Automated valuation platform for vanilla products

by Apolline Bonnerre

Submission date: 09-Sep-2019 07:38PM (UTC+0100)
Submission ID: 110614277
File name: BONNERRE_APOLLINE_01584498.pdf (1.08M)
Word count: 16951
Character count: 80092

Imperial College of Science, Technology and Medecine
London SW72AZ, United Kingdom
Department of Mathematics

Automated Valuation Platform for Vanilla Products

Apolline Bonnerre
CID number: 01584498

Thesis submitted as part of the requirements for the award of the MSc in Mathematics and Finance, Imperial College London, 2018-2019

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Signature and date:

Apolline Bonnerre,
September 10^{th}, 2019

Acknowledgements

I would firstly like to thank my thesis supervisor, Dr Mikko Pakkanen, for his patience , his support and his continous guidance throughout this project.

This work could not have been completed without Mazars London and Mazars Paris quant teams who granted me a great support and precious advice. They made me feel really welcomed in both places and it was a real pleasure to work with them. I would like to particularly thank the team I work with on the project: Antoine Collas, Pierre-Yves Couturier, Abdallah Benchenna and Tima Massaad. It was truly instructive to collaborate with them.

Finally, I would like to express my gratitude to the whole staff of the Mathematics and Finance department of Imperial College. I thank them for their teaching, their expertise and their relevant look on the financial current world they have shared with us.

Contents

List of Figures

List of Tables

Introduction

The objective of this project is to develop a fast and efficient valuation tool for vanilla products.

More precisely, the main goal is to **create a fully automated pricing platform**, focusing on interest rate and FX products. The platform should have a clear architecture, easy to understand and that enables to easily add new products in the future.

A person with no particular knowledge in quantitative finance should be able to easily use this platform. The user will just have to fill the information of the various contracts of his portfolio and to launch the computation. The platform will, without any further intervention of the user, read the contracts, collect the market data needed, handle them and price the products to finally return the value of the portfolio.

The products that will be priced in this project are firstly interest rate products: interest rate swap with every kind of legs (fixed, floating and bullet leg) and cross-currency swap. Foreign exchange forward contract will be also handled. Finally, the platform will also able to price non-linear products: swaptions in priority.

These vanilla products are frequently traded by the banks that the company is auditing, and has been chosen under this criterion. Another platform already exists for more complex products.

The work on the project has been divided in two main parts. The first one, the mathematical and financial part, is the implementation of the valuation tool. This includes the handling of the market data (bootstrapping of the curve and computation or interpolation of the various rates and volatilities) and the creation of the different pricers. Bootstrapping process is not described in this project.

The second part was the creation of the interface which will have the role of intermediary between the user and the pricers. Its main property will be the collection and the storage of the contracts' data.

The collection of the market data is made automatically using tickers from Bloomberg interface thanks to an internal software of the company that has been linked to the platform.

This new platform is coded in Python with an object-oriented architecture. We distinguish three different Python subprojects, the valuation tool, using a few functions of Quantlib library, the interface which is coded using Python Django library, and the API which is in charge of linking the two first entities. This last part is still to be implemented.

In this paper, we will focus, in the first part, on the different pricing processes of the selected vanilla products. The section one deals with the pricing of interest rate swaps in the multiple-curve framework, mainly used on the market at the current time. Every possible variation of parameters will be taken into account, from the swap's schedule consideration to the notional amortisation, so that the platform will be able to price as many kind of swap as possible. In the next chapter, FX forward pricing is studied. This is done considering spread between currencies and adapted to the data available on Bloomberg. Swaption pricing is made in the multiple-curve framework, with the swaption market model. The process will be described both with Bachelier Formula and with Black Formula. The handling of swaption volatility cubes is also explained.

Some contracts are then priced and compared to known prices got from another former pricing tool or results got via other formulas.

The second main part focuses on the implementation of the interface and of the pricing tool. The interface is a Django application, designed aiming at providing a user friendly tool to potential clients. Django application's general structure and then the interface's structure are detailed. Finally the pricing tool architecture is explained describing the main class families that have been implemented.

1 Interest Rate Products

1.1 Current concerns on the OTC derivatives valuation

1.1.1 Modern Multiple-Curve framework

Aiming at pricing interest rate products, it is necessary to model the dynamic of the yield curve term structures, not only to get the discount factors, but also to determine the evolution of the underlying floating rate exchanged in the contract.

The historical and classical method corresponds to the **single curve approach.** In this approach interbank credit and liquidity issues do not matter. The Libor rate is considered as not risky. It is the interbank rate: the rate at which one bank charges the other in case of borrowing for instance. It can be used to deduce risk-free rates: the compounding credit and liquidity risk-free rate. The basis swap spreads are negligible. This spread is defined as the liquidity premium, the rate added to the floating rate (often Libor) when the funding is made by foreign exchanged , for example borrowing in a foreign currency, or if a rate exchanged is riskier than the other. The spread is a supplement of one currency over the other, or of one rate on another due notably to their difference of liquidity and of associate risk. With this approach, in the pricing of interest rate products, the underlying rate of the product is also used as the discounting coefficient.

In the bootstrapping process, which is not detailed in this project, we use, to deduce the yield curve: $C : T \to L(t,T)$ ($L(t,T)$ being the Libor rate, several interest rate products with increasing maturities T. In the single curve framework, discount factors and underlying rate are considered to have the same index rate reference (for example GBP1M, GBP3M and then GBP6M).

However, as explained in [3] by M. Bianchetti, this approach is most of the time no longer suitable. After the financial crisis of 2008, banks can not be considered as default-free anymore, and interbank rate are not credit and liquidity risk-free anymore. Moreover, Libor rates and overnight rate, that were usually closed, started to diverge. The spread between the rates is showed Figure 1 with the example of the Euribor 3M/ Eonia rate spread. It is not stable and not negligible anymore.

The question was then about what could be used as the risk-free rate. The modern multiple-curve approach (MC-approach) is usually used now. As it is known, the longer the rate's tenor is, the riskier the rate is. As a result, in the multiple-curve approach, the funding rate of the company is now used for discounting instead of the underlying rate. Most of the time, it is the OIS rate (for instance GBPOIS, EUROIS etc.). As Alice Gianolio explains it in [4], the OIS rate is associated to a loan of one day or two maximum. Indeed, it is the rate that banks charge other institutions for overnight transactions. The probability of default is, for this really short period, very low. That is why this rate is the best proxy for a risk-free rate. EUROIS is better known as EONIA (Euro Overnight Index Average), GBP as SONIA (Sterling Overnight Index Average). OIS rate

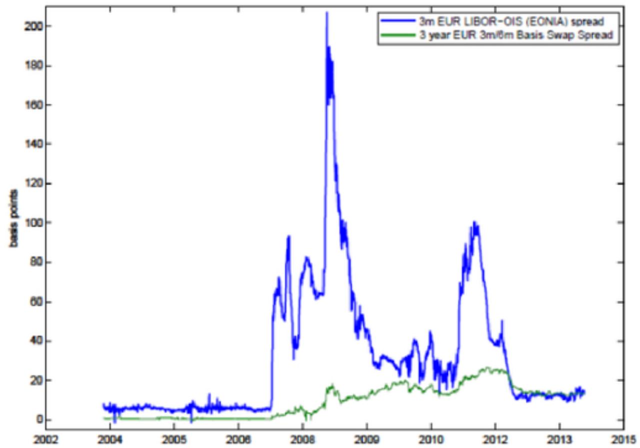


Figure 1: Evolution of the spread between European rates from 2004 to 2014 from [2]

.

is the float rate exchanged in OIS which means Overnight Indexed Swap and so is deduced from the bootstrapping of this quoted product.

In the MC-framework, a single discounting curve is built (EUROIS yield curve for example). Another curve for the underlying rate is also built using a set of market instruments: vanilla interest rate (IR) products homogeneous in the underlying rate (with increasing tenor), the bootstrapping is then made using our unique discounting curve and the bootstrapping rules of the chosen instruments.

This last approach is more complex and demands more efforts to be implemented but describes with more accuracy the current market.

For the pricing of the IR products in this project, the MC-approach will be used. The mathematical formulas are quite similar to the results got with the single-curve(SC) approach. They will be computed for MC framework and then simplified for the SC-framework.

1.1.2 Credit Risk

Another important thing to consider since the financial crisis of 2008 is the counterparty risk. This risk has considerably increased since this event. As explained by N. Choukar [5], it was originally mainly taken into account for the pricing of equity and debt, but it is now also an important element to consider for derivatives pricing.

A counterparty risk is the risk that the counterparty C of the contract defaults before the contract maturity T, as a result he can not respect its payment obligation. This causes economic losses, if at the time of default, the transaction has a positive economic value (payout).

This can be caused by a bankruptcy of C, a failure to pay, an obligation acceleration (has to pay before T, if C does not meet the terms of the contract), or a restructuring of C.

This risk is even more important since banks can not be considered as default-free anymore, and each counterparty of the contract has to protect itself against this risk. New norms (such as IFRS 13) also insist on the importance to consider it.

The valuation adjustement of the derivatives to this risk is made through CVA and DVA which are respectively the Credit Valuation Adjustment and the Debit Valuation Adjustment.

CVA is the price valuing the contract holder exposure in case of default of the counterparty. DVA expresses the risk at which the counterparty is exposed when entering the contract with the holder. Thus, CVA can be seen as a discount I get for exposing myself to the counterparty default risk, and reciprocally DVA is the discount the counterparty gets.

$$V_{\text{RISK}}(t,T) = V_{\text{RiskFree}}(t,T) - \text{CVA}(t) + \text{DVA}(t)$$

CVA depends on the probability of default of the counterparty and on the probable default time, on the net present value at default time and on the loss in case of default (which is linked to the recovery rate, the percentage of debt that is payed back). DVA depends on the reciprocal parameters.

CVA and DVA are not computed in this project but are planned to be added in a future version of the platform.

1.2 General definitions and properties

1.2.1 MC framework notations

In this project, as we adopt the MC framework, two distinct curves will be considered: the discount curve and the curve of the underlying rate. The discount curve is the continuous yield curve based on the funding risk-free rate r_f, (defined by the OIS rate as explained previously).

We will use the following notation:

- We assume a funding account with dynamic: $\begin{cases} \mathrm{d}B_f = r_f B \mathrm{d}t \\ B_f(t) = \exp\left[\int_0^t r_f(u)\mathrm{d}u\right] \end{cases}$, where f indicates the funding rate;

- Q^f the risk neutral probability associated to the numeraire B_f. Indeed, we suppose that the economy is **arbitrage-free and complete** and that (B_f,Q^f) is a numeraire pair of this economy.

- $D_f(t,T) = \frac{B_f(t)}{B_f(T)} = \exp(-\int_t^T r_f(u)\mathrm{d}u)$

- the T maturity zero-coupon bond: a product with a **payoff of one unit of currency** at time T. To express its price at time t, we take the risk-neutral expectation of the discounted payoff:

$$P_f(t,T) = \mathbb{E}_t^{Q_f}\left[D_f(t,T) * 1\right] = \mathbb{E}_t^{Q_f}\left[\exp(-\int_t^T r_f(u)\mathrm{d}u\right]$$

So the discounting curve is $C_t^f = \{T \rightarrow D_f(t,T)\}$, taking all the data from Bloomberg for the valuation time t.

1.2.2 The Forward Rate Agreement (FRA)

To define an Interest rate swap (IRS), it is interesting to firstly describe the Forward Rate Agreement (FRA) product. If we consider the general case, a FRA is a contract in which the holder pays, at a given time T (maturity), an interest rate for a period of time (time R to time T) with a fixed rate K and receives in exchange a rate payment over the same period with a floating rate $L(R,T)$. We entered the contract at a previous time T_0 ($< R$, T). The float rate is a variable rate based on a reference rate.

Financially, the main advantage of this contract is, for one of the party (who pays the fixed rate), to lock the amount to pay, whatever happens to the float rate. For example, in case of an unexpected huge increase, this contract would enable him to pay the initially planned low fix rate, and protects him against default. He can have a better vision of its future expenses.

The **the spot-Libor rate:** $L(R,T)$ is the typical underlying for interest rate products. It is mathematically defined as follows, in SC framework:

$$L_f(R,T) = \frac{1}{\tau(T,R)}(\frac{1}{P_f(R,T)} - 1)$$

Thus $L_f(R,T)$ is fixed at time R which is the reset time, and spans the time period $[R,T]$, $\tau(R,T)$ is the year fraction between time R and time T. More theoretically, the spot-Libor rate is the constant rate at which we have to invest at time R to produce an amount of one unit of currency at time T with an initial value of $P(R,T)$. It is here deduced from C^f, the tenor is $x = T - R$. In **MC framework** $L^x(t,t+x)$ is no longer a risk-free rate and so **can not be replicated by risk-free bonds** as above. We can however define a risky bond $P_x(R,R+x)$, distinct from $P_f(R,R+x)$ such as:

$$P_x(R,R+x) = \frac{1}{1 + L_x(R,R+x)\tau(R,R+x)}$$

Thus, we have three times: the valuation time t that we suppose is the current time, the reset time R and the maturity T. The FRA can be illustrated by the following scheme Figure 2 inspired from D. Brigo course [1]. The difference $\tau = T - R$ is expressed in years, it is a year fraction.

NB: This is represented as a classical case, with the payment occurring at maturity, with the same year fraction for both legs. The valuation time t is of course not fixed, we can price the product for $t > R$, but if $t > T$ the price is 0.

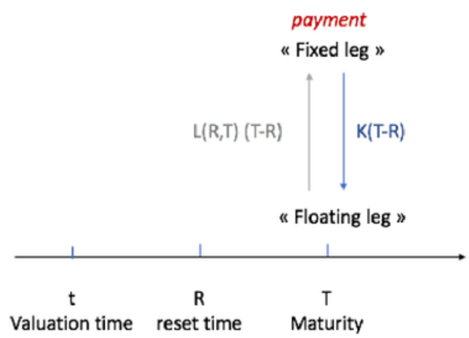


Figure 2: FRA: explanatory scheme

As a consequence, the FRA payoff at time T, with the underlying floating rate L_x is expressed as:

$$\text{FRA}_{\text{payoff}}(T) = \omega N \tau(R,T)(L_x(R,T) - K)$$

where N is the notional, K is the fixed rate, τ is the year fraction previously defined, and $\omega = +/-1$ for respectively a **payer FRA** (pay the fix rate) or the **receiver FRA.**

To compute FRA's price at time t in MC framework, the computation are not as direct as for the SC framework. We are using Theorem 1.1 stated in [3, p. 12], shown thanks to a change of numeraire.

