %	18.59 %	17.66 %	16.85 %	16.28 %	15.96 %
2Y	20.22 %	20.73 %	20.41 %	19.35 %	18.24 %	17.39 %	16.64 %	16.11 %	15.78 %	
3Y	20.83 %	19.90 %	18.69 %	17.60 %	16.92 %	16.28 %	15.83 %	15.52 %		
4Y	18.52 %	17.52 %	16.96 %	16.62 %	16.00 %	15.51 %	15.17 %			
5Y	16.25 %	16.40 %	16.83 %	15.71 %	15.18 %	14.84 %				
6Y	16.95 %	16.64 %	15.55 %	14.87 %	14.52 %					
7Y	16.10 %	14.90 %	14.53 %	14.41 %						
8Y	13.97 %	14.35 %	14.53 %							
9Y	14.94 %	15.00 %								
10Y	14.79 %									
Market	**1Y**	**2Y**	**3Y**	**4Y**	**5Y**	**6Y**	**7Y**	**8Y**	**9Y**	**10Y**
1Y	22.70 %	23.00 %	22.10 %	20.90 %	19.60 %	18.60 %	17.60 %	16.90 %	16.30 %	15.90 %
2Y	22.40 %	21.50 %	20.50 %	19.40 %	18.30 %	17.40 %	16.70 %	16.20 %	15.80 %	
3Y	20.90 %	20.10 %	19.00 %	18.00 %	17.00 %	16.30 %	15.80 %	15.50 %		
4Y	19.50 %	18.70 %	17.70 %	16.80 %	16.00 %	15.50 %	15.10 %			
5Y	18.20 %	17.40 %	16.50 %	15.80 %	15.10 %	14.80 %				
6Y	17.46 %	16.74 %	15.90 %	15.24 %	14.62 %					
7Y	16.72 %	16.08 %	15.30 %	14.68 %						
8Y	15.98 %	15.42 %	14.70 %							
9Y	15.24 %	14.76 %								
10Y	14.50 %									
Difference	**1Y**	**2Y**	**3Y**	**4Y**	**5Y**	**6Y**	**7Y**	**8Y**	**9Y**	**10Y**
1Y	3.36 %	−0.40 %	−0.69 %	−0.09 %	0.08 %	−0.01 %	0.06 %	−0.05 %	−0.02 %	0.06 %
2Y	−2.18 %	−0.77 %	−0.09 %	−0.05 %	−0.06 %	−0.01 %	−0.06 %	−0.09 %	−0.02 %	
3Y	−0.07 %	−0.20 %	−0.31 %	−0.40 %	−0.08 %	−0.02 %	0.03 %	0.02 %		
4Y	−0.98 %	−1.18 %	−0.74 %	−0.18 %	0.00 %	0.01 %	0.07 %			
5Y	−1.95 %	−1.00 %	0.33 %	−0.09 %	0.08 %	0.04 %				
6Y	−0.51 %	−0.10 %	−0.35 %	−0.37 %	−0.10 %					
7Y	−0.62 %	−1.18 %	−0.77 %	−0.27 %						
8Y	−2.01 %	−1.07 %	−0.17 %							
9Y	−0.30 %	0.24 %								
10Y	0.29 %									

Reducing the number of factors to the three biggest gives slightly worse results. The root mean squared error equals:

$$RMSE = \sum_{i,j=1}^{10} \left(\sigma_{ij}^{THEO} - \sigma_{ij}^{MKT}\right)^2 = 0.0034$$

We can also specify another form for the functions Λ_i. Below we present for each specification the matching vectors of eigenvalues and root mean squared errors.

Table 8.15 Vectors of eigenvalues and root mean squared errors

Λ_i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	RMSE
1)	12.21 %	0.31 %	0.05 %	0.01 %	−0.01 %	−0.08 %	−0.18 %	−0.47 %	−1.63 %	−5.85 %	41.527
2)	18.53 %	5.67 %	0.91 %	0.16 %	−0.08 %	−0.26 %	−0.51 %	−0.80 %	−1.67 %	−6.54 %	0.3578
3)	40.09 %	16.56 %	3.21 %	2.27 %	1.38 %	0.59 %	−0.16 %	−0.81 %	−3.27 %	−11.94 %	0.0239
4)	55.37 %	24.26 %	7.99 %	6.90 %	3.42 %	1.51 %	0.48 %	0.06 %	−4.83 %	−11.67 %	0.0226
5)	91.10 %	42.12 %	20.77 %	14.79 %	6.92 %	3.28 %	2.84 %	1.80 %	−7.36 %	−15.88 %	0.0435

1) $\Lambda_i = e^{\delta_i}$
2) $\Lambda_i = \sqrt{e^{\delta_i}}$
3) $\Lambda_i = \delta_i - \ln \delta_i$
4) $\Lambda_i = \sqrt{\delta_i - \ln \delta_i}$
5) $\Lambda_i = 1$

As an alternative calibration algorithm one can use a separated approach with an optimization algorithm, where the target function will minimize the differences between the theoretical and market swaption volatilities. In that case some restrictions for the VCV matrix must be added. First of all the VCV matrix must be positive definite. If that is not the case the algorithm must first reduce VCV matrix by removing eigenvectors associated with negative eigenvalues.

8.3 THE SEPARATED APPROACH WITH OPTIMIZATION

In this section we develop the previously demonstrated separated approach by adding an optimization algorithm. As a target function we set the root mean squared error for the differences between theoretical and market swaption volatilities. We would like to minimize that function but under several restrictions for VCV. We show that the VCV matrix must be positive definite. For that case we have to implement a subalgorithm for reducing the VCV matrix by removing eigenvectors associated with negative eigenvalues. Below we present all the necessary steps for the calibration algorithm, using code consistent with Matlab.

Step 0: Loading initial data and naming the minimization function

Minimization function:

```
function f = CalibrationObjectiveFunction_SeparatedOptim(Lambda);
% Definition of minimization function will be provided after step 6
% Vector of parameters [Lambda] will contain first initial data
% and after that the parameters will be subject to change during optimization
```

Initial data will contain:

1) Matrix of market swaption volatilities [Sig]

```
Sig = . . .
[0.227  0.23   0.221  0.209  0.196  0.186  0.176  0.169  0.163  0.159;
0.224  0.215  0.205  0.194  0.183  0.174  0.167  0.162  0.158  0.154;
0.209  0.201  0.19   0.18   0.17   0.163  0.158  0.155  0.152  0.15;
0.195  0.187  0.177  0.168  0.16   0.155  0.151  0.148  0.147  0.145;
0.182  0.174  0.165  0.158  0.151  0.148  0.145  0.143  0.142  0.14;
```

```
0.1746  0.1674  0.159  0.1524  0.1462  0.1436  0.141  0.1394  0.1384  0.1368;
0.1672  0.1608  0.153  0.1468  0.1414  0.1392  0.137  0.1358  0.1348  0.1336;
0.1598  0.1542  0.147  0.1412  0.1366  0.1348  0.133  0.1322  0.1312  0.1304;
0.1524  0.1476  0.141  0.1356  0.1318  0.1304  0.129  0.1286  0.1276  0.1272;
0.145   0.141   0.135  0.13    0.127   0.1260  0.125  0.125   0.124   0.124; ]
```

2) Date for computations

```
Today='25-Jan-2005';
```

3) Vector of dates [T_Num] and vector of discount factors [B]

VectorOfDates = . . .	B = . . .
['25-Jan-2006';	[0.9774658
'25-Jan-2007';	0.9509789
'25-Jan-2008';	0.9219838
'26-Jan-2009';	0.8911017
'25-Jan-2010';	0.8591725
'25-Jan-2011';	0.8264399
'25-Jan-2012';	0.7930540
'25-Jan-2013';	0.7597502
'27-Jan-2014';	0.7262834
'26-Jan-2015';	0.6944457
'25-Jan-2016';	0.6645450
'25-Jan-2017';	0.6349818
'25-Jan-2018';	0.6068399
'25-Jan-2019';	0.5792752
'27-Jan-2020';	0.5523236
'25-Jan-2021';	0.5273147
'25-Jan-2022';	0.5030900
'25-Jan-2023';	0.4795796
'25-Jan-2024';	0.4567881
'27-Jan-2025'];	0.4346590];

To be consistent with Matlab code we have to transform vector of dates according to function:

```
T_Num = datenum(VectorOfDates);
```

Having loaded all necessary initial data we can we have to compute matrix of parameters **[R]**

Step 1: The Matrix of parameters [R]

```
% Input: Vector of discount factors [B]
% Output: Matrix of parameters [R]
```

Algorithm for step 1:

```
m = 10;  % Number of swaption maturities
M = 20;  % Number of swaption maturities plus number of swaption underlyings
R = [];  % Setting zeros for matrix [R] as initial values
for i = 1 : m
```

```
  for j = i + 1 : M − m + i
    for k = i + 1 : j
      R(i, j, k) = (B(k − 1) − B(k))/(B(i) − B(j));
    end
  end
end
```

