

Modelling Implied Volatility: Neural Network Method

YASHKIR CONSULTING www.yashkir.com

January 13, 2019

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | The problem formulation | 2 |
| 3 | Minimization of the error function | 3 |
| 3.1 | Backpropagation | 3 |
| 3.2 | Neural Network Iteration (Learning) Algorithm | 4 |
| 3.3 | Bond prices expressed through the stochastic integral | 4 |
| 3.4 | Bond price and instantaneous forward rate | 5 |
| 3.5 | Volatility model | 7 |
| 3.6 | Bond price model adapted to numerical simulation | 7 |
| 3.6.1 | Drift Integral | 8 |
| 3.6.2 | Brownian Motion Integral | 8 |
| 3.6.3 | HJM Bond Price simulation formula | 9 |
| 4 | Dual Currency Curve Simulation | 9 |
| 5 | Objective function for calibration | 10 |
| 6 | Calibration Algorithm | 12 |
| 7 | Calibration examples | 14 |
| 7.1 | Single yield curve calibration | 14 |
| 7.2 | Dual Yield curve Calibration | 19 |
| 8 | Summary | 26 |
| 9 | References | 26 |

1 Introduction

Implied volatility surface (Heath et al., 1990) is used for pricing vanilla options given price/strike ratio and time to maturity;

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

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

(Note: within sample n the vector/matrix form would be $\vec{z}_n^l = \vec{a}_n^{l-1} \mathbf{w}_l + \vec{b}_l$. 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

$$a_{nlj} = \sigma(z_{nlj}) \quad (2)$$

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

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

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 (2).

The modelling error for all samples is defined as:

$$E = \frac{1}{2} \sum_n \sum_{i=0}^{I-1} (a_{n,L-1,i} - y_{ni})^2 \quad (4)$$

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 (4).

3 Minimization of the error function

The minimum of the error function can be found using gradient descent algorithm for which partial derivatives of (4) 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 $z_{n,j,L-1}$

$$\delta_{n,j,L-1} = \frac{\partial E}{\partial z_{n,j,L-1}} = \frac{\partial E}{\partial a_{n,j,L-1}} \cdot \frac{\partial a_{n,j,L-1}}{\partial z_{n,j,L-1}} = \frac{\partial E}{\partial a_{n,j,L-1}} \cdot \frac{\partial \sigma(z_{n,j,L-1})}{\partial z_{n,j,L-1}} \quad (5)$$

where

$$\frac{\partial E}{\partial a_{n,j,L-1}} = \frac{\partial}{\partial a_{n,j,L-1}} \left[\frac{1}{2} \sum_k (a_{n,k,L-1} - y_{nk})^2 \right] = a_{n,j,L-1} - y_{nj} \quad (6)$$

$$\frac{\partial \sigma(z_{n,j,L-1})}{\partial z_{n,j,L-1}} \equiv \sigma'(z_{n,j,L-1}) \quad (8)$$

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

$$\delta_{n,j,l} = \frac{\partial E}{\partial z_{n,j,l}} = \sum_k \frac{\partial E}{\partial z_{n,k,l+1}} \frac{\partial z_{n,k,l+1}}{\partial z_{n,j,l}} = \sum_k \delta_{n,k,l+1} \frac{\partial}{\partial z_{n,j,l}} \left(\sum_i a_{n,i,l} w_{i,k,l+1} + b_{k,l+1} \right) \quad (9)$$

$$= \sum_k \delta_{n,k,l+1} \frac{\partial}{\partial z_{n,j,l}} \left(\sum_i \sigma(z_{n,i,l}) w_{i,k,l+1} + b_{k,l+1} \right) = \sum_k \delta_{n,k,l+1} w_{j,k,l+1} \sigma'(z_{n,j,l}) \quad (10)$$

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

$$\frac{\partial E}{\partial b_{jl}} = \frac{\partial E}{\partial a_{njl}} \frac{\partial a_{njl}}{\partial z_{njl}} \frac{\partial z_{njl}}{\partial b_{jl}} = \delta_{njl} \quad (\text{for a single sample } n) \quad (11)$$

$$\beta_{jl} = \frac{1}{N} \sum_{n=0}^{N-1} \delta_{njl} \quad (\text{averaged over all samples}) \quad (12)$$

4. Derivatives by weight matrix components:

Partial derivative within n-th sample: (13)

$$\frac{\partial E}{\partial w_{kjl}} = \frac{\partial E}{\partial z_{njl}} \frac{\partial z_{njl}}{\partial w_{kjl}} = \delta_{njl} \frac{\partial}{\partial w_{kjl}} \left(\sum_i a_{n,i,l-1} + b_{jl} \right) = \delta_{njl} a_{n,k,l-1} \quad (14)$$

Derivative (14) averaged over all samples: (15)

$$v_{kjl} = \frac{1}{N} \sum_{n=0}^{N-1} \delta_{njl} a_{n,k,l-1} \quad (16)$$

5. Gradient descent step:

$$w_{kjl} = w_{kjl} - h \cdot v_{kjl} \quad (17)$$

$$b_{jl} = b_{jl} - h \cdot \beta_{jl} \quad (18)$$

Where h is the iteration step (learning rate)

3.2 Neural Network Iteration (Learning) Algorithm

$$\frac{dB(t, T)}{B(t, T)} = r(t)dt + \sum_{p=0}^2 \sigma_p(t, T) dW_p(t) \quad (19)$$

where T is the bond maturity, σ_p represents the bond price volatility of the stochastic driver number p , and $d\vec{W}(t)$ is a 3-tier Wiener process.

3.3 Bond prices expressed through the stochastic integral

Let $G(B) = \ln B$ and use the Ito's lemma (Itô, 1944) as follows

$$dG = \frac{\partial G}{\partial B} dB + \frac{1}{2} \frac{\partial^2 G}{\partial B^2} (dB)^2 \quad (20)$$

$$\begin{aligned} d(\ln B) &= \frac{1}{B} \left(r \cdot B \cdot dt + \sum_p \sigma_p B \cdot dW_p \right) + \frac{1}{2} \frac{1}{(-B^2)} \left(r(t)B \cdot dt + \sum_p \sigma_p B \cdot dW_p \right)^2 \\ &= r \cdot dt + \sum_p \sigma_p \cdot dW_p - \frac{1}{2} \left(r^2 \cdot (dt)^2 + 2r \cdot dt \sum_p \sigma_p \cdot dW_p + \sum_{p,p'} \sigma_p \sigma_{p'} \cdot dW_p \cdot dW_{p'} \right) \\ &= r \cdot dt + \sum_p \sigma_p \cdot dW_p - \frac{1}{2} r^2 (dt)^2 - r \cdot dt \sum_p \sigma_p \cdot dW_p - \frac{1}{2} \sum_{p,p'} \sigma_p \sigma_{p'} \cdot dW_p \cdot dW_{p'} \quad (21) \end{aligned}$$

