LUISS

MSc. in Corporate Finance
Department of Business and Management

Course of Asset Pricing

Monte Carlo Methods and Market Models for European Swaptions pricing

Prof. Paolo Porchia
SUPERVISOR

Prof. Marco Pirra
CO-SUPERVISOR

Laura Bruno
ID: 706961
CANDIDATE

Academic Year 2019/2020

"What you get by achieving your goals is not as important
as what you become by achieving your goals"
-H. D. Thoreau

Table of Contents

Table of Figures

Introduction

This research is aimed at implementing an algorithm to price European Swaption that can capture current market conditions, which are embedded in macro-financial variables. In the model built in this paper, the calibration primarily takes into account the current term structure and quotes for specific classes of derivatives.

Derivatives started to be traded in the early 90s and, despite the crash related to the 2008 financial crisis, the volume of the notional amount exchanged on OTC market reached 640 trillion (as in June 2019). Those instruments are used from multiple individuals and corporates for speculation and hedging purposes. In particular, fixed income derivatives are mainly used for hedging the risk arising from interest rates, and, not surprisingly, interest rates derivatives are the most liquid sub-category. Those instruments allow investors that are exposed to interest rates, such as issuers of floating-rate securities or investors of callable securities, to be protected against fluctuations in the term structure. Swaptions are exotic options where the underlying is a swap contract, and the buyer of a receiver/payer swaption will have the right at maturity to enter in a receiver/payer swap at a pre-determined strike swap rate. Notwithstanding the diffusion of these instruments, their pricing process poses multiple challenges in its implementation, that will be examined in this paper.

First of all, since the payoff depends on future interest rates, this type of derivatives requires an appropriate process to understand and predict the term structure. Over time, an extensive literature has been developed on term structure modelling, and although there are multiple types of models the most widespread one is the Affine class which will be thoughtfully analyzed.

Furthermore, whilst in equity derivatives, the payoff and the discount factors depend on different variables, in fixed income derivatives both the discount factor and the payoff are built upon the interest rate. This peculiarity causes a non-zero correlation between these two which alters the discounted expected value of future outcomes. Changing the probability measure and obtaining the so-called Equivalent Martingale Measure result, will allow to have a deterministic discount factor outside the expectation operator, and consequently eliminate the problem of correlation. The adoption of a new probability measure, namely the T-forward measure, lays the groundwork for the application of Market Models such as the LIBOR Market Model and

the Swap Market Model. This study will analyze and compare short rates models and market models and their ability to price swaptions reflecting current market conditions.
The chosen model, after a discretization procedure, is implemented through the famous statistical tool of Monte Carlo simulation, which allows describing the true dynamics of interest rates, whom advantages for swaptions will be illustrated.

With the appropriate calibration, the resulting pricing algorithm performed generates an estimator of the price of a European receiver swaption with one year of maturity. On the result are performed sensitivity analyses to explain the variation of the price concerning the main variables: strike and tenor of the underlying swap. The variance of Monte Carlo estimators is determined and appears to be higher for greater tenor and strike values. Finally, to validate the pricing algorithm based on Monte Carlo simulation and LIBOR Market Model its outcome is compared with at-the-money swaptions quoted on the market at the day of evaluation. The Mean Squared Error is additionally investigated.
The research structure follows the logical steps of the pricing process. The first section reviews extensively the most famous Affine models for term structure modelling and the different existing volatility structures, examining the mathematical steps needed to change the probability measure and obtain the solution to the stochastic differential equation for pricing. The second section provides the theoretical knowledge about Monte Carlo methods, underlining the importance of these tools in complex multidimensional integration problems and explaining the advantages of using Monte Carlo estimations. This chapter goes also through variance reduction techniques and the convenience in the application of Monte Carlo simulation to swaption pricing. The third chapter is aimed at performing the pricing, explaining the derivation of the specific discretized stochastic process simulated and its calibration on the current term and volatility structure. The results and the sensitivity analyses are presented together with the model validation. The role of modern pricing techniques in the landscape of global markets is explained and future challenges of the latter techniques are identified. Finally, the conclusions are reported and extensions of the model for future implementation are suggested.