Theorem 1.1 (Forward Measure)**.** *In the MC framework where the risk-free zero coupon bond $P_f(t,T)$ and r_f used as discounting rate, the pricing expression for the asset X holds:*

$$Price_X(t) = P_f(t,T)\mathbb{E}_t^{Q_f^T}\left[Price_X(T)\right] \tag{1.1}$$

where Q_f^T is the risk neutral probability associated with $P_f(t,T)$.

Indeed, we assume that $(P_f(t,T)$,$Q_f^T)$ is another numeraire pair of the complete and arbitrage free economy.

We get the FRA price:

$$\begin{aligned}
\text{FRA}_{\text{price}}(t) &= \mathbb{E}_t^{Q_f}\left[D_f(t,T)\text{FRA}_{\text{payoff}}(T)\right] \\
&= P_f(t,T)\mathbb{E}_t^{Q_f^T}\left[\text{FRA}_{\text{payoff}}(T)\right] \\
&= \omega N \tau(R,T)\mathbb{E}_t^{Q_f^T}\left[L_x(R,T) - K)\right] \\
&= \omega N \tau(R,T)(F_x(t,R,T) - K)
\end{aligned}$$

where $F_x(t, R, T)$ is the forward Libor rate (risky). We define it as $F_x(t, R, T) = \mathbb{E}_t^{Q_f^T}\left[L_x(R, T)\right]$. We can generalise the usual definition of the forward Libor rate with the risky bond P_x:

$$F_x(t, R, T) = \frac{1}{\tau(R, T)}\left[\frac{P_x(t, R)}{P_x(t, T)} - 1\right] \tag{1.2}$$

It can be deduced, and that will be useful in the pricing of the interest rate swap at a fixing date R, that:

$$F_x(R, R, T) = \mathbb{E}_R^{Q_f^T}\left[L_x(R, T)\right] = L_x(R, T)$$

1.3 The Interest Rate Swap (IRS)

1.3.1 Definition and Vanilla Instrument pricing formula

IRS is equivalent to several FRAs occurring along time. As a consequence, we do not consider only three times anymore, but a sequence of future dates: the current time t (the valuation time), and the future date: $[T_\alpha, T_{\alpha+1}, ..., T_\beta]$ at which resets and/or payments will occur. In the IRS the two parties agree to exchange different cash flows at predefined dates. The cash flow are called legs. We talk about *the floating leg*, describing the parameters of the floating rate payments and *the fixed leg*, describing the parameters of the fixed rate payments. In a vanilla swap rate, exchanges are made between one floating leg and one fixed leg. Most of the time they have the same frequency of payment, and the same start date and maturity. The scheme Figure 3, inspired from FRA scheme Figure 2, illustrates clearly the vanilla IRS exchange of payments:

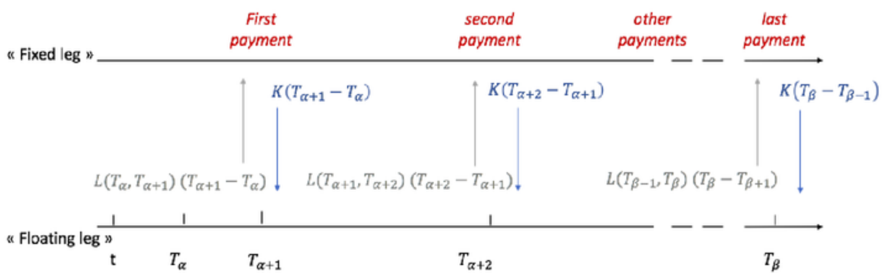


Figure 3: IRS: explanatory scheme

In this thesis, the underlying floating rate will be indexed on the Libor rate. For this vanilla swap, we can deduce a formula in the MC framework of its price at time t, with $t < T_\alpha$ which is the date at which the counterparties enter the swap. This formula is obtained by summing different FRAs corresponding to each payment of the schedule given by $[T_\alpha, T_{\alpha+1}, ..., T_\beta]$. This means summing all the swap payments and discount each cash flows to time t (it is considered here that $t < T_\alpha$).

$$\begin{aligned}\text{IRS}_{\text{price}}(t) &= \sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\mathbb{E}_t^{Q_f^{T_i}}\left[N\omega\tau_i(L_x(T_{i-1},T_i)-K)\right] \\ &= N\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i\omega(\mathbb{E}_t^{Q_f^{T_i}}\left[L_x(T_{i-1},T_i))\right]-K)\end{aligned}$$

As a result:

$$\text{IRS}_{\text{price}}(t) = N\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i\omega(F_x(t,T_{i-1},T_i)-K) \tag{1.3}$$

with the previous notations: the notional N, the fixed rate K , $\omega = 1, -1$ respectively for **payer interest rate swaps** (the holder pays the fixed leg) and for **receiver interest rate swaps** (receives the fixed leg), and τ_i is the year fraction of the period $[T_{i-1}, T_i]$. Let also remind that $Q_f^{T_i}$ is the risk-neutral probability associated with the zero-coupon bond: $P_f(t,T_i)$.

This formula can be simplified in the **SC framework**, since we can replicate the forward Libor rate with the risk-free bond P_f. We would have:

$$\begin{aligned}\text{IRS}_{\text{price}}(t) &= N\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i\omega(F(t,T_{i-1},T_i)-K) \\ &= N\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i\omega(\frac{1}{\tau_i}\left[\frac{P_f(t,T_{i-1})}{P_f(t,T_i)}-1\right]-K) \\ &= N\sum_{i=\alpha+1}^{\beta} \omega\left(P_f(t,T_{i-1})-P_f(t,T_i)-K\tau_i P_f(t,T_i)\right) \\ &= N\omega\left(P_f(t,T_\alpha)-P_f(t,T_\beta)-K\sum_{i=\alpha+1}^{\beta}\tau_i P_f(t,T_i)\right)\end{aligned}$$

In concrete terms, for a company which is funded with a floating rate it is interesting to enter a swap. Indeed, instead of paying or receiving a floating rate which fluctuates, this floating rate will be, thanks to the IR swap, converted into a fixed rate. This is really useful for the company which will not be surprised by a huge increase or decrease in the rate, it brings stability since it enables to better plan the future expenses.

1.3.2 Various parameters to consider

The previous vanilla swap was the simplest case of swaps that can be found. However, several parameters of the swap can vary. Not only can the number of legs increase, but also can the legs parameters vary. Even if the primary purpose of the project is to price vanilla products, it is important for the final platform to be flexible and to enable the user to choose as many parameters as possible for the first version or for future updates.

The Schedule of the legs In the simplest case, the dates are the same for the two legs, and the reset and payment dates correspond respectively to the beginning and to the end of the periods defined by the sequence $[T_\alpha, T_{\alpha+1}, ..., T_\beta]$. However, it is better to distinctly consider **the period dates**, which will define the year fractions τ_i, **the payment dates** (for both type of legs) at each the cash flows are paid and the reset dates (only for the floatig leg) at which the floating rate values are fixed. These three dates series are indeed distinct in most cases of IRS swap deals, and can even be different between the legs.

The **period dates** are firstly defined thanks to the frequency of the exchanged fixed in the contract, the effective date T_α of the contract at which the first exchange will be initialised and the termination date T_β of the contract at which exchanges will end.

NB: *To draw up the legs' schedule, it is crucial to define the dates according to a given calendar (London, United States, Hong Kong etc.) and to use a given business day convention: to know which date to consider if the initial day is not a working day*

The **payment dates** are based on the period dates. They can be the dates of the end of the periods (most of the time), this is called *"in arrears"* payments, but they can also correspond to the beginning of the periods, we talk about *"in advance"* payments. A *lag* can be introduced (positive or negative). It is the number of days between the date of the period (beginning or end) and the payment date.

FOR EXAMPLE: *If the period starts on Wednesday the* 3^{rd} *of July and ends on Friday* 12^{th} *of July, and the lag is of - 2 days: if the payments are in arrears, the payment will take place on Wednesday the* 10^{th} *of July, inversely if they are in advance, the payment will take place on the* 1^{st} *of July.*

The **reset dates** of the floating leg are determined exactly with the same parameters as for the payment dates. If the payment dates are in arrears, the reset dates can be in advance or in arrears. In the case of in advance payment, the reset dates are in advance too. A lag can also be added in the definition of the reset date.

Notionals and Fixed rates amortisation The notional is usually constant and identical for all the legs, nonetheless in a growing number of contracts, an amortisation of the notional value is planned. This means that the value of the notional can change from a payment to another, this variation can be linear or can also not be ruled by any specific law.

The same property can be introduced in the contract for the fixed rates or even for some parameters of the floating legs: the leverage and the spread (cf next paragraphs).

FOR EXAMPLE: *The fixed rate is initially defined at 2.5% and changes 2 years after to become*

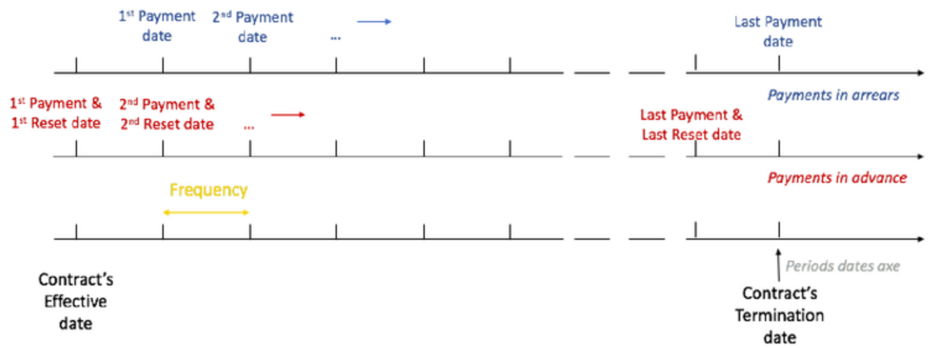


Figure 4: IRS Schedule: period and payment dates

1.5% until the end of the contract.

This will modelled in the platform implementation as *amortisation steps*: a leg's amortisation is characterised by its date and the value of each variable likely to be changed (notional etc.).

Exchange of notional The notional can be exchanged during the swap overall period: at the emission, at maturity or both. Most of the time, if the notional is payed at emission, it is payed back at maturity (end of the contract).

In these cases, these payments have to be taken into account in the swap's price, substracted at the first coupon of the leg and added at the last payment (get the money back). In the case of an amortisation of the notional at each payment of coupons, you have to add the difference between the notionals of the following period and the one of the current period.

For the coupon of the period i, in the amortised notional case: for the concerned leg (before discounting):

$$CashFlow_i = coupon_i + (N_i - N_{i+1})$$

where the $coupon_i$ is the planned "rate" payment and N_i is the notional during the period i.

Spread For the **floating leg**, the floating rate is determined from a risk-free benchmark. However, as explained in the first part (1.1), it is important to take into account an additional risk (sum to the floating rate) reflecting a difference in liquidity and in risk between two rates or two currencies (cross currency swap).

Leverage We can take into account for the floating rate an additional parameter: the leverage. A leverage multiplies the buying power in the market. In general cases, it is characterised by a use of debt (borrowed money) to amplify the return from an investment. The notional is multiplied thanks to the borrowing by a coefficient, the leverage.

1.4 Pricing of the Interest Rate Swap

We wish to include all the previous parameters in the implementation of the IRS pricer to cover as many as possible IRS contracts. The pricing of the IRS consists in **pricing each leg of the swap** (at least two but there can be more) and then summing all the obtained leg prices (discounted cash flows). It does not require any model or specific hypothesis.

The sign of the leg's price will depend on the position of the contract's holder towards the leg:

- positive: if he is short of the leg and has to pay the obtained net present value
- negative: if he is long of the leg and receives the net present value of the leg (NPV).

We have already seen that two types of legs exist in the general case: fixed leg and floating leg. We will study first these two most common cases. Then, we will describe another type of leg: the bullet leg.

1.4.1 Valuation of the Fixed Leg

The value of the fixed leg is the sum of every discounted coupon of the leg.

Once the fixed leg schedule (periods) $[T_\alpha, T_{\alpha+1}, ..., T_\beta]$ and its associated payment schedule are defined: $[T^p_\alpha, T^p_{\alpha+1}, ..., T^p_\beta]$, i_1 and i_2 are determined such that $\{T^p_{i_1}, T^p_{i_2}\} \in [T^p_\alpha, T^p_{\alpha+1}, ..., T^p_\beta]$ and $t \in [T^p_{i_1}, T^p_{i_2}]$. These two interval boundaries are such that $T^p_{i_1}$ is the biggest payment time such that $t > T^p_{i_1}$ and conversely $T^p_{i_2}$ is the smallest payment time such that $t < T^p_{i_2}$. We have the sequence $[i_1, i_2, ..., i_n]$ such that $T^p_{i_n} = T^p_\beta$.
We associate to each period $[T_i, T_{i+1}]$, the cash flow i characterised by:

- The notional N_i, itemized since it can be amortised and so different for each period (with $N_{i_{n+1}} = 0$);
- The discounting risk-free rate (deduced from the C^f yield curve) r^i_f associated to the payment date T^p_i;
- The leg's fixed rate K_i which is indexed since the fixed rate can be amortised too;

To be as general as possible, the pricing formula is implemented as follows. It gives the net present value of the fixed leg at time t.

$$NPV_{\text{fixed}}(t) = \sum_{j=i_2}^{i_n} e^{r^j_f \tau(t,T^p_j)} \left(K_j \tau_j N_j + \mathbb{1}_{NPB}(N_j - N_{j+1})\right) \tag{1.4}$$

where $\mathbb{1}_{NPB} = 1$ if the notional is to be payed back, 0 otherwise; If $i_2 = 0$, and the notional is also exchanged at the swap's emission, we add to $NPV_{\text{fixed}}(t)$: $N_{i_2} e^{r^{i_2}_f \tau(t,T^p_{i_2})}$.