Note that processes dW_p are correlated

$$\langle dW_p(t) dW_{p'}(t) \rangle = \rho_{pp'} dt \quad (22)$$

Keeping in the equation (21) only terms with degree of dt equal or less than 1 we arrive to the following equation for the process $\ln B(t, T)$:

$$d \ln B(t, T) = \left(r(t) - \frac{1}{2} \sum_{pp'} \sigma_p(t, T) \sigma_{p'}(t, T) \rho_{pp'} \right) dt + \sum_{p=0}^2 \sigma_p(t, T) dW_p(t, T) \quad (23)$$

Integration of the equation (23) from s to t leads to the integral expression of the bond price as a function of bond volatilities:

$$\ln B(t, T) - \ln B(s, T) = \int_s^t \left(r(q) - \frac{1}{2} \sum_{pp'} \rho_{pp'} \sigma_p(q, T) \sigma_{p'}(q, T) \right) dq + \int_s^t \sum_p \sigma_p(q, T) dW_p(q) \quad (24)$$

or

$$\frac{B(t, T)}{B(s, T)} = e^{\int_s^t \left(r(q) - \frac{1}{2} \sum_{pp'} \rho_{pp'} \sigma_p(q, T) \sigma_{p'}(q, T) \right) dq + \int_s^t \sum_p \sigma_p(q, T) dW_p(q)} \quad (25)$$

This equation can be further simplified as follows. The equation (25) with $T = t$ becomes:

$$\frac{B(t, t)}{B(s, t)} = e^{\int_s^t \left(r(q) - \frac{1}{2} \sum_{pp'} \rho_{pp'} \sigma_p(q, t) \sigma_{p'}(q, t) \right) dq + \int_s^t \sum_p \sigma_p(q, t) dW_p(q)} \quad (26)$$

Combining (25) and (26) we find (note that $B(t, t) = 1$):

$$B(t, T) = \frac{B(s, T)}{B(s, t)} e^{\left\{ -\frac{1}{2} \sum_{pp'} \rho_{pp'} \int_s^t [\sigma_p(q, T) \sigma_{p'}(q, T) - \sigma_p(q, t) \sigma_{p'}(q, t)] dq + \int_s^t \sum_p (\sigma_p(q, T) - \sigma_p(q, t)) dW_p(q) \right\}} \quad (27)$$

In the last equation (27) the bond price at time t maturing at T is expressed through correlation matrix $\hat{\rho}$, volatility functions $\sigma_p(t, T)$ and Wiener processes $dW_p(t)$.

3.4 Bond price and instantenious forward rate

The instantanious forward rate $f(t, u)$ is defined by the following expression

$$B(t, T) = e^{-\int_t^T f(t, u) du} \quad (28)$$

which is equivalent to

$$f(t, T) = \frac{\partial}{\partial t} \ln B(t, T) \quad (29)$$

The differential of $f(t, T)$ can be calculated using (29) and (23):

$$\begin{aligned} df(t, T) &= d \left(\frac{\partial}{\partial t} \ln B(t, T) \right) = \frac{\partial}{\partial t} (d \ln B(t, T)) = -\frac{\partial}{\partial T} (d \ln B(t, T)) \\ &= -\frac{\partial}{\partial T} \left[r(t) \cdot dt - \frac{1}{2} \sum_{pp'} \rho_{pp'} \sigma_p(t, T) \sigma_{p'}(t, T) dt + \sum_p \sigma_p(t, T) dW_p(t) \right] \end{aligned} \quad (30)$$

The instantaneous forward rate volatility can be introduced as

$$\dot{\sigma}_p(t, T) = \frac{\partial}{\partial T} \sigma_p(t, T) \quad (31)$$

or

$$\sigma_p(t, T) = \int_t^T \dot{\sigma}_p(t, u) du \quad (32)$$

Substitution of $\sigma_p(t, T)$ in the equation (30) with (32) leads to:

$$\begin{aligned} df(t, T) &= \frac{1}{2} \sum_{pp'} \rho_{pp'} \left[\frac{\partial \sigma_p(t, T)}{\partial T} \sigma_{p'}(t, T) + \sigma_p(t, T) \frac{\partial \sigma_{p'}(t, T)}{\partial T} \right] dt - \sum_p \frac{\partial \sigma_p(t, T)}{\partial T} dW_p(t) \\ &= \sum_{pp'} \rho_{pp'} \dot{\sigma}_p(t, T) \left(\int_t^T \dot{\sigma}_{p'}(t, u) du \right) dt - \sum_p \dot{\sigma}_p(t, T) dW_p(t) \end{aligned} \quad (33)$$

Integrate (33) by t (from s to t) and (for convenience) replace T with u . We then have:

$$f(t, u) - f(s, u) = \sum_{pp'} \rho_{pp'} \int_s^t \dot{\sigma}_p(q, u) \left(\int_q^u \dot{\sigma}_{p'}(q, v) dv \right) dq - \sum_p \int_s^t \dot{\sigma}_p(q, u) dW_p(q) \quad (34)$$

Continue by integration the equation (34) by u from t to T):

$$\begin{aligned} \int_t^T f(t, u) du - \int_t^T f(s, u) du &= \\ \sum_{pp'} \rho_{pp'} \int_t^T du \int_s^t \dot{\sigma}_p(q, u) \left(\int_q^u \dot{\sigma}_{p'}(q, v) dv \right) dq - \sum_p \int_t^T du \int_s^t \dot{\sigma}_p(q, u) dW_p(q) \end{aligned} \quad (35)$$

It follows from (28) that

$$\begin{aligned} \int_t^T f(t, u) du &= -\ln B(t, T) \\ \text{and} \\ \int_t^T f(s, u) du &= \int_t^s f(s, u) du + \int_s^T f(s, u) du = -\int_s^t f(s, u) du + \int_s^T f(s, u) du \\ &= \ln B(s, t) - \ln B(s, T) \end{aligned} \quad (36)$$

Combining (35) with (36) we find:

$$\begin{aligned} -\ln B(t, T) - \ln B(s, t) + \ln B(s, T) &= \\ \sum_{pp'} \rho_{pp'} \int_t^T du \int_s^t \dot{\sigma}_p(q, u) \left(\int_q^u \dot{\sigma}_{p'}(q, v) dv \right) dq - \sum_p \int_t^T du \int_s^t \dot{\sigma}_p(q, u) dW_p(q) \end{aligned} \quad (37)$$

or, finally,

$$B(t, T) = \frac{B(s, T)}{B(s, t)} \exp \left\{ - \sum_{pp'} \rho_{pp'} \int_t^T du \int_s^t \dot{\sigma}_p(q, u) \left(\int_q^u \dot{\sigma}_{p'}(q, v) dv \right) dq + \sum_p \int_t^T du \int_s^t \dot{\sigma}_p(q, u) dW_p(q) \right\} \quad (38)$$