Chapter 1

Affine models for term structure modelling

1. Exotic options and European swaptions

The term "exotic options" includes any option type that is different from Vanilla options, from the perspective of payoffs and cash flows structure. This class of derivatives brings many advantages in term of use, because of their complex structure of payoffs and cash flows, they can easily meet different investors' needs. Two main categories of exotic options can be distinguished: path independent are characterized by payoffs that are only function of the underlying asset price at maturity, whereas path dependent's payoffs are functions of the price path of the underlying as whole or in some specific portion. For categorization purposes also the dimension and the order of the option are often taken into account. The former is the number of variables that characterize the payoffs, and the latter is related to the type of function that links those variables and the payoffs. In this paper, I will go through interest rate models and Monte Carlo simulation with the final intent of pricing a type of exotic option on fixed income, that is Bermudian and European Swaption. There exist a considerable body of literature that will be soon introduced, that shows the progress that have been made in term of accuracy to price fixed income derivatives. Notably, several theories have been dedicated to swaptions pricing with some remarkable results during the latest year of the XX century.

2. Structural Affine Models

Structural models for yield curve are aimed not only at characterizing but also at understanding the term structure and its changes, being in this way extremely useful for forecasting purposes. With no doubt, among many types of structural models, the most widespread class is the affine one. An important constraint that is imposed on these models is the no-arbitrage condition, which ensures that prices do not allow no-risk profits. The no-arbitrage condition can also guide statistical estimates in the process of characterization of the yield curve. In this chapter I will analyze briefly the first generation of Affine Models, Vasicek (1977) and CIR (1985) and three important drivers of the yield curve, to focus later on in the chapter on more complex models which rely on the forward neutral probability measure. More specifically the Libor Market Model by Brace, Gatarek and Musiela will be used to simulate the trajectories of future spot Libor rates and price Bermudian Swaptions.

To introduce the Affine setting, Rebonato highlights three factors which determine the structure of the yield curve: Expectation, Risk Premia and Convexity. Expectations are often included in affine models (e.g.: Vasicek) through a mean-reverting component, which is able to capture a long term mean reversion that unfortunately does not fit well the intuition by which expectation might influence the short term part of the yield curve instead of the long term one (Rebonato, 2016).

Vasicek and CIR models represent the first generation of affine models, they both take into account the no-arbitrage condition and the main drivers of the yield curve in a one-factor equation, and the state variable is an affine diffusion under both physical and risk neutral measures. Short rate models, in general, are highly intuitive and flexible for their ability to explore the dynamics of an instantaneous continuously compounded short rate r_t. In the Vasicek model, the short rate increment follows a generic Gaussian Markov process in which the short rate reverts to a long term fixed level γ with a reversion speed of φ (Cox et al., 1985). The SDE is the following:

$$dr_t = \varphi(\gamma - r_t) + \sigma_r dz_t^{\mathbb{P}}$$

Where, in this case, $dz_t^{\mathbb{P}}$ is the Markovian increment in real-world probability measure. The component φ has central importance, precisely when the reversion speed is zero the duration of the security grows linearly with maturity, while at a higher reversion speed the duration grows less with maturity, meaning that for high reversion speeds the security is less sensitive to changes in the yield (Vašíček, 1977).

