M2MO
DEA Laure Elie

Tutors

Yuan LI	(Lunalogic)
Simone SCOTTI	(Paris 7)

Libor Market Model

Version : 03/12/2014

Chi Thanh NGUYEN

Abstract

Master M2MO (**Laure Elie**'s DEA) is one of the most prestige Financial Engineering Master program in France, and Europe. At the end of this Master program, a minimum 6 months internship is required in order to assimilate partly the whole theoretical knowledge acquired from the Master. The intership for me take place at Lunalogic, a consulting & service enterprise for banks and insurances. The main objective at Lunalogic is to have internal libraries studying the financial modelling, with advanced mathematical tools implemented in C++ or C#. However this project is at its own begining phase, i.e models and tools are usually build independantly in form of prototyping (sub projects), ensuring a minimum of accuracy and code qualities. At a second phase, every prototype should be refactorized and integrate into a whole common library. In the internship at Lunalogic, we continu the work of [1] and use the book [3] [4]

Contents

1 About Lunalogic

Position	Name	Contact	Telephone
Exec Director	Fathia Rahmani	fathia.rahmani@lunalogic.com	
HR Director	Isabelle Marouby	isabelle.marouby@lunalogic.com	06.51.44.06.78
Quant Consultant	Yuan Li	revierdenord@gmail.com	06.52.17.69.75
Tech. support	Fastah Rahmani	fastah.rahmani@gmail.com	
Bloomberg supp.	Giulio D'Ambrosio	gdambrosio3@bloomberg.net	+442070733101

2 The Market Model

The aim of market model is to give a model that well price a set of basic instruments. In the LMM case study, the aim is to give a good pricing tool for plain vanilla products, such as Caps, Floors and Swaptions. In this section, firstly show the idea of numeraire change, this idea lies the model and the well-etablished pricing tool for simple instruments.

2.1 Numeraire change and market models

The two important numeraire changes that make the market model of this project are inspired from the Caplet and Swaption Black & Scholes frameworks.

Caplet and forward measure is an option on interest rate derivative. Caplet is analyzed as a european call option, with maturity time T_1 and pay at time T_2 for a certain strike K. The payment of such an option at time T_2 is obsevable at time T_1 and worth

$$\text{price}(T_1) = \tau(T_1, T_2)(L(T_1; T_1, T_2) - K)^+$$

where $L(T_1, T_1, T_2)$ is the observed Libor at time T_1. This payoff at the present time can be obtainted by discounting

$$\text{price}(0) = \mathbb{E}^B \left[\frac{B(0)}{B(T_2)} \tau(T_1, T_2)(L(T_1; T_1, T_2) - K)^+ \right]$$

with $B(t)$ the money market account. Since the zerocoupond $P(t, T_2)$ is a tradable instrument, and always have positive values, one can change the numeraire, and the pricing formula becomes

$$\text{price}(0) = \tau(T_1, T_2)P(0, T_2)\mathbb{E}^{T_2} \left[(L(T_1; T_1, T_2) - K)^+ \right]$$

Note that by using $P(t, T_2)$ as numeraire, $\mathbb{E}^{T_2}$ is defined as T_2-forward measure, and under such probability, the forward rate follow a martingale, i.e

$$\frac{dL(t; T_1, T_2)}{L(t; T_1, T_2)} = \sigma_L(t)dW_t^{T_2} \qquad\qquad t \leq T_1 < T_2$$

End then the fundamental Black price formula give at any time $0 \leq t < T_1$

$$\text{price}(t) = \tau(T_1, T_2)P(0, T_2) \left[L(t; T_1, T_2)\mathcal{N}(d_+) - K\mathcal{N}(d_-) \right]$$

with

$$d\pm = \frac{\ln \frac{L(t)}{K} \pm \frac{1}{2} \int_t^{T_1} \sigma_L^2(s)ds}{\sqrt{\int_t^{T_1} \sigma_L^2(s)ds}}$$

Swaption and swap measure is the option to enter into a interest rate swap. As for swap interest rate, swaption contract is settled on a set of payment dates $T_\alpha..T_i..T_\beta$ where T_α and T_β are start and end of swap instrument. The swaption's maturity date is the swap's start date T_α. The price value of such an option is caculable at time T_α by discounting

$$\text{price}(T_\alpha) = \left[S_{\alpha,\beta}(T_\alpha) - K \right]^+ C_{\alpha,\beta}(T_\alpha) \qquad C_{\alpha,\beta}(t) = \sum_{\alpha}^{\beta-1} \tau_i P(t, T_{i+1})$$

Where $C_{\alpha,\beta}(t)$ is the swap's annuity measure at time t. This is simply the payment sum of the fix leg on a swap configuration. Note that annuity is indeed a tradable product, as a sum of zerocoupons, always have positive value, so those can be used as numeraire. By the same way as discounting the caplet price, the present value of swaption is

$$\text{price}(0) = \mathbb{E}^B \left[\frac{B(0)}{B(T_2)} (S_{\alpha,\beta}(T_\alpha) - K)^+ C_{\alpha,\beta}(T_\alpha) \right]$$

and by change to numeraire $C_{\alpha,\beta}(t)$, under the associated probability $\mathbb{E}^{\alpha,\beta}$

$$\text{price}(0) = C_{\alpha,\beta}(0)\mathbb{E}^{\alpha,\beta} \left[(S_{\alpha,\beta}(T_\alpha) - K)^+ \right]$$

Since the numeraire change framework ensure the martingale properties, the swap interest rate $S_{\alpha,\beta}$ under the measure $\mathbb{E}^{\alpha,\beta}$ follows the martingale, in the way that it can be modelled by equation

$$\frac{dS_{\alpha,\beta}(t)}{S_{\alpha,\beta}(t)} = \sigma_{\alpha,\beta}(t) dW_{\alpha,\beta}(t)$$

Again, by the Black pricing formula, for any time $0 \leq t < T_\alpha$

$$\text{price}(t) = C_{\alpha,\beta}(t) \left[S_{\alpha,\beta}(t)\mathcal{N}(d_+) - K\mathcal{N}(d_-) \right]$$

with

$$d\pm = \frac{\ln \frac{S_{\alpha,\beta}(t)}{K} \pm \frac{1}{2} \int_t^{T_\alpha} \sigma^2_{\alpha,\beta}(s) ds}{\sqrt{\int_t^{T_\alpha} \sigma^2_{\alpha,\beta}(s) ds}}$$

These two pricing approachs introduce to two different ideas of modelling for differents basic instrument. That also show the incompatibilities issues of these two derivative's familly. The evidence come from the fact that the measure $\mathbb{E}^{T_2}$ associated to the numeraire $P(t, T_2)$ is different to the measure $\mathbb{E}^{\alpha,\beta}$ associated to the numeraire $C_{\alpha,\beta}(t)$.

2.2 Lognormal Forward Libor model

As its name, the Lognormal Forward Libor Model (or simply Libor Market Model), is the model that describ the dynamic of forward rate. The idea is if one know all values of forward rate, one can calculate easily the price of many related fixed income derivative as caps, floors, swaptions.

Model configuration is firstly defined by a timeline setting. Although the mathenatical definition of forward rate is a continuous functions, its implementation can only be done through a discrete configuration. There are different choice for time discretizetion, more or less dense, but we choose the simplest one : the dates coincide with the settlement date of Libor. In the US market, that is every 3MO, in the European market, that is every 6MO. By giving a maximum horizon, we discretize our times to a set of date $\xi = \{T_0, ..T_i, ..T_N\}$ [1]. After defining the timeline, we give some conventions for related values.