3.5 Volatility model

It is common approach to assume that the volatility function is a decaying function of time ¹

$$\dot{\sigma}_p(q, r) = \hat{\sigma}_p e^{-\kappa_p(r-q)} \quad (39)$$

$$\sigma_p(t, T) = \int_t^T \dot{\sigma}_p(q, T) dq = \frac{\hat{\sigma}_p}{\kappa_p} \left(1 - e^{-\kappa_p(T-t)} \right) \quad (40)$$

$$\hat{\sigma}_p = \text{const} \quad (41)$$

3.6 Bond price model adapted to numerical simulation

Using (39) in (38) we obtain

$$B(t, T) = \frac{B(s, T)}{B(s, t)} \exp \left\{ - \sum_{pp'} \rho_{pp'} \int_t^T du \int_s^t \hat{\sigma}_p e^{-\kappa_p(u-q)} \left(\int_q^u \hat{\sigma}_{p'} e^{-\kappa_{p'}(v-q)} dv \right) dq + \sum_p \int_t^T du \int_s^t \hat{\sigma}_p e^{-\kappa_p(u-q)} dW_p(q) \right\} \quad (42)$$

Equation (42) can now be used to build numerical simulation algorithm. It contains dependence of bond prices at time t for maturities T on bond prices at previous time s with the drift term $D(s, t, T)$ and the Brownian motion $\Phi(s, t, T)$ integral:

$$B(t, T) = \frac{B(s, T)}{B(s, t)} e^{D(s, t, T) + \Phi(s, t, T)} \quad , \text{ where} \quad (43)$$

$$D(s, t, T) = - \sum_{pp'} \rho_{pp'} \int_t^T du \int_s^t \hat{\sigma}_p e^{-\kappa_p(u-q)} \left(\int_q^u \hat{\sigma}_{p'} e^{-\kappa_{p'}(v-q)} dv \right) dq \quad (44)$$

$$\Phi(s, t, T) = \sum_p \int_t^T du \int_s^t \hat{\sigma}_p e^{-\kappa_p(u-q)} dW_p(q) \quad (45)$$

¹See (Chiarella and Kwon, 2001)

3.6.1 Drift Integral

The calculation of the drift integral (44) is straitforward. The result is:

$$D(s, t, T) = \sum_{pp'} \rho_{pp'} \frac{\hat{\sigma}_p \hat{\sigma}_{p'}}{\kappa_{p'}} \left\{ \frac{1 - e^{-(\kappa_p + \kappa_{p'})(T-t)} + e^{-(\kappa_p + \kappa_{p'})(T-s)} - e^{-(\kappa_p + \kappa_{p'})(t-s)}}{(\kappa_p + \kappa_{p'})^2} - \frac{1 - e^{-\kappa_p(T-t)} + e^{-\kappa_p(T-s)} - e^{-\kappa_p(t-s)}}{(\kappa_p)^2} \right\} \quad (46)$$

3.6.2 Brownian Motion Integral

The components $dW_p(t)$ of the Wiener process are correlated, therefore each of them depends linearly on three independent normally distributed random drivers ε_p :

$$dW_p(t) = \sum_{p'} \gamma_{pp'} \varepsilon_{p'} \sqrt{dt} \quad (47)$$

$$\langle \varepsilon_p \varepsilon_{p'} \rangle = \delta_{pp'} \quad \langle \varepsilon_p^2 \rangle = 1$$

where linear form γ can be derived from correlation matrix ρ . From (22) and (47) we find:

$$\langle W_p(q) W_{p'}(q) \rangle = \left\langle \sum_n \gamma_{pn} \varepsilon_n \sqrt{dq} \sum_m \gamma_{p'm} \varepsilon_m \sqrt{dq} \right\rangle = \sum_n \gamma_{pn} \gamma_{p'n} dq = \rho_{pp'} dq$$

therefore

$$\sum_n \gamma_{pn} \gamma_{p'n} = \rho_{pp'} \quad (48)$$

Given correlation matrix ρ , it follows from (48) that:

$$\gamma = \begin{pmatrix} 1 & 0 & 0 \\ \rho_{01} & \sqrt{1 - \rho_{01}^2} & 0 \\ \rho_{02} & \frac{\rho_{12} - \rho_{01}\rho_{02}}{\sqrt{1 - \rho_{01}^2}} & \sqrt{1 - \rho_{02}^2 - \frac{(\rho_{12} - \rho_{01}\rho_{02})^2}{1 - \rho_{01}^2}} \end{pmatrix} \quad (49)$$

The Brownian motion integral (45) with account of (47) and (48) takes the following form:

$$\Phi(s, t, T) = \sum_{pp'} \left(\gamma_{pp'} \hat{\sigma}_p \int_t^T du \int_s^t e^{-\kappa_p(u-q)} \sqrt{dq} \right) \varepsilon_{p'} \quad (50)$$

The Brownian motion process (50) is therefore a simple sum of three parts driven by Gaussian random drivers ε_p . To find corresponding volatilities we calculate the total variance of the

Brownian motion integral

$$\begin{aligned}
& \langle \Phi(s, t, T) \Phi(s, t, T) \rangle = \\
& = \left\langle \sum_{pp'} \left(\gamma_{pp'} \hat{\sigma}_p \int_t^T du \int_s^t e^{-\kappa_p(u-q)} \sqrt{dq} \right) \varepsilon_{p'} \cdot \sum_{np'} \left(\gamma_{np'} \hat{\sigma}_n \int_t^T du \int_s^t e^{-\kappa_n(u-q)} \sqrt{dq} \right) \varepsilon_{p'} \right\rangle \\
& = \sum_{pp'n} \gamma_{pp'} \gamma_{np'} \hat{\sigma}_p \hat{\sigma}_n \int_s^t \frac{e^{-\kappa_p(T-q)} - e^{-\kappa_p(t-q)}}{-\kappa_p} \cdot \frac{e^{-\kappa_n(T-q)} - e^{-\kappa_n(t-q)}}{-\kappa_n} dq \\
& = \sum_n v_n(s, t, T)
\end{aligned} \tag{51}$$

where

$$\begin{aligned}
v_n(s, t, T) = & \sum_{pp'} \frac{\gamma_{pn} \gamma_{p'n} \hat{\sigma}_p \hat{\sigma}_{p'}}{\kappa_p \kappa_{p'} (\kappa_p + \kappa_{p'})} \left(1 + e^{-(\kappa_p + \kappa_{p'})(T-t)} - e^{-(\kappa_p + \kappa_{p'})(T-s)} - e^{-\kappa_p(T-t)} \right. \\
& \left. + e^{-\kappa_p(T-s) - \kappa_{p'}(t-s)} - e^{-\kappa_{p'}(T-t)} + e^{-\kappa_p(t-s) - \kappa_{p'}(T-s)} - e^{-(\kappa_p + \kappa_{p'})(t-s)} \right)
\end{aligned} \tag{52}$$

is the variance of the n^{th} factor of the Brownian motion integral.