The risk premia is the excess return required from investors to bear some specific level of risk, and the compensation related to each risk factor per unit risk is determined in the following way:

$$u_j^{t,T} = \frac{\partial P_t^T}{\partial x_j} \sigma_j^t \lambda_j^t$$

Where $\frac{\partial P_t^T}{\partial x_j}$ is the price sensibility of the specific bond to the j^{th} risk factor, σ_j^t is the volatility of the risk factor, and λ_j^t is the market price of one unit of that risk. Considering all the risk factors that characterize a security, the expected return at time t can be defined as the sum between the compensations of all the risk factors:

$$\frac{E[dP_t^T]}{P_t^T} = r_t + \sum_{j=1}^{n} \frac{1}{P_t^T} \frac{\partial P_t^T}{\partial x_j} \sigma_j^t \lambda_j^t$$

The market price of risk λ_j^t is assumed to be constant in the first generation of Affine Models. A number of questions regarding this assumption remain to be addressed, and even though this approach is particularly straight forward it does not capture the shown trend of positive excess return when the yield curve is steep and zero or negative excess returns when the curve is flat or downward sloping (Rebonato, 2014).

Convexity captures the non-linear relationship between yields and prices and might be responsible for the shape of the term structure for long maturities. The above mentioned non-linear effect in the Vasicek model is observed in the volatility of the yield, which depends quadratically on the volatility of the state variable and on the sensitivity term (Rebonato, 1999). The no-arbitrage assumption, that represents an important landmark in pricing is translated in the equation below, which in simple words states that the return on a security should equate the sum of the compensation for every source of risk (Kim & Wright, 2005).

$$\frac{E_t^{\mathbb{P}}\left[P_{t+dt}^{T-dt}\right] - P_t^T}{P_t^T} = r_t dt + \frac{1}{P_t^T}\left(\sum\nolimits_i \frac{\partial P_{t+dt}^{T-dt}}{\partial x_i} \lambda_t^i(\vec{x}) \sigma_{x_i}\right) dt$$

At this point it is important to underline that there is a trade-off regarding the type of variables (or factors) on which the model should depend. From one perspective, choosing variables which derive from macroeconomic equilibrium models surely simplifies the intuition behind the model, yet it is important to double-check the robustness of the macroeconomic assumptions behind them. From the other perspective choosing variables that come from statistical and econometric analyses, such as Principal Component analysis (Rebonato et al., 2014), adds opaqueness and introduces the problem of overparameterization (Rebonato, 2016).

3. Ho-Lee, Hull-White and Black, Derman, Toy models

Models such as Vasicek and Cox, Ingersoll and Ross are not able to exactly fit the dynamic of the current yield curve, luckily other studies carried by Ho and Lee, Hull and White and Black Derman and Toy proposed new models that were not only able to fit the term structure, but once revisited and extended were also able to fit the observed volatility structure.

Ho Lee model, from 1986, is the most straight forward and the first one able to fit the term structure of interest rates. The short rate follows a stochastic process with drift θ_t and diffusion σ.

$$dr_t = \theta_t dt + \sigma dW$$

Where the drift θ_t is chosen exactly to fit the term structure. Those parameters are found using a bootstrapping procedure that starts from zero-coupon bonds' prices, and with the use of a searching algorithm, the first parameter θ_0 is found so that the price of the first ZCB is returned from the risk-neutral tree. Afterwards, the second parameter is found, and so on (Ho & Lee, 1986).

The estimate for θ_t can also be obtained in a less time-consuming process from the forward rates structure. Recalling that the forward rate is equal to the short rate plus the slope of the spot curve:

$$f(0,t) = r(0,t) + t\frac{\partial r(0,t)}{\partial t}$$

Applying the pricing equation from the Ho Lee model is obtained the following equation for the θ_t term.

$$\theta_t = \frac{\partial f(0,t)}{\partial t} + \sigma^2 t$$

Despite the intuitive nature of the Ho Lee model, there are some flaws due to its simplicity. In fact, the model allows a positive probability of negative interest rates because of the symmetrical distribution. Furthermore, it uses the empirical volatility, computed from

historical interest rates assuming a flat volatility structure (Veronesi, 2005a). Therefore, it tends to overprice low maturity caps, floors and swaption and underprice long maturity ones.
The Hull White model, introduced in 1990, extends the Vasicek model in order to fit the term structure. With respect to the Ho Lee model, it presents a mean reversion component.