- horizon = N
- Timeline T_i , i = 0,1, ... N+1, T_0 = 0, Timeline size = N+2

[1]In the Bloomberg terminal one can get a yield curve for 50YR Max. There are no sens to define a timeline longer than 50YR

- Time Fractions $\tau_i = T_{i+1} - T_i$, i=0,1,... N.
- The i-th Libor value at time t : $L_i(t) = L(t, T_i, T_{i+1})$, Number of LIBORs = N+1

This timeline discretization at the first part facilite the whole implementation. Henceforth, every calculated values can be stored in a container, and every calculation loop iterate through time indices instead of time value.

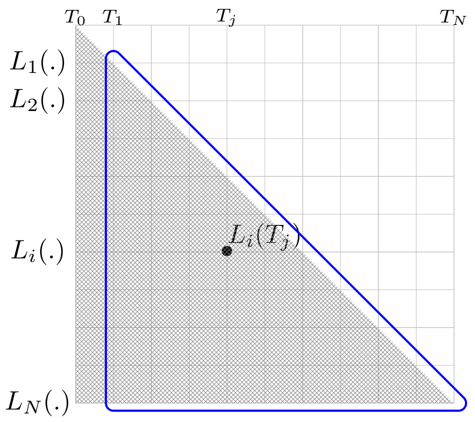


Figure 1: Lower triangular matrix storing Libor values. Volatilities use the same matrix format, but values used are only in blue zone

- A LIBOR is a process that is identified by its maturity time T_i, its values is calculated at every time $T_j \leq T_i$. Thus libors values can be stored in a lower triangular matrix $(N+1) \times (N+1)$, where (i, j) indices get the i-th Libor at time T_j
- A Vanilla Swaption is an instrument firstly defined by a pair of indices (α, β), meaning the expiry time T_α and tenor $T_\beta - T_\alpha$. Note that T_β is the end date of the swaption and $T_\alpha < T_\beta$ has to be ensured
- The deterministic volatility is defined in our implementation as piecewise volatilities. It is defined by three lower triangular matrix h, g and shift matrix, all of same format as matrix used for Libor values. Note that the first row and first column of volatility matrices are not used since at time zero, all libor are know by extracting data from the quoted market.
- correlation model is a little more complicated, but we can say that the correlation between two libor $L_i(t)$,$L_j(t)$ is modelled by a correlation value ρ_{ij}. Roughly speaking, it is also a symetric matrix of size $(N+1) \times (N+1)$.

Forward libor dynamic is described by lognormal processes. For every settlement date T_i, under the numeraire $P(t, T_{i+1})$, the associated forward libor $L(t, T_i, T_{i+1})$ follow the dynamic

$$\frac{dL_i(t)}{L_i(t)} = \sigma_i(t) dW^{T_{i+1}}(t) \qquad dW^{T_{i+1}}(t) dW^{T_{i+1}}(t) = \hat{\rho} dt$$

After translating the brownian $N-i$ times, under the numeraire $P(t,T_{N+1})$, for every date T_i, the dynamic of the forward libor remain a lognomal process, but this time with drift under the terminal probability

$$\frac{dL_i(t)}{L_i(t)} = \mu_i(t)dt + \sigma_i(t)dW^{T_{N+1}}(t) \qquad dW^{T_{N+1}}(t)dW^{T_{N+1}}(t) = \hat{\rho}dt$$

Since the Brownian is now the unique one under the terminal measure, we can ommit the subscript for simplify the notations. Furthurmore, we introduce other indices used for indicating the multidimentional brownian and its correlation. The forward libor dynamic finally become

$$\frac{dL_i(t)}{L_i(t)} = \mu_i(t)dt + \sigma_i(t)\hat{\rho}^{\frac{1}{2}}dW(t) \qquad \mu_i(t) = \sum_{k=i+1}^{N+1} \frac{\tau_j L_j(t)}{1+\tau_j L_j(t)}\sigma_i(t)\sigma_j(t)\hat{\rho}_{ij} \tag{1}$$

Correlation should be a function of time. But a first implementation can simplify by a function constant in time. [3]p.247 mention that the correlation has to satisfy some financial feature

- $0 \le \hat{\rho}_{ij} \le 1$ for all i,j, a symetric matrix
- $\hat{\rho}_{kk} = 1$ for all k
- When moving away from the prinicipal diagonal, $\hat{\rho}_{ij}$ decrease
- When moving along a sub-diagonal from top to bottom, $\hat{\rho}_{ij}$ increase

There are several way of modelling the correlation. The essential use should be a parameterized one proposed in [9]

$$\hat{\rho}_{ij} = e^{-\beta|T_i - T_j|}$$

Decorrelation An issue in term of computation time araise when model use fully the number of factor. Imagine for a 15YR horizon, in the european market, that say there are 30 Libors to simulate. If the model try to simulate fully 30 random factor, every sampling has to fill a 30 lower matrix, multiply with the number of sample, that could be very time comsuming. A mechanism of reduction factor is proposed through the correlation matrix. Assuming the correlation matrix ρ is symetric definite positive,by SVD factorization, it can be written as

$$\hat{\rho}_{ij} = PDP^t$$

Where D is the diagonal matrix containing all positive eigenvalues, P is a real unitary matrix ($PP^t = P^tP = I$), containing eigenvectors associated to eigenvalues. Let $A = PD^{\frac{1}{2}}$,

$$AA^t = \hat{\rho}_{ij} \qquad A^tA = D$$

By removing the smallest entries in the matrix $D^{\frac{1}{2}}$ and the associated column in the matrix P to get $\tilde{D}^{\frac{1}{2}}$ and $\tilde{P}$, a reduced correlation matrix is given by

$$\rho = \tilde{P}\tilde{D}\tilde{P}^t \qquad \rho^{\frac{1}{2}} = \tilde{P}\tilde{D}^{\frac{1}{2}}$$

The Libor dynamic becomes

$$\frac{dL_i(t)}{L_i(t)} = \mu_i(t)dt + \sigma_i(t)\rho^{\frac{1}{2}}dW(t) \qquad \mu_i(t) = \sum_{k=i+1}^{N+1} \frac{\tau_j L_j(t)}{1+\tau_j L_j(t)}\sigma_i(t)\sigma_j(t)\rho_{ij} \tag{2}$$

Volatility and time homogeneity feature
Rebonato in [8] explain that volatility in a normal market condition have a humped shaped. Furthermore for a reason of stability through times, volatility should be modelled as a function of time to maturity. This property is indeed time homogeneous. A parameterized volatility model aslo known as "abcd" volatility having a perfect time homogeneous properties and can match many type of volatility shape