Step 2: The Matrix of covariances [VCV] as a function of parameters [Lambda]

```
% Input: (1) Matrix of parameters [R]
%        (2) Vector of dates [T_Num]
%        (3) Matrix of market swaption volatilities [Sig]
%        (4) Vector of initial parameters [Lambda]
% Output: Matrix of covariances [VCV] as a function of parameters [Lambda]
```

Algorithm for step 2:

```
VCV = []; % Setting zeros for matrix [VCV] as initial values
% Diagonal elements of matrix VCV
for k = 1 : m
  VCV(k, k) = yearfrac(Today, T_Num(k))*Sig(k, 1)^2/Lambda(k);
end
s = 1;
for i = 1 : m
  for j = i + 1 : m
    Sum = 0;
    for l = i + j − 2*s + 1 : j + 1
      for k = i + j − 2*s + 1 : j + 1
        SumTemp = R(i + j − 2*s, j + 1, k)*R(i + j − 2*s, j + 1, 1)*VCV(k − 1, l − 1);
        Sum = Sum + SumTemp;
      end
    end
    VCV(i + j − 2*s, j) = (yearfrac(Today, T_Num(i + j − 2*s))*Sig(i + j − 2*s, i + 1)^2 −
     Lambda(i + j − 2*s)*(Sum − 2*R(i + j − 2*s, j + 1, i + j − 2*s + 1)*VCV(i + j −
     2*s, j)*R(i + j − 2*s, j + 1, j + 1)))/(2*Lambda(i + j − 2*s)*R(i + j − 2*s, j + 1, i + j −
     2*s + 1)*R(i + j − 2*s, j + 1, j + 1));
    VCV(j, i + j − 2*s) = VCV(i + j − 2*s, j);
    end
    s = s + 1;
  end
```

Step 3: The Vector of eigenvalues [L] and the Matrix of eigenvectors [E] as a function of parameters [Lambda]

```
% Input: Matrix of covariances [VCV] as a function of parameters [Lambda]
% Output: (1) Vector of eigenvalues [L] as a function of parameters [Lambda]
%         (2) Matrix of eigenvectors [E] as a function of parameters [Lambda]
```

Algorithm for step 3:

```
[E,X] = eig(VCV);
L = diag(X);
```

Step 4: The modified covariance matrix [VCV_M] as a function of parameters [Lambda]

```
% Input: (1) Vector of eigenvalues [L] as a function of parameters [Lambda]
%        (2) Matrix of eigenvectors [E] as a function of parameters [Lambda]
% Output: Modified covariance matrix [VCV_M] as a function of parameters [Lambda]

% Step 4 contains sub-algorithm for eliminating eigenvectors associated with negative eigenvalues
```

Algorithms for step 4:

```
for i = 1 : m
  if L(i) < 0
    L_check(i) = 1;
  else
    L_check(i) = 0;
  end
end

% Matrix [E_sqrL] constructed by multiplying eigenvectors by square root of associated positive eigenvalues
for i = 1 : m
  if L_check(i) == 0
    for j = 1 : m
      E_sqrL(j, i) = E(j, i)*sqrt(L(i));
    end
  else
    for j = 1 : m
      E_sqrL(j,i) = 0;
    end
  end
end
VCV_M = E_sqrL*E_sqrL'; % symbol' denotes transposition
```

Step 5: Calculation of theoretical swaption volatilities [Sig_theo]

```
% Input: (1) Matrix of parameters [R]
%        (2) Matrix of modified covariance [VCV_M]
% Output: Matrix of theoretical swaption volatilities [Sig_theo]
```

Algorithm for step 5:

```
Sig_theo=[];
for k = 1:m
  for N = k + 1 : m + 1
    Sum = 0;
    for l = k + 1:N
      for i = k + 1:N
        SumTemp = R(k, N, i)*VCV_M(i − 1, l − 1)*R(k, N, l);
        Sum = Sum + SumTemp;
      end
    end
    Sig_theo(k,N-k)=sqrt(Sum*Lambda(k)/yearfrac(Today,T_Num(k)));
  end
end
```

Step 6: RSME between theoretical and market swaption volatilities

```
% Input: (1) Matrix of theoretical swaption volatilities [Sig_theo]
%        (2) Matrix of market swaption volatilities [Sig]
% Output: RSME between theoretical and market swaption volatilities
```

Algorithm for step 6:

```
RSME = 0;
for i = 1:m
  for j = 1 : m − i + 1
    RSME_Temp = (Sig_theo(i, j)-Sig(i, j))^2;
    RSME = RSME+RSME_Temp;
  end
end
f = RSME;  % function f will be used as a minimization function
```

For the purpose of optimization we set initial values of parameters [Lambda] as:

```
Lambda0 = [1 2 3 4 5 6 7 8 9 10];
```

Having that we use Matlab function @fminsearch dedicated for nonlinear optimization:

```
[Lambda, f] = fminsearch(@CalibrationObjectiveFunction_SeparatedOptim, Lambda0);
```

Results of computations:

1) Parameters [Lambda]

Lambda =

2.1848
3.4058
3.3030
2.7767
2.4863
2.5656
2.7588
2.5325
1.6783
0.4483

2) Vector of eigenvalues [L]

L =

−0.0083
0.0000
0.0070
0.0119
0.0216
0.0290
0.0706
0.1367
0.2696
0.4927

3) Matrix of eigenvectors

E =

0.3073	0.1854	0.5379	−0.1028	−0.4686	0.5464	0.1659	−0.0605	0.1412	−0.0452
−0.3571	−0.6145	−0.1988	0.3945	−0.1505	0.4219	0.1730	−0.0667	0.2305	−0.1076
0.2648	−0.4985	−0.1433	−0.6669	−0.1641	−0.1958	−0.0486	−0.2608	0.2673	−0.0944
−0.3471	−0.1183	0.6026	0.1006	0.1183	−0.1818	−0.4200	−0.5105	0.0325	−0.0846
0.4086	0.0957	−0.3578	0.4094	−0.3322	−0.0654	−0.1795	−0.5742	−0.2140	0.0769
−0.4287	0.2228	−0.2551	−0.4496	0.0612	0.4319	0.0629	−0.3769	−0.3991	0.0611
0.3816	−0.3520	0.2430	0.0668	0.5225	0.0929	0.4200	−0.1711	−0.4204	0.0541
−0.2701	−0.1437	0.1694	−0.0167	−0.5571	−0.4390	0.3956	0.0861	−0.4587	0.0009
0.1275	−0.3141	0.0304	−0.0581	−0.1240	0.2421	−0.6213	0.3915	−0.5069	−0.1030
−0.0426	−0.1531	0.0753	−0.0311	−0.0632	0.0361	−0.0936	0.0400	0.0729	0.9732

4) Matrix of covariances [VCV]

VCV =

0.0236	0.0216	0.0120	0.0014	−0.0078	−0.0058	−0.0148	−0.0135	−0.0226	−0.0187
0.0216	0.0295	0.0194	0.0020	−0.0104	−0.0218	−0.0217	−0.0292	−0.0338	−0.0483
0.0120	0.0194	0.0397	0.0259	−0.0002	−0.0163	−0.0321	−0.0325	−0.0443	−0.0408
0.0014	0.0020	0.0259	0.0548	0.0400	0.0134	−0.0034	−0.0210	−0.0102	−0.0402
−0.0078	−0.0104	−0.0002	0.0400	0.0666	0.0528	0.0289	0.0200	0.0021	0.0309
−0.0058	−0.0218	−0.0163	0.0134	0.0528	0.0713	0.0600	0.0393	0.0321	0.0192
−0.0148	−0.0217	−0.0321	−0.0034	0.0289	0.0600	0.0709	0.0554	0.0260	0.0136
−0.0135	−0.0292	−0.0325	−0.0210	0.0200	0.0393	0.0554	0.0807	0.0486	−0.0104
−0.0226	−0.0338	−0.0443	−0.0102	0.0021	0.0321	0.0260	0.0486	0.1246	−0.0526
−0.0187	−0.0483	−0.0408	−0.0402	0.0309	0.0192	0.0136	−0.0104	−0.0526	0.4691

5) Matrix of modified covariances [VCV_M]

VCV_M=

0.0244	0.0207	0.0127	0.0005	−0.0067	−0.0069	−0.0138	−0.0142	−0.0222	−0.0188
0.0207	0.0305	0.0186	0.0031	−0.0116	−0.0205	−0.0229	−0.0284	−0.0341	−0.0482
0.0127	0.0186	0.0403	0.0251	0.0007	−0.0172	−0.0313	−0.0331	−0.0441	−0.0409
0.0005	0.0031	0.0251	0.0558	0.0388	0.0146	−0.0045	−0.0202	−0.0106	−0.0400
−0.0067	−0.0116	0.0007	0.0388	0.0680	0.0513	0.0302	0.0190	0.0026	0.0308
−0.0069	−0.0205	−0.0172	0.0146	0.0513	0.0728	0.0586	0.0403	0.0316	0.0194
−0.0138	−0.0229	−0.0313	−0.0045	0.0302	0.0586	0.0721	0.0545	0.0264	0.0135
−0.0142	−0.0284	−0.0331	−0.0202	0.0190	0.0403	0.0545	0.0813	0.0484	−0.0103
−0.0222	−0.0341	−0.0441	−0.0106	0.0026	0.0316	0.0264	0.0484	0.1248	−0.0526
−0.0188	−0.0482	−0.0409	−0.0400	0.0308	0.0194	0.0135	−0.0103	−0.0526	0.4691