$$dr_t = (\theta_t - \gamma^* r)dt + \sigma dW \qquad (4.0)$$

The parameter θ_t can be as well estimated with a bootstrapping procedure or directly from the forward curve with the same procedure of the previous model, with the following result (Hull & White, 1990).

$$\theta_t = \frac{\partial f(0,t)}{\partial t} + \sigma^2 f(0,t) + \frac{\sigma^2}{2\gamma^*}(2 - e^{-2\gamma^* t})$$

In 1994 the same model was broadened to a two factor model, to give a better shape of the term structure. Unfortunately, it still allows negative values for the short rate. On the positive side, one can choose the parameters γ^* and σ to best fit the forward volatility structure.

The Black, Derman and Toy (BDT) model, introduced in 1990 as well, applies a transformation to the short rate defining a new variable, $z_t = \ln(r_t)$. The logarithmic transformation gives the variable a distribution with positive skewness that changes the result of the estimate. This procedure gives a zero probability to negative interest rates, however assigns higher probabilities to high levels of interest rates and lower probability to low interest rates. The stochastic process followed by the increment of the variable z_t is the following (Black et al., 1990):

$$dz_t = \left[\theta_t + \frac{\sigma'_t}{\sigma_t} z_t\right] dt + \sigma_t dW$$

When volatility is considered constant the mean reversion component drops to zero, and the model becomes simply a logarithmic version of the Ho Lee model [Hull, 2018 p. 738]. This model tends to underprice for all maturities derivatives such as Caps, Floors and Swaptions. The model allows fitting exactly not only the term structure but also the volatility structure.

However, the simple version of the BDT model does not fit the volatility structure, because when substituting the diffusion term σ with σ_i the resulting tree is not recombining anymore (Veronesi, 2005a).

4. Flat and Forward volatilities

Notwithstanding the fact that the above models are able to fit the term structure and consequently correctly price bonds, they still do not correctly price the type of interest rate derivatives that are being analyzed in this paper. This is due to the lack of matching with the volatility implied by market prices of those securities. The implied volatility is indeed the volatility implied by the dollar price, that if applied to the pricing formula gives back the exact same price quoted on market. Often derivatives quotes are expressed in terms of implied volatility, as for swaption contracts. If we consider different caps, one for each maturity, starting from their prices we can extract with the use of the Black formula the implied volatility for each cap. The issue related to this specific volatility is that each one is able to price only a single cap because with this process it is of necessity assumed a different flat volatility structure for each cap. However each cap has different caplets at different maturities, and recalling that the price of the cap is equal to the sum of the single caplets' prices, it is inconsistent to price with different volatilities two caplets with identical maturities coming from two different caps having different maturities. Accordingly, it is reasonable to extract from implied volatilities the structure of Forward volatilities so that to each point in time corresponds a volatility that is able to price all the caplets for that maturity. This is implemented through a bootstrapping procedure where the starting point is the cap with the closest maturity, whose flat volatility corresponds to the first step forward volatility. Generalizing, the dollar value of the i-th caplet is found in the following way from the dollar price of the cap (Veronesi, 2005a);

$$T_i \; caplet \; dollar \; value = Cap(T_i) - \sum_{j=1}^{i-1} Caplet(T_j, r_{K,i}, \sigma_f^{Fwd}(T_j))$$

Where $\sigma_f^{Fwd}(T_i)$ is the forward volatility for maturity i, nonetheless our unknown. Once the value of the caplet of interest is found it is possible to find the value of the forward volatility from the Black formula. Then the procedure is repeated for each maturity.

