Modelling Implied Volatility: Neural Network Method

YASHKIR CONSULTING www.yashkir.com

January 15, 2019

Contents

1 Introduction **1**

2 The problem formulation **2**

3 Minimization of the error function **3**
3.1 Backpropagation 4
3.2 Neural Network Iteration (Learning) Algorithm 5

4 Historical data preparation **6**

5 Dual Currency Curve Simulation **7**

6 Objective function for calibration **7**

7 Calibration Algorithm **9**

8 Calibration examples **11**
8.1 Single yield curve calibration 11
8.2 Dual Yield curve Calibration 16

9 Summary **23**

10 References **23**

1 Introduction

Implied volatility surface (Heath et al., 1990) is used for pricing vanilla options given price/strike ratio S/K and time to maturity τ. As an example, consider a call option according to Black-Scholes model:

$$C_t = S_t \cdot N(d_1) - K \cdot e^{-r\tau} \cdot N(d_2) \tag{1}$$

$$d_1 = \frac{\log \frac{S_t}{K} + (r + 1/2\sigma^2) \cdot \tau}{\sigma\sqrt{\tau}} \tag{2}$$

$$d_2 = d_1 - \sigma\sqrt{\tau} \tag{3}$$

To simplify historical data usage we introduce the "discounted" strike price

$$\kappa = K \cdot e^{-r\tau} \tag{4}$$

which transforms equation (1) into:

$$C_t = S_t \cdot N(d_1) - \kappa \cdot N(d_2) \tag{5}$$

$$d_1 = \frac{\log \frac{S_t}{\kappa} + 1/2 \cdot \sigma^2 \cdot \tau}{\sigma\sqrt{\tau}} \tag{6}$$

$$d_2 = d_1 - \sigma\sqrt{\tau} \tag{7}$$

It is common practice to quote the option price in terms of implied volatility σ. Therefore, the raw market data has the following format

Historical data:

Date	Strike	Time-to-maturity	Implied volatility
t_0	κ_0	τ_0	σ_0
$\cdots$	$\cdots$	$\cdots$	$\cdots$
t_{i-1}	κ_{i-1}	τ_{i-1}	σ_{i-1}
t_i	κ_i	τ_i	σ_i
$\cdots$	$\cdots$	$\cdots$	$\cdots$

Using historical raw data we construct the data block for the model training as follows:

Training data

κ_0	τ_0	σ_0	σ_1
$\cdots$	$\cdots$	$\cdots$	$\cdots$
κ_i	τ_i	σ_{i-1}	σ_i
κ_{i+1}	τ_{i+1}	σ_i	σ_{i+1}
$\cdots$	$\cdots$	$\cdots$	$\cdots$

In short, the i^{th} sample is: $\vec{x}_i = \{\kappa_i, \tau_i, \sigma_{i-1}, \sigma_i\}$. Note that we included the price (implied volatility) of the previous date in order to account for day-to-day correlation.

2 The problem formulation

Given historical data (option price, spot price, strike price, time to maturity) for certain period of time, to build a model for implied volatility as function of spot/strike ratio κ and time to maturity T. File with historical data in general case can be formatted as follows:

Input data file:

x_{00}	$\cdots$	x_{0j}	$\cdots$	$x_{0,J-1}$	y_{00}	$\cdots$	y_{0i}	$\cdots$	$y_{0,I-1}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n0}	$\cdots$	x_{nj}	$\cdots$	$x_{n,J-1}$	y_{n0}	$\cdots$	y_{ni}	$\cdots$	$y_{n,I-1}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$x_{N-1,0}$	$\cdots$	$x_{N-1,j}$	$\cdots$	$x_{N-1,J-1}$	$y_{N-1,0}$	$\cdots$	$y_{N-1,i}$	$\cdots$	$y_{N-1,I-1}$

In this file the number of samples is N, the number of input variables is J, the number of output values is I. To build a model we will use deep (with the number of layers L) neural network

structure. The first (input) layer has J nodes, hidden layers may have different number of nodes, but we'll assume that the number of nodes in all layers is the same J, except in output layer with I nodes.
The index n indicates a sample, parameters $\vec{x}_n$ contain the spot price/strike ratio, time-to-maturity, and some more. Values $\vec{y}_n$ contain historical implied volatilities, and may have some other variables.
The propagation of the input signal x_{nj} (or $\mathbf{x}$) is as follows:

1. Nodes of the input layer ($l = 0$) "emit" values $a_{n0j} \equiv x_{nj}$

2. Layer l ($l = 1 \cdots L-1$):
   Output signals from nodes (neurons) of the layer $(l-1)$ are combined as biased weighted average values for inputs into nodes j of the layer l

$$z_{nlj} = \sum_{k=0}^{J-1} a_{n,l-1,k} w_{lkj} + b_{lj} \tag{8}$$

   (Note: within sample n the vector/matrix form would be $\vec{z}_{nl} = \vec{a}_{n,l-1}\mathbf{w}_l + \vec{b}_l$. Weight matrices $\mathbf{w}_l$ and biases $\vec{b}_l$ will be found as a result of the neural network iterative "training".)
   The output of the j^{th}-neuron of the layer l is emitted as nonlinear transformation of $\vec{z}_{n,l}$

$$a_{nlj} = \sigma(z_{nlj}) \tag{9}$$

   where function $\sigma()$ is, for example, a sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{10}$$

3. Output layer $(L-1)$: ...
   As the result of the propagation of the input signal x_{nj} through neural network we have $a_{n,L-1,i}$ $(i = 0 \cdots, I-1)$, see the equation (9).
   The modelling error for all samples is defined as:

$$E = \frac{1}{2} \sum_n \sum_{i=0}^{I-1} (a_{n,L-1,i} - y_{ni})^2 \tag{11}$$

The problem, therefor, can be formulated as follows:

> To find neural network parameters (weight matrices $\mathbf{w}_l$ and biases $\vec{b}_l$) by minimizing the error function E (11).

3 Minimization of the error function

The minimum of the error function can be found using gradient descent algorithm for which partial derivatives of (11) by components of $\mathbf{w}_l$ and $\vec{b}_l$ must be calculated using backpropagation procedure.

3.1 Backpropagation

1. Output layer $(L-1)$.
   Derivatives by weighted averages $z_{n,L-1,j}$

$$\delta_{n,L-1,j} = \frac{\partial E}{\partial z_{n,L-1,j}} = \frac{\partial E}{\partial a_{n,L-1,j}} \cdot \frac{\partial a_{n,L-1,j}}{\partial z_{n,L-1,j}} = \frac{\partial E}{\partial a_{n,L-1,j}} \cdot \frac{\partial \sigma(z_{n,L-1,j})}{\partial z_{n,L-1,j}} \tag{12}$$

   where

$$\frac{\partial E}{\partial a_{n,L-1,j}} = \frac{\partial}{\partial a_{n,L-1,j}} \left[\frac{1}{2} \sum_k \left(a_{n,L-1,k} - y_{nk} \right)^2 \right] = a_{n,L-1,j} - y_{nj} \tag{13}$$

$$\frac{\partial \sigma(z_{n,L-1,j})}{\partial z_{n,L-1,j}} \equiv \sigma^{'}(z_{n,L-1,j}) \tag{14}$$

   where

$$\sigma^{'}(z) = \sigma(z)(1 - \sigma(z)) \tag{15}$$

2. Hidden layers $(l = L-2, \cdots, 1)$; backpropagation:

$$\delta_{n,l,j} = \frac{\partial E}{\partial z_{nlj}} = \sum_k \frac{\partial E}{\partial z_{n,l+1,k}} \frac{\partial z_{n,l+1,k}}{\partial z_{n,l,j}} = \sum_k \delta_{n,l+1,k} \frac{\partial}{\partial z_{nlj}} \left(\sum_i a_{nli} w_{i,l+1,k} + b_{l+1,k} \right) \tag{16}$$

$$= \sum_k \delta_{n,l+1,k} \frac{\partial}{\partial z_{nlj}} \left(\sum_i \sigma(z_{nli}) w_{i,l+1,k} + b_{l+1,k} \right) = \sum_k \delta_{n,l+1,k} w_{j,l+1,k} \sigma^{'}(z_{nlj}) \tag{17}$$