6) Root mean squared error for differences between theoretical and market swaption volatilities

RSME = 4.6066e-005

7) Theoretical swaptions volatilities [Sig_theo]

Sig_theo=

0.2308	0.2301	0.2212	0.2091	0.1961	0.1861	0.1760	0.1690	0.1630	0.1590
0.2280	0.2150	0.2057	0.1940	0.1833	0.1740	0.1671	0.1620	0.1580	
0.2105	0.2010	0.1903	0.1800	0.1701	0.1630	0.1580	0.1550		
0.1968	0.1870	0.1772	0.1680	0.1600	0.1550	0.1510			
0.1839	0.1740	0.1652	0.1580	0.1510	0.1480				
0.1765	0.1674	0.1591	0.1524	0.1462					
0.1686	0.1608	0.1531	0.1468						
0.1604	0.1542	0.1470							
0.1525	0.1476								
0.1450									

8) Market swaptions volatilities [Sig]

Sig =

0.2270	0.2300	0.2210	0.2090	0.1960	0.1860	0.1760	0.1690	0.1630	0.1590
0.2240	0.2150	0.2050	0.1940	0.1830	0.1740	0.1670	0.1620	0.1580	
0.2090	0.2010	0.1900	0.1800	0.1700	0.1630	0.1580	0.1550		
0.1950	0.1870	0.1770	0.1680	0.1600	0.1550	0.1510			
0.1820	0.1740	0.1650	0.1580	0.1510	0.1480				
0.1746	0.1674	0.1590	0.1524	0.1462					
0.1672	0.1608	0.1530	0.1468						
0.1598	0.1542	0.1470							
0.1524	0.1476								
0.1450									

Comments

If we analyse the eigenvalues and eigenvectors we can see that we have obtained only one negative eigenvalue and of very small absolute value. If we take the absolute values of all eigenvalues, the negative eigenvalue will have eighth biggest value from the set of ten values. What is more, the first three biggest eigenvalues (0.4927, 0.2696, 0.1367) have much bigger values than other eigenvalues. The eigenvectors associated with first three biggest eigenvalues are presented on Figure 8.3:

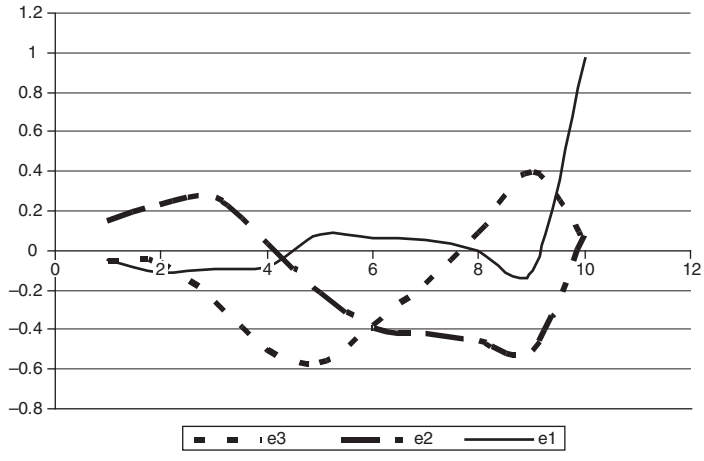


Figure 8.3 Eigenvectors associated with first three biggest eigenvalues for optimized Λ_i.

Although the eigenvectors do not have the typical humps presented in many books, these values generate very small differences between the theoretical and swaption volatilities.

The biggest differences are denoted for swaptions with one year length underlying swaps. There are practically no other significant differences in the volatilities for other maturities and underlying lengths. This suggests that this kind of calibration may be widely used in practice for valuation of various interest rate derivatives.

Let us see a graphical representation of optimized parameters Λ_i [Lambda].

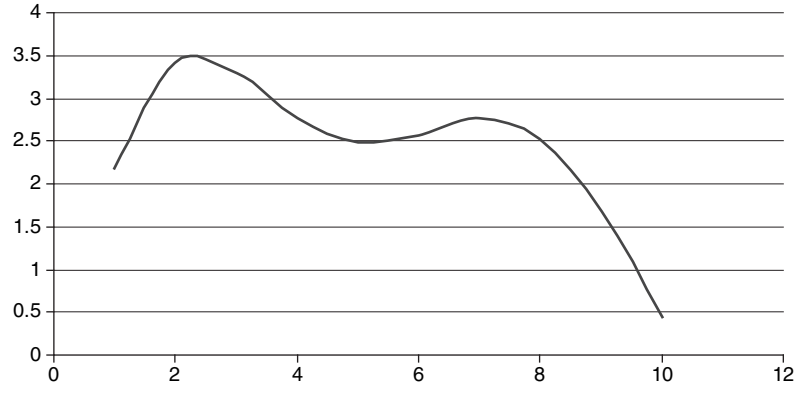


Figure 8.4 Parameters Λ_i obtained through optimization.

The values obtained through optimization are much different than for any arbitrary chosen function Λ_i presented in section 8.2. For such optimized parameters Λ_i give much a better RSME levels than previously presented. Recall that setting $\Lambda_i = \delta_i$ we had a RSME = 0.0013, setting $\Lambda_i = \sqrt{\delta_i}$ we have had a RSME = 0.0019, and in result of optimization we have obtained a RSME = 0.000046066.

Having presented the separated approach with optimization let us move on now to another widely used approach to the calibration to swaptions – which is called the locally single factor approach.

8.4 THE LOCALLY SINGLE FACTOR APPROACH

The locally single factor approach is based upon the assumption that the covariance matrices Φ^i are of single factor which means that $\varphi^i_{kl} - \varphi^{i-1}_{kl} = \varphi^i_k \varphi^i_l$ and $\varphi^{-1}_{kl} = 0$. Using this assumption we can write:

$$\varphi^k_{k+1} = \sqrt{\delta_k \sigma^2_{k,k+1} - \sum_{j=0}^{k-1} \left(\varphi^j_{k+1}\right)^2}, \quad \varphi^k_N = \frac{\sqrt{\delta_k \sigma^2_{k,N} - \sum_{j=0}^{k-1} \left(\sum_{i=k+1}^{N} R^i_{kN}(0) \varphi^j_i \right)^2} - \sum_{i=k+1}^{n-1} R^i_{kN}(0)\, \varphi^k_i}{R^N_{kN}(0)}$$

where

$$\delta_k = \delta(0 \div T_k) = \delta(k)$$

The matrix of market swaption volatilities can be arranged in a slightly different way.

$$\Sigma^{MKT} = \begin{bmatrix} \sigma^{MKT}_{0,1} & \sigma^{MKT}_{0,2} & \sigma^{MKT}_{0,3} & \cdots & \sigma^{MKT}_{0,m} & - & - & - & - \\ - & \sigma^{MKT}_{1,2} & \sigma^{MKT}_{1,3} & \cdots & \sigma^{MKT}_{1,m} & \sigma^{MKT}_{1,m+1} & - & - & - \\ - & - & \sigma^{MKT}_{2,3} & \cdots & \sigma^{MKT}_{2,m} & \sigma^{MKT}_{2,m+1} & \sigma^{MKT}_{2,m+2} & - & - \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ - & - & - & \cdots & \sigma^{MKT}_{m-1,m} & \sigma^{MKT}_{m-1,m+1} & \sigma^{MKT}_{m-1,m+2} & \cdots & \sigma^{MKT}_{m-1,M} \end{bmatrix}_{m \times M}$$

Such arrangement allows us to construct a calibration algorithm in a little easier for practical implementation. The interpretation of the subscripts are presented below:

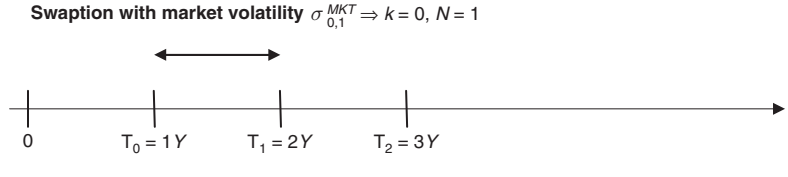


Figure 8.5 Interpretation of swaption volatility.

Let us go to a practical example. We take market data for a swaption and appropriate discount factors only up to three years.