5. Probability measures

The price of a derivative is equal to the discounted expectation of payoffs under the risk neutral measure. The most common measure of value, that is the one used under the traditional risk neutral measure, is the risk free rate of return. The risk neutral pricing methodology starts from a generic stochastic processes:

$$dr_t = m(r_{t,}t)dt + s(r_t, t)dX_t \tag{4.1}$$

Where dX_t is a Wiener component, and the drift *m* and the diffusion *s* are functions of the short rate and of the time. Considering now a portfolio made of two derivatives dependent on the short rate, f and g. The latter securities follow the processes

$$\frac{df}{f} = \mu_1(r_{t,}t)dt + \sigma_f(r_{t,}t)\, dX_t \tag{4.2}$$

$$\frac{dg}{g} = \mu_2(r_{t,}t)dt + \sigma_g(r_{t,}t)\, dX_t \tag{4.3}$$

where μ_1, μ_2, σ_1 and σ_2 are functions of the short rate and of the time. Now an instantaneous riskless portfolio can be constructed, with $\sigma_g g$ unit of f, and $-\sigma_f f$ unit of g (Hull, 2018b). The process that this portfolio follows can be obtained applying Ito's lemma, imposing the no-arbitrage condition, so that the portfolio return must equal the risk free rate, one can obtain the PDE that every security must satisfy. This steps, similar to the Black and Scholes setting, bring to the Fundamental Pricing Equation.

$$rV = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial r}m^*(r,t) + \frac{1}{2}\frac{\partial^2 V}{\partial r^2}s(r,t)^2$$

With the boundary condition of $V(r,T) = h_T$, where h_T is the payoff. The Feynman-Kac theorem provides us with a general solution to the PDE,

$$V(r_t, t) = \mathbb{E}^*[e^{-\int_t^T R(r_u)du}h(r_t,t)|r_t]$$

The expectation operator is with respect to the risk neutral interest rate process, that has the risk neutral adjusted drift (Veronesi, 2005b).

Regrettably, the Fundamental Pricing Equation is not easily applicable to interest rate derivatives, whose pricing is the aim of this thesis. In fact, in that case both the discount factor and the payoff depend on the interest rate, consequently there is a positive correlation element that should be taken into account.

$$V(r,t;T) = \mathbb{E}^* \left[e^{-\int_t^T r_u du} h_T \right] = \mathbb{E}^* \left[e^{-\int_t^T r_u du} \right] * \mathbb{E}^*[h_T] + cov(e^{-\int_t^T r_u du}, h_T)$$

The change of numeraire allows a simplification of the pricing formula in this situation. Assuming the security $Z(r,t;T)$ as a numeraire, the normalized value of the derivatives becomes $\tilde{V}(r,t;T) = \frac{V(r,t;T)}{Z(r,t;T)}$, where V(r,t;T) is the dollar value. The normalized value satisfies the following partial differential equation

$$0 = \frac{\partial \tilde{V}}{\partial t} + \frac{\partial \tilde{V}}{\partial r}(m^*(r,t) + \sigma_Z(r,t)s(r,t)) + \frac{1}{2}\frac{\partial^2 \tilde{V}}{\partial r^2} s(r,t)^2$$

where $\sigma_Z = \frac{1}{Z}\frac{\partial Z}{\partial r} s(r,t)$ is the diffusion of the numeraire. The equivalent martingale measure result shows that for a given numeraire, and a particular choice of the market price of risk, the normalized price $\tilde{V}(r,t;T)$ is a martingale for all the securities (Musiela & Rutkowski, 1997). The change of measure in the case of constant market price of risk is applied in the stochastic process of interest with a parallel shift of the drift. For instance, considering again the Vasicek process it is possible to observe that when moving from the real world to the traditional risk neutral world changes the reversion level by $\Delta\gamma = \lambda_0 \sigma_r$ (Veronesi, 2005b)

$$dr_t = \varphi(\gamma - r_t + \lambda_0 \sigma_r) + \sigma_r dz_t^{\mathbb{Q}}$$