$$\Phi(T-t) = (a + b(T-t))e^{-c(T-t)} + d \tag{3}$$

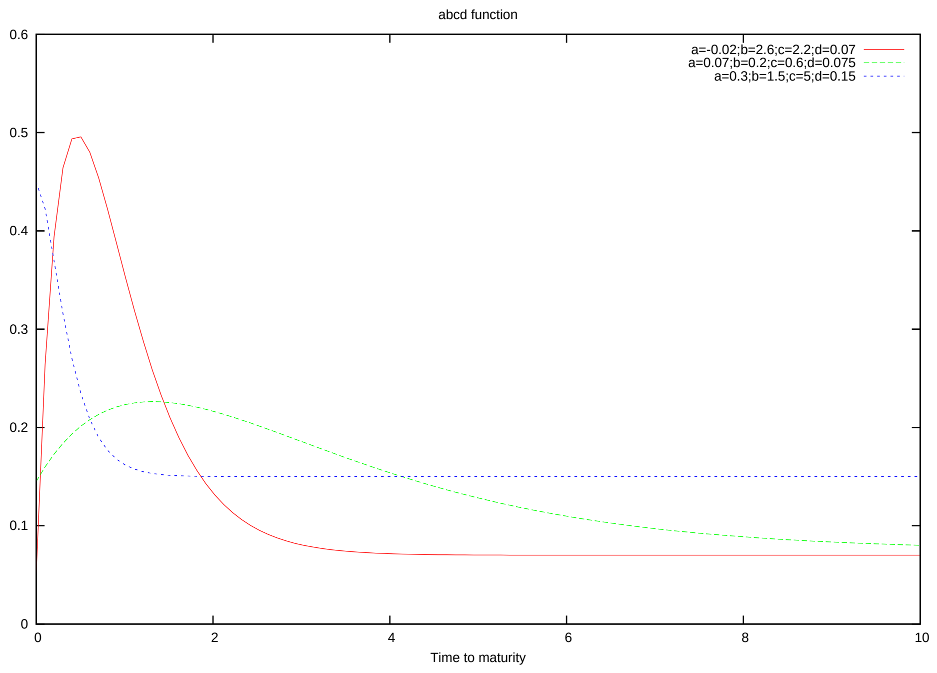


Figure 2: Different humped volatility can be modelled by abcd function

Piecewise volatility
The simple parameterized functional form 3 however is too simple for volatility in LMM model. Firstly, the true volatility should not have such the perfect shape. Secondly, the fewness of parameters make model a small number of degree of freedom, thus difficult to calibrate. Rebonano has proposed several ways for sophistifying the volalility model. In our implementation, we choose a simple but enoughly general approach, called *hgVolatility*. The idea is to separate general volatility function into two part h and g . The h part reflect the time homognenuous properties, and the g part give more degree of freedom to the model. That facilite the calibration task and allow model to be more flexible. In this way, h and g are two independant lower triangular matrices (the same structure as libors). The matrix h will be initiate by an appropriated "*abcd*" values, where "*abcd*" can be calibrated or given by traders. The matrix g is the center of the volatility calibration task. The good result is to have *abcd* values such that g matrix have

values approximatively 1.

$$\sigma_{ij} = h_{ij} g_{ij} \qquad \begin{cases} g_{ij} = g_i(T_j) \\ h_{ij} = h_i(T_j) \end{cases} \quad j \leq i \qquad \begin{cases} g_{0j} = g_{i0} = 0 \\ h_{0j} = h_{i0} = 0 \end{cases} \quad \forall i, j$$

2.3 Approximation Rebonato's formula

As we have seen the issue from incompatibility of forward swap model and the forward rate model. In [7], authors explain a link between these two model, through an approximative formula relying the caplet's implied volatility and the swaption's implied volatility. The idea start from the forward swap rate model, under the swap measure, the forward swap rate follows a martingale

$$\frac{dS_{\alpha,\beta}(t)}{S_{\alpha,\beta}(t)} = \sigma_{\alpha,\beta}(t) dW_{\alpha,\beta}(t)$$

in the way that

$$[\sigma_{\alpha,\beta}(t)]^2 = \frac{d < S_{\alpha,\beta}, S_{\alpha,\beta} >_t}{[S_{\alpha,\beta}(t)]^2}$$

Since from the swap rate equation 6, the volatility dynamic becomes

$$[\sigma_{\alpha,\beta}(t)]^2 = \frac{1}{[S_{\alpha,\beta}(t)]^2} \sum_{i,j=\alpha}^{\beta-1} w_i(t) w_j(t) d < L_i, L_j >_t \qquad w_k(t) = \frac{\sum_{k=\alpha}^{\beta-1} \tau_k P(t, T_{k+1})}{C_{\alpha,\beta}(t)}$$

or more detailed

$$[\sigma_{\alpha,\beta}(t)]^2 = \frac{1}{[S_{\alpha,\beta}(t)]^2} \sum_{i,j=\alpha}^{\beta-1} w_i(t) w_j(t) L_i(t) L_j(t) \rho_{L_{ij}}(t) \sigma_{L_i}(t) \sigma_{L_j}(t) dt$$

P.Jackel and R.Rebonato studied the quantity

$$\zeta_{ij}(t) = \frac{1}{[S_{\alpha,\beta}(t)]^2} \sum_{i,j=\alpha}^{\beta-1} w_i(t) w_j(t) L_i(0) L_j(0)$$

and constate that its variations through time are negligible, thus one can approximated those by values at time zero. The swaption's volatility dynamic become

$$[\sigma_{\alpha,\beta}(t)]^2 \approx \frac{1}{[S_{\alpha,\beta}(0)]^2} \sum_{i,j=\alpha}^{\beta-1} w_i(0) w_j(0) L_i(0) L_j(0) \rho_{L_{ij}}(t) \sigma_{L_i}(t) \sigma_{L_j}(t) dt$$

The swaption's implied volatility is now calculable by the approximation formula

$$T_\alpha [\sigma_{\alpha,\beta}^{BLACK}]^2 \approx \frac{1}{[S_{\alpha,\beta}(0)]^2} \sum_{i,j=\alpha}^{\beta-1} w_i(0) w_j(0) L_i(0) L_j(0) \int_0^{T_\alpha} \rho_{L_{ij}}(t) \sigma_{L_i}(t) \sigma_{L_j}(t) dt \tag{4}$$

2.4 Market Information concerning Libor Model

The bijective relation between forward Libor and zero coupon

$$L_i(t) = L(t, T_i, T_{i+1}) = \frac{1}{\tau_i}\left[\frac{P(t,T_i)}{P(t,T_{i+1})} - 1\right] \qquad P(t,T_{i+1}) = \frac{P(t,T_i)}{1+\tau_i L_i(t)} \tag{5}$$

After having these informations, i.e zero-coupond prices or all Forward Libor Rate, one can compute the related swap rate starting at date T_α, ends at date T_β

$$S_{\alpha,\beta}(t) = \frac{P(t,T_\alpha) - P(t,T_\beta)}{\sum_{\hat{i}=\hat{\alpha}}^{\hat{\beta}-} \tau_{\hat{i}} P(t, T_{\hat{i}++})} = \frac{\sum_{i=\alpha}^{\beta-1} \tau_i P(t,T_{i+1})}{C_{\alpha,\beta}(t)} L_i(t) \tag{6}$$

For the initial forward LIBOR (which is different than the spot LIBOR)

$$L_i(0) = L_i = \frac{1}{\tau_i}\left[\frac{ZC_i}{ZC_{i+1}} - 1\right] = \frac{1}{\tau_i}\left[\frac{Num_{i+1}}{Num_i} - 1\right]$$

Then

$$\begin{cases} Num_0 = 1 \\ Num_{i+1} = Num_i(\tau_i L_i + 1) \end{cases} \qquad \begin{cases} ZC_0 = 1 \\ ZC_{i+1} = \frac{ZC_i}{\tau_i L_i + 1} \end{cases}$$

The forward discount factor (useful in Swap configuration) is defined for a $t \le T_\alpha < T_k$

$$\mathrm{FP}(t, T_\alpha, T_k) = \frac{P(t,T_k)}{P(t,T_\alpha)} = \prod_{i=\alpha}^{k-1} \frac{P(t,T_{i+1})}{P(t,T_i)} = \prod_{i=\alpha}^{k-1} \frac{1}{1+\tau_i L(t,T_i,T_{i+1})}$$

The swap rate is computed by [2]

$$S_{\alpha,\beta}(t) = \frac{P(t,T_\alpha) - P(t,T_\beta)}{\sum_{\hat{i}=\hat{\alpha}}^{\hat{\beta}-} \tau_{\hat{i}} P(t, T_{\hat{i}++})} = \frac{\sum_{i=\alpha}^{\beta-1} \tau_i P(t,T_{i+1})}{A_{\alpha,\beta}(t)} L_i(t)$$