Example 8.2 Locally single factor calibration

Table 8.16 presents initial market data

Table 8.16 Market data – swaption volatility, discount factors

Swaption volatility		Underlying swap length		Discount factor
		$1Y$	$2Y$	
Option maturity	T0 = 1Y	22.70 %	23.00 %	0.9774658
	T1 = 2Y	22.40 %	21.50 %	0.9509789
	T2 = 3Y			0.9219838

For k = 0, N = 1 we have

$$\varphi_1^0 = \sqrt{\delta_0 \sigma_{0,1}^2 - \sum_{j=0}^{-1} \left(\varphi_1^j\right)^2} = \sqrt{\delta_0}\sigma_{0,1} = \sqrt{1} \cdot 22.70\% = 22.70\,\%.$$

For k = 0, N = 2 we have

$$\varphi_2^0 = \frac{\sqrt{\delta_0 \left(\sigma_{0,2}\right)^2 - \sum_{j=0}^{-1} \left(\sum_{i=1}^{2} R_{0,2}^i(0)\,\varphi_i^j\right)^2} - \sum_{i=1}^{1} R_{0,2}^i(0)\,\varphi_i^0}{R_{0,2}^2(0)} =$$

$$= \frac{\sqrt{\delta_0 \left(\sigma_{0,2}\right)^2} - R_{0,2}^1(0)\,\varphi_1^0}{R_{0,2}^2(0)} = \frac{\sqrt{1} \cdot 23.00\,\% - 0.4774 \cdot 22.70\,\%}{0.5226} = 23.27\,\%.$$

For k = 1, N = 2 we have

$$\varphi_2^1 = \sqrt{\delta_1 \sigma_{1,2}^2 - \sum_{j=0}^{0} \left(\varphi_2^j\right)^2} = \sqrt{\delta_1 \left(\sigma_{1,2}\right)^2 - \left(\varphi_2^0\right)^2} = \sqrt{2 \cdot 22.40\,\%^2 - 23.27\,\%^2} = 21.49\,\%.$$

Finally we obtain

$$\varphi_{1,1}^0 - \varphi_{1,1}^{-1} = \varphi_1^0 \varphi_1^0 \Rightarrow \varphi_{1,1}^0 = \left(\varphi_1^0\right)^2 = 0.227^2 = 0.051529$$
$$\varphi_{1,2}^0 - \varphi_{1,2}^{-1} = \varphi_1^0 \varphi_2^0 \Rightarrow \varphi_{1,2}^0 = \varphi_1^0 \varphi_2^0 = 0.227 \cdot 0.2327 = 0.052832$$
$$\varphi_{2,2}^0 - \varphi_{2,2}^{-1} = \varphi_2^0 \varphi_2^0 \Rightarrow \varphi_{2,2}^0 = \left(\varphi_2^0\right)^2 = 0.2327^2 = 0.054168$$
$$\varphi_{2,2}^1 - \varphi_{2,2}^0 = \varphi_2^1 \varphi_2^1 \Rightarrow \varphi_{2,2}^1 = \left(\varphi_2^1\right)^2 + \varphi_{2,2}^0 = 0.2149^2 + 0.054168 = 0.100352$$

Going further we generate matrices Φ^i for the rest of the swaptions. These is shown below.

Table 8.17 Matrix with elements φ_k^i.

i/k	0	1	2	3	4	5	6	7	8	9
1	22.70 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
2	23.27 %	21.49 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
3	20.48 %	20.85 %	21.36 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
4	17.65 %	19.60 %	20.64 %	20.03 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
5	14.90 %	17.69 %	17.94 %	20.83 %	19.14 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
6	14.08 %	14.32 %	16.91 %	17.55 %	19.74 %	21.05 %	0.00 %	0.00 %	0.00 %	0.00 %
7	12.06 %	14.16 %	13.14 %	16.95 %	16.72 %	21.51 %	20.24 %	0.00 %	0.00 %	0.00 %
8	12.35 %	13.04 %	13.32 %	12.88 %	16.83 %	16.53 %	21.56 %	18.91 %	0.00 %	0.00 %
9	11.57 %	13.72 %	12.84 %	13.51 %	9.73 %	17.39 %	16.28 %	21.67 %	17.09 %	0.00 %
10	12.11 %	12.62 %	14.82 %	10.53 %	15.66 %	7.05 %	16.47 %	15.65 %	22.13 %	12.92 %

Table 8.18 Matrices Φ^i

i = 0	1	2	3	4	5	6	7	8	9	10
1	5.15 %									
2	5.28 %	5.42 %								
3	4.65 %	4.77 %	4.20 %							
4	4.01 %	4.11 %	3.62 %	3.12 %						
5	3.38 %	3.47 %	3.05 %	2.63 %	2.22 %					
6	3.20 %	3.28 %	2.88 %	2.49 %	2.10 %	1.98 %				
7	2.74 %	2.81 %	2.47 %	2.13 %	1.80 %	1.70 %	1.46 %			
8	2.80 %	2.87 %	2.53 %	2.18 %	1.84 %	1.74 %	1.49 %	1.52 %		
9	2.63 %	2.69 %	2.37 %	2.04 %	1.72 %	1.63 %	1.40 %	1.43 %	1.34 %	
10	2.75 %	2.82 %	2.48 %	2.14 %	1.81 %	1.71 %	1.46 %	1.50 %	1.40 %	1.47 %
i = 1	1	2	3	4	5	6	7	8	9	10
2		10.04 %								
3		9.25 %	8.54 %							
4		8.32 %	7.70 %	6.96 %						
5		7.27 %	6.74 %	6.10 %	5.35 %					
6		6.35 %	5.87 %	5.29 %	4.63 %	4.03 %				
7		5.85 %	5.42 %	4.90 %	4.30 %	3.73 %	3.46 %			
8		5.68 %	5.25 %	4.74 %	4.15 %	3.61 %	3.34 %	3.23 %		
9		5.64 %	5.23 %	4.73 %	4.15 %	3.59 %	3.34 %	3.22 %	3.22 %	
10		5.53 %	5.11 %	4.61 %	4.04 %	3.51 %	3.25 %	3.14 %	3.13 %	3.06 %
										…
i = 10	1	2	3	4	5	6	7	8	9	10
10										21.04 %

Having this data, the instantaneous volatility vectors will have the following values:

Table 8.19 Instantaneous volatility vectors for locally single factor calibration

Index	$\gamma_1(t)$	$\gamma_2(t)$	$\gamma_3(t)$	$\gamma_4(t)$	$\gamma_5(t)$	$\gamma_6(t)$	$\gamma_7(t)$	$\gamma_8(t)$	$\gamma_9(t)$	$\gamma_{10}(t)$
1	22.70 %	23.27 %	20.48 %	17.65 %	14.90 %	14.08 %	12.06 %	12.35 %	11.57 %	12.11 %
2	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
...										
10	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %

So in this case the instantaneous volatilities will be equal to the market swaption volatilities $\sigma_{i,i+1}^{MKT}$ for $i = 0, T_1, T_2, \ldots, T_9$.

Another popular way of calibration is calibration to swaptions using the historically computed correlations of forward rates.

8.5 CALIBRATION WITH HISTORICAL CORRELATIONS OF FORWARD RATES

Before we present the calibration algorithm we have to do preliminary computations. We have to compute a matrix of historical correlations.

The matrix of historical correlations will be computed using daily EUR interest rates taken from deposits and IRS (in both cases mid rates). The time series starts at 29-10-1999 and ends at 21-01-2005. Below is a graph presenting an example of historical EUR interest rates for 3 month deposit and 5Y IRS.

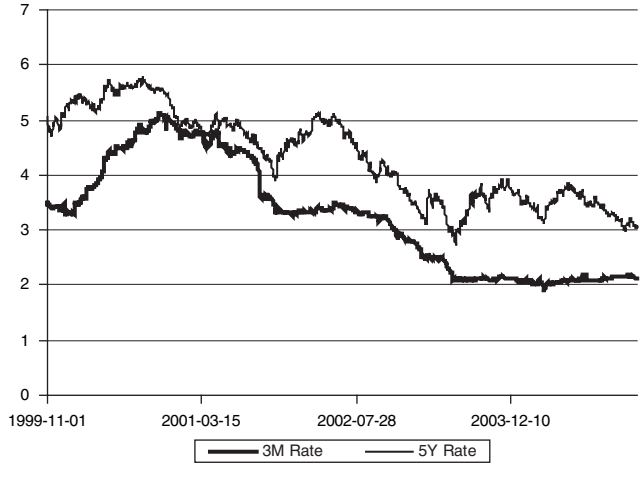


Figure 8.6 Historical EUR interest rates for 3 month deposit and 5 year IRS.

We will have to compute correlations between the rates of return from forward rates for EUR. We select the following intervals for forward rates.

3M-6M	4Y-5Y	9Y-10Y	15Y-16Y
6M-9M	5Y-6Y	10Y-11Y	16Y-17Y
9M-1Y	6Y-7Y	11Y-12Y	17Y-18Y
1Y-2Y	7Y-8Y	12Y-13Y	18Y-19Y
2Y-3Y	8Y-9Y	13Y-14Y	19Y-20Y
3Y-4Y		14Y-15Y	

All market data used in calculations was taken from Reuters.

The set of data consisted of daily-quoted closing bid and ask par rates for euro from $t = 29$ October 1999 to 20 January 2005 for the following tenors T_i

3M	2Y	6Y	10Y
6M	3Y	7Y	12Y
9M	4Y	8Y	15Y
1Y	5Y	9Y	20Y

The rates for tenors from 3M to 1Y came from the deposit market; the rates for tenors from 2Y to 20Y came from the IRS market.