3.6.3 HJM Bond Price simulation formula

Equations (43), (46), (52), derived above, provide basis for numerical bond price simulation in the following form

$$\ln B(t, T) = \ln B(s, T) - \ln B(s, t) + D(s, t, T) + \sum_{n=0}^2 \sqrt{v_n(s, t, T)} \cdot \varepsilon_n(t) \tag{53}$$

with sufficiently small time step $dt = t - s$ (1 day, for example).

•

The interest rate r_t at time t (time to maturity $T - t$) and the zero-coupon bond value are related as

$$B(t, T) = \frac{1}{(1 + r_t)^{T-t}} \tag{54}$$

therefore yield rates at time t for maturity $\tau = T - t$ are calculated as

$$r_\tau(t) = e^{-\ln B(t, T)/(T-t)} - 1 \tag{55}$$

4 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 ε'_n for curve A is correlated

to corresponding driver ε_n'' for curve B . Therefore the dual-curve model takes the following form (see equation 53 for zero-coupon-bond values $B(t, T)$):

$$\ln B(t, T) = \ln B(s, T) - \ln B(s, t) + D(s, t, T) + \sum_{n=0}^2 \sqrt{v_n(s, t, T)} \cdot \varphi_n(t) \quad (56)$$

$$\ln B^*(t, T) = \ln B^*(s, T) - \ln B^*(s, t) + D^*(s, t, T) + \sum_{n=0}^2 \sqrt{v_n^*(s, t, T)} \cdot \varphi_n^*(t) \quad (57)$$

Here, the equation (56) is for the base rate model (A) and the equation (57) (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:

$$\varphi_n(t) = \varepsilon_n' \xi_n + \varepsilon_n'' \sqrt{1 - \xi_n^2} \quad (58)$$

$$\varphi_n^*(t) = \varepsilon_n' \sqrt{1 - \xi_n^2} + \varepsilon_n'' \xi_n \quad (59)$$

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

$$\begin{aligned} \langle \varphi_n^2 \rangle &= 1 & \langle \varphi_n \varphi_{m \neq n} \rangle &= 0 & \langle \varphi_n^{*2} \rangle &= 1 & \langle \varphi_n^* \varphi_{m \neq n}^* \rangle &= 0 \\ \rho^* &= \langle \varphi_n \varphi_n^* \rangle = 2\xi_n \sqrt{1 - \xi_n^2} \end{aligned} \quad (60)$$

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

5 Objective function for calibration

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

$$\begin{aligned} \hat{r}_{\tau_k}(t_i) \\ \hat{r}_{\tau_k}^*(t_i) \end{aligned} \quad (61)$$

are available for certain period of time $\{t_0 \dots T_H\}$ for both yield curves. Here t_i are dates and τ_k are maturities ($k = 0, 1, 2, \dots$ 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 (4) with the following initial condition:

$$\ln B(0, t_p) = -\overset{o}{r}_{t_p}(t_0) \cdot t_p \quad (62)$$

$$\ln B^*(0, t_p) = -\overset{o}{r}_{t_p}^*(t_0) \cdot t_p \quad (63)$$

where t_p are bond maturity times (corresponding to T in (56), 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 interpolation 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

$$\begin{aligned} \ln B_j(t_i, t_p) \\ \ln B_j^*(t_i, t_p) \end{aligned} \quad (64)$$

Next, for a given set of historical yield rate maturities τ_k we use (64) and (55) to simulate corresponding yield rates

$$\begin{aligned} r_j(t_i, t_p) \\ r_j^*(t_i, t_p) \end{aligned} \quad (65)$$

The calibration procedure must reduce as much as possible the aggregated mismatch between simulated (65) and historical (61) 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

$$f(r, \hat{r}, \theta) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(-\frac{(r - \hat{r})^2}{2\theta^2}\right) \quad (66)$$

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

$$P(\vec{z}) = \prod_{i,k,j} \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(-\frac{[r_j(t_i, t_p) - \hat{r}(t_i, t_p)]^2}{2\theta^2}\right) \quad (67)$$

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

$$\vec{z} = [\vec{\kappa}^A, \vec{\sigma}^A, \vec{\rho}^{A*}, \theta, \vec{\kappa}^B, \vec{\sigma}^B, \vec{\rho}^{B*}, \xi] \quad (68)$$

Taking logarithm of (67) and dividing by total number of fitting points M , we obtain the log-likelihood function

$$F(\vec{z}) = -\ln \sqrt{2\pi\theta^2} - \frac{1}{M} \sum_{i,k,j} \frac{[r_j(t_i, t_p) - \hat{r}(t_i, t_p)]^2}{2\theta^2} \quad (69)$$

The number of fitting points M is the total number of sum items in (69). 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 (69) 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 $-F(\vec{z})$:

$$\Omega(\vec{z}) = \sum_k \{\max(0, z_k^{min} - z_k) + \max(0, z_k - z_k^{max})\} \cdot \Upsilon \quad (70)$$

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

$$\begin{aligned} Q(\vec{z}) = \ln \sqrt{2\pi\theta^2} + \frac{1}{M} \sum_{i,k,j} \frac{[r_j(t_i, t_p) - \hat{r}(t_i, t_p)]^2}{2\theta^2} \\ + \sum_k \{\max(0, z_k^{min} - z_k) + \max(0, z_k - z_k^{max})\} \cdot \Upsilon \end{aligned} \quad (71)$$

6 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 (71) 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 j correspond to the scenario j , time point t_i and $n = 0, \dots, 2$ or $n = 0, \dots, 5$. Components of ϵ_{jin} will then be used in formulas (58) and (59).

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 iteration 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).