2.5 Translation probability toward Libors dates

The aim is to be able to express the T_{i+1}-forward measure and its associated Brownian to the T_i-forward measure and its associated measure. From the numeraire change formula, we have

$$\frac{dQ^{T_{i+1}}}{dQ^{T_i}}|_{\mathcal{F}_t} = \frac{P(0,T_{i+1})}{P(0,T_i)} \frac{P(t,T_i)}{P(t,T_{i+1})} = \mathbf{c}(\tau_i L_i(t) + 1)$$

If we can express the right hand side as an exponential process, then we can apply the Girsanov theorem for getting the Brownian translation relation. In fact

$$\frac{d(\mathbf{c}(\tau_i L_i(t)+1))}{\mathbf{c}(\tau_i L_i(t)+1)} = \frac{\tau_i dL_i(t)}{\tau_i L_i(t)+1} = \frac{\tau_i L_i(t) s_i(t) B_i(t)}{\tau_i L_i(t)+1} dW^{T_{i+1}} = \mu_i(t) dW^{T_{i+1}}$$

Remark that the last equation has a solution as geometric brownian motion :

$$\mathbf{c}(\tau_i L_i(t) + 1) = \exp\left\{\int_0^t \mu_i(s) dW_s^{T_{i+1}} - \frac{1}{2}\int_0^t \mu_i^2(s) ds\right\}$$

[2] The iteration through indices $\hat{i}$ and i is different when the swap settlement dates are different for fixed legs and floatting legs. $\hat{i}$ iterate through fixed legs while i iterate through floatting legs. Their steps can be different depending on Fixed/Float Tenor configuration

Plug into the probability equation, we can now apply the Girsanov theorem

$$\frac{dQ^{T_{i+1}}}{dQ^{T_i}} = \exp\left\{\int_0^t \mu_i(s)dW_s^{T_{i+1}} - \frac{1}{2}\int_0^t \mu_i^2(s)ds\right\} \qquad dW^{T_i} = dW^{T_{i+1}} - \tilde{\mu}(t)dt$$

Or in a general formula

$$dW^{T_k} = dW^{T_{k+1}} - \frac{\tau_k L_k(t)}{\tau_k L_k(t)+1} s_k(t)B_k(t)dt$$

3 Deterministic volatility calibration

This section explain how process the calibration algorithm in our LMM Model, what are input data, what are ouputs, how to test and say that our calibration is good. Firstly, we will explain what market data we need and how to retreive them. We then explain the market data issu and how to resolv this by our algorighm with a set of numerical techniques (interpolation, solver ...). At the end, we explain how we test our algorithm before using real market data, and how is tested with the real market data.

3.1 Bloomberg and Market Data

The first step to do with the calibration is to retreive market information. At Lunalogic RD team, we have a Bloomberg terminal and all of our data comes from this. We explain here how we retreive data with Bloomberg. In a general way, a full yield curve for horizon 50YR does not exist. One has to use quoted instruments and AOA formula, interpolation tools to fully build the interest rate curve. These instruments frequently used are

- Quotation from EURIBOR, used for short horizon
- FRA, used for midle horizon
- Vanilla Interest rate Swap, used for long horizon

There exist many method to build interest rate curve, that we will not focus here. We suppose this curve is already built. In reality, Bloomberg propose tools to build this curve, that one can take a look at [2] or use the bloomberg tool ICVS (Swap Curve Builder).

Note that all of our study are european-based. So when going to Bloomberg, we choose EUR currency, the fixed/float tenor are only choosen by 6M/1YR as standard vanilla european swap configuration. Two main data we need for the calibration are Initial Libor curve and the Black Volatility of ATM Swaption matrix.

Initial Libor is calculated from a zerocoupond set, which is available from Bloomberg (not directly). In Bloomberg terminal, we use

ICVS $\longrightarrow$ 14) EUR $\longrightarrow$ 21) STRIPPED CURVE $\longrightarrow$ EXPORT TO EXCEL

That give the discount curve for 50YR horizon. After getting this curve, we interpolate linearly at dates multiple of 6MO, and eliminate all elements that are not at date multiple of 6MO. Values from this curve are used as discount factors, in equations 5, they are $P(0,T_i)$, and we can have every $L(0,T_i,T_{i+1})$ from this.

Swaption ATM Black's Vol and skew is available from Bloomberg by ticker VCUB. After going into this page, user have to choose the right currency, the right tenor type (6M for european swaps), then refresh information. One can get Black's Vol quote for ATM swaptions, OTM or ITM swaptions and the correspondant strikes matrix by giving the moneyness information. We get the derivative skew matrix by finite difference method

$$\mathbf{derivative}(skew) = \frac{quote(\mathbf{ATM{+}5bp}) - quote(\mathbf{ATM\text{-}5bp})}{\mathbf{10bp}}$$

The post treatment is similar as Libor quotes, we eliminate all experities that are not multiple of 1YR. The difference is that for swaptions quote, we only use dates multiple of 1YR. Note that since our LMM model has fixed a horizon as a maximum of date, then only swaptions having the end date smaller then this horizon can be choosen. Thus only the upper anti triangular part of swaption quote matrix will be used for the calibration.

Data check Before using the data, we always check if the data is coherent, i.e Libor quotes is coherent with swaptions. Thanks to analytical swap rate formula 6 and the Libor quotes, we recompute our swap rate matrix, and compare with the ATM swaptions strike quote. The ideal case has to match exactly the same values. But in reality, there are a difference about 1%-5%, that is negligible and we accept this difference since we didn't send time to rigorously build our yield curve.

3.2 Algorithm description

The main task of our calibration is to calibrate the g matrix. In this section, we introduce the analysis of volatilities into market data quotation. Based on this analysis, three calibration algorithms are implemented : cascade, local, global. The last one allow a regularized solution when market data do not allow a solution. At the end, we also explain a quick implementation for calibrating h matrix, or indeed the *abcd* parameters.

Market data and volatilities dependency
Although many technical note talk about calibration LMM model with Cap/Caplet quotes from market, we choose to use Swaption quotes from market. The reason is Swaption informations are richer than Cap/Caplet market. Swaption market have more quotes, and allow also calibrate correlation. In this paragraph, we reexplain the P.Jackel and R.Rebonato's analysis in [7].

When calibrating using Implied Swaption Volatilities quoted on the market, a problem arise if swaptions has different configuration of fixed tenor and float tenor. See from the figure example, on an horizon 5 year, with float/fixed tenor are 6M/1YR, we have only 10 swaption's data from the market, while the model has 55 parameters (elements of g matrix). The excessive number of parameters lead to infinity of solutions. Then we use a special calibration's process to avoid this problem. The idea is to choose parsimoniously a set of appropriated parameters in the g matrix, in the way that there are the same number of parameters to be calibrated than the number of swaption's quotation. After every calibration step (minimization iteration) an interpolation process are launched in order to taking in acount for the whole matrix.