Any missing records in the data, separately for bid and ask series, were interpolated using the formula

$$\widetilde{R}(t, t, T_i) = R(t-1, t-1, T_i)\left(1 + P_{t;t-1}\right)$$

where t indexes consecutive days, i indexes tenors T_i, $\widetilde{R}(t, t, T_i)$ is the interpolated par rate for day t and tenor T_i, $R(t-1, t-1, T_i)$ is the par rate taken from the previous day with the same tenor T_i and $P_{t;t-1}$ is the percentage market move, averaged across all bid or ask percentage changes for the rest of tenors

$$P_{t;t-1} = \frac{1}{n_{J^t}} \sum_{i \in J^t}^{n_{J^t}} p^i_{t;t-1} = \frac{1}{n_{J^t}} \sum_{i \in J^t}^{n_{J^t}} \frac{R(t, t, T_i) - R(t-1, t-1, T_i)}{R(t-1, t-1, T_i)} \times 100\,\%$$

In the above formula, J^t is the set of all i's for which data exists for day t, n_{J^t} is the total number of elements in set J^t and $p^i_{t;t-1}$ is the percentage change in $R(t, t, T_i)$, i.e. the change of par rate between day t and day $t-1$ for tenor T_i.

After interpolating missing records, we obtained mid rates by averaging bid and ask rates for each tenor T_i

$$\mathbf{r}^i_m = \frac{\mathbf{r}^i_b + \mathbf{r}^i_a}{2}$$

where $\mathbf{r}^i_m$, $\mathbf{r}^i_b$ and $\mathbf{r}^i_a$ are vectors of mid, bid and ask rates, respectively.

The next step is to calculate par rates for the non-standard tenors: 11Y, 13Y, 14Y, 16Y, 17Y, 18Y and 19Y. We use linear interpolation

$$R(t, t, T_i) = \frac{R\left(t, t, T_{i_2}\right)\left(T_i - T_{i_1}\right) + R\left(t, t, T_{i_1}\right)\left(T_{i_2} - T_i\right)}{T_{i_2} - T_{i_2}}.$$

In the above expression, $R(t, t, T_i)$ is the par rate being interpolated for tenor T_i and $R\left(t, t, T_{i_1}\right)$, $R\left(t, t, T_{i_2}\right)$ are known par rates for tenor T_{i_1} and T_{i_2}, respectively. If, for example, $T_i = 17\text{Y}$, then $T_{i_1} = 15\text{Y}$ and $T_{i_2} = 20\text{Y}$.

The discount factors were calculated as follows.

As par rates for euros are quoted by Reuters in the Act/360 convention, the discount factors up to 1Y were determined using the following formula

$$B(t, T_i) = \left(1 + R(t, t, T_i) \times \frac{T^j}{360}\right)^{-1}$$

where $B(t, T_i)$ denotes the discount factor for day t and for tenor T_i. In this case we have $T_i = 90, 180, 270$ and 360.

For tenors 2Y and above, discount factors were calculated using the standard bootstrapping technique

$$B(t, T_i) = \frac{1 - R(t, t, T_i) \times \sum_{k=1}^{j-1} B(t, T_k)}{1 + B(t, T_k)}$$

The discount factors for future periods were calculated using the relationship:

$$B(i, T_i, T_{i+1}) = \frac{B(t, T_{i+1})}{B(t, T_i)}$$

where $B(t, T_i, T_{i+1})$ denotes the discount factor for future period $T_i \div T_{i+1}$.

Now that all the discount factors have been calculated for the future periods, we are able to calculate forward rates. In continuous compounding, forward rates f for tenors up to 1Y may be expressed as

$$B(t, T_i, T_{i+1}) = e^{-\frac{T_i}{360} F\left(t, T_i, T_{i+1}\right)} \Rightarrow L(t, T_i, T_{i+1}) = -\frac{360}{T^j} \ln B(t, T_i, T_{i+1})$$

where $T_i = 90$, 180, 270 and 360. For tenors 2Y and above, we can write

$$B(t, T_i, T_{i+1}) = e^{-\frac{1Y}{360} \times F\left(t, T_i, T_{i+1}\right)} \Rightarrow L(t, T_i, T_{i+1}) = -\ln B(t, T_i, T_{i+1}).$$

Next we calculate rates of return s from the forward rates. We defined the rates of return as follows

$$s_t^i = \ln \frac{L(t, T_i, T_{i+1})}{L(t-1, T_i, T_{i+1})}.$$

Finally we calculate correlations between the rates of return from the forward rates

$$Corr(\mathbf{s}^i, \mathbf{s}^k) = \frac{\sum_{t=1}^{N} (s_t^i - \bar{s}^i) \times (s_t^k - \bar{s}^k)}{\sqrt{\sum_{t=1}^{N} (s_t^i - \bar{s}^i)^2 \times \sum_{i=1}^{N} (s_t^k - \bar{s}^k)^2}}. \tag{8.2}$$

In the above formula $\bar{s}^i$ denotes the average of all rates of return in the series corresponding to tenor T_i.

The resulting correlation matrix is presented below.

Table 8.20 Historical correlation matrix of forward rates

	3M–6M	6M–9M	9M–1Y	1–2Y	2–3Y	3–4Y	4–5Y	5–6Y	6–7Y	7–8Y	8–9Y	9–10Y
3M–6M	1.000	0.128	0.214	0.159	0.223	0.200	0.155	0.082	0.046	0.020	0.074	−0.008
6M–9M	0.128	1.000	−0.072	0.179	0.199	0.200	0.215	0.065	0.062	0.056	0.069	−0.025
9M–1Y	0.214	−0.072	1.000	−0.016	0.202	0.164	0.126	0.050	0.041	0.115	0.023	−0.004
1Y–2Y	0.159	0.179	−0.016	1.000	0.513	0.379	0.354	0.259	0.134	0.124	0.155	−0.005
2Y–3Y	0.223	0.199	0.202	0.513	1.000	0.208	0.194	0.168	0.128	0.204	0.059	0.002
3Y–4Y	0.200	0.200	0.164	0.379	0.208	1.000	0.149	0.142	0.164	0.072	0.110	0.013
4Y–5Y	0.155	0.215	0.126	0.354	0.194	0.149	1.000	−0.176	0.133	0.019	0.161	−0.010
5Y–6Y	0.082	0.065	0.050	0.259	0.168	0.142	−0.176	1.000	−0.121	0.260	0.055	0.007
6Y–7Y	0.046	0.062	0.041	0.134	0.128	0.164	0.133	−0.121	1.000	−0.128	0.062	0.015
7Y–8Y	0.020	0.056	0.115	0.124	0.204	0.072	0.019	0.260	−0.128	1.000	−0.359	−0.011
8Y–9Y	0.074	0.069	0.023	0.155	0.059	0.110	0.161	0.055	0.062	−0.359	1.000	−0.709
9–10Y	−0.008	−0.025	−0.004	−0.005	0.002	0.013	−0.010	0.007	0.015	−0.011	−0.709	1.000
14–15Y	0.071	0.131	0.086	0.308	0.310	0.296	0.222	0.249	0.180	0.193	0.123	0.018
19–20Y	0.041	−0.003	0.066	0.126	0.113	0.167	0.114	0.128	0.199	0.126	0.075	0.023

	10Y–11Y	11Y–12Y	12Y–13Y	13Y–14Y	14Y–15Y	15Y–16Y	16Y–17Y	17Y–18Y	18Y–19Y	19Y–20Y	10Y–11Y	11Y–12Y
3M–6M	1.000	0.128	0.214	0.159	0.223	0.200	0.155	0.082	0.046	0.020	0.074	−0.008
6M–9M	0.128	1.000	−0.072	0.179	0.199	0.200	0.215	0.065	0.062	0.056	0.069	−0.025
9M–1Y	0.214	−0.072	1.000	−0.016	0.202	0.164	0.126	0.050	0.041	0.115	0.023	−0.004
1Y–2Y	0.159	0.179	−0.016	1.000	0.513	0.379	0.354	0.259	0.134	0.124	0.155	−0.005
2Y–3Y	0.223	0.199	0.202	0.513	1.000	0.208	0.194	0.168	0.128	0.204	0.059	0.002
3Y–4Y	0.200	0.200	0.164	0.379	0.208	1.000	0.149	0.142	0.164	0.072	0.110	0.013
4Y–5Y	0.155	0.215	0.126	0.354	0.194	0.149	1.000	−0.176	0.133	0.019	0.161	−0.010
5Y–6Y	0.082	0.065	0.050	0.259	0.168	0.142	−0.176	1.000	−0.121	0.260	0.055	0.007
6Y–7Y	0.046	0.062	0.041	0.134	0.128	0.164	0.133	−0.121	1.000	−0.128	0.062	0.015
7Y–8Y	0.020	0.056	0.115	0.124	0.204	0.072	0.019	0.260	−0.128	1.000	−0.359	−0.011
8Y–9Y	0.074	0.069	0.023	0.155	0.059	0.110	0.161	0.055	0.062	−0.359	1.000	−0.709
9–10Y	−0.008	−0.025	−0.004	−0.005	0.002	0.013	−0.010	0.007	0.015	−0.011	−0.709	1.000
14–15Y	0.071	0.131	0.086	0.308	0.310	0.296	0.222	0.249	0.180	0.193	0.123	0.018
19–20Y	0.041	−0.003	0.066	0.126	0.113	0.167	0.114	0.128	0.199	0.126	0.075	0.023

The results seem to be a little unexpected. We have obtained a lot of negative historical correlations. Perhaps this is due to the fact that market of forward rates is not as effective as everybody thinks. The presence of negative correlations creates a possibility of statistical arbitrage. That means one can construct instruments that will use the information about negative historical correlations and make an arbitrage opportunity if any counterparty does not have information about the presented fact.