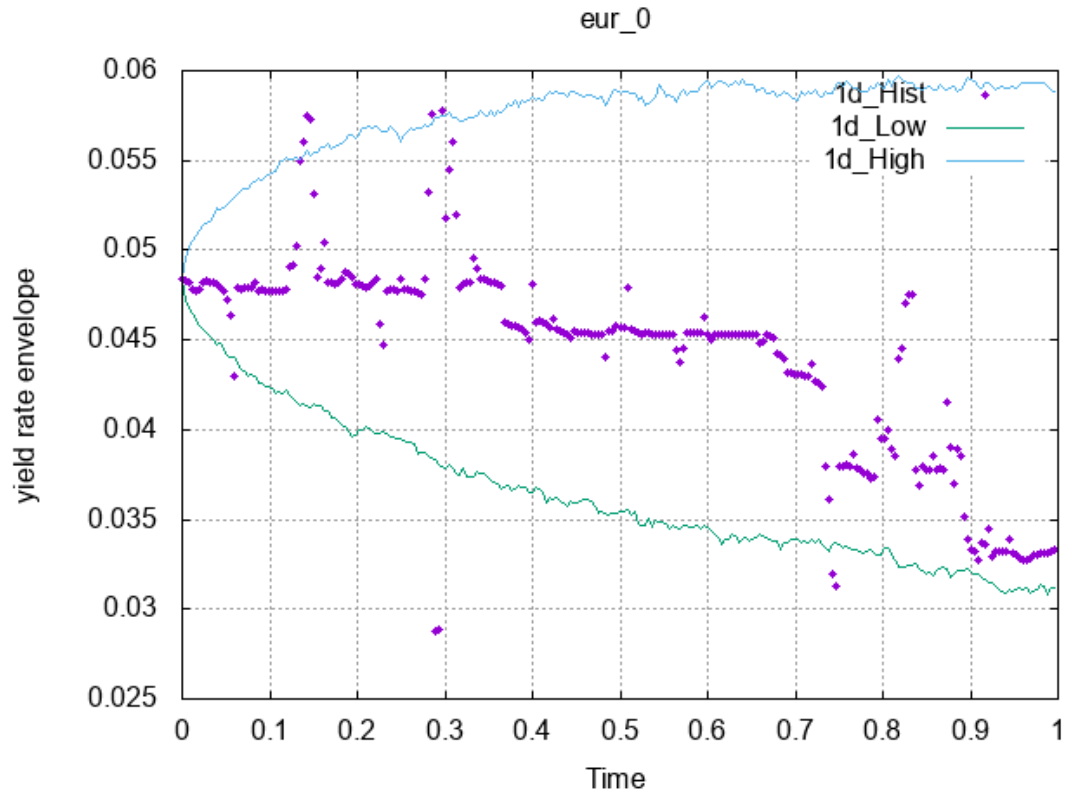


Figure 1: Excel front end for HJM model calibration

7 Calibration examples

7.1 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 |

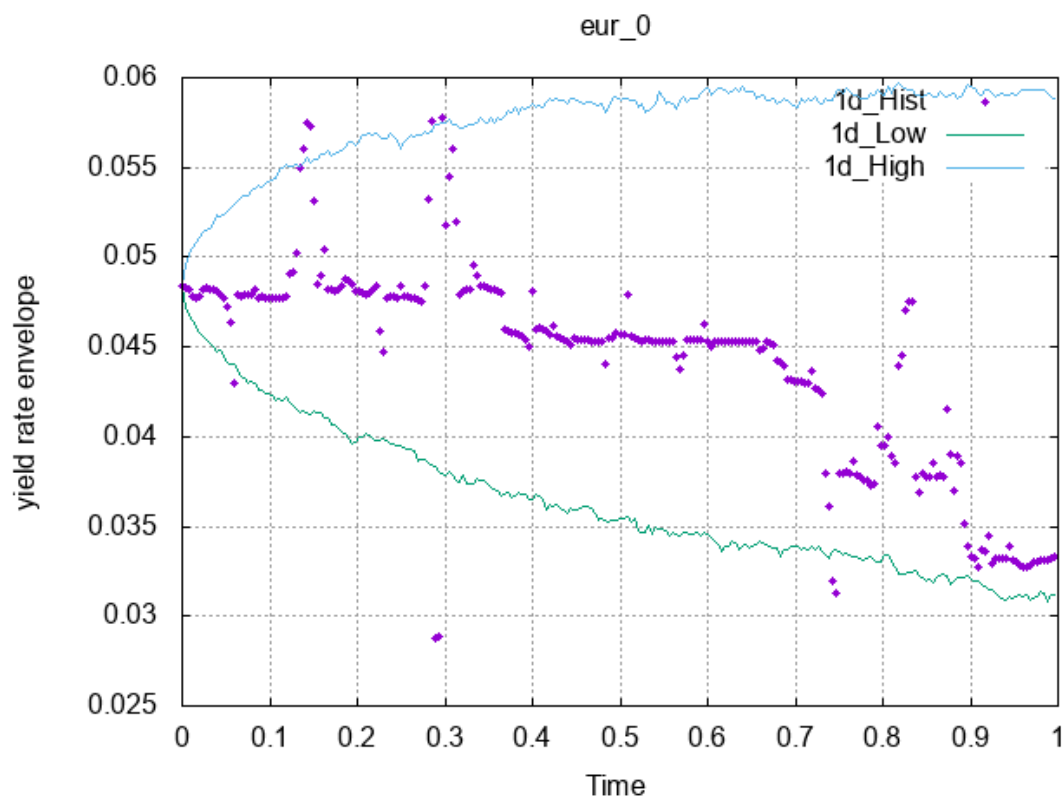


Figure 2: Object function for calibration of GBP yield curve on market data of 2011