Let's take a look in detail into each swaptions. Since a swaption depend to a set of determined Libors (see 6), each Libor is modelled by a set of parameters, we can deduces the dependant parameters. For example on a setting of 5YR horizon, with float/fixed tenor 6M/1YR, the swaption 3YRx2YR depend to Libors L_6, L_7, L_8, L_9. Since the swaption's maturity is 3YR=T_6, the

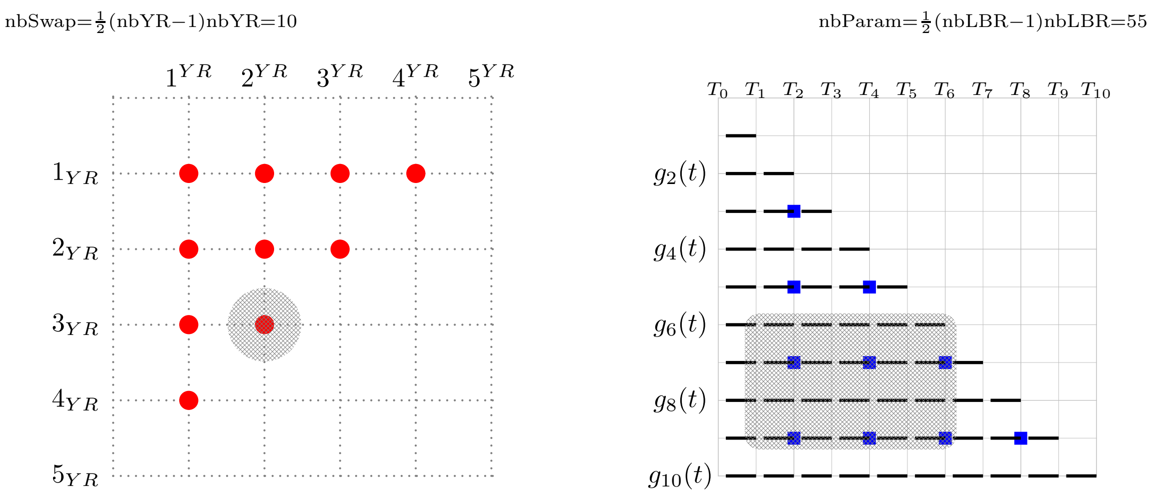

nbSwap=$\frac{1}{2}$(nbYR−1)nbYR=10
nbParam=$\frac{1}{2}$(nbLBR−1)nbLBR=55
1^{YR} 2^{YR} 3^{YR} 4^{YR} 5^{YR}
1_{YR} 2_{YR} 3_{YR} 4_{YR} 5_{YR}
T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}
$g_2(t)$ $g_4(t)$ $g_6(t)$ $g_8(t)$ $g_{10}(t)$

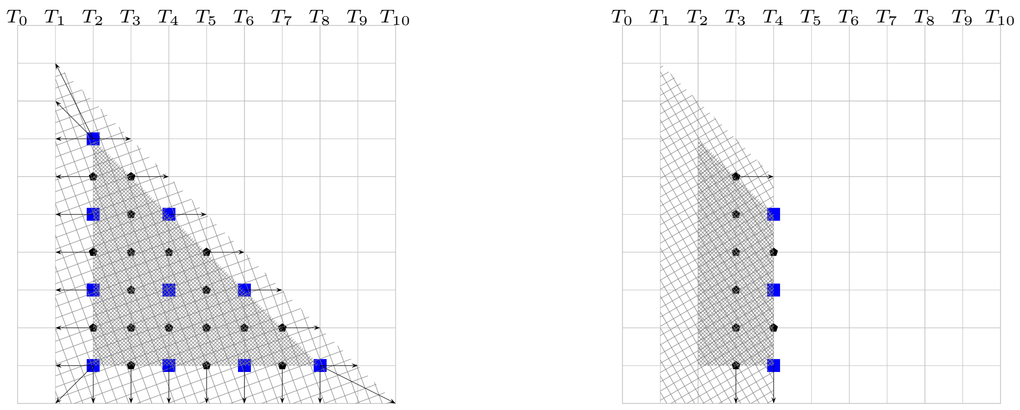
Figure 3: Dependency of swaption quotes and volatilities parameters

Swaption	Libor Dep.	Full Vol Dep.	Sparse Vol Dep.
1YRx1YR	L_2, L_3	$g_{ij,2\leq i\leq 3,1\leq j\leq 2}$	g_{32}
1YRx2YR	$L_2, ...L_5$	$g_{ij,2\leq i\leq 5,1\leq j\leq 2}$	g_{32}, g_{52}
1YRx3YR	$L_2, ...L_7$	$g_{ij,2\leq i\leq 7,1\leq j\leq 2}$	g_{32}, g_{52}, g_{72}
2YRx1YR	L_4, L_5	$g_{ij,4\leq i\leq 5,1\leq j\leq 4}$	g_{52}, g_{54}
2YRx2YR	$L_4, ...L_7$	$g_{ij,4\leq i\leq 7,1\leq j\leq 4}$	$g_{52}, g_{54}, g_{72}, g_{74}$
3YRx1YR	L_6, L_7	$g_{ij,6\leq i\leq 7,1\leq j\leq 6}$	g_{72}, g_{74}, g_{76}

Table 1: Dependency of Swaption and Nodes in g matrix

swaption depend only to parameters anterior to time T_6, i.e $g_{ij,6\leq i\leq 9,1\leq j\leq 6}$. From Rebonato approximation formula 4, only $g_{ij,6\leq i\leq 9,1\leq j\leq 6}$ has impact to the Implied Vol of swaption 3YRx2YR. In general, we choose parsimoniously parameters by jumping indices step fixed tenor/float tenor. In our implementation, we choose $g_{72}, g_{74}, g_{76}, g_{92}, g_{94}, g_{96}$ as principale dependant parameters, the others will be deduced by interpolation-extrapolation. An explicite example of swaptions 1YR/6M and parameter's dependance for horizon 4YR are described in the table 1. These choice of parameters make the number of unknow parameters equal to the number of equation, thus the calibration system become solvable.

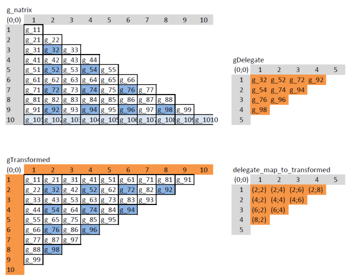
Interpolation - Extrapolation volatility matrix
From the dependency analysis, the calibration problem is now reduced into finding nodes elements in g matrix instead of finding the whole g matrix. Element out of nodes are found by interpolattion, extrapolation. Thanks to the dependency analysis, we can calibrate by using a ”stripping” algorithm, i.e by moving in the swaption quote matrix left-to-right and top-to-bottom, we move parrallelly in the g matrix top-to-bottom then left-to-right (see [7]). Our interpolation-extrapolation has to ensure this strictly dependance, i.e a newly found node in g matrix is related to a swaption node, and these interpolation-extrapolation have to not modify



T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}
T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

Figure 4: Interpolation and Extrapolation for volatility matrix

values that found precedently. For example in figure 4 the node g_{74} in the stripping algorithm correspond to the swaption 2YRx2YR, the change if this node has to modify only and only g_{73}, g_{63} and g_{64}.



g_natrix

(0;0)	1	2	3	4	5	6	7	8	9	10
1	g_11									
2	g_21	g_22								
3	g_31	g_32	g_33							
4	g_41	g_42	g_43	g_44						
5	g_51	g_52	g_53	g_54	g_55					
6	g_61	g_62	g_63	g_64	g_65	g_66				
7	g_71	g_72	g_73	g_74	g_75	g_76	g_77			
8	g_81	g_82	g_83	g_84	g_85	g_86	g_87	g_88		
9	g_91	g_92	g_93	g_94	g_95	g_96	g_97	g_98	g_99	
10	g_101	g_102	g_103	g_104	g_105	g_106	g_107	g_108	g_109	g_1010

gDelegate

(0;0)	1	2	3	4	5
1	g_32	g_52	g_72	g_92	
2	g_54	g_74	g_94		
3	g_76	g_96			
4	g_98				
5					