Let us denote the computed correlation matrix by Ψ. Our goal is to obtain the VCV matrices. We may use for that universal volatility function presented in Chapter 4 'Swaption Pricing and Calibration'. Using this we specify formulae for φ_{m+1}^m:

$$\varphi_1^0 = \sigma_{01}, \quad \varphi_{k+1}^k = \sqrt{\delta_k \sigma_{k,k+1}^2 - \sum_{j=0}^{k-1} \left(\varphi_{k+1}^j\right)^2}$$

and the following quadratic equation for φ_N^m:

$$\left(\varphi_N^m R_{mN}^N(0)\right)^2 + 2\varphi_N^m R_{mN}^N(0) \sum_{i=m+1}^{N-1} R_{mN}^i(0)\psi_{iN}\varphi_i^m - \delta_m \sigma_{mN}^2 =$$

$$-\sum_{j=0}^{m-1}\sum_{l=m+1}^{N}\sum_{i=m+1}^{N} R^i_{mN}(0)\varphi^j_l\psi_{il}\varphi^j_i R^l_{mN}(0) - \sum_{l=m+1}^{N-1}\sum_{i=m+1}^{N-1} R^i_{mN}(0)\varphi^m_l\psi_{il}\varphi^m_i R^l_{mN}(0).$$

Let us present a practical example showing the necessary steps to compute the required VCV matrices:

Example 8.3 Calibration with historical correlations

Table 8.21 presents the initial market data for our example. The data is identical as for locally single factor calibration.

Table 8.21 Market data - swaption volatility, discount factors

Swaption volatility		Underlying swap length		Discount fact.
		1Y	2Y	
Option maturity	T0 = 1Y	22.70 %	23.00 %	0.9774658
	T1 = 2Y	22.40 %	21.50 %	0.9509789
	T2 = 3Y			0.9219838

For k = 0, N = 1 we have

$$\varphi^0_1 = \sqrt{\delta_0\sigma^2_{0,1} - \sum_{j=0}^{-1}\left(\varphi^j_1\right)^2} = \sqrt{\delta_0}\sigma_{0,1} = \sqrt{1}\cdot 22.70\,\% = 22.70\,\%.$$

For k = 0, N = 2 we have

$$\left(\varphi^0_2\right)^2\left(R^2_{0,2}(0)\right)^2 + 2\varphi^0_2 R^2_{0,2}(0)\sum_{i=1}^{1} R^i_{0,2}(0)\,\psi_{i,2}\varphi^0_i - \delta_0\left(\sigma_{0,2}\right)^2 =$$

$$\sum_{j=0}^{-1}\sum_{l=1}^{2}\sum_{i=1}^{2} R^i_{0,2}(0)\,\varphi^j_l\psi_{i,l}\varphi^j_i R^l_{0,2}(0) - \sum_{l=1}^{1}\sum_{i=1}^{1} R^i_{0,2}(0)\,\varphi^0_l\psi_{i,l}\varphi^0_i R^l_{0,2}(0)$$

$$\left(\varphi^0_2\right)^2\left(R^2_{0,2}(0)\right)^2 + 2\varphi^0_2 R^2_{0,2}(0)\,R^1_{0,2}(0)\,\psi_{1,2}\varphi^0_1 - \delta_0\left(\sigma_{0,2}\right)^2 + R^1_{0,2}(0)\,\varphi^0_1\psi_{1,1}\varphi^0_1 R^1_{0,2}(0) = 0 \Rightarrow$$

$$0.5226^2\cdot\left(\varphi^0_2\right)^2 + 2\cdot 0.5226\cdot 0.4774\cdot 0.5133\cdot 22.70\,\%\cdot\left(\varphi^0_2\right) - 1\cdot(23.00\,\%)^2$$
$$+ 0.4774^2\cdot 22.70\,\%^2\cdot 1 = 0$$

$$0.2731\cdot\left(\varphi^0_2\right)^2 + 0.0581\cdot\left(\varphi^0_2\right) - 0.0412 = 0 \Rightarrow \varphi^0_2 = -50.90\,\% \text{ or } \varphi^0_2 = 29.61\,\%$$

For our calibration case we take the positive value $\varphi^0_2 = 29.61\,\%$.

For k = 1, N = 2 we have

$$\varphi^1_2 = \sqrt{\delta_1\sigma^2_{1,2} - \sum_{j=0}^{0}\left(\varphi^j_2\right)^2} = \sqrt{\delta_1\left(\sigma_{1,2}\right)^2 - \left(\varphi^0_2\right)^2} = \sqrt{2\cdot 22.40\,\%^2 - 29.61\,\%^2} = 11.26\,\%$$

Finally we obtain:

$$\varphi_{1,1}^0 - \varphi_{1,1}^{-1} = \varphi_1^0\varphi_1^0 \Rightarrow \varphi_{1,1}^0 = \left(\varphi_1^0\right)^2 = 0.227^2 = 0.051529$$
$$\varphi_{1,2}^0 - \varphi_{1,2}^{-1} = \varphi_1^0\varphi_2^0 \Rightarrow \varphi_{1,2}^0 = \varphi_1^0\varphi_2^0 = 0.227 \cdot 0.2961 = 0.067209$$
$$\varphi_{2,2}^0 - \varphi_{2,2}^{-1} = \varphi_2^0\varphi_2^0 \Rightarrow \varphi_{2,2}^0 = \left(\varphi_2^0\right)^2 = 0.2961^2 = 0.087661$$
$$\varphi_{2,2}^1 - \varphi_{2,2}^0 = \varphi_2^1\varphi_2^1 \Rightarrow \varphi_{2,2}^1 = \left(\varphi_2^1\right)^2 + \varphi_{2,2}^0 = 0.1126^2 + 0.087661 = 0.100340$$

We can go further and compute the whole matrix with elements φ_N^m. For this we can specify parameters *a,b,c* in the following way:

$$a = (R_{mN}^N(0))^2$$
$$b = 2R_{mN}^N(0) \sum_{i=m+1}^{N-1} R_{mN}^i(0)\psi_{iN}\varphi_i^m$$

$$c = \sum_{j=0}^{m-1}\sum_{l=m+1}^{N}\sum_{i=m+1}^{N} R_{mN}^i(0)\varphi_l^j\psi_{il}\varphi_i^j R_{mN}^l(0) + \sum_{l=m+1}^{N-1}\sum_{i=m+1}^{n-1} R_{mN}^i(0)\varphi_l^m\psi_{il}\varphi_i^m R_{mN}^l(0) - \delta_m\sigma_{mN}^2$$

and compute

$$\Delta = b^2 - 4ac$$
$$(\varphi_N^m)_1 = \frac{-b-\sqrt{\Delta}}{2a}$$
$$(\varphi_N^m)_2 = \frac{-b-\sqrt{\Delta}}{2a}$$

i.e. solutions of a quadratic equation. However one can find that for some specific market data (swaption volatilities) and historical correlations the parameter Δ may have a negative value. This is the case for φ_3^2 for our market data taken from 25 January 2005. Let us see that ($k = 2,\ N = 3$):

$$\varphi_3^2 = \sqrt{\delta_2\sigma_{2,3}^2 - \sum_{j=0}^{1}\left(\varphi_3^j\right)^2} = \sqrt{\delta_2\sigma_{2,3}^2 - \left(\varphi_3^0\right)^2 - \left(\varphi_3^1\right)^2} = \sqrt{3 \cdot 20.90\,\%^2 - 34.63\,\%^2 - 29.34\,\%^2}$$
$$= \sqrt{-7.50\,\%}$$

We have obtained square root from a negative value and the whole algorithm will collapse for that set of market data. On the other hand this situation is not a general rule which means that for another set of data the algorithm may give proper results.

We now present another calibration algorithm to co-terminal swaptions.

8.6 CALIBRATION TO CO-TERMINAL SWAPTIONS

Another widely used technique for BGM calibration is calibration to co-terminal swaptions. Figure 8.7 presents the idea of that calibration.