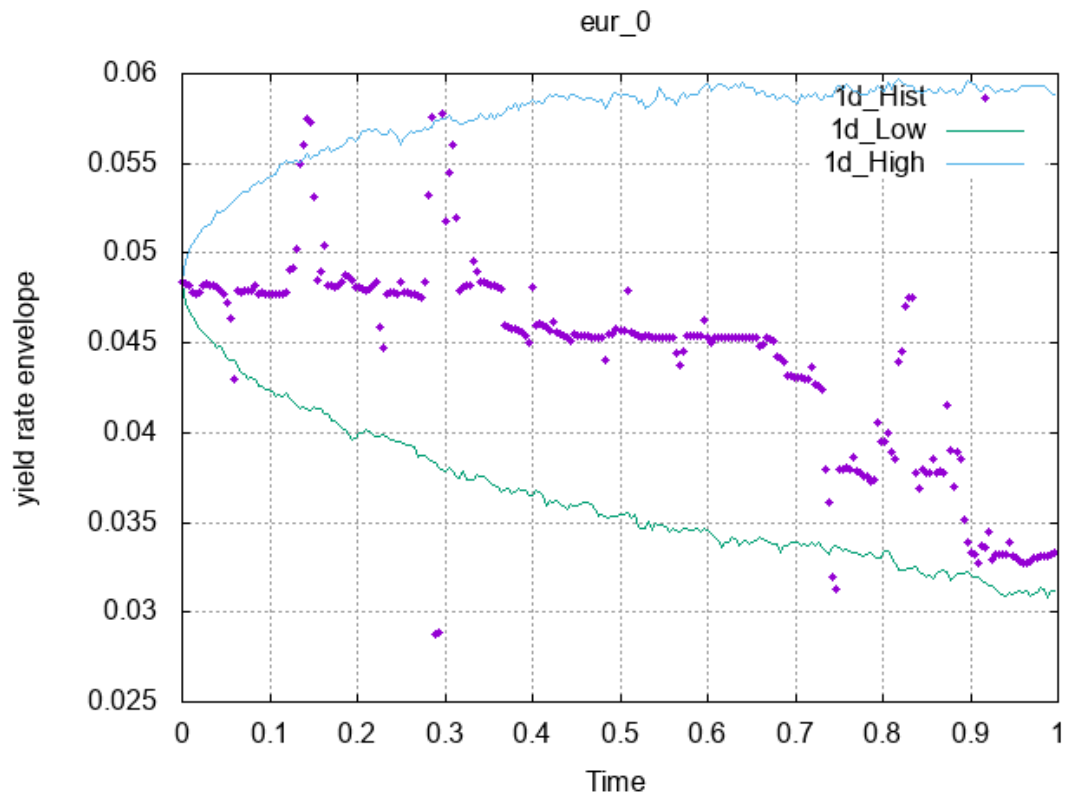


Figure 3: Simulation of GBP overnight (1 day) rate on optimal HJM parameters. Historical data 2011: dots, upper and lower percentiles of simulated rates: lines

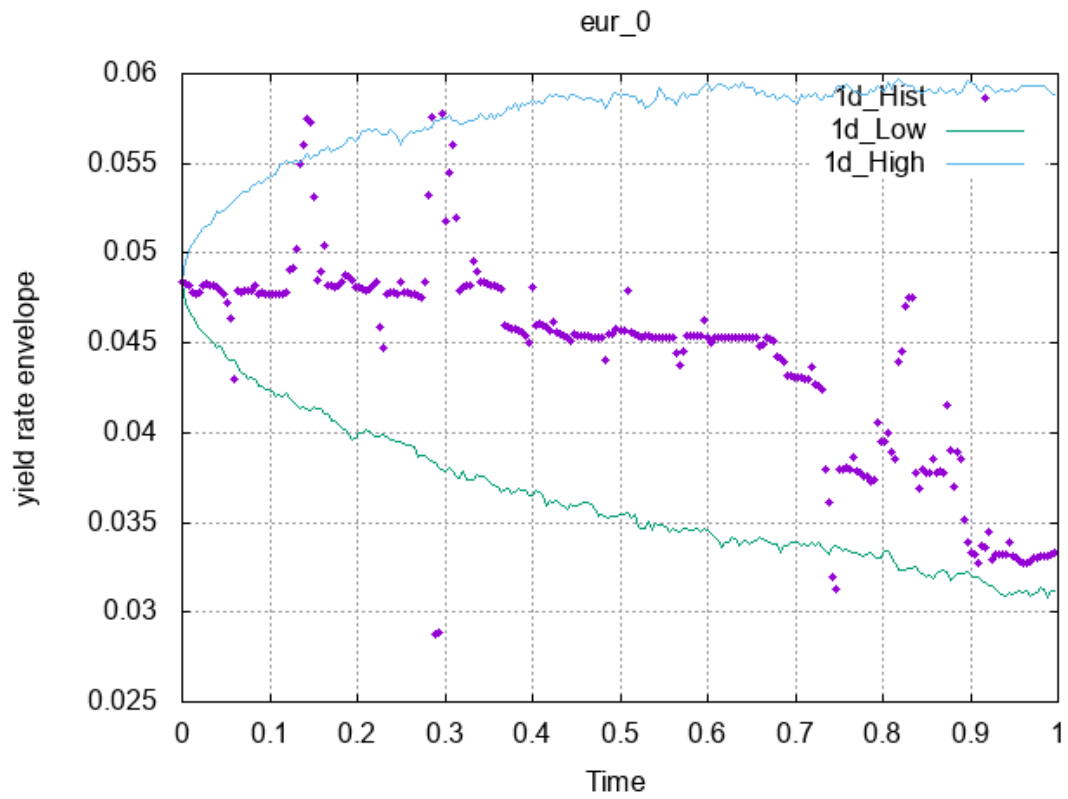


Figure 4: Simulation of GBP rate (maturity 1 year) on optimal HJM parameters. Historical data 2011: dots, upper and lower percentiles of simulated rates: lines

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.

7.2 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 |

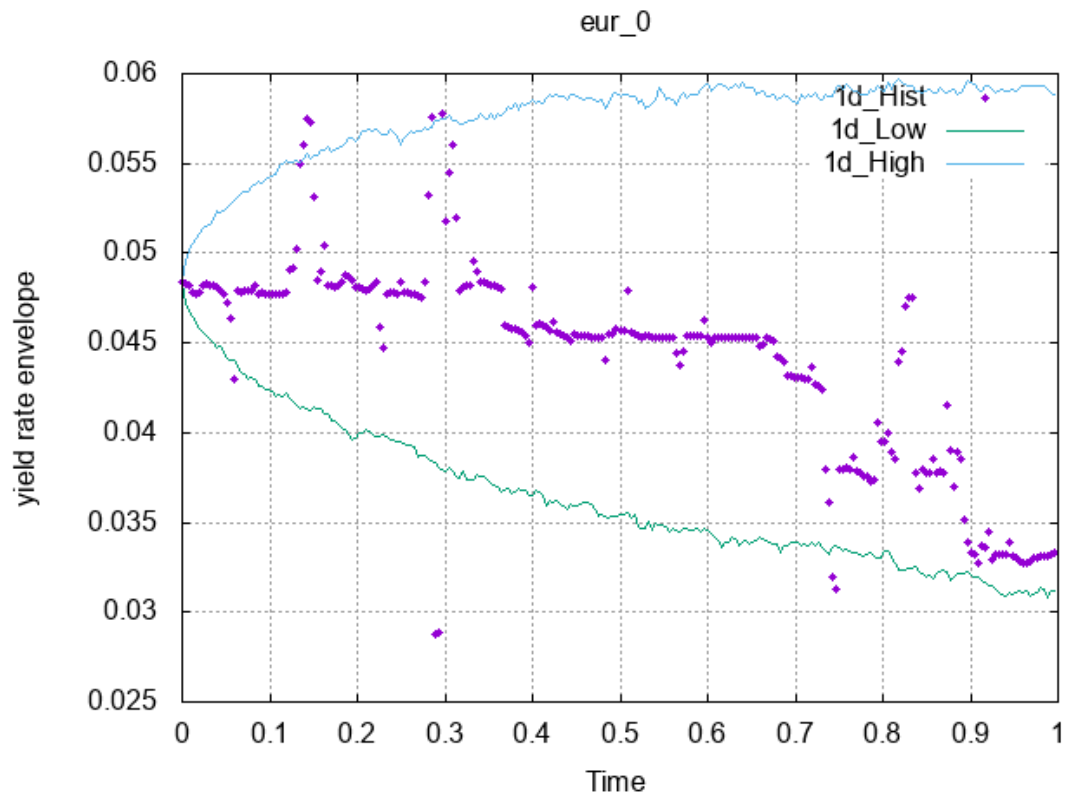


Figure 5: Object function for calibration of GBP and EUR yield curves on market data of 2011