3. Derivatives by bias variables $\vec{b}_l \quad (l = L-1, \ldots 1)$:

$$\frac{\partial E}{\partial b_{lj}} = \frac{\partial E}{\partial a_{nlj}} \frac{\partial a_{nlj}}{\partial z_{nlj}} \frac{\partial z_{nlj}}{\partial b_{lj}} = \delta_{nlj} \quad \text{(for a single sample n)} \tag{18}$$

$$\beta_{lj} = \frac{1}{N} \sum_{n=0}^{N-1} \delta_{nlj} \quad \text{(averaged over all samples)} \tag{19}$$

4. Derivatives by weight matrix components:

   Partial derivative within n-th sample: (20)

$$\frac{\partial E}{\partial w_{lkj}} = \frac{\partial E}{\partial z_{nlj}} \frac{\partial z_{nlj}}{\partial w_{lkj}} = \delta_{nlj} \frac{\partial}{\partial w_{lkj}} \left(\sum_i a_{n,l-1,i} + b_{lj} \right) = \delta_{nlj} a_{n,l-1,k} \tag{21}$$

   Derivative (21) averaged over all samples: (22)

$$v_{lkj} = \frac{1}{N} \sum_{n=0}^{N-1} \delta_{nlj} a_{n,l-1,k} \tag{23}$$

5. Gradient descent step:

$$w_{lkj} = w_{lkj} - h \cdot v_{lkj} \tag{24}$$

$$b_{lj} = b_{lj} - h \cdot \beta_{lj} \tag{25}$$

   Where h is the iteration step (learning rate)

3.2 Neural Network Iteration (Learning) Algorithm

There are two data blocks: N samples of learning (calibration) data and N^* samples of validation data.

1. Generate random values for matrix $\mathbf{w}$ and vector $\vec{b}$ as normal deviates
2. Fill in the input layer ($l = 0$) with input parameters $\vec{x}$:

$$a_{n0j} = x_{nj} \quad n = 0 \cdots N-1 \tag{26}$$
$$a^*_{n0j} = x^*_{nj} \quad n = 0 \cdots N^*-1 \tag{27}$$

Iteration process p:

1. Forward propagation through neural network:

$$z_{nlj} = \sum_{k=0}^{J-1} a_{n,l-1,k} w_{lkj} + b_{lj} \tag{28}$$
$$a_{nlj} = \sigma(z_{nlj}) \tag{29}$$
$$n = 0 \cdots (N-1), \quad l = 0 \cdots (L-1), \quad j = 0 \cdots (J-1)$$

$$z^*_{nlj} = \sum_{k=0}^{J-1} a^*_{n,l-1,k} w_{lkj} + b_{lj} \tag{30}$$
$$a^*_{nlj} = \sigma(z^*_{nlj}) \tag{31}$$
$$n = 0 \cdots (N^*-1), \quad l = 0 \cdots (L-1), \quad j = 0 \cdots (J-1)$$

2. Error values calculated:

$$E_p = \frac{1}{2N} \sum_{n=0}^{N-1} \sum_{i=0}^{I-1} (a_{n,L-1,i} - y_{ni})^2 \tag{32}$$
$$E^*_p = \frac{1}{2N^*} \sum_{n=0}^{N^*-1} \sum_{i=0}^{I-1} (a^*_{n,L-1,i} - y^*_{ni})^2 \tag{33}$$

3. Derivatives (by weighted average variables $\vec{z}$) of the error function E are calculated:

$$\delta_{n,L-1,i} = (a_{n,L-1,i} - y_{ni}) \cdot \sigma'(z_{n,L-1,i}) \tag{34}$$
$$n = 0 \cdots (N-1), \quad i = 0 \cdots (I)$$

$$\delta_{n,l,j} = \sum_k \delta_{n,l+1,k} w_{j,l+1,k} \sigma'(z_{nlj}) \tag{35}$$
$$n = 0 \cdots (N-1), \quad l = (L-1) \cdots 1, \quad j = 0 \cdots (J-1)$$

4. Derivatives by bias parameters $\vec{b}$ (averaged over all samples) are calculated:

$$\beta_{lj} = \frac{1}{N} \sum_{n=0}^{N-1} \delta_{n,l,j}, \quad l = 1 \cdots (L-1), \quad j = 0 \cdots (J-1) \tag{36}$$

5. Derivatives by components of the matrix **w** (averaged over all samples) are calculated:

$$v_{lkj} = \frac{1}{N}\sum_{n=0}^{N-1} \delta_{nlj} a_{n,l-1,k} \tag{37}$$
$$l = 1 \cdots (L-1), \quad j = 0 \cdots (J-1), \quad k = 0 \cdots (J-1)$$

6. Gradient descent step:

$$w_{lkj} = w_{lkj} - h \cdot v_{lkj} \tag{38}$$
$$b_{lj} = b_{lj} - h \cdot \beta_{lj} \tag{39}$$
$$l = 1 \cdots (L-1), \quad j = 0 \cdots (J-1), \quad k = 0 \cdots (J-1)$$

Next iteration p+1

4 Historical data preparation

5 Dual Currency Curve Simulation

Discount factors for both yield curves (we will use the notation of A and B (the first yield curve, or "base" curve: A, and the second yield curve, or "quoted" curve: B) must be simulated simultaneously with account of their correlation. Each random driver $\varepsilon_n^{'}$ for curve A is correlated to corresponding driver $\varepsilon_n^{"}$ for curve B. Therefore the dual-curve model takes the following form (see equation **??** for zero-coupon-bond values $B(t,T)$):

$$\ln B(t,T) = \ln B(s,T) - \ln B(s,t) + D(s,t,T) + \sum_{n=0}^{2} \sqrt{v_n(s,t,T)} \cdot \varphi_n(t) \tag{40}$$

$$\ln B^*(t,T) = \ln B^*(s,T) - \ln B^*(s,t) + D^*(s,t,T) + \sum_{n=0}^{2} \sqrt{v_n^*(s,t,T)} \cdot \varphi_n^*(t) \tag{41}$$

Here, the equation (40) is for the base rate model (A) and the equation (41) (with symbols *) is for the quoted (B) rate model. To satisfy HJM model conditions for each rate model the stochastic drivers φ_n and φ_n^* are constructed as follows:

$$\varphi_n(t) = \varepsilon_n^{'}\xi_n + \varepsilon_n^{"}\sqrt{1-\xi_n^2} \tag{42}$$

$$\varphi_n^*(t) = \varepsilon_n^{'}\sqrt{1-\xi_n^2} + \varepsilon_n^{"}\xi_n \tag{43}$$

Variables $\varepsilon_n{}^{',"}$ are independent normally distributed random numbers, and ξ_n are factors defining degree of correlation between the two rate models. We have

$$\begin{aligned} &\left\langle \varphi_n^2 \right\rangle = 1 \quad \left\langle \varphi_n \varphi_{m\neq n} \right\rangle = 0 \quad \left\langle \varphi_n^{*2} \right\rangle = 1 \quad \left\langle \varphi_n^* \varphi_{m\neq n}^* \right\rangle = 0 \\ &\rho^* = \left\langle \varphi_n \varphi_n^* \right\rangle = 2\xi_n\sqrt{1-\xi_n^2} \end{aligned} \tag{44}$$

Correlation parameter ρ^* varies from -1 to $+1$ which implies the following restriction on ξ_n: $-1/\sqrt{2} \leq \xi_n \leq 1/\sqrt{2}$.