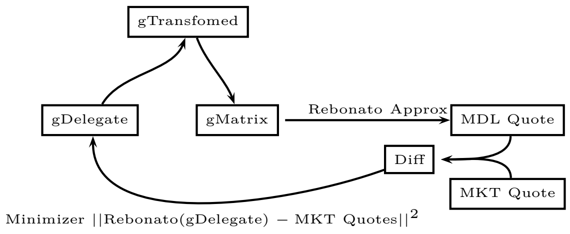
gTransformed

(0;0)	1	2	3	4	5	6	7	8	9	10
1	g_11	g_21	g_31	g_41	g_51	g_61	g_71	g_81	g_91	
2	g_22	g_32	g_42	g_52	g_62	g_72	g_82	g_92		
3	g_33	g_43	g_53	g_63	g_73	g_83	g_93			
4	g_44	g_54	g_64	g_74	g_84	g_94				
5	g_55	g_65	g_75	g_85	g_95					
6	g_66	g_76	g_86	g_96						
7	g_77	g_87	g_97							
8	g_88	g_98								
9	g_99									
10										

delegate_map_to_transformed

(0;0)	1	2	3	4	5
1	(2;2)	(2;4)	(2;6)	(2;8)	
2	(4;2)	(4;4)	(4;6)		
3	(6;2)	(6;4)			
4	(8;2)				
5					

(a) Matrix Mapping



(b) Calibration pipeline

Figure 5: G Matrix Mapping

Matrix Mapping
Due to the different natural matrices direction (swaption quote matrix versus g matrix), we implement a structure of "matrix mapping" helping to manage all these desorder indices. The gDelegate matrix is a upper anti-triangular matrix, store all nodes values of gMatrix. Note that this matrix has the exact same direction as the swaption quote matrix when processing the stripping algorithm. gDelegate matrix fetch value into nodes of gTransformed matrix. This latter always have the same "natural" direction, but have more values, and all interpolation, extrapolation will be done in this matrix. After interpolation, extrapolation, values in gTransfomred matrix are transformed into g matrix, which is the volatility matrix of the LMM model (see 5a). This representation allow to define a consensual data structure faciliting the calibration algorithm. Now, parameters of the calibration problems are all stored in the gDelegate matrix,

and the whole general calibration algorithm is described as in the pipeline 5b.

From theses above analysis and defined elementary tools, we are able to implement three calibration algorithm with the same routine as in 5b. The only different is how much data we take for each minimization loop.

- Global : The whole swaption quote matrix and the whole gDelegate matrix is put entirely into the solver.
- Local : The whole calibration is divided into line stripping algorithm. Each time one line of the swaption matrix quote and one line of the gDelegate matrix are put into the solver, in the top-to-bottom direction.
- Cascade : The calibration is divided into emement stripping algorithm. Each time only one element of the swaption matrix quote and one element of the gDelegate matrix are put into the solver, in the direction left-to-right and top-to-bottom.

3.3 Regularization

As we can see, the separation of volatility function σ into h and g has the advantage that is more flexible. h matrix reflect perfectly the time homogeneuous property, while g matrix allow to add more degree of freedom to the volatility model. The downside of this separation is that if elements in g matrix varies two much, the product $\sigma_{ij} = h_{ij} g_{ij}$ also varies and the existence of h loose its sens. Furthermore in a Caplet quote analysis in [8], Rebonato show that for the existing of solution, the variance of Black volatility has to be an increasing function in maturity.

$$\text{Variance}(T_i) = T_i \sigma^2_{\text{BLACK}}(T_i)$$

It is not exactly the same condition for swaption matrix, but when analyse the VCUB quotation, we've checked that the calibration behave badly when variance do not have a good shape. For resolving all theses issues, we put two penalties in our calibrator. User have the choice to active or unactive theses regularizator independant to the goodness of market data. Regularization can be added to the calibrator in the general form by adding the penalties term

$$\textbf{Cost Function} = (\textbf{MDL Quote} - \textbf{MKT Quote}) + \text{Pel}_T + \text{Pel}_M$$

that we will explain how to compute the penalties term as below.

Time homogeneity penalty
This penaly regularize the parameters in the sens that force the time homogeneuous property. In the ideal case

$$g(t,T) = \Phi(T-t) = \Phi(\tau) \qquad\qquad \forall 0 \le t \le T \quad \tau = T - t$$

In order to have this property, we force the derivative of g in relation to $\tau = T - t$ to be small

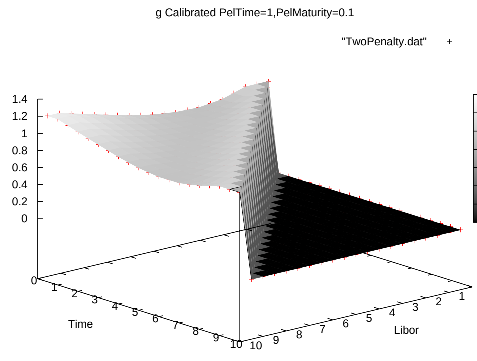
$$|\frac{\partial g}{\partial \tau}(t,T)| < \text{Const} \qquad\qquad \tau = T - t$$

This constraint translate into the minimun variation on the diagonals of g matrix. Since elements nodes on a diagonal in g matrix correspond to a column in gDelegate matrix, we compute the

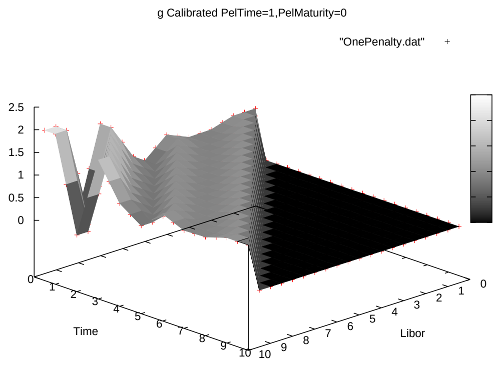
Time homogeneuous penalty by finite difference method

$$\text{Pel}_T = \mathbf{c}_T(\sum_j \sum_i \text{gDelegate}_{(i+1,j)} - \text{gDelegate}_{(i,j)})$$

where $\mathbf{c}_T$ is a constant regrouping penalty ratio, normalization coefficient, and the denominator part of the finite difference formula. Normally the finite difference denominator and the nomalization coefficient are not to be changed, user change only penalty coefficient, which is equivalently to change $\mathbf{c}_T$.



(a) $\mathbf{c}_T$=1,$\mathbf{c}_M$=0.1



(b) $\mathbf{c}_T$=1,$\mathbf{c}_M$=0

Figure 6: Penalties Setting on a virtual calibration test of 10YR : a) Smoothness on maturity b) Without smoothness on maturity

Smoothness on maturities

The second penalty is to force the smoothness to g function in relation to the maturity.

$$|\frac{\partial^2 g}{\partial T^2}(t,T)| < \text{Const}$$

This constraint translate into forcing the smooth variation on a columns of g matrix. Since elements nodes on a column in g matrix correspond to a row in gDelegate matrix, we compute the Maturity smoothness penalty again by finite difference method

$$\text{Pel}_M = \mathbf{c}_M(\sum_i \sum_j \text{gDelegate}_{(i,j-1)} + \text{gDelegate}_{(i,j+1)} - 2\text{gDelegate}_{(i,j)})$$

As the case for Time homogeneuous penalty, if the coefficient set to zero, the penalty will not be setted.

Note that while the first penalty is added to have the good financial feature, the second penalty is added in order to have the correctness of the solution. If only time homogeneuous penalty is added, the calibrator do everything to satisfy the constant diagonal property of the matrix g. An undesirable behavior comes out that calibrator increase one diagonal, and decrease the others. This solution can match the quoted price, but as we see in figure 6b, that is a very irregular solution. When adding a second penalty, we have a smooth solution as shown ins figure 6a

3.4 Tests and numerical results

Before performing a calibration on real market data, we built a set of virtual test, and analyse the quality of our calibrators. A good calibrator has to ensure the accuracy and the stability of the result in relation to the data.