Mark to Market: MtM To have an idea of the value of the product, the mark to market can also be used. Its computation is also implemented in the valuation platform for the swap's legs.

It represents a measure of the fair value of the leg, providing a realistic appraisal of its price. To get the MtM value, the **accrued interests (AC)** are subtracted to the net present value.

$$\text{MtM} = \text{NPV} - AC \tag{1.5}$$

The accrued interests are defined as interests that have been earned but their time of payment has not occurred yet. That means that the accrued interests are not zero if the valuation time is after the effective date, after the start of period but before the payment date of this period: $T_\alpha \leq T_{i_2} < t < T^p_{i_2} < T_{i_3} \leq T_\beta$.

We can compute fixed leg accrued interests (AC) as follows, with the previous notations:

$$\text{AC}_{\text{fixed}}(t) = N_{i_2}\tau(T_{i_2}, t)K_{i_2} \tag{1.6}$$

There is no discount term since it precedes the valuation date t.

1.4.2 Valuation of the Floating Leg

In a first time, we define, as for the fixed leg, the floating leg's periods schedule $[T_\alpha, T_{\alpha+1}, ..., T_\beta]$, the payments schedule $[T^p_\alpha, T^p_{\alpha+1}, ..., T^p_\beta]$. and the resets schedule $[T^R_\alpha, T^R_{\alpha+1}, ..., T^R_\beta]$.

Floating rate The floating rate for each coupon i is determined at the reset date T^R_i for the whole period $[T_i; T_{i+1}]$.

There are two distinct cases:

- The reset date is before the valuation date, which means:

$$T_\alpha \leq T_{i_2} < T^R_{i_2} < t < T^p_{i_2} < T_{i_3} \leq T_\beta$$

  In that case the floating rate is already known and can be directly found in the past market data. It is called the fixing rate. This usually concerns only the first period $[T_{i_2}; T_{i_3}]$.

- The reset date is after the valuation date: $t < T^R_i$. In this case, we need to use the bootstrapped yield curve representing the dynamic term structure of P_x of the swap underlying Libor rate L_x, and we compute the corresponding forward Libor rate F_X used in the pricing formula.

Adapting equation (1.2) for each leg schedule's period:

$$F_x(t, T^R_i, T^R_i + x) = \frac{1}{\tau(T^R_i, T^R_i + x)}\left[\frac{P_x(t, T^R_i)}{P_x(t, T^R_i + x)} - 1\right]$$

Keeping the same notations as for the fixed leg, we get the pricing formula for valuation time t:

$$\mathrm{NPV}_{\mathrm{float}}(t) = \sum_{j=i_2}^{i_n} e^{r_f^j \tau(t,T_j^p)} \left((F_x(t, T_j^R, T_j^R + x) + S_j)\tau_j N_j L_j + \mathbb{1}_{NPB}(N_j - N_{j+1}) \right) \qquad (1.7)$$

where S_i is the spread value for the period $[T_i; T_{i+1}]$ and L_i is the leverage value for the same period.

We can identically define the accrued interest for the floating leg:

$$\mathrm{AC}_{\mathrm{float}}(t) = N_{i_2} \tau(T_{i_2}, t) L_{i_2} (F_x(t, T_{i_2}^R, T_{i_2}^R + x) + S_{i_2}) \qquad (1.8)$$

$F_x(t, T_{i_2}^R, T_{i_2}^R + x)$ being replaced by the fixing rate given in the historical market date if $T_{i_2}^R < t$.

1.4.3 The Bullet Leg: Definition & Valuation

A **bullet leg** is a leg for which the counterparties agree to periodically receive or to pay (respectively long or short position), a certain amount of money. It is different from the fixed leg in the fact that we do not pay a fixed rate multiplied by a year fraction but we pay a fixed amount.

The period dates and payment dates are determined with the same method as for the fixed and floating leg.

An amortisation, modification of the amount payed or received between the periods, can be set up identically than for the notional or for the fixed rate. There is no notional exchange considered for the bullet leg.

We get the price of the bullet leg by doing some appropriate changes in the fixed leg NPV formula. We sum all the coupons i and keep the same notations as for the other legs. Let define: A_i, the amount exchanged at payment time T_i^p, without amortisation this amount is constant, equal to A.

$$\mathrm{NPV} = \sum_{j=i_2}^{i_n} e^{r_f^j \tau(t,T_j^p)} A_j$$

2 Foreign Exchange Products

Let firstly define the FX forward contract and explain how it is valuated. Then we will focus on the determination of the FX forward rate required in the pricing process. This will be done first thanks to Bloomberg estimations of the forward rate,and then thanks to its definition and the bootstrapping of cross-currency swaps.

2.1 Definitions of the the Foreign Exchanged Forward Contract (FX Forward)

A FX Forward is a contract in which the two parties agree on exchanging two amounts of currencies at a specific time in the future, at a fixed exchange rate.

This enables the counterparties to be protected against future unexpected or adverse movements in the FX rate, they avoid any currency fluctuation.

In a FX Forward contract, the two counterparties will without upfront cost agree on the amount exchanged and the date. The FX Forward is seen as the exchange of two notionals amount (not necessarily identical) at a future time.

2.2 Foreign Exchange rate

The Foreign exchange rate is decisive to determine the value of a given FX forward rate. It is the exchange rate at which two banks agree to exchange one currency for another. It is named using the name of the two currencies exchanged .

FOR EXAMPLE: *To explain more precisely, let suppose that we want to buy today one unit of pound (GBP), we give in exchange a equivalent amount in US dollars (USD). The price of one unit of pound in dollars will be the spot exchange rate GBPUSD:*

$$1£ * GBPUSD = \text{the equivalent value in \$}$$

In GBPUSD, GBP is the ***base currency****, USD the* ***quote currency*** *and GBPUSD is the amount of dollars needed to buy one unit of GBP.*

To estimate the equivalent value in pounds but at a future date, the **forward exchange rate** is used. It is the foreign exchange rate at which two banks agree today to make an exchange of currency at a future date. Forward exchange rate is defined on a non-arbitrage condition, the **covered interest rate parity**. Indeed, an investor should earn the same payoff investing in a local currency asset (with local interest rate) than investing in a foreign currency asset (with the foreign currency interest rate) and convert it then in the domestic currency at the forward exchange rate.

Indeed, the value should be the same at time t, if your exchange currency at time T and discount to time t, or if you discount from T to time t first and then exchange currency at time t.

As a result, as stated by M. Taylor in [6], the non-arbitrage condition is expressed by:

$$D_f(t,T)F(t,T) = Z_f(t,T)Spot(t) \tag{2.1}$$

with the local discount factor based on the risk free rate: D_f, the foreign discount factor Z_f, the foreign spot rate between the two currencies : $Spot(t)$ (t being valuation time) and the forward exchange rate $F(t,T)$.

For example: *the forward exchange rate for EURUSD can be expressed:*

$$F_{\text{EURUSD}}(t,T) = Spot_{\text{EURUSD}}(t)\frac{DF(€)}{DF(\$)}$$

However, this formula is not really accurate anymore: a spread between the currencies has now to be taken into account. Every bank in the world are not considered as default-free anymore and currencies are not equally risk-free and liquid. Funding market trades are now made at a premium to a globally considered risk-free currency (USD or EURO). This premium depends on the local currency. Values of spread relative to USD liquidity benchmark are given in Figure 5.

13:47 18DEC03 GARBAN-INTERCAPITAL UK04138 ICAB1
Basis Swaps - All currencies vs. 3m USD LIBOR - Also see <ICAB2>

	REC/PAY EUR	REC/PAY JPY	REC/PAY GBP	REC/PAY CHF	REC/PAY SEK
1 Yr	+3.125/+1.125	-02.00/-05.00	+02.50/-01.50	+1.50/-1.50	-3.75/-6.75
2 Yr	+3.000/+1.000	-02.00/-05.00	+02.50/-01.50	+1.00/-2.00	-2.50/-5.50
3 Yr	+3.000/+1.000	-01.75/-04.75	+02.25/-01.75	+0.25/-3.00	-1.00/-4.00
4 Yr	+3.000/+1.000	-01.75/-04.75	+02.00/-02.00	-0.75/-3.75	-0.25/-3.25
5 Yr	+2.750/+0.750	-02.00/-05.00	+01.75/-02.25	-1.25/-4.25	+0.25/-3.25
7 Yr	+2.750/+0.750	-02.50/-05.50	+00.75/-03.25	-1.50/-4.50	+0.25/-2.75
10Yr	+2.750/+0.750	-04.50/-07.50	-00.75/-04.75	-1.50/-4.50	+0.25/-2.75
15Yr	+4.125/+0.125	-10.50/-13.50	-02.25/-06.25	-0.75/-4.75	+2.50/-2.50
20Yr	+4.125/+0.125	-15.25/-18.25	-02.50/-06.50	-0.25/-4.25	+2.50/-2.50
30Yr	+4.125/+0.125	-23.25/-26.25	-02.50/-06.50	*FOR 3M V 6M EUR/EUR <ICAB4>**	
	DKK	NOK	CAD	CZK	PLN
1 Yr	-1.00/-5.00	-4.00/-8.00	+12.50/+08.50	+02.00/-07.00	+05.00/-12.00
2 Yr	-1.50/-4.50	-4.00/-8.00	+13.50/+09.50	+01.50/-06.50	+05.00/-12.00
3 Yr	-1.25/-4.25	-4.00/-8.00	+14.25/+10.25	+01.50/-06.50	+05.00/-12.00
4 Yr	-1.00/-4.00	-3.75/-7.75	+15.25/+11.25	+01.50/-06.50	+03.00/-10.00
5 Yr	-0.25/-3.25	-3.75/-7.75	+16.25/+12.25	+01.50/-06.50	+03.00/-10.00
7 Yr	+0.25/-3.00	-3.75/-7.75	+17.00/+13.00	+01.50/-06.50	+03.00/-10.00
10Yr	+0.25/-3.00	-3.75/-7.75	+17.00/+13.00	+01.50/-06.50	+03.00/-10.00
15Yr	+1.25/-3.75	-3.50/-8.50	+17.25/+13.25		
20Yr	+1.25/-3.75	-3.50/-8.50	+17.25/+13.25		

Call Brendan McVeigh or Marcus Kemp or Simon Payne on +44 (0)20 7463 4520

Figure 5: Cross-currency basis spreads relative to USD liquidity benchmark from [7]

For Example:*A cross currency basis swap of 5 years (5 Yr) with USD3M vs EUR3M is fair if a spread of +2.750 basis point is added to the EUR floating rate if USD is received.*

Taking into account this spread changes the previous formula to:

$$F_{\text{EURUSD}}(t,T) = Spot_{\text{EURUSD}}(t)\frac{DF(€\_\$)}{DF(\$)}$$

where $DF(€\_\$)$ is an adapted discount factor taking into account the currency's spread. In the case of EURUSD this spread will be negative.

The adapted discount factor curve, required when computing forward rate from the previous formula, can be computed from market quoted FX forward rates or market quoted cross-currency swaps as we will see later in this section.

2.3 Pricing of the FX Forward contract

To price a FX forward contract, one takes the difference between the discounted values of the two amounts of currencies.

In the implementation, two currencies are considered: the long currency and the short currency. If the first currency is safer than the second, the second amount is converted into the safest currency. Else, if both are exotic currencies, it is necessary to convert both in a safer currency. Indeed exchange rates between exotic currencies are sometimes not quoted in the market since the exchanges are too rare.

As a result:

$$\text{FX\_Forward}_{\text{price}}(t,T) = D_3(t,T)\left[\frac{N_1}{F_{3,1}(t,T)} - \frac{N_2}{F_{3,2}(t,T)}\right] \quad (2.2)$$

T is the contract maturity and t the valuation time. Index 1 is for long currency, and index 2 is for short currency. 3 is the index for the reference currency: the safer currency in which we convert the both cash flows. If currency 3 and 1 or 2 are the same, the forward rate is 1. $D_3(t,T)$ is the forward discount rate for time T associated to the funding risk-free rate in currency 3. N is the amount exchanged, $F_{x,y}$ is the FX forward rate with base currency x and quote currency y.

The valuation currency is here, currency 3, the multiplication of the above price by a foreign spot rate can be made to convert it in another valuation currency.

2.4 Estimation of the FX Forward rate

2.4.1 Interpolation

It is possible with the function 'FRD' on Bloomberg to collect quoted values for a given forward rate (for instance EURGBP). However, only values of FX forward rate $F(t,T)$ with small maturity T are availble, then they are deduced by extrapolation.

So, it is possible to collect this data, and then interpolate (or extrapolate) in two dimensions to deduce the required value $F_{x,y}(t,T)$, which is a point of the function $F : T \to F(t,T)$.

Linear, flat and cubic interpolation are implemented in the platform.

2.4.2 Determination of Adapted Discount Factors & use of the Forward rate definition

Another way to compute the FX forward rate is simply to compute it using its definition given in equation (2.1). To be consistent with the market, we need to estimate the adpated discount factor. The yiedcurves, with tenor OIS adapted, to another currency, reliable. This currency should be available for every base currencies, we take so the yield curve relative to the USD liquidity.

With the function ICVS on Bloomberg, we get the data presented as in Figure 1.

With the forward rate instrument, we deduce adapted discount factors by reversing its definition formula:

$$D^*_{BASE\_USD}(t,T) = \frac{F_{BASE,USD}(t,T)}{Spot_{BASE,USD}(t)} D_{USD}(t,T)$$

Maturity	Market Rate	Source
1W	0.981249	FXFWD
2W	0.980675	FXFWD
...	...	...
12M	0.9474575	FXFWD
2Y	-0009	CCY_BASIS_SWAP
..	...	...
30Y	-O.001263	CCY_BASIS_SWAP

Table 1: Bloomberg data for a given couple BASE_USD: here CHF_USD for t: 31/12/2018

where $D^*_{BASE\_USD}(t,T)$ is the discount factor adapted to USD currency.

USDOIS yield curve being already computed by bootstrapping of vanilla swaps for their IRS pricing (this is not detailed in this project), and spot rate being a known market data, we can compute the curve $C^*_{BASE_USD} : T \to D^*_{BASE\_USD}(t,T)$ for small value of T.