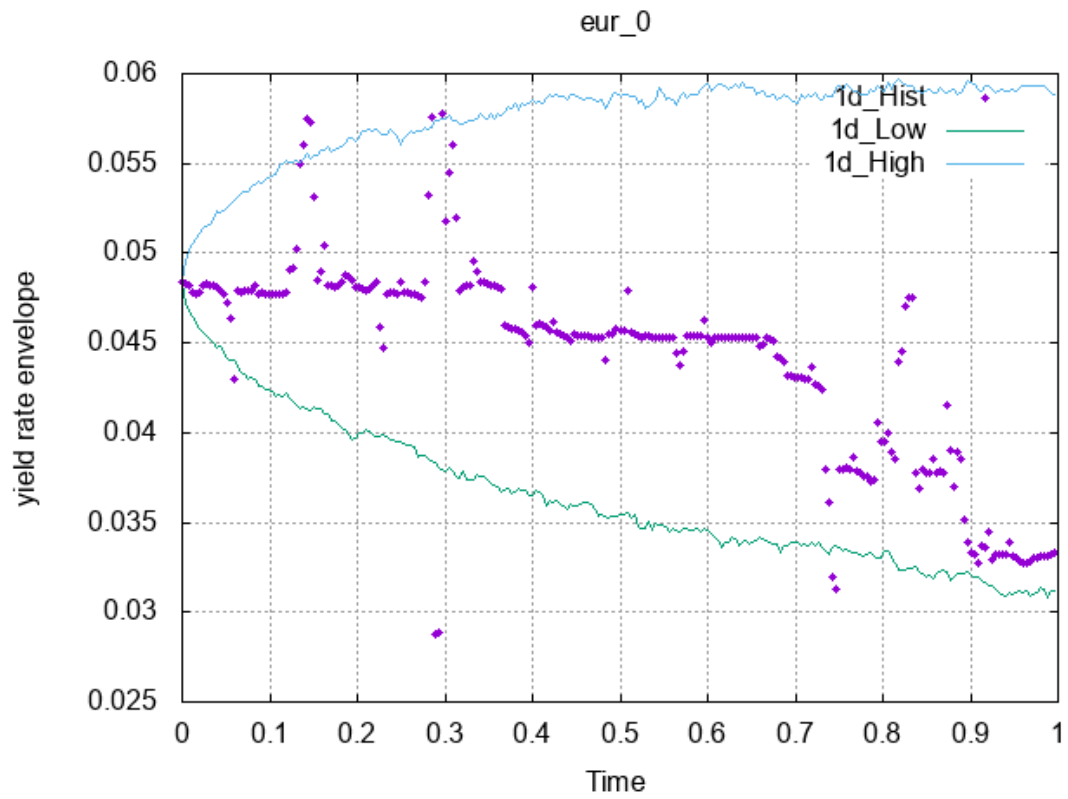


Figure 6: Simulation of dual GBP/EUR overnight (1 day) rate on optimal HJM parameters. Historical GBP data 2011: dots, upper and lower percentiles of simulated GBP rates: lines

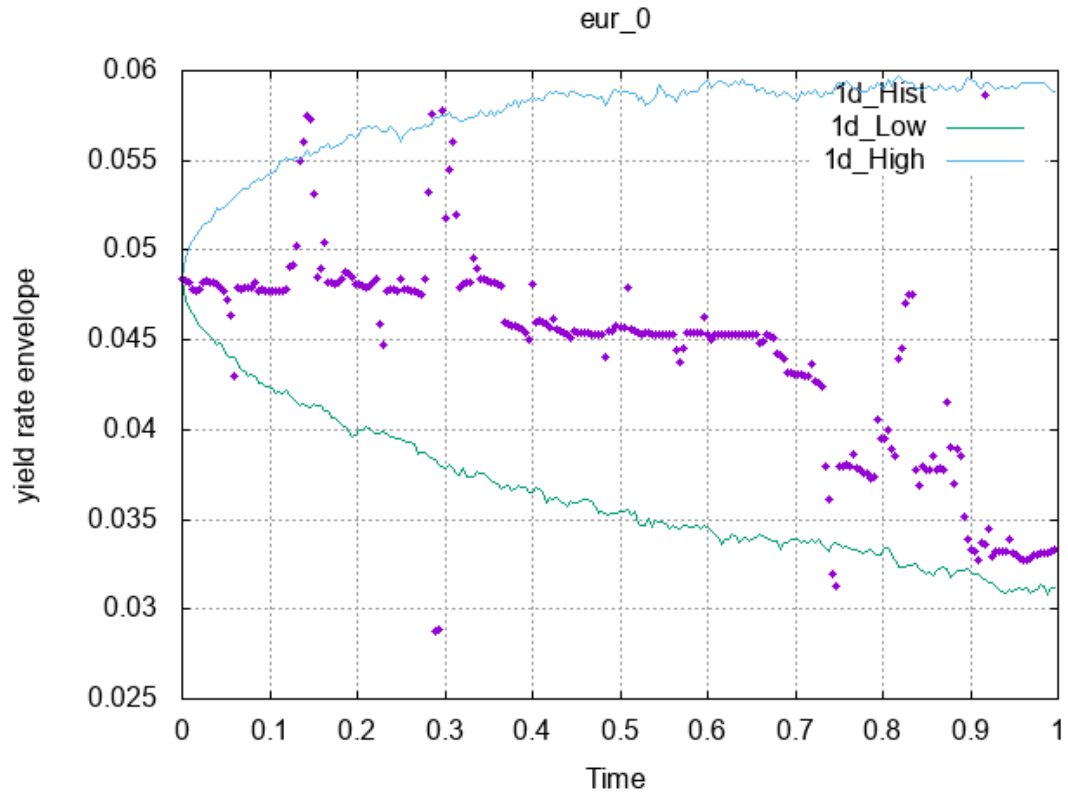


Figure 7: Simulation of dual GBP/EUR curves (maturity 1 year) rate on optimal HJM parameters. Historical GBP data 2011: dots, upper and lower percentiles of simulated GBP rates: lines