6 Objective function for calibration

Next step is to "fit" the model to historical data. Assume that the historical yield rates

$$\begin{aligned} &\hat{r}_{\tau_k}(t_i) \\ &\hat{r}^*_{\tau_k}(t_i) \end{aligned} \tag{45}$$

are available for certain period of time $\{t_0 \cdots T_H\}$ for both yield curves. Here t_i are dates and τ_k are maturities ($k = 0, 1, 2...$ correspond to (for example) 1 day, 1 week, 1 month, etc.).
Taking starting point for rate simulation as $r_{\tau_k}(t_0)$ and $r^*_{\tau_k}(t_0)$ we perform simulation using Section (5) with the following initial condition:

$$\ln B(0,t_p) = -\overset{o}{r}_{t_p}(t_0) \cdot t_p \tag{46}$$

$$\ln B^*(0,t_p) = -\overset{o^*}{r}_{t_p}(t_0) \cdot t_p \tag{47}$$

where t_p are bond maturity times (corresponding to T in (40), the small enough step size (we'll use $t_p - t_{p-1} = 1$ day, for simplicity), and yield rates $\overset{o}{r}_{t_p}(t_0)$ and $\overset{o^*}{r}_{t_p}(t_0)$ are obtained by iterpolation of historical yields $\hat{r}_{\tau_k}(t_0)$ and $\hat{r}^*_{\tau_k}(t_0)$.

As a result, we will have simulated log bond values for given scenario j

$$\begin{matrix} lnB_j(t_i, t_p) \\ lnB_j(t_i, t_p) \end{matrix} \tag{48}$$

Next, for a given set of historical yield rate maturities τ_k we use (48) and (??) to simulate corresponding yield rates

$$\begin{matrix} r_j(t_i, t_p) \\ r_j^*(t_i, t_p) \end{matrix} \tag{49}$$

The calibration procedure must reduce as much as possible the aggregated mismatch between simulated (49) and historical (45) yield rates at all time points $\{t_0 \cdots T_H\}$, all yield types τ_k, and all Monte Carlo scenarios.
For this purpose we will use the maximum likelihood criterium (see ((Fisher, 1922))). For given scenario j, time point t_i, and maturity t_p the probability of the (random) simulated rate to fit the corresponding historical rate is given by Gaussian distribution function

$$f(r, \hat{r}, \theta) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(-\frac{(r-\hat{r})^2}{2\theta^2}\right) \tag{50}$$

where variable θ is unknown. Overall probability (likelihood value) is therefore:

$$P(\vec{z}) = \prod_{i,k,j} \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(-\frac{\left[r_j(t_i, t_p) - \hat{r}(t_i, t_p)\right]^2}{2\theta^2}\right) \tag{51}$$

where array $\vec{z}$ consists of parameters (22 in total):

$$\vec{z} = \left[\vec{\kappa}^A, \vec{\sigma}^A, \vec{\rho}^{A*}, \theta, \vec{\kappa}^B, \vec{\sigma}^B, \vec{\rho}^{B*}, \vec{\xi}\right] \tag{52}$$

Taking logarithm of (51) and dividing by total number of fitting points M, we obtain the log-likelihood function

$$F(\vec{z}) = -\ln\sqrt{2\pi\theta^2} - \frac{1}{M}\sum_{i,k,j} \frac{\left[r_j(t_i, t_p) - \hat{r}(t_i, t_p)\right]^2}{2\theta^2} \tag{53}$$

The number of fitting points M is the total number of sum items in (53). Note, that the variance θ^2 is not the model parameter.
The calibration task is now reduced to the search of the objective function maximum in $\vec{z}$-space. Maximum of likelihood function (53) can be found using some numerical minimization algorithm for the objective function $-F(\vec{z})$. We use standard Nelder–Mead method (downhill simplex or amoeba method)((Aldrich, 1997)). This method is for unconstrained minimization, therefore we introduce constraints by adding penalty function to $-F(\vec{z})$:

$$\Omega(\vec{z}) = \sum_k \left\{\max(0, z_k^{min} - z_k) + \max(0, z_k - z_k^{max})\right\} \cdot \Upsilon \tag{54}$$

where constraints $\vec{z}_k^{min} < z_k < z_k^{max}$, and penalty weight Υ are user-defined.
The calibration objective function expression takes finally the following form:

$$\begin{aligned} Q(\vec{z}) = &\ln\sqrt{2\pi\theta^2} + \frac{1}{M}\sum_{i,k,j} \frac{\left[r_j(t_i, t_p) - \hat{r}(t_i, t_p)\right]^2}{2\theta^2} \\ &+ \sum_k \left\{\max(0, z_k^{min} - z_k) + \max(0, z_k - z_k^{max})\right\} \cdot \Upsilon \end{aligned} \tag{55}$$

7 Calibration Algorithm

Calibration of the model is based on historical data, such as (for example) daily quoted yield GBP rates from 2001-01-02 to 2018-10-26 (1 day, 1 week, 1 month, 3 months, 6 months, 1 year). This data is used in the following format:

1	7	21	42	63	126	252
0.0581094	0.0583031	0.0591828	0.0591156	0.0590531	0.0584922	0.0579141
0.060975	0.06	0.0593813	0.05915	0.0590328	0.05835	0.057625
0.0557125	0.0579375	0.0589375	0.05875	0.0584922	0.0575125	0.0567969
...	...	...	...	...	...	...
0.006845	0.00706	0.0072038	0.0075175	0.0080806	0.0090831	0.0105656
0.0068488	0.007055	0.0072163	0.0075425	0.0080869	0.0091225	0.0107
0.0068425	0.0070838	0.0072475	0.0075388	0.0081019	0.0091138	0.0105338

Table 1: Historical (2001-01-02 to 2018-10-26) GBP yield rates input file. The first line (header)contains maturities (in days). Each row contains rates for given date (for maturities: 1 day, 21 days, etc.; as in the header)

Corresponding data for EUR curves:

1	7	21	42	63	126	252
0.0484125	0.0484500	0.0484375	0.0484000	0.0483250	0.0479000	0.0468750
0.0483250	0.0483500	0.0483688	0.0481938	0.0480125	0.0472125	0.0461688
0.0482250	0.0481625	0.0478688	0.0474938	0.0473500	0.0462625	0.0452188
...	...	...	...	...	...	...
-0.0045557	-0.0043800	-0.0040643	-0.0036743	-0.0034771	-0.0032257	-0.0020871
-0.0045500	-0.0043700	-0.0040586	-0.0036614	-0.0034900	-0.0032300	-0.0020871
-0.0045529	-0.0043714	-0.0040543	-0.0036671	-0.0035486	-0.0032657	-0.0020914

Table 2: Historical (2001-01-02 to 2018-10-26) EUR yield rates input file. Each row contains rates for given date (for maturities: 1 day, 21 days, etc.; see the header)

The following time scale is set up:

$$\begin{aligned} \Delta t = 1 &\text{ (time step is 1 day)} \\ t_{start} &\text{ (starting date corresponding to t=0)} \\ t_{fit} &\text{ (number of days for fitting model to historical rates)} \\ t_{max} = t_{fit} + \tau_{max} &\text{ (number of days for simulation)} \\ \tau_{max} &\text{ (maximal maturity)} \\ \text{(Time unit is 1 year)} & \end{aligned}$$

For calibration procedure the initial values of parameters and their constraints are user-defined. For example,

for currency A (GBP):			
Parameter	Starting value	Minimum	Maximum
κ_0	**1.2**	0.80	1.5
κ_1	**1.2**	0.80	1.5
κ_2	**0.050**	0.01	0.15
σ_0	**0.040**	0.002	0.05
σ_1	**0.002**	0.001	0.05
σ_2	**0.002**	0.001	0.01
ρ_{01}	**-0.20**	-0.03	0.20
ρ_{02}	**-0.10**	-0.20	0.10
ρ_{12}	**-0.10**	-0.50	0.10
θ	**0.100**	0.002	0.25

for currency B (EUR):			
Parameter	Starting value	Minimum	Maximum
κ_0	**1.2**	0.80	1.5
κ_1	**1.2**	0.80	1.5
κ_2	**0.050**	0.01	0.15
σ_0	**0.040**	0.002	0.05
σ_1	**0.002**	0.001	0.05
σ_2	**0.002**	0.001	0.01
ρ_{01}	**-0.20**	-0.03	0.20
ρ_{02}	**-0.10**	-0.20	0.10
ρ_{12}	**-0.10**	-0.50	0.10
ξ_0	**-0.10**	-0.25	0.05
ξ_1	**-0.17**	-0.40	0.00
ξ_2	**-0.50**	-0.90	0.00

The minimum of the objective function (55) is searched using simplex Nelder-Mead algorithm which requires setting the following parameters

Parameter	Comment
Υ	**Penalty weight**
$\vec{\delta}$	**Nelder-Mead iteration step** $(0 < \delta_k \leq 1)$
ϑ	**The terminating limit for the variance of object function values**
c_q	**Convergence check iteration period**
n_Q	**Maximal number of object function evaluation**

The objective function evaluation is based on N Monte Carlo scenarios. For each scenario the time dynamics of the bond price requires generation of 3 (for single curve) or 6 (for dual curve case) draws from normal random distribution at every time step. To ensure the smooth objective function dependence on $\vec{z}$ we generate first the random driver cube ϵ_{jin} where indexes jin correspond to the scenario j, time point t_i and $n = 0, ..., 2$ or $n = 0, ..., 5$. Components of ϵ_{jin} will then be used in formulas (42) and (43).
As a result of the minimization procedure we obtain the optimal values of model parameters $\vec{x}^*$.
In addition, the simulated rates at the last iteraition are used to build graphs of the time dependence of historical rates and corresponding low and high percentiles of simulated rates.

The calibration algorithm is realized in a C^{++} application with its Excel front-end (see Figure 1).

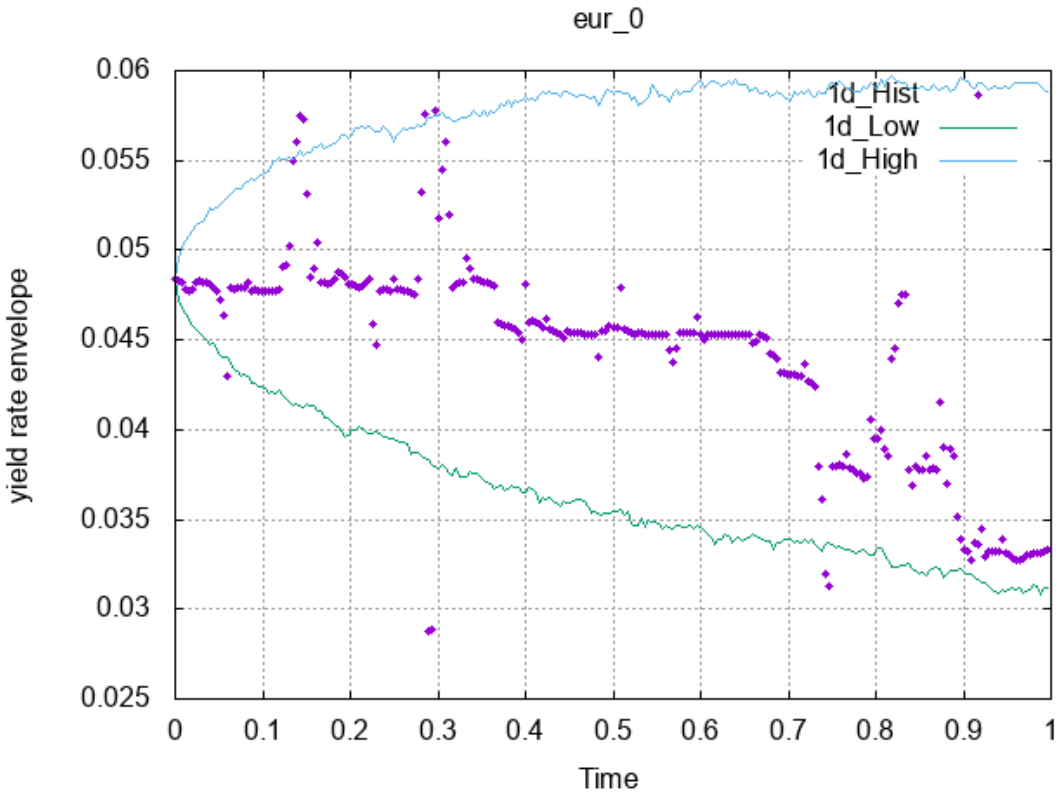

Figure 1: Excel front end for HJM model calibration

8 Calibration examples

8.1 Single yield curve calibration

In this section we present results of the model calibration for single yield curve (GBP). The C^{++} application was used with the following input data.
Control parameters:

0	Starting date for reading data from historical rates files
252	Number of lines/days to read from historical rates files beginning from starting day
0	Single curve calibration
252	Number of days in one year - convention
0.95	Confidence level
5000	Number of Monte Carlo scenarios
1	Penalty weight
gbp_2001_2018.in	File for historical rates in GBP
hjm_A.in	File with initial parameter values, and corresponding constraints, for GBP
0.25	Nelder-Mead iteration step
0.000001	Terminating limit for the variance of object function values
10	Convergence check period
300	Maximal number of object function evaluations
out_data_A.csv	File for GBP output of visualization of calibration results
obj_fn.csv	File for object function values versus iterations
report.txt	Calibration summary output file
1	Yes for graphical output
1	Maturities indexed 0, 1 and 7 will be displayed in graphical output