For bigger T, we do the same thing with cross-currency basis swaps.

A cross currency basis swap (XCCYBS) is a swap which exchanges cash flows in one currency for another. A dealer-to-dealer cross currency basis swap, which means, quoted on the market, is a vanilla swap with two floating legs in a different currency. In our case, as we chose USD benchmark, one leg is a USD leg which exchanges for example USD3M. The other leg exchanges an interbank rate of another foreign currency with tenor 3M too (EUR3M, GBP3M etc.). A corresponding spread is added to the floating rate of the non USD leg. By convention, USD leg is received and forward currency leg is payed. An exchange of notional (initially 1) is also done one at the spot value date and one at maturity.

The two legs are priced using the data already computed for the adapted curve, computed thanks to quoted FX Forward rates.

By minimisation of error with Newton-Raphson method, the adapted discount factor at maturity T_β is deduced. We get :

$$\text{XCCYBS}(t) - (\text{leg}_{\text{USD}}(t) - \text{leg}^*_{\text{Foreign}}(t, D^*_{\text{Foreign\_USD}}(t,T_\beta))Spot_{\text{Foreign\_}USD}(t)) < 1 * 10^{-10}$$

where $\text{leg}_{\text{USD}}(t)$ is the price at t of the floating leg exchanging USD3M (discounted with USDOIS), $\text{leg}^*_{\text{Foreign}}(t, D^*_{\text{Foreign\_USD}}(t,T_\beta))$ is the price at t of the foreign currency leg discounted with foreign_USD adjusted rate. This second value depends on $D^*_{\text{Foreign\_USD}}(t,T_\beta)$ adjusted that we want to determine (T_β being the maturity of cross currency swap).

3 Swaptions

So far, the products that have been studied are linear and so do not require a particular model to be priced. Let now focus on the pricing of a more complex product which is not linear: the swaption.

3.1 The derivative's properties

3.1.1 Definitions

A Swaption is a contract which gives its holder the right but not the obligation to enter an interest rate swap at a future time, called the **exercise date**, with a previously determined strike (fixed rate). As a consequence, the swaption is by definition an option on an IR swap. With a payer swaption, you have the right to enter a payer IRS (short of the fixed leg), and reciprocally with a receiver swaption.

It is often mentionned "Expiry x Tenor swaption": the **expiry** is the time between the valuation date and the exercise date, whereas the **tenor** is the time between the exercise date and the underlying swap's maturity.

Vanilla Swaptions As for the swap, the priority is to price vanilla swaptions. Thus, we consider for the priced swaption, that the underlying swap is vanilla, which means that some parameters seen in 1.4 will be simplified. As a result, the underlying swaps will be similar to the instruments used on the market to estimate swaption volatilities, and this will make easier the pricing process.

The underlying swap contract that can be entered at the exercise date:

- has only two legs: one fixed leg & one floating leg, one (indifferently fixed or floating) with a long position and the other with a short position.
- has the same emission date and maturity for the two legs, (and so the same conventions and calendar)
- has no amortisation of the fixed rate, which is so constant and no amortisation of the notional.
- has no exchange of the notional

Moreover, we consider that the exercise date is one of the reset date.

3.1.2 Payoff Formula

We have expressed in the first part the simplest expression of the price of an interest rate swap at pricing time t. The exercise time T_e of the swaption is usually the first reset date. Here, we take $T_e \in \{T_\alpha^R, T_{\alpha+1}^R, ..., T_\beta^R\}$. Indeed if $T_e > T_\alpha$, T_e can be considered as the first reset date of an equivalent swap whose schedule is the truncated initial swap's schedule.

The forward swap rate Let define the **forward swap rate** $S_{\alpha,\beta}(t)$: the value of K such that the price at t of the IRS is fair (equal to 0).

To do so, let transform the vanilla swap price formula at time T_e:

$$\begin{aligned}
\mathrm{IRS}_{\mathrm{price}}(T_e) &= N \sum_{i=\alpha+1}^{\beta} P_f(T_e,T_i)\omega\tau_i(F_x(T_e,T_{i-1},T_i)-K) \\
&= N \sum_{i=\alpha+1}^{\beta} \omega P_f(T_e,T_i)\tau_i \left(\frac{\sum_{i=\alpha+1}^{\beta} P_f(T_e,T_i)\tau_i}{\sum_{i=\alpha+1}^{\beta} P_f(T_e,T_i)\tau_i} F_x(T_e,T_{i-1},T_i)-K\right) \\
&= N\omega C_f^{\alpha,\beta}(T_e)\left(\frac{\sum_{i=\alpha+1}^{\beta} P_f(T_e,T_i)\tau_i}{C_f^{\alpha,\beta}(T_e)} F_x(T_e,T_{i-1},T_i)-K\right) \\
&= N\omega C_f^{\alpha,\beta}(T_e)\left(S_x^{\alpha,\beta}(T_e,P_f(T_e,.))-K\right)
\end{aligned}$$

Thus, in the MC framework, at time t, we define:

$$\begin{cases}
\mathrm{IRS}_{\mathrm{price}}(t) = N\omega C_f^{\alpha,\beta}(t)\left(S_x^{\alpha,\beta}(t,P_f(t,.))-K\right) \\
S_x^{\alpha,\beta}(t,P_f(t,.)) = \frac{\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i}{C_f^{\alpha,\beta}(t)} F_x(t,T_{i-1},T_i) \text{ , the Forward Swap rate} \\
C_f^{\alpha,\beta}(t) = \sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i \text{ , the Annuity}
\end{cases} \tag{3.1}$$

The forward swap rate here depends on the valuation time t, on the forward Libor rate of the underlying swap's floating rate for the different swap's periods , and on the risk-free discounting bond for the different maturity T_i: $P_f(t,T_i)$. The Annuity is going to be used as numeraire in the next section.

We can simplify the forward swap rate definition in the SC framework, indeed we computed:

$$\mathrm{IRS}_{\mathrm{price}}(t) = N\omega\left(P(t,T_\alpha)-P(t,T_\beta)-K\sum_{i=\alpha+1}^{\beta}\tau_i P(t,T_i)\right)$$

As a consequence:

$$\begin{cases}
S^{\alpha,\beta}(t) = \frac{P(t,T_\alpha)-P(t,T_\beta)}{\sum_{\alpha+1}^{\beta} P(t,T_i)\tau_i} = \frac{P(t,T_\alpha)-P(t,T_\beta)}{C^{\alpha,\beta}(t)} \\
\mathrm{IRS}_{\mathrm{price}}(T_e) = N\left(\omega(S^{\alpha,\beta}(T_e)-K)\right)\sum_{\alpha+1}^{\beta} P(T_e,T_i)\tau_i
\end{cases} \tag{3.2}$$

To get the swaption payoff, we add the optionnality on the swap price, so on the sum (and not in the sum like for a cap or a floor). The holder enters the swap if the overall value of the swap at the exercise time T_e is positive. If the swap value at T_e is negative the swaption will expire without begin exercised.

As a consequence, adding a discount term, the swaption payoff at time t is in MC framework:

$$\mathrm{Swaption}_{\mathrm{Payoff}}(t) = \epsilon N D_f(t,T_e)\left(\omega(S_f^{\alpha,\beta}(T_e,P_f(T_e,.))-K)\right)^+ C_f^{\alpha,\beta}(T_e) \tag{3.3}$$

where $\epsilon := 1/-1$, if the swaption's holder has respectively a long position or a short position for the swaption.

It is easier now to understand the hypothesis on the vanilla swap to have one long leg and one short leg. Indeed, if the two legs have a short position, the overall payoff will be in every case zero since $(-S_f^{\alpha,\beta}(T_e, P_f(T_e,.)) - K) < 0$ (the forward swap rate is usually positive). Conversely, if the two legs have a long position, the option is always exercised at time T_e, the price of the swaption is thus the price of the swap $(S_f^{\alpha,\beta}(T_e, P_f(T_e,.)) + K > 0)$

This formula can not be decomposed additively, we need as a results to study the joint action of the rates involved in the contract.

3.1.3 Price formula

The swaption's price is the risk neutral expectation of its payoff. Let express it and simplify it in the **MC framework** with a change of numeraire.

$$
\begin{aligned}
\text{Swaption}_{\text{price}}(t) &= \epsilon N \mathbb{E}_t^{Q_f}[D_f(t,T_e)(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)) - K))^+ C_f^{\alpha,\beta}(T_e)] \\
&= \epsilon N \mathbb{E}_t^{Q_f}\left[\frac{B_f(t)}{B_f(T_e)}(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)) - K))^+ C_f^{\alpha,\beta}(T_e)\right] \\
&= \epsilon N \mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[\frac{C_f^{\alpha,\beta}(t)}{C_f^{\alpha,\beta}(T_e)}(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)) - K))^+ C_f^{\alpha,\beta}(T_e)\right] \\
&= \epsilon N C_f^{\alpha,\beta}(t) \mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)) - K))^+\right]
\end{aligned}
$$

where $Q_f^{\alpha,\beta}$ is the risk neutral probability associated to the annuity numeraire $C_f^{\alpha,\beta}(t)$.

As the forward swap rate can be expressed with the annuity numeraire at the denominator in equation (3.1):

$$
S_x^{\alpha,\beta}(t, P_f(t,.)) = S_{x,f}^{\alpha,\beta}(t) = \frac{\sum_{i=\alpha+1}^{\beta} P_f(t,T_i)\tau_i}{C_f^{\alpha,\beta}(t)} F_x(t, T_{i-1}, T_i)
$$

It is a martingale under $Q_f^{\alpha,\beta}$ as stated in D. Brigo course [1]. We assume that $(C_f^{\alpha,\beta}, Q_f^{\alpha,\beta})$ is a numeraire pair of the arbitrage-free and complete economy.

This is also valid under the **SC framework** (3.2) in which the price will be expressed as:

$$
\text{Swaption}_{\text{price}}(t) = \epsilon N C_f^{\alpha,\beta}(t) \mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S^{\alpha,\beta}(T_e) - K))^+\right]
$$

3.2 Swaption Pricing Models

This swaption payoff is non linear and so a model is needed to compute its price. Indeed to get its price, the previous risk-neutral expectation of the payoff has to be computed and so supposition on the probability distribution of the forward swap rate is needed.

A famous model to price Caps and Floors is the Libor Market Model. This model can be used to price swaption but it would require some approximation (like drift freezing which is not explained in this project). A more appropriate model for the swaptions is the Swaption Market Model (SMM). It is the one that has been chosen to be used in this project.

3.2.1 Swaption Market Model: SMM

The SMM models the quantity of the forward swap rate $S^{\alpha,\beta}_{x,f}$.

The swaption market model presents several advantages compared to the modelisation of for example the short rate r.

The most important is that the quantity modeled is associated to the market payoff and is not abstract as r. Moreover, for the calibration, the implied volatility of the swap rate model can be specified and is deduced from Market swaption prices (cf subsection on the market data). The calibration of this model is more intuitive since the parameter (volatility) is meaningful and found on the market. Our pricing will fit more accurately the market data.

The historical most famous formula used in the SMM to price swaptions is **Black formula**. Another famous formula is based on **Bachelier's model** which supposes a normal distribution for the process $(S^{\alpha,\beta}_{x,f}(t))_{t>0}$. Thus, it appears that the SMM understands the swaption as a call or put option on the forward swap rate with strike K.

3.2.2 Bachelier Formula

In Bachelier's model, it is supposed that the forward swap model follows a **normal distribution**.

We suppose its dynamic is:

$$\mathrm{d}S^{\alpha,\beta}_{x,f} = \sigma^{\alpha,\beta}_{x,f}(t)\mathrm{d}W^{\alpha,\beta}_t$$

where $W^{\alpha,\beta}(t)$ is a Brownian motion under $Q^{\alpha,\beta}_f$. As $S^{\alpha,\beta}_{x,f}$ has no drift, it is a martingale under the probability measure $Q^{\alpha,\beta}_f$. Moreover, this variable follows a normal distribution, thus it can have negative values. This sometimes happens since the financial crisis of 2008.

Consequently, we have: $S^{\alpha,\beta}_{x,f}(T_e) = S^{\alpha,\beta}_{x,f}(t) + \int_t^{T_e} \sigma^{\alpha,\beta}_{x,f}(s)\mathrm{d}W_s$ where $S^{\alpha,\beta}_{x,f}(t)$ is deterministic.

We can conclude on the parameters of the normal distribution:

$$\begin{cases} \mu = \mathbb{E}\left[S^{\alpha,\beta}_{x,f}(T_e)\right] = \mathbb{E}\left[S^{\alpha,\beta}_{x,f}(t)\right] = S^{\alpha,\beta}_{x,f}(t) \\ Var\left[S^{\alpha,\beta}_{x,f}(T_e)\right] = \mathbb{E}\left[(S^{\alpha,\beta}_{x,f}(T_e) - \mu)^2\right] = \mathbb{E}\left[(\int_t^{T_e} \sigma^{\alpha,\beta}_{x,f}(s)\mathrm{d}W_s)^2\right] = \int_t^{T_e} (\sigma^{\alpha,\beta}_{x,f}(s))^2\mathrm{d}s \quad \text{(Ito's Isometry)} \end{cases}$$

Thus,

$$S^{\alpha,\beta}_{x,f} \sim \mathcal{N}\left(S^{\alpha,\beta}_{x,f}(t), \int_t^{T_e} (\sigma^{\alpha,\beta}_{x,f}(s))^2\mathrm{d}s\right) \tag{3.4}$$

Let define:

$$\begin{cases} Y := \frac{S^{\alpha,\beta}_{x,f}(T_e) - S^{\alpha,\beta}_{x,f}(t)}{v^{\alpha,\beta}_{x,f}(T_e)\sqrt{T_e - t}} \sim \mathcal{N}(0,1) \\ (v^{\alpha,\beta}_{x,f}(T_e))^2 := \frac{1}{T_e - t}\int_t^{T_e} (\sigma^{\alpha,\beta}_{x,f}(s))^2\mathrm{d}s \end{cases}$$

In the case where $\sigma^{\alpha,\beta}_{x,f}(t)$ is constant, we have: $v^{\alpha,\beta}_{x,f}(T_e) = \sigma^{\alpha,\beta}_{x,f}$.