If the market price of risk is positive, the shift will be positive as well, meaning that extra return is required. This results in a higher reversion level and consequently higher levels of rates. Now that it is clear how the probability measure changes the drift in the simple Vasicek model, it is possible to observe the important result of the Equivalent Martingale Result. Considering the two securities analyzed in equation 4.2 and 4.3, and choosing g as numeraire and $\lambda_0 = \sigma_g$ as the market price of risk, then applying Ito's Lemma to the normalized price $\frac{f}{g}$ it is possible to prove that the latter is a martingale (equation 4.2). This is called the forward-risk neutral world with respect to the security g (Hull, 2018a).

$$df = (r + \sigma_f \sigma_g) f dt + \sigma_f \, dX_t$$

$$dg = (r + {\sigma_g}^2) dt + \sigma_g \, dX_t$$

$$d\frac{f}{g} = (\sigma_f - \sigma_g) \, \frac{f}{g} dX_t \qquad (4.2)$$

The price of the derivative $\frac{f}{g}$ is given by:

$$\frac{f_0}{g_0} = \mathbb{E}_g{}^* \left[\frac{f_T}{g_T} \right]$$

Likewise, the normalized price of the portfolio considered before $\tilde{V}(r,t;T)$, recalling that the chosen numeraire was $Z(r,t;T)$, is equal to:

$$\tilde{V}(r,t;T) = \mathbb{E}_f{}^*[h_t]$$
$$V(r,t;T) = Z(r,t;T)\ \mathbb{E}_f{}^*[h_t]$$

The important result is that now the discount factor is outside the expectation operator. This allows further manipulation that will be shown in the next paragraph.

6. Forward neutral models

The LIBOR Market Model from Brace, Gatarek and Musiela (1997), uses the Equivalent Martingale Measure result to price fixed income derivatives using the dynamics of the LIBOR forward rate as a starting point. Considering a derivative whose payoffs are function of the LIBOR, the present value of the contract today is equal to

$$V(r,t;T) = Z(r,t;T)\mathrm{N}\Delta\mathbb{E}_f{}^*[r_n(\tau,T) - r_K]$$

Where the expectation is taken with respect to the T-forward risk neutral dynamics. Recalling that the forward rate is equal to the expected future spot rate, which is itself the specific strike rate that makes the value of the derivative equal to zero. Since the forward rate has to move towards the spot rate while maturity is approaching, It is also possible to state that the T-forward rate is a martingale and it is assumed to follow a lognormal drift-less diffusion process under the T-forward risk neutral dynamics. Since as just mentioned the forward rate converges to the spot rate, the LIBOR rate has a log normal distribution as well (Alan Brace et al., 1997).

$$r_n(\tau,T) \sim LogN\left(f_n(0,\tau,T);\ \int_0^\tau \sigma_f(t)^2 dt\right)$$

This key assumption is not only useful because it yields the straight forward Black Formula, but it is also useful in its applications to Monte Carlo method and to more complex derivatives depending on the LIBOR rate, such as Swaptions.

When pricing derivatives depending on a single LIBOR rate, the forward volatility structure extracted from caps as explained in paragraph 4 is considered. The latter is enough to have a full characterization of the process of the forward rate, and can be assumed equal to the volatility of the forward rate. Knowing the lognormal distribution, even if the final payoff is more complex it is possible to simulate the final LIBOR rate on which the payoff depends on using a Monte Carlo simulation, where each simulated path s can be obtained with the following formula (Veronesi, 2005b):

$$r_n^s = e^{\log(f(0,\tau,T)) - \frac{1}{2}\sigma_f^{Fwd}(T)^2\tau + \sigma_f^{Fwd}(T)\sqrt{\tau}\varepsilon^s}$$

Once all the trajectories are simulated the value of the derivative can be easily obtained discounting the payoffs and plugging them in the Monte Carlo estimator formula. However, the application of this model to derivatives whose payoffs depend on multiple LIBOR rates might require some adjustments that are explained in the next paragraph.