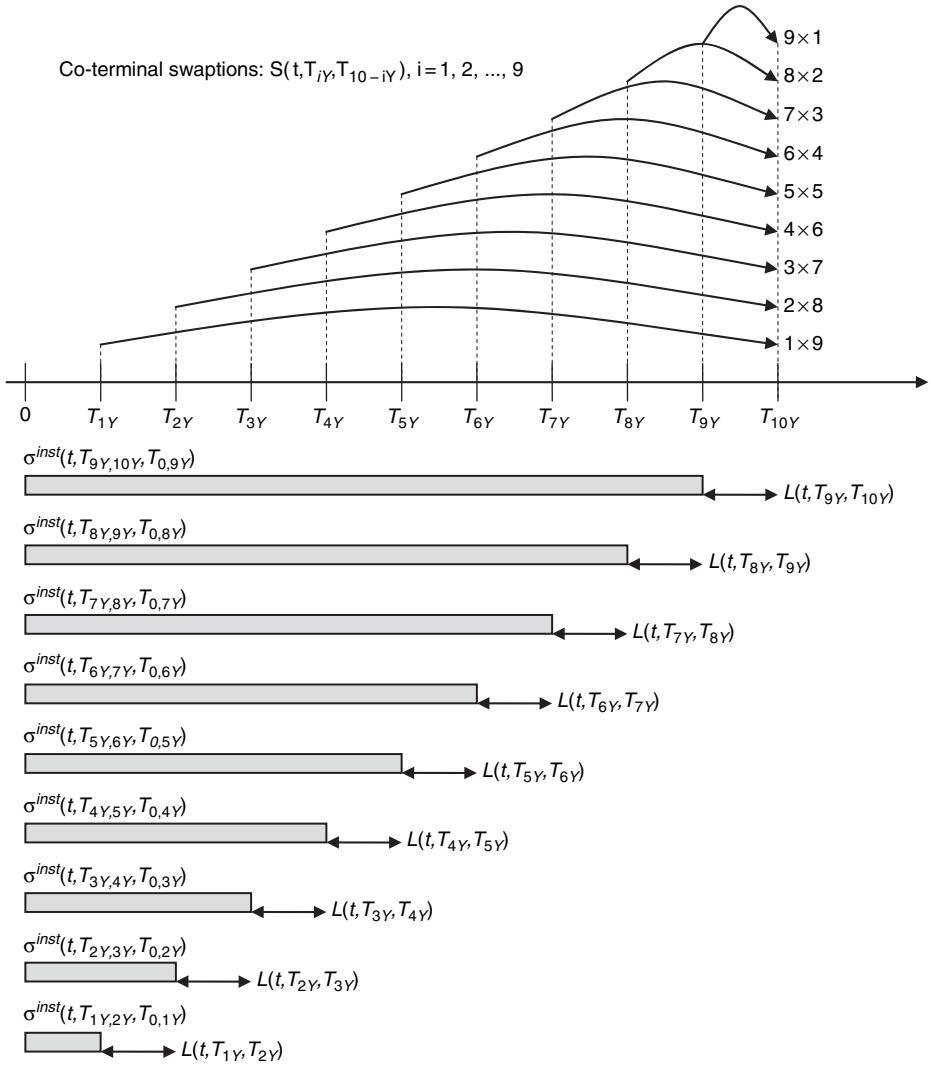


Figure 8.7 Set of co-terminal swaptions and instantaneous volatilities of forward rates.

We can see in the figure a set of nine co-terminal swaptions. Co-terminal swaptions are swaptions which expire at the same time (in our case T_{10Y}) and maturities increases through time, e.g. by one year per swaption. In our case we consider the following co-terminal swaptions: $9 \times 1,\ 8 \times 2,\ 7 \times 3,\ 6 \times 4,\ 5 \times 5,\ 4 \times 6,\ 3 \times 7,\ 2 \times 8,\ 1 \times 9$. Table 8.22 presents also a set of forward rates $L\left(t, T_{iY}, T_{i+1Y}\right)$ and corresponding instantaneous volatilities of $\sigma^{inst}\left(t, T_{iY,i+1Y}, T_{0,iY}\right)$ which are constant through periods $0 \div iY$.

The first step of the calibration will be deriving the forward swap rates for the considered co-terminal swaptions.

Step 1: Derivation of the forward swap rates for a co-terminal swaption

Because we have derived forward swap rates in one of our previous exercises, we present now only the results of the derivation. Table 8.22 below presents forward swap rates and additionally forward rates together with ATM swaption volatilities.

Table 8.22 Forward rates, swap rates and swaption ATM volatilities

Forward rate		Swap rate		Swaption volatility	
$L(0, T_{9Y}, T_{10Y})$	4.534 %	$S(0, T_{9Y}, T_{10Y})$	4.534 %	$\sigma^{swpt}(0, T_{9Y}, T_{10Y})$	15.24 %
$L(0, T_{8Y}, T_{9Y})$	4.520 %	$S(0, T_{8Y}, T_{10Y})$	4.527 %	$\sigma^{swpt}(0, T_{8Y}, T_{10Y})$	15.42 %
$L(0, T_{7Y}, T_{8Y})$	4.312 %	$S(0, T_{7Y}, T_{10Y})$	4.452 %	$\sigma^{swpt}(0, T_{7Y}, T_{10Y})$	15.30 %
$L(0, T_{6Y}, T_{7Y})$	4.152 %	$S(0, T_{6Y}, T_{10Y})$	4.372 %	$\sigma^{swpt}(0, T_{6Y}, T_{10Y})$	15.24 %
$L(0, T_{5Y}, T_{6Y})$	3.906 %	$S(0, T_{5Y}, T_{10Y})$	4.271 %	$\sigma^{swpt}(0, T_{5Y}, T_{10Y})$	15.10 %
$L(0, T_{4Y}, T_{5Y})$	3.675 %	$S(0, T_{4Y}, T_{10Y})$	4.161 %	$\sigma^{swpt}(0, T_{4Y}, T_{10Y})$	15.50 %
$L(0, T_{3Y}, T_{4Y})$	3.400 %	$S(0, T_{3Y}, T_{10Y})$	4.039 %	$\sigma^{swpt}(0, T_{3Y}, T_{10Y})$	15.80 %
$L(0, T_{2Y}, T_{3Y})$	3.102 %	$S(0, T_{2Y}, T_{10Y})$	3.906 %	$\sigma^{swpt}(0, T_{2Y}, T_{10Y})$	16.20 %
$L(0, T_{1Y}, T_{2Y})$	2.747 %	$S(0, T_{1Y}, T_{10Y})$	3.758 %	$\sigma^{swpt}(0, T_{1Y}, T_{10Y})$	16.30 %

Let us remember also results of the weights used for forward swap derivation that will be used in the calibration.

Table 8.23 Weights for swap rates

	$2y$	$3y$	$4y$	$5y$	$6y$	$7y$	$8y$	$9y$	$10y$
$9y \times 1y$									1
$8y \times 2y$								0.509736	0.490264
$7y \times 3y$							0.347558	0.332573	0.319869
$6y \times 4y$						0.266116	0.255067	0.24407	0.234747
$5y \times 5y$					0.216549	0.208489	0.199832	0.191217	0.183912
$4y \times 6y$				0.18453	0.176589	0.170017	0.162957	0.155931	0.149975
$3y \times 7y$			0.160158	0.154976	0.148307	0.142787	0.136858	0.130958	0.125955
$2y \times 8y$		0.142109	0.137398	0.132953	0.127231	0.122496	0.11741	0.112347	0.108056
$1y \times 9y$	0.127793	0.123949	0.11984	0.115962	0.110972	0.106842	0.102405	9.80E-02	0.094247

Now we are ready to move to step 2 of the co-terminal calibration. This step is very similar to the bootstrapping technique widely used in the market.

Step 2: The Bootstrapping technique for instantaneous volatility

We start our algorithm from the derivation of instantaneous volatility for the forward rate $F(t, T_{9Y}, T_{10Y})$. We assume that such volatility will be constant through all the period before our forward rate will be known, i.e. in the period: $0 \div 9Y$. We will use previously computed approximations of swaption volatility by instantaneous volatility of forward rates. For the first period the computations will be straightforward, and the volatilities for all other periods $0 \div 8Y$ for rate $F(t, T_{8Y}, T_{9Y})$, $0 \div 7Y$ for rate $F(t, T_{7Y}, T_{8Y})$ and so on, will be computed based on the results obtained in the previous steps. We also assume, that all instantaneous correlations between the forward rates are equal to one. Let us start our computations:

Period: $0 \div 9Y$ for forward rate $F(t, T_{9Y}, T_{10Y})$

The market swaption volatility will be approximated by:

$$\begin{aligned}\sigma^{swpt}\left(t, T_{9Y}, T_{10Y}\right)^2 &= \frac{w_{10Y}^{10Y,10Y}(0)^2 L\left(0, T_{9Y}, T_{10Y}\right)^2}{\delta_{0,9Y} S\left(0, T_{9Y}, T_{10Y}\right)^2}\delta_{0,9Y}\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)^2 \\ &= \frac{1^2 \cdot 4.534^2 \cdot 1}{4.534^2}\cdot\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)^2 = \sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)^2 \\ &\Rightarrow \sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right) = 15.25\,\%\end{aligned}$$

Period: $0 \div 8Y$ for forward rate $F\left(t, T_{8Y}, T_{9Y}\right)$

$$\begin{aligned}\sigma^{swpt}\left(t, T_{8Y}, T_{10Y}\right)^2 &= \frac{w_{9Y}^{9Y,10Y}(0)^2 L\left(0, T_{8Y}, T_{9Y}\right)^2}{S\left(0, T_{8Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right)^2 \\ &+ \frac{w_{10Y}^{9Y,10Y}(0)^2 L\left(0, T_{9Y}, T_{10Y}\right)^2}{S\left(0, T_{8Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)^2 \\ &+ 2\cdot\frac{w_{9Y}^{9Y,10Y}(0)\, w_{10Y}^{9Y,10Y}(0) L\left(0, T_{8Y}, T_{9Y}\right) L\left(0, T_{9Y}, T_{10Y}\right)}{S\left(0, T_{8Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) \\ &\times \sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right).\end{aligned}$$

We have obtained a quadratic equation with unknown $\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right)$. The value of instantaneous volatility $\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)$ has been obtained from the previous step. We can rearrange the above equation into a classic quadratic equation. Putting in the real market data we can write:

$$\begin{aligned}&A_{8Y,9Y}\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right)^2 + B_{8Y,9Y}\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) + C_{8Y,9Y} = 0 \Leftrightarrow \\ &\Leftrightarrow 0.0005309\cdot\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right)^2 + 0.0001561\cdot\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) - 0.0000373 = 0 \Leftrightarrow \\ &\Leftrightarrow \sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) = 15.06\,\% \quad \text{or} \quad \sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) = -75.85\,\%\end{aligned}$$

We take into account only positive values. We show below the required computations in the next period. Other computations will be very similar.