General virtual test
As shown in figure 7, our virtual test is explained as below. Everything will firstly be tested in a virtual context. We initially have a gMatrix, supposed to be a *true parameters*. Using this *true parameters* and our model, we generate the virtual market data by our pricer. Then we perturbe our true parameters into a false parameters set, with a certain noises, to hide the *true parameters*. Virtual market data then is put into the calibrator as a input. The calibrator in an exact calibration (without regularization) has to find out the initial *true parameters*. Each time whem we do a perturbation, we use the pseudo random generator implemented in Boost library, and we retain the seed of this generator. Doing so allow us to reproduce exactly the behavior. In virtual test, as we known the *true parameters*, we can calculated two error types after calibration, the error of parameters (difference between calibrated and *true parameters*), and the error of quotation (difference between market quotations and model quotations).

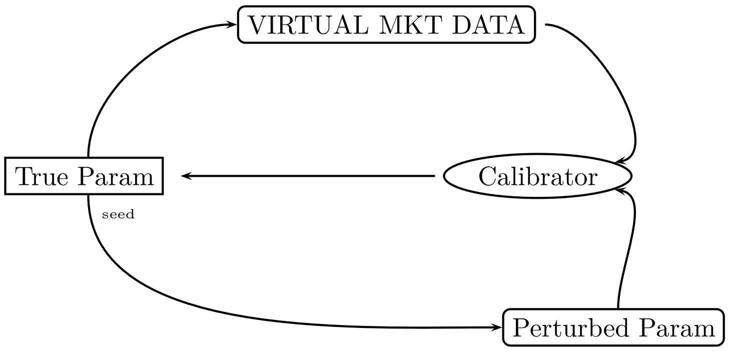


Figure 7: Virtual Test Pipeline

Virtual test exact calibration
As described above, we initiate our *true parameters*, which is the g matrix, with elements non all constants. We set a variation of 20% around 1. We generate the virtual market data then perturb this *true parameters* to a factor of 50% different. Put all those into our three calibrator, the result is satisfied.We have turned our tests many times, with different seed (in order to have different randomness of the perturbation). All three calibrators always exactly find the initial *true parameters* (error is about 1e-12%). We also analyse the time consuming of three calibrator in a particular case : horizon 10YR. As shown in the figure 8, we see the evolution of calibration for the three calibrator. Even our calibration was not optimally implemented, but we see qualitatively the good evolution of the curve. Approximatively the time comsuming of the Global calibrator is $\circ(n^2)$. The Local calibrator, run more quicker than the Global calibrator. It is difficult to compare time computing of Cascade calibrator with Local calibrator, because each time we increase the horizon, we add a diagonal to the price matrix.

Virtual test regularized calibration
Tests for regularized calibration are setted exactly with the same pipeline with exact calibration,

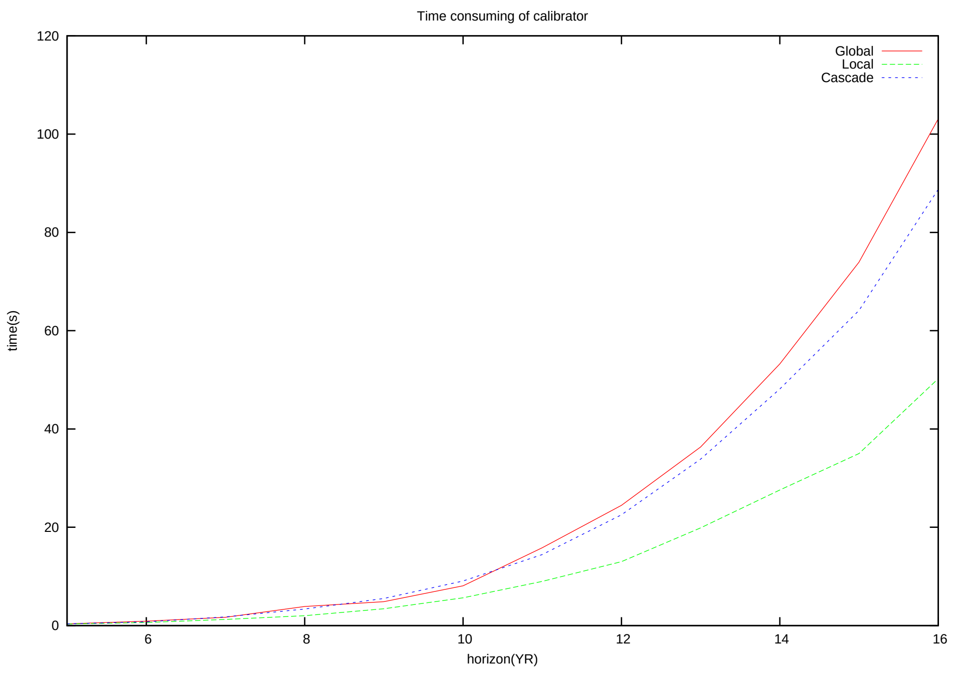
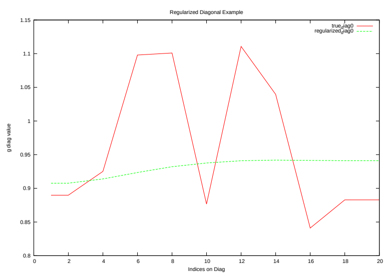


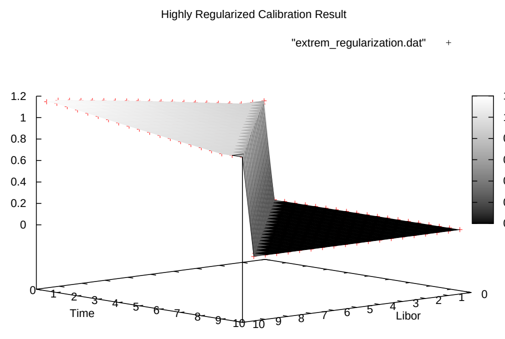
Figure 8: Calibrator time consuming comparison

except the calibrator is setted to have two penalies coefficients, and only Global calibrator can set the penalties. We've run different test, with variuous penalties coefficients and different seeds. Note that in the regularized test, we can never find exactly the initial *true parameters* because penalties are put in. When setting the penalties, calibrator intruduce error to the solution, but this later is ensured to be smooth. There are two extremun cases for the regularized calibrator. When the two penalty coefficients are setted to zero, that become the exact calibrator. When the two penalty coefficients are setted to high values, the calibrator give a highly inaccureate solution, but perfectly constant in diagonal and perfectly linear in column direction (see figure 9b). Each regularized calibration result have of course diagonal varying less than the initial *true parameter*, and the regularized diagonals has to be enter in the midle of the *true parameter* diagonal (see figure 9a).