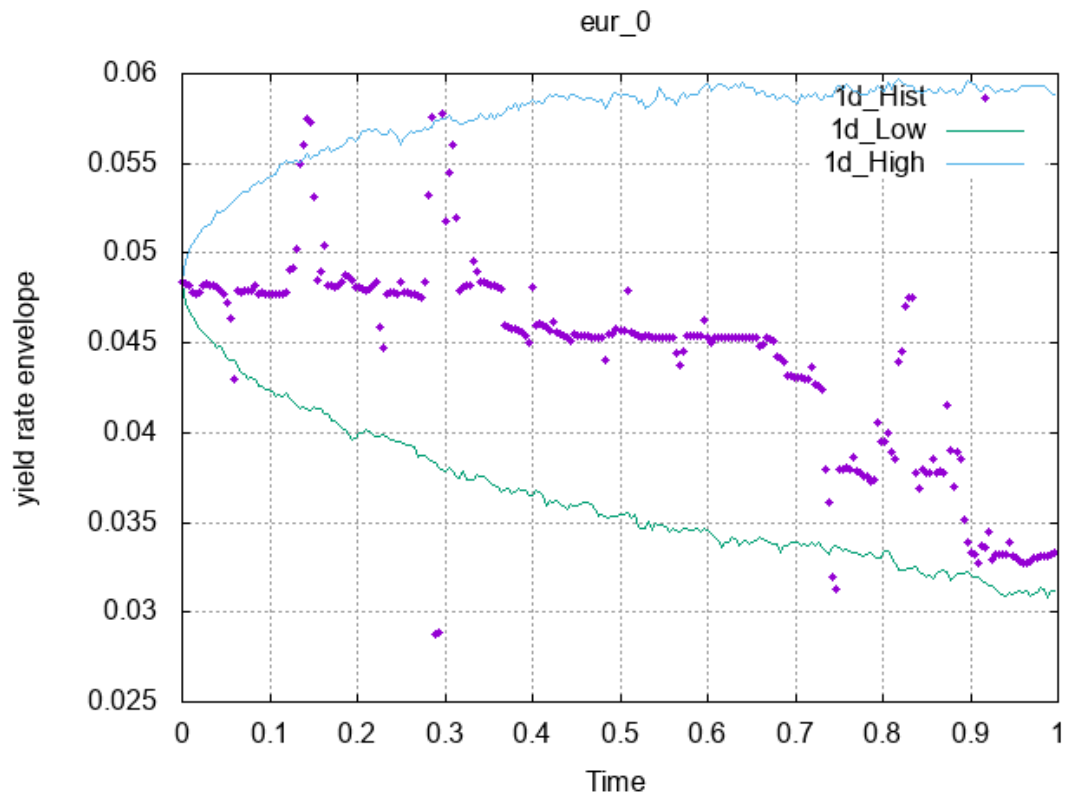


Figure 8: Simulation of dual GBP/EUR overnight (1 day) rate on optimal HJM parameters. Historical EUR data 2011: dots, upper and lower percentiles of simulated EUR rates: lines

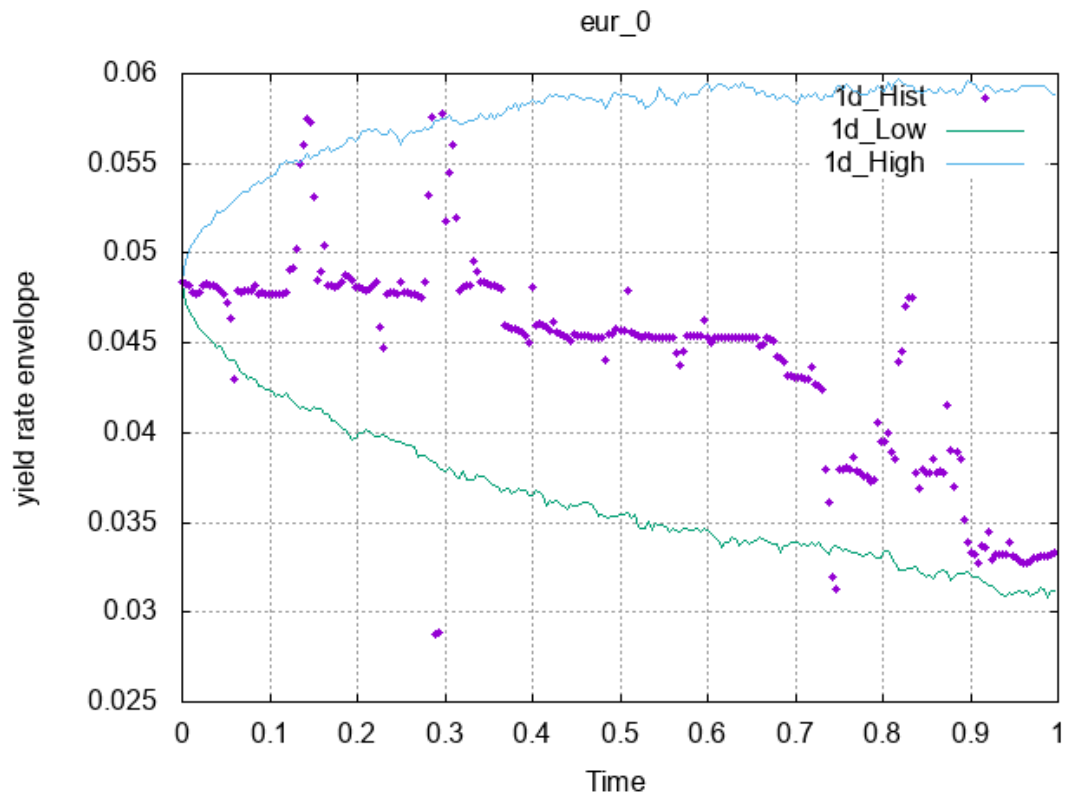


Figure 9: Simulation of dual GBP/EUR (maturity 1 year) rate on optimal HJM parameters. Historical EUR data 2011: dots, upper and lower percentiles of simulated EUR rates: lines

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

8 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.

9 References

- Aldrich, J.
1997. R. a. fisher and the making of maximum likelihood 1912 – 1922. *Statistical Science*, 12(3):162–176.
- Chiarella, C. and O. K. Kwon
2001. Classes of interest rate models under the hjm framework. *Asia-Pacific Financial Markets*, 8(1):1–22.
- 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.
- Itô, K.
1944. Stochastic integral. *Proceedings of Imperial Academy*, (20):519–524.