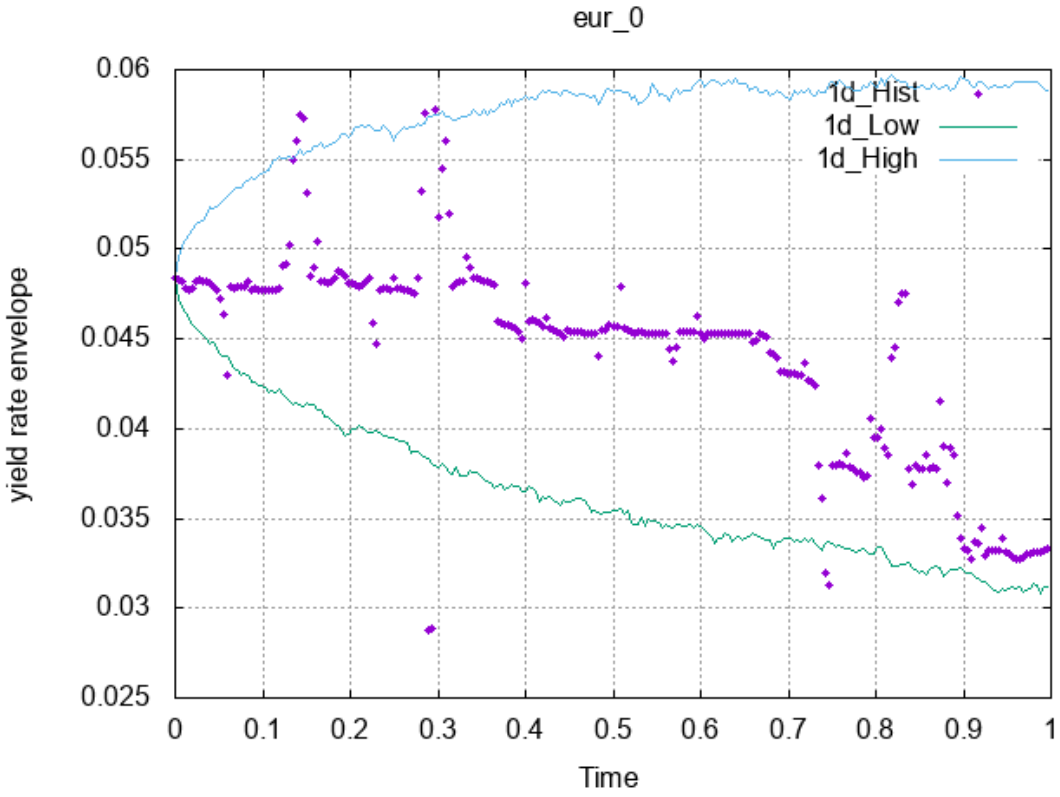

Figure 2: Object function for calibration of GBP yield curve on market data of 2011

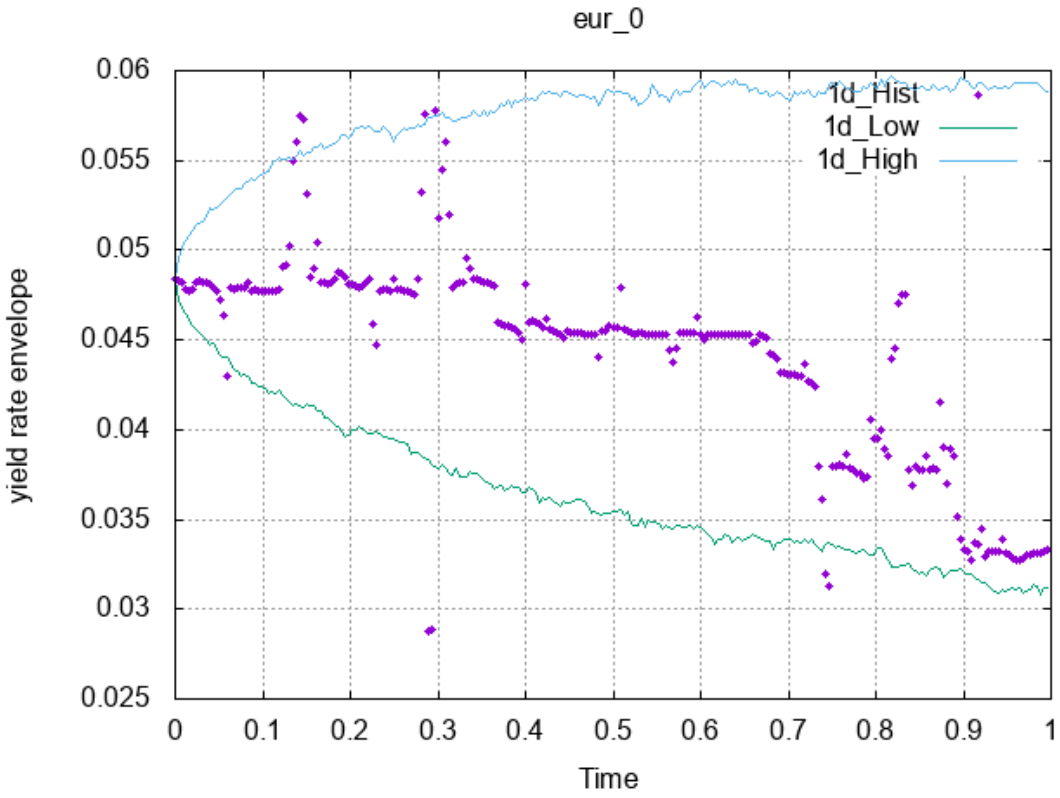

Figure 3: Simulation of GBP overnight (1 day) rate on optimal HJM parameters. Historical data 2011: dots, upper and lower percentiles of simulated rates: lines