Then, if we use the distribution hypothesis in the swaption's price formula, we get for the computation of the expectation:

$$\mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)-K))^+\right]=\mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_{x,f}^{\alpha,\beta}(t)+Yv_x^{\alpha,\beta}(T_e)\sqrt{T_e-t}-K))^+\right]$$
$$=\int_{-\infty}^{+\infty}(\omega(S_{x,f}^{\alpha,\beta}(t)+yv_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}-K))^+\phi(y)\mathrm{d}y$$

where: $\phi : x \to \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ is the normal probability density function.

It is clear that:

$$\omega(S_{x,f}^{\alpha,\beta}(t)+yv_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}-K)>0$$
$$\longleftrightarrow \frac{K-S_{x,f}^{\alpha,\beta}(t)}{v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}}<y \qquad \text{if w = 1}$$
$$\frac{K-S_{x,f}^{\alpha,\beta}(t)}{v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}}>y \qquad \text{if w = -1}$$
$$\longleftrightarrow \omega d > -\omega y$$

Let define $d := \frac{S_{x,f}^{\alpha,\beta}(t)-K}{v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}}$ and $\Phi(x) : x \to \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2}\mathrm{d}y$, the normal cumulative distribution function.

As a result, by symmetry of $\phi(x)$:

$$\mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)-K))^+\right]=\int_{-\infty}^{\omega d}\left(\omega(S_{x,f}^{\alpha,\beta}(t)-K)+uv_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}\right)\phi(u)\mathrm{d}u$$
$$=\omega(S_{x,f}^{\alpha,\beta}(t)-K)\Phi(\omega d)+v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}\int_{-\infty}^{\omega d}y\phi(y)\mathrm{d}y$$
$$=\omega(S_{x,f}^{\alpha,\beta}(t)-K)\Phi(\omega d)+v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}\phi(d)$$

Indeed:

$$\int_{-\infty}^{\omega d}y\phi(y)\mathrm{d}y=\frac{1}{\sqrt{2\pi}}\left[e^{-\frac{1}{2}y^2}\right]_{-\infty}^{\omega d}=\phi(\omega d)=\phi(d)\text{ by symmetry of }\phi$$

As a result, with Bachelier's model, the swaption's price is express as follows:

$$\text{Swaption}_{\text{NormalPrice}}(t)=\epsilon NC_f^{\alpha,\beta}(t)(S_{x,f}^{\alpha,\beta}(t)-K)\omega\Phi(\omega d)+v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}\phi(d) \tag{3.5}$$

3.2.3 Black Formula

The second model, really famous is Black model. It supposes that the forward swap rate follows a log normal distribution.

It is assumed that the forward swap rate dynamic is:

$$\mathrm{d}S_{x,f}^{\alpha,\beta} = \sigma_{x,f}^{\alpha,\beta}(t) S_{x,f}^{\alpha,\beta}(t) \mathrm{d}W_t^{\alpha,\beta}$$

where $W_t^{\alpha,\beta}$ is a Brownian motion under $Q_f^{\alpha,\beta}$. As $S_{x,f}^{\alpha,\beta}$ has no drift in this case too, it is a martingale under $Q_f^{\alpha,\beta}$. As this variable follows a log normal distribution, it only has positive values, which can cause some problems at the present time since the forward swap rate can be negative (cf Black shifted model subsection 3.2.4).

We can approximate: $\sigma_{x,f,\text{normal}}^{\alpha,\beta}(t) \sim \sigma_{x,f,\text{Black}}^{\alpha,\beta}(t) S_{x,f}^{\alpha,\beta}(t)$. As $S_{x,f}^{\alpha,\beta}(t) < 1$, usually $\sigma_{x,f,\text{normal}}^{\alpha,\beta}(t) < \sigma_{x,f,\text{Black}}^{\alpha,\beta}(t)$. As a consequence, normal implied volatilities are usually available in bps on Bloomberg and Black implied volatilities are expressed in percentage.

To express the swaption's price, we need again to express the risk neutral expectation. Let find an expression for $S_{x,f}^{\alpha,\beta}$ and its distribution parameters.

By definition of the log-normal distribution, we can find $X(T_e)$ which follows a normal distribution with mean μ and variance σ^2 such that in distribution $X(T_e) = \ln(S_{x,f}^{\alpha,\beta}(T_e))$, if $S_{x,f}^{\alpha,\beta}(T_e)$ follows a log-normal distribution of mean μ and of variance σ^2.

By Ito's formula:

$$\begin{aligned} \mathrm{d}\ln S_{x,f}^{\alpha,\beta} &= \frac{1}{S_{x,f}^{\alpha,\beta}} \mathrm{d}S_{x,f}^{\alpha,\beta} - \frac{1}{2} \frac{1}{(S_{x,f}^{\alpha,\beta})^2} \mathrm{d}S_{x,f}^{\alpha,\beta} \mathrm{d}S_{x,f}^{\alpha,\beta} \\ &= \sigma_{x,f}^{\alpha,\beta}(t) \mathrm{d}W_t^{\alpha,\beta} - \frac{1}{2} (\sigma_{x,f}^{\alpha,\beta}(t))^2 \mathrm{d}t \end{aligned}$$

By integration:

$$\ln \frac{S_{x,f}^{\alpha,\beta}(T_e)}{S_{x,f}^{\alpha,\beta}(t)} = \int_t^{T_e} \sigma_{x,f}^{\alpha,\beta}(s) \mathrm{d}W_s^{\alpha,\beta} - \frac{1}{2} \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s$$

As a result:

$$S_{x,f}^{\alpha,\beta}(T_e) = S_{x,f}^{\alpha,\beta}(t) \exp \left[\int_t^{T_e} \sigma_{x,f}^{\alpha,\beta}(s) \mathrm{d}W_s^{\alpha,\beta} - \frac{1}{2} \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s \right]$$

As in the previous par we can compute the expectation and the variance (using It's isometry) of the normal variable: $X(T_e) = \int_t^{T_e} \sigma_{x,f}^{\alpha,\beta}(s) \mathrm{d}W_s^{\alpha,\beta} - \frac{1}{2} \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s$

We can conclude that:

$$S_{x,f}^{\alpha,\beta}(T_e) \sim \mathcal{LN} \left(-\frac{1}{2} \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s, \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s \right) \tag{3.6}$$

Let define :

$$\begin{cases} Y := \left(\ln \frac{S_{x,f}^{\alpha,\beta}(T_e)}{S_{x,f}^{\alpha,\beta}(t)} + \frac{1}{2} (v_x^{\alpha,\beta}(T_e))^2 (T_e - t) \right) \frac{1}{v_{x,f}^{\alpha,\beta}(T_e) \sqrt{T_e - t}} \sim \mathcal{N}(0,1) \\ (v_{x,f}^{\alpha,\beta}(T_e))^2 := \frac{1}{T_e - t} \int_t^{T_e} (\sigma_{x,f}^{\alpha,\beta}(s))^2 \mathrm{d}s \end{cases}$$

It is now possible to express the swaption's price, keeping the same notation that for 4.2.2.

$$\mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)-K))^+\right] = \mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_{x,f}^{\alpha,\beta}(t)e^{\left(Yv_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}-\frac{1}{2}((v_x^{\alpha,\beta}(T_e))^2(T_e-t)\right)}-K)^+\right]$$
$$= \int_{-\infty}^{+\infty}(\omega(S_{x,f}^{\alpha,\beta}(t)e^{\left(Yv_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}-\frac{1}{2}((v_{x,f}^{\alpha,\beta}(T_e))^2(T_e-t)\right)}-K))^+\phi(y)\mathrm{d}y$$

It is clear that, as previously:

$$\omega(S_{x,f}^{\alpha,\beta}(t)e^{\left(Yv_x^{\alpha,\beta}(T_e)\sqrt{T_e-t}-\frac{1}{2}((v_{x,f}^{\alpha,\beta}(T_e))^2(T_e-t)\right)}-K)>0$$
$$\longleftrightarrow -\omega y<\omega d_-$$

where

$$d_- := \frac{\ln\frac{S_{x,f}^{\alpha,\beta}(t)}{K}-\mu}{V}, \mu := -\frac{1}{2}((v_{x,f}^{\alpha,\beta}(T_e))^2(T_e-t), V := v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}$$

$$\mathbb{E}_t^{Q_f^{\alpha,\beta}}\left[(\omega(S_x^{\alpha,\beta}(T_e,P_f(T_e,.)-K))^+\right] = \int_{-\infty}^{\omega d_1}(\omega(S_{x,f}^{\alpha,\beta}(t)\exp\left(-\omega v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}u+\mu\right)-K))\phi(u)\mathrm{d}u$$
$$= \omega\left(S_{x,f}^{\alpha,\beta}(t)\int_{-\infty}^{\omega d_1}\exp\left(-\omega v_{x,f}^{\alpha,\beta}(T_e)\sqrt{T_e-t}y+\mu\right)\phi(y)\mathrm{d}y - K\Phi(\omega d_-)\right)$$
$$= \omega\left(S_{x,f}^{\alpha,\beta}(t)e^{\frac{1}{2}V^2+\mu}\int_{-\infty}^{\omega d_1}e^{-\frac{1}{2}(\omega(y+\omega V))^2}\phi(y)\mathrm{d}y - K\Phi(\omega d_-)\right)$$
$$= \omega\left(S_{x,f}^{\alpha,\beta}(t)\Phi(\omega(d_-+V)) - K\Phi(\omega d_-)\right)$$

As a result, using Black formula, the swaption's price is expressed as follows:

$$\text{Swaption}_{\text{BlackPrice}}(t) = \epsilon NC_f^{\alpha,\beta}(t)\omega\left(S_{x,f}^{\alpha,\beta}(t)\Phi(\omega d_+) - K\Phi(\omega d_-)\right) \tag{3.7}$$

where $d_+ = d_- + V$

3.2.4 Shifted Black Formula

At current time, forward interest rate can be negative resulting in missing implied volatility in the collected Bloomberg surfaces. Indeed in the case of a Log Normal distribution for the forward swap rate we have a relation of proportionality between the implied volatility and the rate: $\sigma_{x,f}^{\alpha,\beta}(t) = \frac{\text{Volatility}}{S_{x,f}^{\alpha,\beta}(t)}$. If $S_{x,f}^{\alpha,\beta}(t) < 0$, as volatility is positive we would have the implied volatility: $\sigma_x^{\alpha,\beta}(t) = 0$ or $\to -\infty$ which is not possible. To avoid this problem, we can consider a shifted Black model, let h be the shift value, the dynamic of the forward swap rate is in that case modelled by:

$$\mathrm{d}S_{x,f}^{\alpha,\beta} = \sigma_{x,f}^{\alpha,\beta}(t)(S_{x,f}^{\alpha,\beta}(t)+h)\mathrm{d}W_t^{\alpha,\beta}$$

Thus the price becomes:

$$\text{Swaption}_{\text{BlackShiftedPrice}}(t) = \epsilon NC_f^{\alpha,\beta}(t)\omega\left((S_{x,f}^{\alpha,\beta}(t+h)\Phi(\omega d_+) - (K+h)\Phi(\omega d_-)\right) \tag{3.8}$$

3.3 Required Market Data: Swaption Volatility Cube

In each of the previous models, the formula uses the implied volatility as a required parameter to get the price of our derivative. The implied volatilities are market data that can be found on Bloomberg.

3.3.1 Data Description

As for the swaptions, the swaption volatilities are functions of the expiry, the tenor and the strike. Thus, we have chosen to create volatility cubes.

On Bloomberg, swaption implied volatility surfaces can be found for given strikes (function VCUB) (absolute strike or relative stike). These volatilies are computed from swaptions priced at time t on the market with either Black formula or Bachelier formula (also called normal model). Their underlying swap are vanilla swaps for which we can choose the tenor of the floating rate and the discount tenor (either OIS for multicurve or the same tenor as the floating rate for single curve). This vanilla swaps are similar to the underlying swaps of the swaptions that are planned to be priced by the platform. As a result, a stochastic model for the volatilies is not necessary in our case given that the implied volatilities available on the market are good approximation of the volatility of our products.

Thus, we have the equality (supposing of course that the swaptions on both sides of the equation have the same underlying swap and same pricing parameters):

$$\text{Swaption}_{\text{MarketPrice}}(t, T_e, T_\beta, X) = \text{Swaption}_{\text{BlackPrice}}(\sigma_{Black})$$

$$\text{Swaption}_{\text{MarketPrice}}(t, T_e, T_\beta, X) = \text{Swaption}_{\text{NormalPrice}}(\sigma_{normal})$$

Where T_β is the maturity of the underlying swap. The right hand side of the equations is the price computed with the respective formulas.

From these equalities, we could deduce the volatility. For our valuation platform, as the market data of volatilities are already available, we just need to collect them automatically like the other previous data.

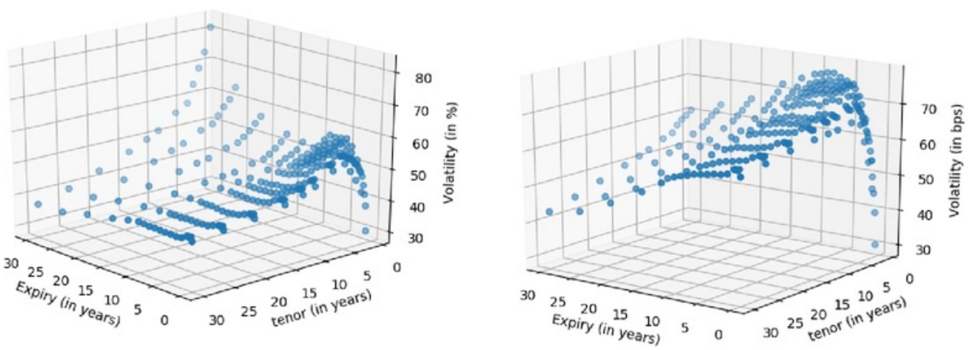
For different relative strikes: $[ATM\pm200bps, ATM\pm100bps, ATM\pm50bps, ATM\pm25bps, ATM]$ (ATM means at the money), the surfaces are picked up and written in a csv file, as in Table 2 where the last columns are the tenors.