Period: $0 \div 7Y$ for forward rate $F\left(t, T_{7Y}, T_{8Y}\right)$

$$\begin{aligned}\sigma^{swpt}\left(t, T_{7Y}, T_{10Y}\right)^2 &= \frac{w_{8Y}^{8Y,10Y}(0)^2 L\left(0, T_{7Y}, T_{8Y}\right)^2}{S\left(0, T_{7Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{7Y,8Y}, T_{0,7Y}\right)^2 \\ &+ \frac{w_{9Y}^{8Y,10Y}(0)^2 L\left(0, T_{8Y}, T_{9Y}\right)^2}{S\left(0, T_{7Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right)^2 \\ &+ \frac{w_{10Y}^{8Y,10Y}(0)^2 L\left(0, T_{9Y}, T_{10Y}\right)^2}{S\left(0, T_{7Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)^2 \\ &+ 2\cdot\frac{w_{8Y}^{8Y,10Y}(0)\, w_{9Y}^{8Y,10Y}(0) L\left(0, T_{7Y}, T_{8Y}\right) L\left(0, T_{8Y}, T_{9Y}\right)}{S\left(0, T_{8Y}, T_{10Y}\right)^2}\sigma^{inst}\left(0, T_{7Y,8Y}, T_{0,7Y}\right) \\ &\times \sigma^{inst}\left(0, T_{8Y,9Y}, T_{0,8Y}\right) + 2\cdot\frac{w_{8Y}^{8Y,10Y}(0)\, w_{10Y}^{8Y,10Y}(0) L\left(0, T_{7Y}, T_{8Y}\right) L\left(0, T_{9Y}, T_{10Y}\right)}{S\left(0, T_{8Y}, T_{10Y}\right)^2} \\ &\times \sigma^{inst}\left(0, T_{7Y,8Y}, T_{0,7Y}\right)\sigma^{inst}\left(0, T_{9Y,10Y}, T_{0,9Y}\right)\end{aligned}$$

$$+2\cdot\frac{w_{9Y}^{8Y,10Y}(0)\,w_{10Y}^{8Y,10Y}(0)\,L(0,T_{8Y},T_{9Y})\,L(0,T_{9Y},T_{10Y})}{S(0,T_{8Y},T_{10Y})^2}\sigma^{inst}\left(0,T_{8Y,9Y},T_{0,8Y}\right)$$
$$\times\,\sigma^{inst}\left(0,T_{9Y,10Y},T_{0,9Y}\right)$$

Once again we have obtained a quadratic equation with unknown $\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)$. In this step the values of the instantaneous volatilities $\sigma^{inst}\left(0,T_{9Y,10Y},T_{0,9Y}\right)$ and $\sigma^{inst}\left(0,T_{8Y,9Y},T_{0,8Y}\right)$ have been obtained from previous steps. We can rearrange above equation into a classic quadratic equation. Substituting in real market data we can write:

$$A_{7Y,8Y}\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)^2+B_{7Y,8Y}\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)+C_{7Y,8Y}=0\Leftrightarrow$$
$$\Leftrightarrow 0.0002246\cdot\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)^2+0.0001365\cdot\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)-0.0000257=0\Leftrightarrow$$
$$\Leftrightarrow\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)=15.06\,\%\text{ or }\sigma^{inst}\left(0,T_{7Y,8Y},T_{0,7Y}\right)=-105.55\,\%$$

Again we take into account only positive values.

The results for other periods are presented in Table 8.24.

Table 8.24 Calibration results to co-terminal swaptions

i	$A_{iY,i+1Y}$	$B_{iY,i+1Y}$	$C_{iY,i+1Y}$	$\sigma^{inst}\left(0,T_{iY,i+1Y},T_{0,iY}\right)$	
$9Y$				15.24 %	
$8Y$	0.0005309	0.0001561	−0.0000373	15.59 %	−45.00 %
$7Y$	0.0002246	0.0001365	−0.0000257	15.06 %	−75.85 %
$6Y$	0.0001221	0.0001105	−0.0000194	15.06 %	−105.55 %
$5Y$	0.0000716	0.0000883	−0.0000143	14.49 %	−137.91 %
$4Y$	0.0000460	0.0000713	−0.0000140	17.60 %	−172.61 %
$3Y$	0.0000296	0.0000594	−0.0000114	17.62 %	−218.04 %
$2Y$	0.0000194	0.0000482	−0.0000101	19.38 %	−267.58 %
$1Y$	0.0000123	0.0000387	−0.0000071	17.36 %	−331.63 %

Our results seem to be sensible. We can observe the typical hump of volatility between years 1 and 3. The result is similar to that obtained via calibration to caps.

8.7 CONCLUSIONS

This chapter was dedicated to presenting calibration algorithms to caps and swaptions. All the algorithms were non parametric. We have started with a separated approach. Based on the market data taken from particular working day we have obtained results of the calibration as a set of covariance matrices. Many of the matrices had negative eigenvalues so we have transformed them by eliminating the eigenvectors associated with the negative eigenvalues. Our results seem to be right for some of the variants of the calibration. For two parameters of Λ_i we have presented the algorithms of calibration in more detail. For these two cases we have showed how to compute instantaneous volatility vectors. We also compared the

root mean squared errors between the theoretical and market swaption volatilities. The best results occurred for two forms of parameters Λ_i when equal to time δ_i specified as a day count fraction and next when equal to a squared root of time $\sqrt{\delta_i}$.

In the next section the separated approach was developed further. As a target function we have set a root mean squared error for differences between theoretical and market swaption volatilities. We have minimized that function under several restrictions for VCV. We have postulated that the VCV matrix must be positive definite. For that case we have implemented a subalgorithm for reducing VCV matrix by removing eigenvectors associated with negative eigenvalues. Results here were very good. There were much better than for any arbitrary chosen function Λ_i.

In the chapter we have also presented an approach called locally single factor. After defining initial assumptions and the constituting algorithm we have moved to results. In this simple calibration we have obtained instantaneous volatilities of forward rates equal to one year swaptions. This will be always the case as long as we are interested in obtaining instantaneous volatilities for one-year forward rates and we choose for calibration swaptions with underlying swaps length also equal to one year and paying once a year.

After that we have moved into calibration to swaptions given an exogenously instantaneous correlation matrix of forward LIBOR rates. The correlations were computed from historical data. We have presented the method of obtaining historical correlations and also computed the correlation matrix for particular working day. The results are completely unexpected. We have seen forward correlations that are close to one. Unfortunately there were a lot of negative correlations. One reason may be that the historical data is not totally accurate. The data was taken from one of most popular financial services as closing prices for LIBOR's and IRS quotations. We used a bootstrapping technique to obtain zero coupon rates and after that forward zero coupon rates. The data has daily frequency and it is possible that some marginal computation technique during interpolation caused it to have too much influence on the final results. Nevertheless one should be very careful if it was decided to use this technique in practice. It is worthy to mention that some other research obtains quite good results of historical correlations but they used data from zero coupon bonds. We still think that taking data from IRS market should be more appropriate because of the nature of LIBOR Market Model which is dedicated mainly to the money market.

The last part of the chapter was dedicated to calibration to co-terminal swaptions. Co-terminal swaptions were such swaptions which expire at the same time and maturities increased through time, e.g. by one year per swaption. We have presented all necessary steps for calibration and computed the final result. All the results seem to be very sensible. We have observed a typical hump of volatility between years 1 and 3. The results were also very similar to obtained via calibration only to caps using assumption of volatility dependency only on time to maturity.

Please be aware that the presented algorithms should be used in practice for those interest rate instruments which by nature are close to the swaption market. A classical example may be any Bermudan swaption. An example of pricing a Bermudan swaption using calibration to co-terminal swaption is presented in Chapter 10.

We have presented many variations of non-parametric calibration to swaptions. This is a good time now to move onto parametric calibration algorithms to caps and swaptions simultaneously based on optimization techniques. In the next chapter we present at the beginning non parametric calibration to caps and swaptions based on the Rebonato approach and then simultaneous parametric calibration to caps and swaptions.