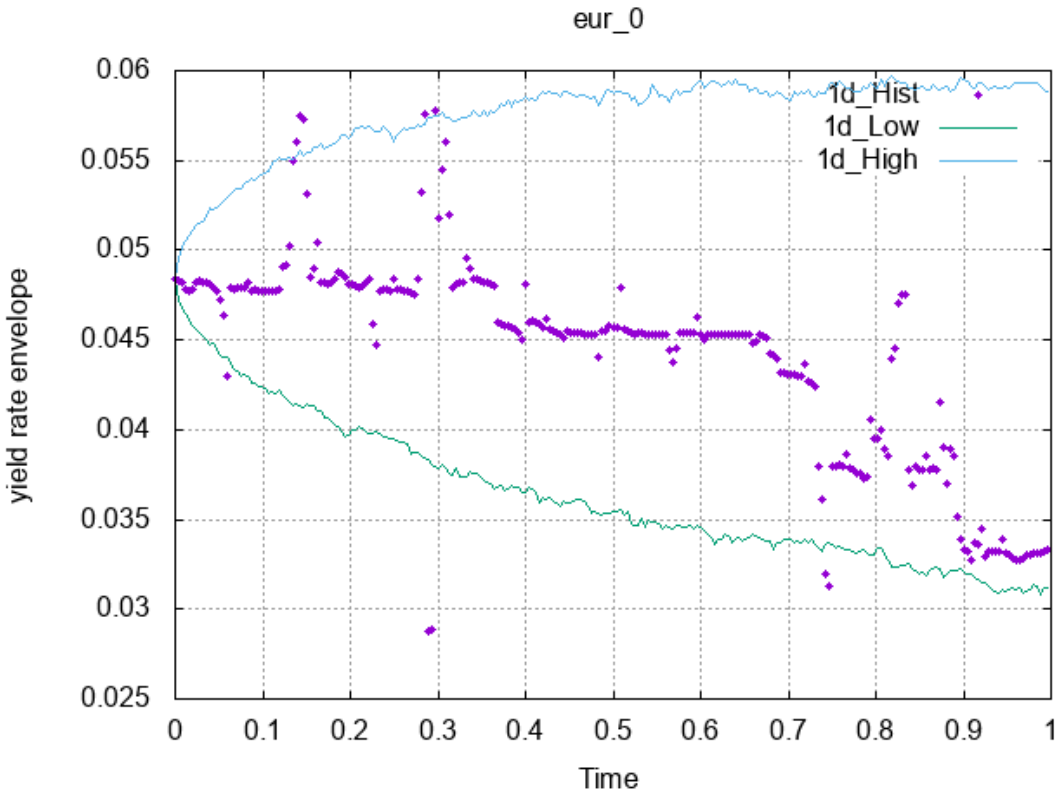

Figure 4: Simulation of GBP rate (maturity 1 year) on optimal HJM parameters. Historical data 2011: dots, upper and lower percentiles of simulated rates: lines

Calibration report:

```
Calibration run
Started: 11_December_2018_01_07_54_AM
Data length: 252 days
Days per year: 252
Confidence level: 0.95
Number of Monte-Carlo scenarios: 5000
Penalty weight: 1
Currency A file: gbp_2001_2018.in
Iteration step size: 0.25
Terminating limit: 1e-06
Convergence check period: 10
Number of object function calculations: 300
Constraints A:
   0.8000   1.5000
   0.8000   1.5000
   0.0100   0.1500
   0.0020   0.0500
   0.0010   0.0500
   0.0010   0.0100
  -0.0300   0.2000
  -0.2000   0.1000
  -0.5000   0.1000
   0.0020   0.2500

A starting point:
kappa0 kappa1 kappa2 sigma0 sigma1 sigma2   rho01  rho02   rho12   theta:
1.200  1.200  0.050  0.040  0.002  0.002  -0.200  -0.100  -0.100  0.100

Starting F(x) = -1.19424
err dev at start=1.97033%

A Optimal point:
kappa0 kappa1 kappa2 sigma0 sigma1 sigma2   rho01  rho02   rho12   theta:
1.3185 1.2123 0.0553 0.0330 0.0052 0.0020  -0.1610 -0.118  0.001  0.0165

Return code IFAULT = 2
Resulting  F(x*) = -2.55852
Resulting err dev=1.67629%
Number of iterations = 303
Number of restarts =   0
Ended: 11_December_2018_08_50_19_AM
```

Results presented in this section demonstrate clearly that model parameters obtained in the calibration procedure allow to simulate rates whose percentile envelope (5% to 95%) includes historical yield rates.

8.2 Dual Yield curve Calibration

In this section we present results of the model calibration for dual yield curves (GBP and EUR).
The C^{++} application was used with the following input data.
Control parameters:

0	Starting date for reading data from historical rates files
252	Number of lines/days to read from historical rates files starting from starting day
1	Dual curve calibration
252	Number of days in year - convention
0.95	Confidence level
1000	Number of Monte Carlo scenarios
10	Penalty weight
gbp_2001_2018.in	File for historical rates in GBP
hjm_A.in	File with initial parameter values, and corresponding constraints, for GBP
eur_2001_2018.in	File for historical rates in EUR
hjm_B.in	File with initial parameter values, and corresponding constraints, for EUR
1	Nelder-Mead iteration step
0.000001	Terminating limit for the variance of object function values
10	Convergence check period
500	Maximal number of object function evaluations
out_data_A.csv	File for GBP output of visualization of calibration results
out_data_B.csv	File for EUR output of visualization of calibration results
obj_fn.csv	File for object function values versus iterations
report.txt	Calibration summary output file
1	Yes for graphical output
1	Maturities indexed 0, 1 and 7 will be displayed in graphical output

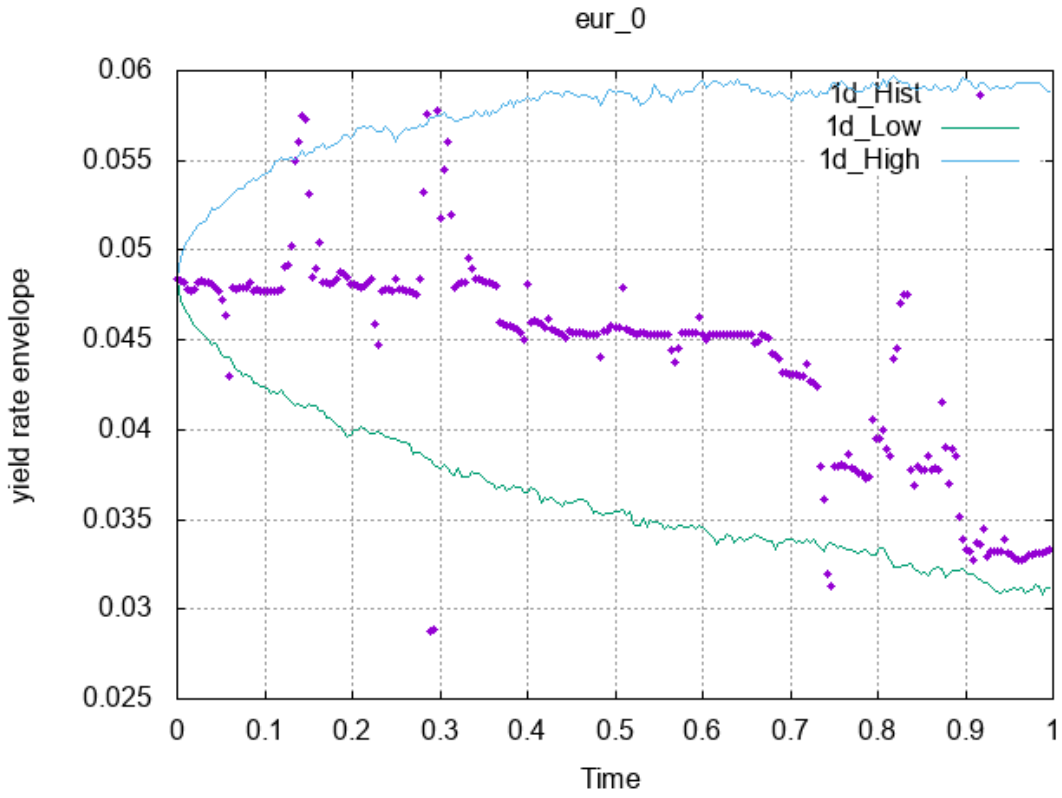

Figure 5: Object function for calibration of GBP and EUR yield curves on market data of 2011