The file is read by the platform and then a dataframe is created with two levels of indexes to represent the volatility cube. The year fraction are computed taking into account the index parameters. We devide volatility value by 100 or by 10000 respectively for Black and normal implied volatility.

We can plot swaption volatilty surfaces for each given value of strike respectively for normal and Black model. The plot of GBP3M volatilities at the money is in Figure 6.

Strike	Expiry	1Yr	2Yr	3Yr	4Yr	5Yr	...	30Yr
-2	1 Mo	[values]	...					
-2	3 Mo	[values]	...					
...	...	...						
0	1 Mo	[values]	...					
...	...	...						
2	1 Mo	[values]	...					

Table 2: Market data csv files for swaptions volatility



Black implied volatility

Bachelier implied volatility

Figure 6: Swaption volatility surfaces, at the money for GBP3M

As we can see for low expiry year fractions (in the short term), volatility is quite high, it even reaches record high exceeding 80% for the short tenor (few months) for Black implied volatility. It can also be noticed that globally, swaptions with an underlying swap with a short tenor, whatever its expiry is, has a swaption volatility rather high. This is due to the fact that the positivity of the swap depends only on the value of the floating rate on a short period. The positivity is so difficult to assure as the floating rate can vary a lot, and as a conclusion, the swaption's price varies a lot too.

3.3.2 3D Interpolation

Either the required volatility matches with a dataframe pillar, or it is not in the given data and we have to interpolate to estimate its value. Relative strike are used in this example which means that we take volatility for the strike:

$$\text{Strike} = K - S_{x,f}^{\alpha,\beta}(t)$$

where K is the fixed rate.

Different interpolation methods are possible in 3D dimension. Cubic, linear and flat are implemented.

4 Results

4.1 Pricing of the Swaps

To test the swap pricing, different swap contracts are used, firstly vanilla swap contracts and then more complex swap contracts (with notably a notional amortisation). The data are then collected for the pricing date: 31/12/2017 and the products are priced in GBP in the MC-framework (cash flows discounted with GBPOIS). To deduce the rates we use a logCubic interpolation method on the discount factors of the bootstrapped curves.

Let describe the tested contracts of simple swaps in the Table 3. Only few coupons are paid. This test is done to check globally if the schedules are correctly established and if the pricing is efficient. The Target calendar (usually employed in the euro zone) is used and the business day convention is the "modified following" convention. The day count fraction is "Act/365".

NB: *the modified following convention follows the rule under which if the transaction day is not a banking day, the next banking day is taken, but if the month changes, the preceding day of the transaction date is taken.*

Swap ID	SWAP1	SWAP2	SWAP3
Effective start date	29/09/2017	29/09/2018	26/10/2017
Termination date	29/03/2018	29/03/2019	26/01/2018
Index Rate	GBP6M	GBP6M	GBP3M
Notional	45176194,04	43 221 350.55	11288331.00
Fixed Rate	0.0381	0.0381	0.0513
Fixed & Floating Leg Frequency	6M	6M	3M
Position Fixed leg	Payer	Receiver	Payer

Table 3: Vanilla swap parameters

The main difference between first two swaps is on the effective date, SWAP1 starts before the valuation date whereas SWAP2 starts after this date. Fixing rates (historically quoted) will be used for SWAP1, and only forward Libor rate will be used in the valuation of SWAP2. SWAP3 has notably a different frequency.

The prices computed by the valuation platform are compared to prices computed by a former reliable valuation platform used by the company in Table 4.

Swap ID	SWAP1	SWAP2	SWAP3
Valuation Platform price	- 742 723	644 607	- 133901
Former Tool price	- 742 557	- 644 184	- 133900
Comparison in bps	0.0367	0.0784	0.0011

Table 4: Truncated (unit) prices of the vanilla swaps, estimated on the 31/12/2017 in GBP

The difference in bps is computed with the following formula:

$$\text{Diff}_{\text{bps}} = \frac{\mid (\text{theoretic price} - \text{computed price}) \mid}{N\tau(t, T_\beta)} * 10000 \tag{4.1}$$

where N is the notional of the product and T_β its maturity. $\tau(t, T_\beta)$ is the year fraction between t and T_β. It is used to compare the prices of a product, for example the price an audit company computes and the price the audited bank gives.

For the pricing of the vanilla swaps, the relative difference in bps between the two prices is about 10^{-2} bps so about 10^{-6}. The objective given was about 1bps. So these results can be considered as satisfactory. Besides, the schedules are identical with the two pricers.

We can then try to price more complex contracts with amortisation and exchange of notionals. The priced swaps are described in the Table 5 and Table 6. The convention used are the same as for the simple swaps and are the same for both legs.

The swap whose ID is SWAP4 has a linear amortisation for the notional values and an arbitrary amortisation for the fixed rate values. The values are plotted in Figure 7.

As previously, the results are compared to the values got with the former valuation tool. The notional used in the equation (4.1) is the initial value of the floating leg's notional. The times of computation must also be checked for these more complex contracts.

The results are printed in the Table 7. The difference in bps is really low for all the legs: inferior to 0.3 bps at maximum. Moreover, the computation time for each swap is about half a second which is rather fast.

4.2 Pricing of the FXForwards

As it has been seen in the FX product pricing section, there is a difference between adapted discount factors (to USD liquidity) and the usual discount factors got from the corresponding quote currency OIS rate curve. Let see an example with EUROIS and EUR_USD curves. We compute the discount factors for regular dates (maturity from 1 year to 10 years), the results are displayed in Table 8.

Let price a FX forward contract in **EUR** on the $t = 31/12/2018$. This contract exchanges **EUR for CHF** (Swiss franc). The calendar is the 'Target calendar' and the business day convention is

SWAP3			
Start date		20/07/2016	
End date		24/07/2020	
Notional (initial value)		50000000	
Fixed leg		Floating Leg	
Frequency	6 Months	Frequency	6 Months
Position	long	Position	short
Notional Exchange	None	Not. Exchange	None
Fixed rate	0.06	Index	GBP6M
		Reset Lag	-2 Days
		Spread	0.01
		Leverage	1
Notional Amortisation			
Step dates		Notional Value	
Step date		40 000 000	
Step date		30 000 000	

Table 5: Parameters of SWAP3

SWAP4			
Start date		26/01/2004	
End date		26/01/2019	
Notional (initial value)		11288331	
Stub		Long end	
Fixed leg		Floating Leg	
Frequency	6 Months	Frequency	3 Months
Position	short	Position	long
Notional Exchange	Both	Not. Exchange	Both
Fixed rate	0.0513	Index	GBP3M
		Reset Lag	0 Days
		Spread	0
		Leverage	1

Table 6: Parameters of SWAP4

still 'modified following'. The day count fraction is "Act/365".

We get the **notional amount** of 1141000 units of CHF in exchange of giving 1000000 units of EUR currency.

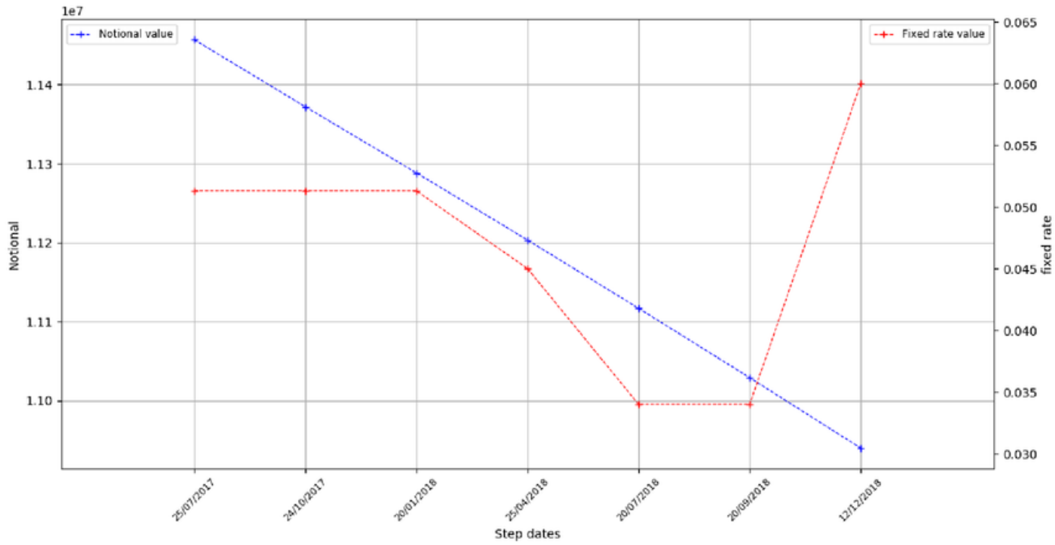


Figure 7: SWAP4's fixed rate and notional linear amortisation

Swap ID	SWAP3		SWAP4	
Legs	Fixed Leg	Floating Leg	Fixed Leg	Floating Leg
Valuation Platform price	7152686	- 2070523	- 11699104	11303014
Former Tool price	7152823	- 2071329	- 11699347	11302298
Comparison in bps	0.0106	0.0356	0.2002	0.5927
Computation time average (whole swap)	0.47s		0.58s	

Table 7: Truncated (unit) prices of the complex swaps, estimated on the 31/12/2017 in GBP

We will take several maturity dates and compute the forward rate that we get with the valuation platform and compare it to the one computed by the "calculator Forward FX" on Bloomberg.

Then we will compare the price computed with the valuation platform to the one we would have with the FX forward rate of Bloomberg (BBG) with formula described equation (4.1) (with the notional in euro and for T_β value the maturity of the forward contract. We do not focus here on the discount factor values that we consider correct.

The results are detailed in the Table 9.

As it is displayed, the difference between the interpolated Bloomberg forward rates and the the forward rates computed by the pricing tool is really low. The impact on the price of the contract is ,using the relative comparison in bps, of order of magnitude 1 bps for short maturity to less than 1 bps when the maturity increases. We can so consider that the price returned by the valuation

Maturity (year)	Discount Factor EUROIS	Discount Factor EUR_USD
1	1.00356	1.00657
2	1.00618	1.01256
3	1.00643	1.01536
4	1.00333	1.01562
5	0.99747	1.01276
6	0.98896	1.00682
7	0.97794	0.99931
8	0.96488	0.98833
9	0.95072	0.97678
10	0.93510	0.96411

Table 8: Discount factors deduced from EUROIS and EUR_USD curves (round number to 10^{-5})

	$F_{EURCHF}(t,T)$		Contract Price		
Maturity T	Platform	BBG	Platform	BBG	Comparison (bps)
29/03/2021	1.1154	1.1158	23057.1418	22700.6952	1.5668
29/03/2022	1.1102	1.1106	27870.16729	27481.0844	1.1830
29/03/2023	1.1036	1.1036	33921.9504	33933.7759	0.0275
29/03/2024	1.0957	1.0953	41172.1525	41467.3858	0.5553

Table 9: FX Forward rate $F_{EURCHF}(t,T)$ for different maturity date T (round numbers)

platform is relatively correct.

4.3 Pricing of the Swaptions

To present the testing of the swaptions pricing, the contract of one swaption: SWAPN1 is priced with Bachelier model and then with Black model. The results are compared, as for the swaps, with the computed results from another pricing tool. We will price it at a given strike and then at the money.

The pricing is done with the "Target" calendar and the "modified following" business day convention. The day count fraction is "Act/365". It is priced in GBP at the valuation date t: 31/12/2018. The contract of the swaption is described in the Table 10. The Bloomberg data are collected with relative strikes.

The results got with Bachelier and Black model are presented in the table 11. The difference in bps is computed with the same formula as the equation (4.1). The notional is the one used for both legs, the maturity T_β is, in the swaption case, the exercise date of the swaption T_e.

The results are rather good for both Bachelier and Black model, if we compare them to the

SWAPN1			
Exercise date	20/07/2020	Swap Maturity	24/07/2025
Swaption type	Receiver	Position	Long
Notional	50000000	Discount Index	GBPOIS
Fixed Leg Frequency	3M	Floating Leg Frequency	3M
Strike (ATM + 0.01)	0,02335	Floating leg Index	GBP3M
		Reset lag	0 day

Table 10: Parameters of the contract of the swaption: SWAPN1

SWAPN1	Bachelier Model	Black Model	Comparison of the models (10^{-2} bps)
Valuation Platform price	2592509	2592715	2,640
Former tool price	2592046	2592019	0.348
Comparison in bps (10^{-2})	5,973	8.961	

Table 11: Truncated price of SWAPN1 at 31/12/2018 in GBP with Bachelier & Black models

results of the former pricing tool. The comparison is about 10^{-2} which is lower than what we wanted (less than 1 bps). The difference between the two tools are due to the interpolation since the values are computed differently and due to the discount factor and forward rate computation.

The price for Bachelier and Black are also quite similar too (about 10^{-2}) which is an additionnal confirmation that the prices are correct.

The pricing of this swaption takes on average 0.60 seconds. This is quite short especially if we consider that some of this time is used to bootstrap the curves. Pricing a portfolio of swaptions will be so quite fast.

Let look at the results at the money. This means, as a cube of volatility with relative strikes is used, that we need: $K = S_{x,f}^{\alpha,\beta}(t)$. Thus, we take $K = 0,013347164$.

It is generally known that the price of a call option at the money can be approximated by:

$$Call_{price}(S,T) = 0.4S\sigma\sqrt{T}$$

, where S is the stock price and T the maturity.

Comparing the call option price Black-Scholes formula and Black formula for the swaption, we can adapt it, we get the **ATM formulas**:

$$\begin{cases} \text{Swaption}_{\text{BlackPrice}}(t) = 0.4NS_{x,f}^{\alpha,\beta}(t)C_f^{\alpha,\beta}(t)\sigma_{x,f,\text{Black}}^{\alpha,\beta}(t)\sqrt{T_e} & \text{with Black model} \\ \text{Swaption}_{\text{NormalPrice}}(t) = 0.4NC_f^{\alpha,\beta}(t)\sigma_{x,f,\text{normal}}^{\alpha,\beta}(t)\sqrt{T_e} & \text{with Bachelier model} \end{cases}$$

Let's check this equality for the two models in Table 12.