By varying the penalty coefficients, we see how behave the errors and then can choose the best ones for the real calibration. Figure 10 we se the error variation with differents penalty coefficients setting, from zero to one. On figure 10b, we se the quotation error varies in the natural sens, i.e monotonically in both penalty coefficients. Increasing time homogeneuous or maturity penalty both increase quotation error. Figure 10b show the interesting variation of parameters's error in relation to penalties choices. Notice that parameters's error do not varies in the natural sens. For low time homogeneuous penalty and high maturity smoothness penalty, parameters's error magnitude is significant. This behavior can by understood as the logic of our implementation. The time homogeneuous penalty allow to have the good financial feature, but generate calibration error. The maturity smoothness help to control the error generated by time homogeneuous penalty. That is seen at the line when time homogeneous penalty equal to 1, the error line decrease in function of maturity smoothness penalty. When time homogeneous is small, error increase strongly with maturity penalty. That show the maturity smoothness



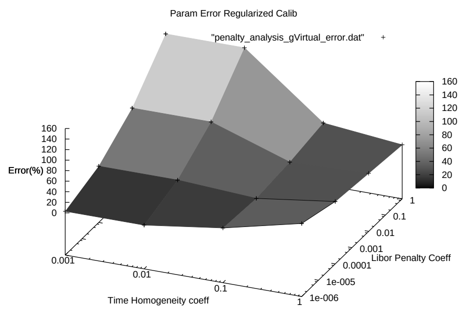
(a) Regularized diagonal result



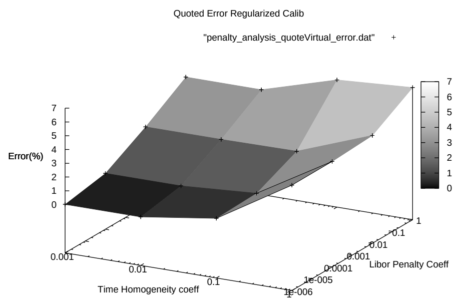
(b) Highly regularized result

Figure 9: Example of regularized calibration result

penalty dominate the time homogeneity penalty. This analyis tell us to always set two penalty together, with time homogeneous higher than maturity smoothness penalty.



(a) Error of param



(b) Error of Quotation

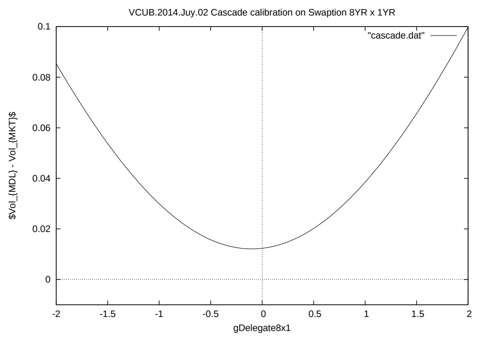
Figure 10: Regularization analysis

3.5 Numerical result

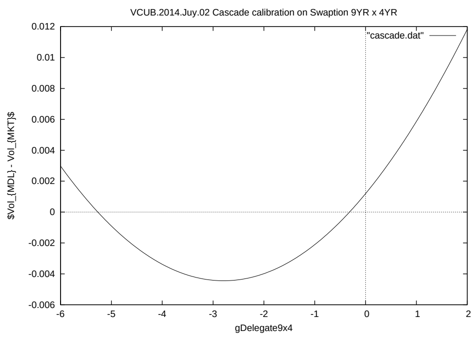
We try to test our calibrators with many data files, over 5 years (since 2008), extracted every 3 months. The objective is to be able to process the calibration with error lower than 10bp for the pratical results. However, in pratice, real data make that there aren't exist a solution for the calibration problem. We can show the the non existence of the solution, and then compare the errors performance of our three calibrators. At the end, we will show how to choose the best penalties for a regularized calibration.

Cascade Calibrator as analysis tools

Theoretically, if the real solution existe, the cascade calibrator will be able to quickly find the exact solution of the calibration problem. But when testing with the data extracted from market, even if we accept the negative volatilities parameters, we can fall in the case there are not existe a real solution. We show that by taking the VCUB of 2014.July.02, then run our cascade calibrator. We see that the first swaption that have difficulty to calibrate is the 8YRx1YR one. We stop the cascade calibrator before this swaption, and create a grid of values of volatility parameter, going from -2 to 2, and we check the variation of the cost function. As shown in figure 11a, the cost



(a) Non exact solution



(b) Negative solution

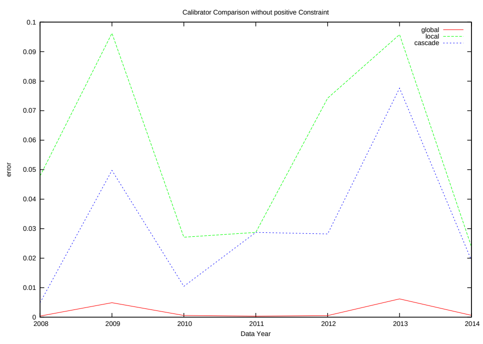
Figure 11: Cascade Calibration Analysis

function of our calibration problem is a quadratic variation in relation to volatility parameter. The fact this curve do not reach zero show that the model's Black Vol can not reach the market's Black Vol. In 11b, the minimization problem admits negatives solutions, so the cascade calibrator find the greater negative one. In these two cases, the cascade calibrator is not useful for pratical results, but it can be used as an analysis tool for seeing when, and where the solution does not go well.

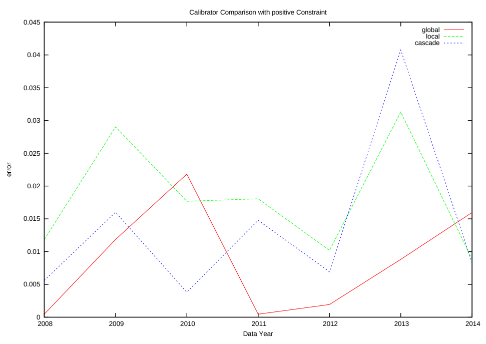
Comparison calibrators

We compare our three calibrators for two cases :

- accept the negatives parameters
- constraint positives parameters



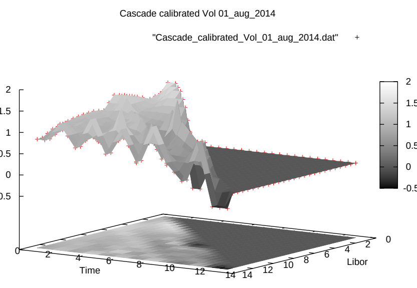
(a) Without positive constraint



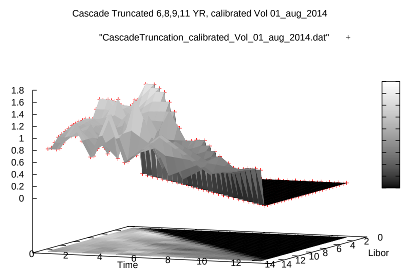
(b) With positive constraint

Figure 12: Calibrator's comparison

As shown in 12a, without positive constraint, the global calibrator has the best result as error, the cascade calibrator is slightly better than the local one. However, in case of setting the positive constraint, the cascade and local calibrator goes better, but the global calibrator degrade.



(a) Cascade Calibration, negative values found at swaptions (7,1),(12,1)



(b) Exogeneous Calibration for horizon 12YR, no negative values in gMatrix

Figure 13: Calibration 12YR with market data 01-aug-2014

References

[1] T. Adrien, *Off-cycle internship at Lunalogic*, 2013.

[2] Bloomber, *Buiding the Bloomberg Interest Rate Curve - Definition and Methodology*, 2012.

[3] D. Brigo and M. Fabio, *Interest Rate Models - Theory and pratice*, Springer New York, 2nd ed., 2006.

[4] B. Damiano and M. Massimo, *An empirically efficient analytical cascade calibration*, 2005.

[5] P. Elvis, *Learn Simple And Compound Interest*, 2014.

[6] M. Marek and R. Marek, *Martingale Methods in Financial Modelling*, Springer New York, 2nd ed., 2009.

[7] J. Peter and R. Riccardo, *Linking caplet and swaption volatilites in a BGM framework : approximate solution and empirical evidence*, Journal of Computational Finance, (2003).

[8] R. Rebonato, *Modern Pricing of Interest-Rate Derivatives : The LIBOR Market Model and Beyond*, Princeton University Press, 1st ed., 2002.

[9] R. Rebonato, K. McKay, and R. White, *The SABR/LIBOR Market Model*, Wiley, 1st ed., 2009.