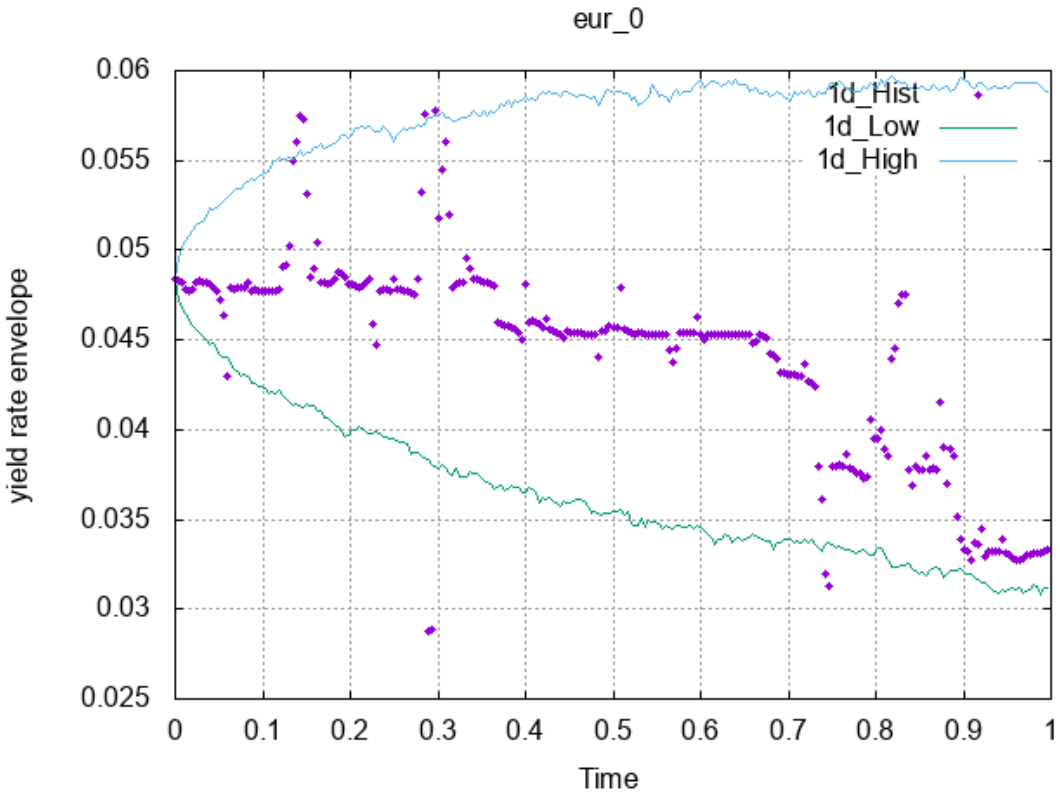

Figure 6: Simulation of dual GBP/EUR overnight (1 day) rate on optimal HJM parameters. Historical GBP data 2011: dots, upper and lower percentiles of simulated GBP rates: lines

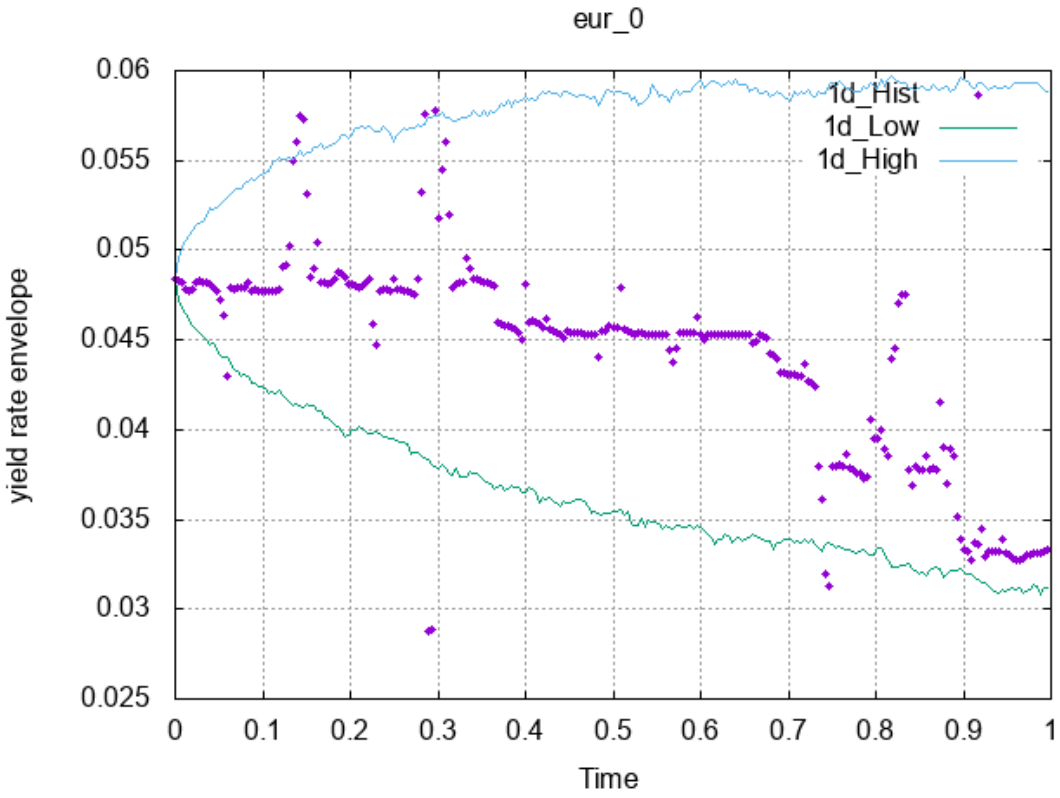

Figure 7: Simulation of dual GBP/EUR curves (maturity 1 year) rate on optimal HJM parameters. Historical GBP data 2011: dots, upper and lower percentiles of simulated GBP rates: lines