As it is shown in the table, the price computed with the valuation platform are quite close to the results got with the ATM formula above. The lightly bigger number of bps with an order of

SWAPN1	Bachelier Model	Black Model	Comparison models (in bps)
Valuation Platform price	857672	858310	0,08218
ATM Formula	868015	888057	2.58
Comparison in bps	3.829	1.30	

Table 12: Truncated prices of SWAPN1, ATM on the 31/12/2018 in GBP with Bachelier & Black models

magnitude of the unit, can be understood by the fact that the formula gives an approximation, and is not a closed formula of the price. Both models have a similar result for the ATM price of SWAPN1 with a relative difference about 8.10^{-2} bps.

5 Architecture and Implementation of the Tool

5.1 Architecture and Libraries

As presented in the introduction, the valuation platform is entirely coded in Python, with an object-architecture.

It is composed of two main entities: the interface and the pricing tool, they will be linked thanks to a third application which will connect the user request to the pricing tool functions. Data are transferred via XML files (see section 6.3).

The global architecture of the platform is illustrated in the Figure 8.

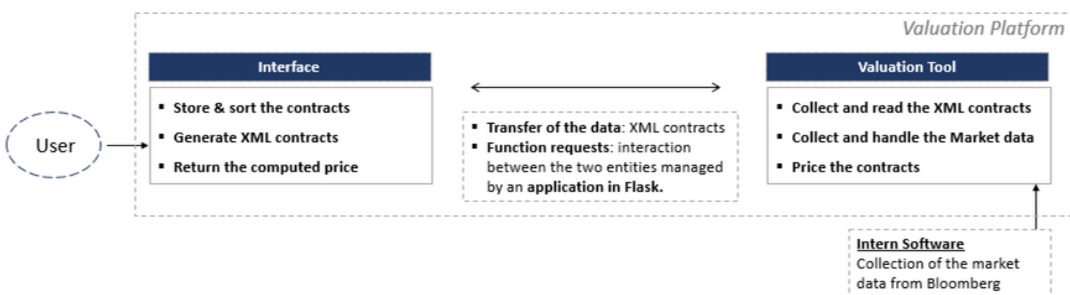


Figure 8: Global architecture of the platform

Three main libraries are used :

- Quantlib for the pricing tool,
- Django for the interface,
- Flask for the API which is in charged of linking the two first entities.

The figure explains how the three parts are linked. In parallel, the pricing tool has been connected to an intern software that collects the market data thanks to Bloomberg's tickers directly on Bloomberg platform. This part will not be explained in this project.

Commercialisation of the tool This tool is planned to be sold to potential clients. As a consequence, the code has to be cleared and documented. Moreover every change in the code has to be checked so that other parts of the implementation are not affected by wrong code addition.

Some Python libraries are used to reach that goal:

- **Flake8**, this library enables to control the code syntax, for example, it notices the forgetting of spaces, the too many characters etc.
- **Sphinx** generates automatic documentation on the different functions and classes. It is just required to complete for each function the arguments, the output and the goal.
- **Pytest**: Unitests are implemented for each part of the code: Pricers, curves' bootstrapping, interpolations etc., they have their own data set distinct from the platforms contracts. They check if functions previously correctly implemented have not been changed by mistake.
- **Logging** is also used, as a result at each contract pricing, chosen intermediary results are printed in a log file (updated at each running), it is really useful during the development period to debug the code.

The project is loaded on a git. The git plays the role of a version control. Each time the project is committed aiming at pushing it on the git, all the unitests are launched and the syntax inspection is done by flake8. This is implemented thanks to a pre-commit "hook" file.

5.2 Interface

5.2.1 Functional specifications

The interface is playing the role of intermediary between the valuation tool which prices the products and the user. It has to be user-friendly and efficient to use.

The principal function of the interface is to collect Portfolios' contracts and to return their price. To make the platform more user-friendly, other secondary functions have to be satisfied. They are described in the Table 13 presenting the interface specifications.

Several Python packages are used to make the platform pratical and visually pleasant to use. **Bootstrap 4** enables to get design more sophisticated html files with for instance a toolbar and other practical components. **Django-tables2** is imported to create functional tables with filters, action buttons, and possibility to order data. **Crispy-form** is a package to design forms with a customised layout. The forms are useful to collect data for a given product for example.

Functions	Criteria
Principal Function	
To Book a contract and get its price	Few interventions of the user, The contracts can be: FX Forwards, a IR Swaps (every type of legs) a cross-currency swaps, swaptions
Secondary Functions	
To stock and store the data correctly	A contract belongs to a portfolio, A portfolio belongs to a client.
To choose the pricing parameters	Pricing parameters include interpolation methods and variables for the market data
To check, modify, or delete data previously loaded	Contracts or pricing parameters
To clearly visualize the data	Present them by category
Generate XML files	The contracts and the pricing parameters' Format adapted to the valuation tool readers
To present the results	
To be usable by different person	To Respect confidentiality Both usable by an audit company with several clients, & by a company pricing only its own portfolios.

Table 13: Interface specifications

5.2.2 Organisation of a Django application

A django application presents four main files which are important for the implementation, as M. Lorant explained it in [9]:

- the models (models.py)
- the views (views.py)
- the URLS (urls.py)
- the templates folder containing all the html files.

I have chosen to add two others : tables.py to implement the tables to present the data and forms.py to code all the forms used to collect or update data.

Models The names of the files speak for themselves. In models.py, we implement the models, which correspond most of the time to a table in the database (SQLite) (if the model class is not abstract). Each model is a Python class, and its parameters are the class attributes. The models organisation and links between one another are crucial since they determine the objects on which we work, which data are stored and how data are linked.

The views are functions or classes whose action will depend on the request. They are stored in views.py. On the server we distinguish two types of request : POST and GET. In the case of the **GET request**, the view collects the information and selects the html template that will appears on the window of the application. Each view is associated to an url address on the server which can contain information. These information can be arguments of the view function.

For the **Post request** (triggered by a click on a button for example), the view function determines which information to collect on the current page and store them in the correct data table of the database. It can also send the user to another URL address and so to a new view function.

A view can create, update, delete instances of one model. Generic Views (inheriting for example from CreateView) are designed in django to reach respectively the previous mentionned purposes. They are linked directly to a model, to a form and to a template, and collect automatically the information filled in the forms. They create or update the model with the accurate attributes' values.

The URLS and views are linked thanks to a code implemented in urls.py, most of the time the information required by the view functions are contained in the URL address.

5.2.3 Organisation of the Interface

Definition of the Interface's Models: The models correspond to the data stored. The models of this interface are so based on the categories of data given in the contracts and required to get the contract price. One class of model corresponds of course to contracts, the inhereted class are the different types of priced contracts in the valuation tool. These subclasses are linked with a one-to-one or one-to-multiple relations to other models which brings additionnal parameters to the contracts.

For Example: *"swaplegs" is a contract that will complete the IRS swap. The leg instances associated to the IRSwap can be infinite, the link between the swap and the legs is one to multiple: we said that IRswap is a* ***foreign key*** *associated to the leg (attribute of the leg).*

To respect the interface requirements, we also create a class "portfolio" and a class "account". That helps to better organise and store the data.

The schemes in Figure 9 explains exactly how the models are inherited and linked (with one-to-multiple relations).

Every instance of each class is **uniquely determined** by an ID. This ID is respectively the portfolio ID or the contract ID . For the legs of the swap, which is a particular case, we choose to identify them both with the contract ID and a leg ID (the leg id being not officially defined in the

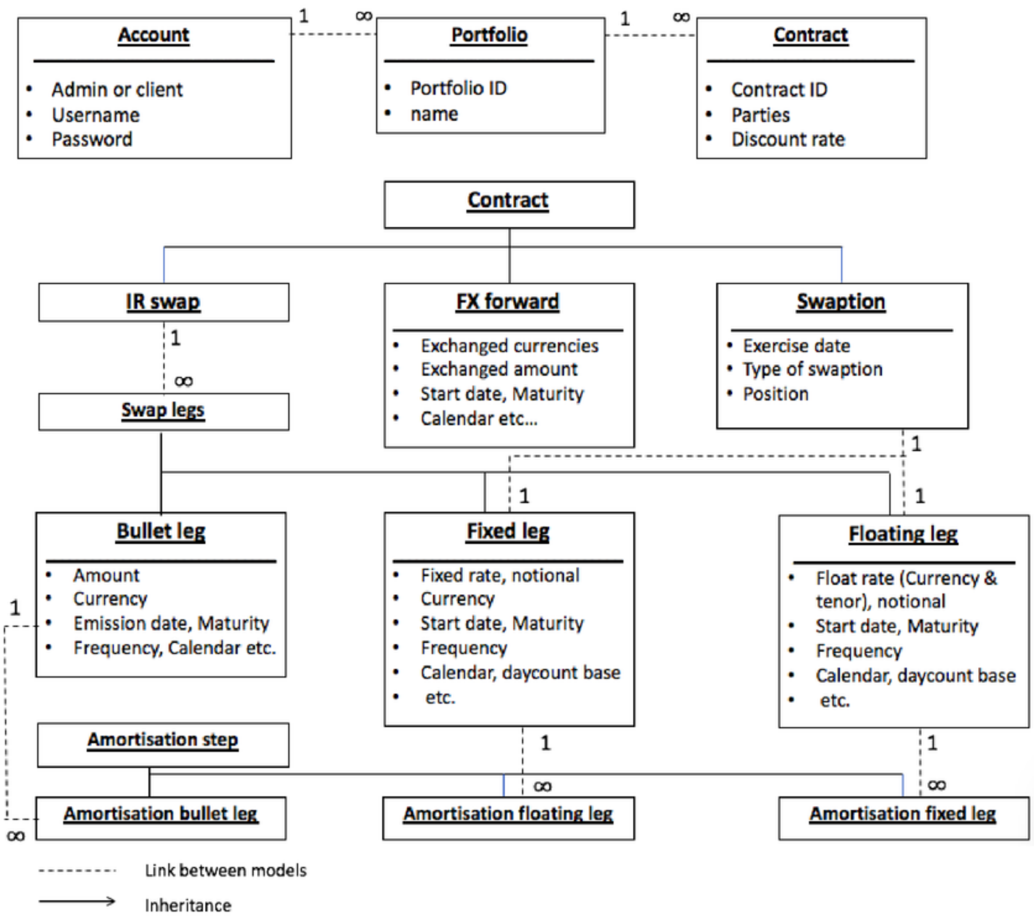


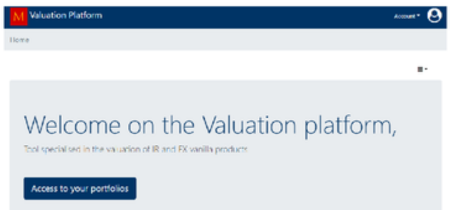
Figure 9: Interface's models

contract signed by the two counterparts and so is not unique). These enables to collect data easily just by mentioning this ID.

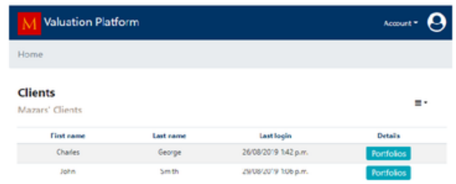
Accessibility As the final tool is to be sold to potential clients, we firstly consider two access modes to the platform : an Administrator mode and a client mode. The Administrator will have access to the whole dataset and can execute every function on it (add new portfolios, new deals, change pricing parameters etc.). This first mode is ideal for an auditor who will have several clients and several company to audit.

Someone using the "client mode" will have only access to his own portfolios and deals. If the user tries to access products that are not his, he will reach a 'denied access' error page.

For the two different user modes, there are notably two different home pages (first page reached once logged in) shown in Figure 10.



Home page "Client mode"



Home page "Admin mode"

Figure 10: Homepages of the interface's two access modes

This is implemented thanks to the model "Account" linked to the django model "User". This class is already designed in Django library and enables notably to define secured passwords. The authentication function is already implemented as well.

We consider obviously that each time the product is sold, the customer will get an application with an empty and secured database.

Interface views: We will distinguish two types of views in this project. The first type is to present a recap of the data registered, and the second is to register new data or to modify the data already registered.

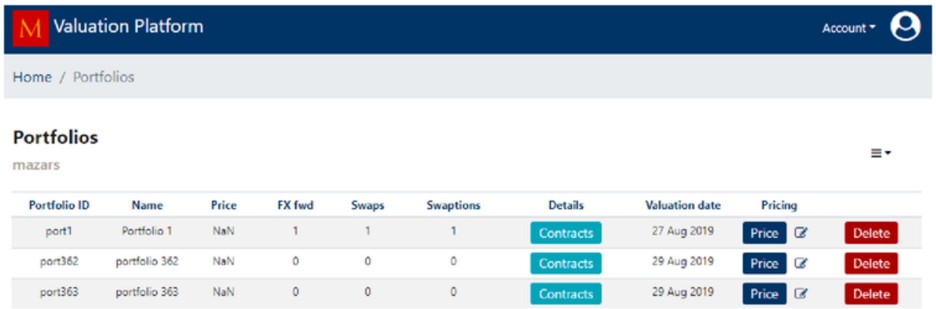
To present the data, tables are used. One table presents the clients, another the portfolios for a given client. There are also a table for the products of a given portfolio and one specific to the IRS swap to present the list of the product's legs.

Besides, these tables are also a mean to add some functions to the interface. Buttons have been added in certain columns to delete or to access to more details concerning the item. In the latter case, for the product tables, they send the user to an update page to possibly modify the data of the product.