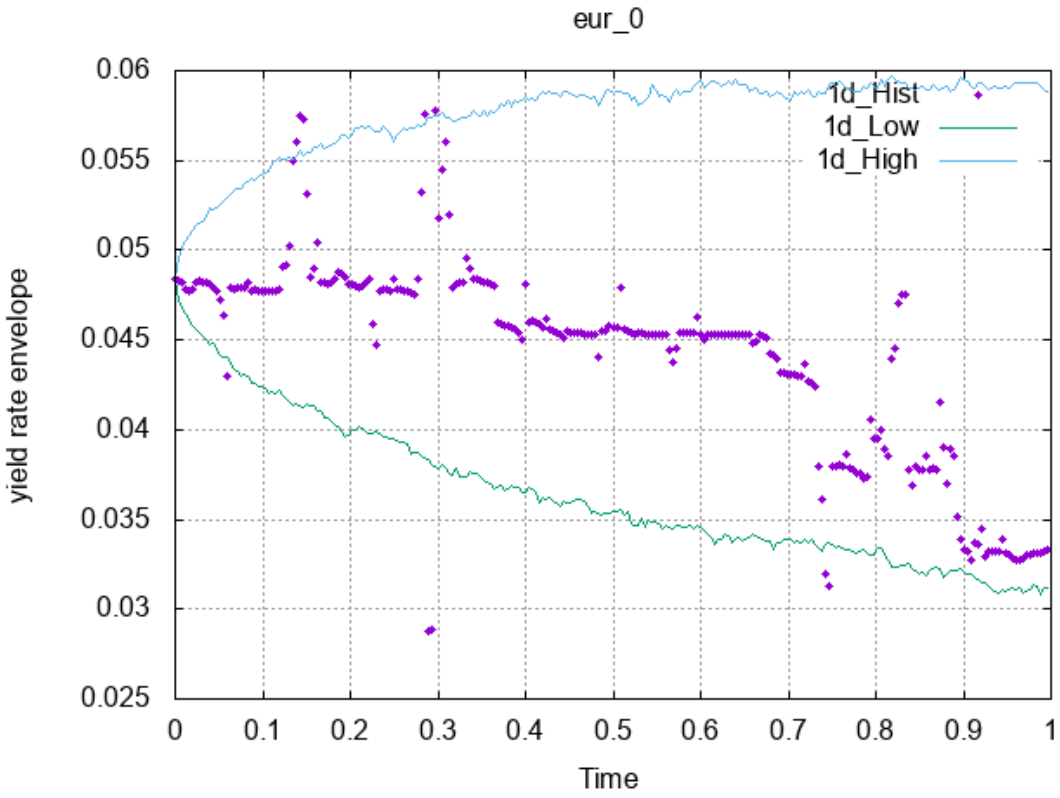

Figure 8: Simulation of dual GBP/EUR overnight (1 day) rate on optimal HJM parameters. Historical EUR data 2011: dots, upper and lower percentiles of simulated EUR rates: lines

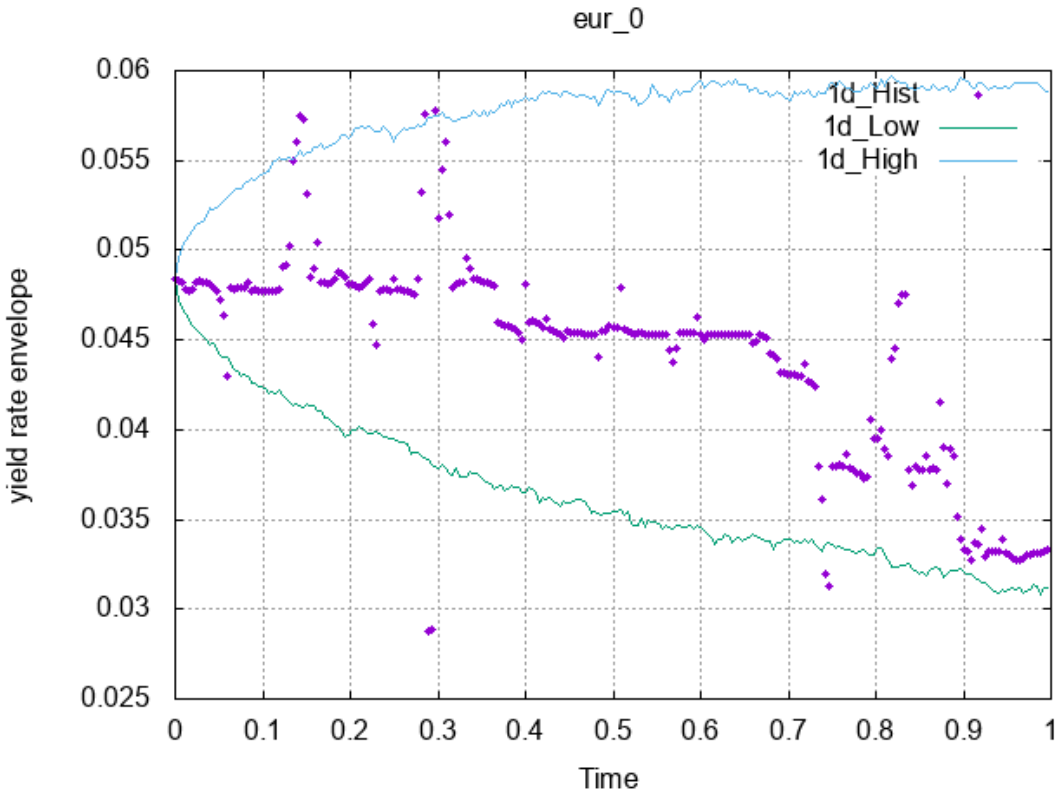

Figure 9: Simulation of dual GBP/EUR (maturity 1 year) rate on optimal HJM parameters. Historical EUR data 2011: dots, upper and lower percentiles of simulated EUR rates: lines

Calibration report for dual curves calibration:

```
Started: 20_December_2018_11_16_12_PM
Data length: 252 days,Days per year: 252,Confidence level: 0.95
Number of Monte-Carlo scenarios: 1000
Penalty weight: 10
Currency A file: gbp_2001_2018.in,Currency B file: eur_2001_2018.in
Iteration step size: 1
Terminating limit: 1e-06
Convergence check period: 10
Number of object function calculations: 500
Constraints A       Constraints B
0.100 0.500       0.300 1.200
0.100 0.500       0.500 1.200
0.030 0.150       0.100 0.1500
0.010 0.050       0.002 0.010
0.005 0.010       0.005 0.030
0.001 0.010       0.005 0.020
-0.04 0.100       -0.030 0.050
-0.10 0.100       -0.140 0.010
-0.50 0.000       -0.500 0.000
0.010 0.100       -0.300 0.010
                  -0.250 0.000
                  -0.700 -0.100
A starting point:
kappa0 kappa1 kappa2 sigma0 sigma1 sigma2   rho01 rho02  rho12   theta:
0.20   0.20   0.07   0.02   0.008   0.005  -0.02   0.02  -0.20   0.05
B starting point:
kappa0 kappa1 kappa2 sigma0  sigma1 sigma2   rho01  rho02   rho12:
1.00   1.00   0.20   0.007   0.015   0.010  -0.01   -0.07  -0.20
ksi0    ksi1    ksi2:
-0.20  -0.15  -0.50
Starting F(x) = -1.04178
Err dev at start=1.32305%
A Optimal point:
kappa0 kappa1 kappa2 sigma0 sigma1 sigma2 rho01  rho02   rho12  theta:
0.2811 0.2731 0.0768 0.0059 0.0093 0.0090 0.0106 0.0289 -0.2561 0.0110
B Optimal point:
kappa0 kappa1 kappa2 sigma0 sigma1 sigma2 rho01  rho02   rho12  theta:
0.8278 0.9211 0.1481 0.0092 0.0135 0.0179 -0.0052 -0.0863 -0.1929
ksi0      ksi1    ksi2:
-0.1551 -0.1395 -0.5218
Return code IFAULT = 2
Resulting  F(x*) = -3.10626
Resulting err dev=1.05731%
  Number of obj fn evaluations = 514
Number of restarts =   0
Ended: 21_December_2018_10_15_32_AM
```

9 Summary

The tri-factor HJM interest rate model calibration algorithm was developed in general case of the dual curve modelling. For the dual-curve model the correlation factors for different currencies were introduced. The calibration algorithm is based on the maximum likelihood criterium: yield rate simulated by Monte-Carlo process fit to the historical data. Minimization of the objective function is based on simplex (Nelder-Mead) method. The C^{++} application for calibration was developed and tested.

10 References

Aldrich, J.
1997. R. a. fisher and the making of maximum likelihood 1912 – 1922. *Statistical Science*, 12(3):162–176.

Fisher, R.
1922. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society A*, (222):594–604.

Heath, D., R. Jarrow, and A. Morton
1990. Bond pricing and the term structure of interest rates: A discrete time approximation. *Journal of Financial and Quantitative Analysis*, (25):419–440.