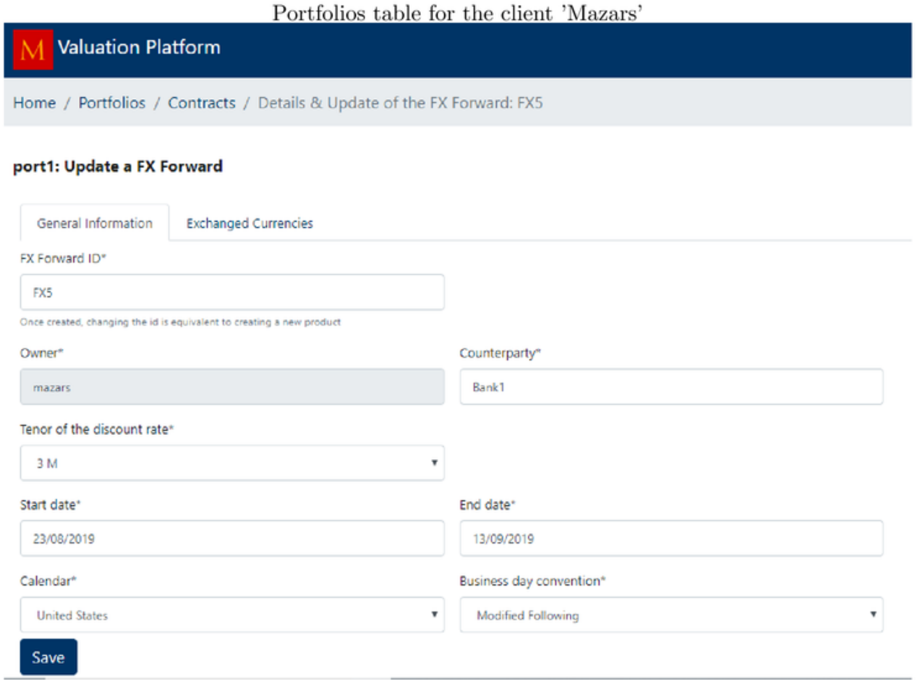
As for the creation and updating of the data, generic views are used. They are inherited from Django classes: CreateView and UpdateView. As explained previously this views link a model to a form and to an HTML template. The forms for the products are of two types. One form is simplified to create or update vanilla products. A lot of fields are defined by default and does not appear in the template. The other forms are more complex and enable the user to choose all the parameters of the products (one field for each parameter).

After clicking on a submit button of a form, the user goes automatically to another view, in our application case, most of the time, he comes back to the table containing the updated data.

The screenshots in the Figure 11 present two examples of the views/templates of the interface.



Portfolios table for the client 'Mazars'



Form to update a FX Forward Product (FX5)

Figure 11: Different types of views of the interface

HTML base template The HTML basis template is used as a base for every template. On the top, a tool bar enables the user to log out or to access to its personal parameters and to change them.

Another tool bar (in gray) appears when you go further in the application. Its tab indicates on which page the user is, and offers the possibility to go back to the previous page.

When the user navigates on the application, the pages succession reflects the organisation of the models. Indeed in the "administrator mode", the succession is as follows:

$$\text{Home Page} \xrightarrow{\text{Choice of the Client}} \text{Portfolios Table} \xrightarrow{\text{Choice of the Portfolio}} \text{Product Table} *$$
$$* \xrightarrow{\text{Choice of the Product}} \text{Product's Page (update form for ex.)}$$

Different functions are available on each pages. They appears in a dropdowns menu triggered by clicking on the button with three strips on the right top corner. For the portfolios table page it is possible to create a new portfolio. On the product table page, the user will have the possibility to create and add different kinds of financial products to the portfolio (swaps, swaptions, FX forward). He can also specify the pricing parameters of the portfolio.

The **pricing function** is in the portofios table where you can choose to price the portfolio, or on the product table page where you can do the same thing, the price in the table will be then updated.

URLS As explained previously URLS determine the view function that will be executed and so the infomation that will appear on the interface. Most of the time, they contain data which can be arguments of view functions or context data appearing in the template.
Their syntax reflects the succession of pages in the interface.

For example:
URL structure: 'portfolios/<client_name>';
Concrete case: 'portfolios/client1';
This URL will send the user (admin or client1) to a table listing all the portfolios of client 1.

URL structure: 'contracts/<client_name>/<portfolio_id>/fxforward/add';
Concrete case: 'contracts/client2/port230/fxforward/add';
This URL will send the user (admin or client2 or else the access will be denied) to the creation form of a Fx forward. Once the FX forward is created, it will be added to the portfolio of the company 'client2' whose ID is 'port230'.

5.3 Transfer of the data: XML files

To transfer the data from the interface, where they were booked in the database by the user, to the valuation tool, files coded in XML are used. There are generated in the interface by **generation functions** included in each model class and put in a folder accessible to the valuation tool. Then **readers** are implemented in the tool to read the XML files and to store the data aiming at using them in the pricers.

Templates of XML files have been designed for each model of the Django application: every contract type, every leg type, every amortisation and also for the pricing parameters which has to be specified to the valuation tool.

The node tags are determined by the name of the model's attributes. Each attibute has a corresponding node. The node organisation is similar to the form organisation. They correspond to all the parameters required to correctly price the product.

To **generate the XML contracts**, the code , for each contract of a portfolio, selects the accurate xml templates. The template is scanned and for each node if an attribute of the model has the same name (node tag = attribute name), its value is written in the node text.

An example of XML contract is given in the Figure 12. It has been made for the swaption contract.

```
▼<trade>
  ▼<trade_header>
     <deal_id>SWAN1</deal_id>
     <party_1>Client Mazars</party_1>
     <party_2>Bank1</party_2>
     <discount_tenor>OIS</discount_tenor>
   </trade_header>
  ▼<swaption>
    ▼<swap_information>
       <!-- Legs to be added here -->
     </swap_information>
    ▼<exercise_information>
       <exercise_date>24/07/2019</exercise_date>
       <type>physical</type>
       <!-- physical or cash -->
       <position>long</position>
     </exercise_information>
   </swaption>
 </trade>
```

Figure 12: XML template of the swaption contract

This file is composed of a main node: "the trade node" which gathers the general information of the trade:

- The two involved counterparts

- The valuation date and currency
- the discount curved tenor (OIS here)

This node is common to every contract. The other sub nodes contained in the part "swaption" are specific to the contract. In the final XML contract read by the pricing tool, the legs' nodes will be introduced at the fitting place.

The reading of the contract is done by the pricing tool: the reader functions scan the XML and store the values (and convert them in the proper type: float, quantlib date etc.) in the corresponding variables used by the pricer later. Thus, the readers create the instruments corresponding to the contacts and parse their information. The instances will have all the properties mentioned by the user.

5.4 Pricing Tool

5.4.1 Global Organisation of the Pricing Tool

As described before, the pricing tool is the other main entity of the valuation platform. It is composed of three main class families: The pricers, the reader/instruments, and the market data classes.

The structure of the pricing tool is clearly presented in Figure 13.

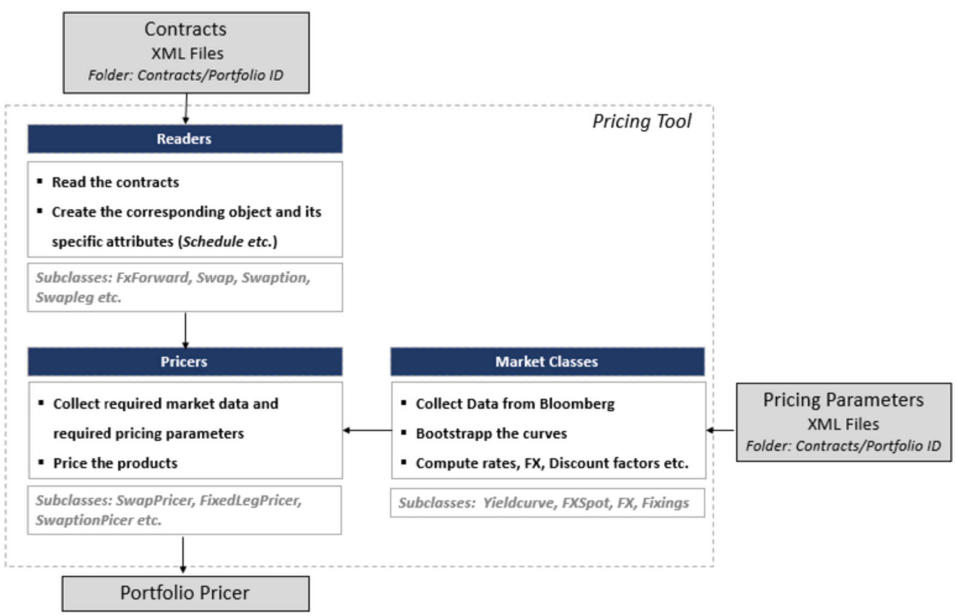


Figure 13: Architecture of the Pricing Tool

5.4.2 Readers/Instruments

When the portfolio generation is launched by the pricing tool, all the XML files are read by the accurate reader. There is a reader for each product (swaption, FX forward, IR swap) and also for the legs and the pricing parameters.

One instance of the relevant instruments is created and also its attributes like the schedule of the swap.

No information concerning pricing is introduce in the instrument's parameters. Indeed, it is, as a result, possible to price one instrument with different interpolation methods, discounting tenors etc. without modifying it, without changing the instrument's attributes.

Concerning the structure of the code, each specific reader is a subclass of a basis class *TradeReader*. This mother class is in charge of collecting all the information common to all the products: those contained in the "Trade node" detailed above.

5.4.3 Pricers

A pricer has been implemented for each kind of products. They are subclasses of a mother class *Pricer*,which is a "Pricing Library". For each type of instruments, a different pricer is called. Then the pricer collects, from the market, the data required and the parsed pricing parameters. As a result, with the instrument's parameters and the market data, the pricer uses the adapted formulas and return the correct price. It computes also intermediary results (it is possible to have access to the cash flows' values for a swap leg for example).

A different pricer is implemented for the swap (and for each swap leg type), for the fx forward and for the swaption. The swap pricer calls the leg's pricers and then sums the results received for each leg.

5.4.4 Market: The Collection and Analysis of Market Data

A class "Market" gathers in dictionaries every path to collected market data. They are stored in csv files after being collected on Bloomberg. One instance of the Market class matches one reference date (used as valuation date). Market's methods are functions to collect the targeted files (corresponding to one particular rate index for example) and to create a 'marketdata' instance. Indeed Market Data is a mother class for each type of data. Market Data are Yield_curves, Fixings, FXspot, FxForward etc. A market data subclass, depending on the type, contains methods for parsing data, calibrating curves, computing associated rates (discount factor, forward ...) etc. To calibrate curves when the case occurs, calibration pricers are used. They are really specific to vanilla products quoted on the market. They price the instruments and determines the rate value by comparing the computed price to the quoted price. This determination is made by using Newton-Raphson algorithm.

5.4.5 Conventions, Tools, Static parameters

To compute the price of the contracts, the time and dates need to be precisely defined and computed. The payment dates or reset dates for a swap for example need to be, in particular really precise.

These dates depends on the calendar chosen, but also on the business day convention. The **calendar** is different according to the country in which the trade has been concluded for example. We distinguish United Kingdom calendar, Hong Kong calendar, the United States calendar (etc.) in which the business days, the bank holidays and other characteristics can be different. There are also different business day conventions. These conventions fix the rule we adopt if a given date is not a working day. We can take then the following day or the previous one for example. The daycount base is also important to determine: a year can have 360 days or 365,25 days etc.

Instead of implementing all these rules in the platform, Quantlib library is used. It is quite pratical and liable for this kind of issues as presented by G. Balaraman in [8].

Every static parameters, and shared functions (interpolation, date and period computations, XML files parsing functions etc.) are stored in files containing all the tools of the platform.

5.4.6 Unitests

As explained previously, unitests are implemented for each pricer, market curve bootstrapping and interpolation functions. They primarily check that the development of the tool does not wrongly impact the previous functions that were correctly implemented.

Conclusion

A valuation tool has been implemented providing a new efficient and practical way to price vanilla products such as the FX forwards, interest rate swaps and the swaptions. It is fast and convenient for the user to store new contracts, filling them on the interface, and to price them. The price computed are correct and precise and so can the tool start to be used in the short term.

As a result the initial objectives of this project have been reached. However, this is still a first version of the platform: some innovations are planned to be implemented for other versions to come. We wish to improve the platform to price more complex products (more complex swaptions, caps and floors) but also to improve the interface adding new features to print for example intermediary results.

It was a multidisciplinary project that enables me both to use financial skills to price the products but also to discover application implementation. It is really interesting to notice that it is possible to implement a tool to price vanilla products which is pleasant and easily usable by people without specific quantitative finance skills.

References

[1] D.Brigo. *Interest Rate Models with Credit Risk, Collateral, MSc Mathematics and Finance*, Imperial College, London, 2016-2017.

[2] Z. Grbac, A. Papapantoleon, J. Schoenmakers, D. Skovmand. *Affine Libor Models with Multiple Curves: Theory, Examples and Calibration*. SIAM Journal on Financial Mathematics, 2015.

[3] F. Ametrano, M. Banchetti. *Everything You always Wanted to Know about Multiple Interest Rate Curve Bootstrapping But Were Afraid To Ask*, 2013.

[4] A. Gianolio. *On the pricing of Bermudian swaptions in the multi-curve LIBOR Market Model*, Delft University of Technology, 2016.

[5] N. Choukar. *CVA-DVA Tour dhorizon des mthodologies de calcul*, Internal document, Mazars, 2017.

[6] M. Taylor. *Currency Forward or FX Forward Introduction and Pricing Guide* available on `https://finpricing.com/lib/FxForward.html`, FinPricing, 2018.

[7] W. Boenkost , W M. Schmidt. *Cross currency swap valuation*, HfB-Business School of Finance Management, 2004.

[8] Goutham Balaraman, Luigi Ballabio. *Quantlib Python Cookbook*, available for sale on `http://leanpub.com/quantlibpythoncookbook`, 2017.

[9] M. Lorant, M. Xhonneux. *Developpez votre site web avec le framework django*, OpenClassroom, available on: `https://openclassrooms.com/fr/courses`, 2019.

IMPERIAL COLLEGE LONDON

THESIS DECLARATION FORM

(for candidates whose degree will be awarded by **Imperial College**)

Declaration: I hereby confirm that the work contained in this thesis is my own work unless other wises stated.

Name Apolline Bonnerre.. **CID** …01584498…..…………

Title of Thesis......Automated Valuation Platform for Vanilla Products……………………........................

Month and year of submission for examination……September 2019…………………………………

Date….9/09/2019…………………………

Signature of Candidate……Apolline Bonnerre………

Automated valuation platform for vanilla products

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55