7. Forward Rates distribution under the LIBOR Market Model

A swaption gives the option to enter into a swap contract with swap rate equal to a predetermined strike rate, at the exercise date. If at the exercise date the current swap rate on the market is less convenient (higher/lower swap rate observed on the market in case of a payer/receiver) the option will be exercised. Thus, the payoff value at the exercise date can be computed as the value of the exchange of two coupon bonds: one for the fixed leg characterized by a coupon rate equal to the option strike, and one for the floating leg that is at-par by definition. However to compute the present value of this bond is necessary to have as many discount factors as the number of payment of the bond. It is obvious that the final payoff of this derivative depends on multiple future LIBOR rates. For this reason under the T-forward risk neutral measure there is only one forward rate that is a martingale, while all the other rates are no more log-normally distributed. Therefore only one Forward rate will be drift less, with the other rates having a more complex stochastic process.

By choosing as numeraire the $\bar{T}$-forward rate by which for every i+1 $\bar{T} < T_{i+1}$ is true, that means choosing as numeraire the smallest of the discount factors of interest, the process followed by the increment of the forward rate is (Alan Brace et al., 1997):

$$\frac{df_n(t,T_i,T_{i+1})}{f_n(t,T_i,T_{i+1})} = \left(\sum_{j=\bar{\iota}}^{i} \frac{\Delta f_n(t,T_j,T_{j+1})\sigma_f^{i+1}(t)\sigma_f^{j+1}(t)}{1+\Delta f_n(t,T_j,T_{j+1})}\right)dt + \sigma_f^{i+1}(t)dX_t$$

This process will be applied to the forward rate in Monte Carlo simulations explained in the third chapter. Furthermore, the LMM is not the only Market Model that can be used to price swaptions, and when explaining the implementation of a pricing algorithm for swaptions in Chapter 3 I will also go through the advantages of using the LMM instead of other models.

8. Volatility Structures

In the simplest application of the model, the instantaneous volatility was considered to be constant in each forward rate, independently of the time t at which that specific forward rate is observed. However, the forward volatility structure extracted from caplets is characterized by a bump which is not coherent with the assumption just explained. Furthermore, in a swaption the payoff is not only function of a single LIBOR, therefore the simplistic assumption made in the previous paragraph is not feasible anymore. Alternatively, in the application of the model that this paper follows there are two other assumptions to be made, for the volatility of the forward rate used in the final the simulation.

The first assumption is that the volatility of the forward rates has a dependence only with the time to maturity, through a specific function $S(\cdot)$. The second assumption is that the function is constant in each period. These assumptions give as a result a semi-linear volatility structure with S_i being the volatility of the forward rate $f_n(t, T_i, T_{i+1})$ (Brigo & Mercurio, 2001b). The result of these assumption is summarized by the following equation:

$$\sigma_k(t) = \sigma_{k,\beta(t)} := S_k$$

To extract this volatility structure is sufficient to have the forward volatility structure mentioned in paragraph 4, and then apply a bootstrapping procedure. The first step of the bootstrapping is similar to the one already presented for Forward volatilities:

$$S_1 = \sigma_f^{Fwd}(i\Delta)$$

While the other steps are obtained with the following formula, assuming that $t = 0$:

$$S_i = \sqrt{\frac{1}{\Delta}\left(\sigma_f^{Fwd}(T_{i+1})T_i - \sum_{j=1}^{i-1} S_j^2 \Delta\right)}$$

Yet convenient, this volatility structure has some flaws. In fact, when the Forward volatility declines fast the function S drops to zero by its own nature (Brigo & Mercurio, 2001b). This function will be used in the Monte Carlo simulation run for the empirical analysis in support of this paper, and its application will be better explained in the last chapter when dealing with calibration of the model.

Chapter 2
Monte Carlo Methods in finance

1. An introduction to Monte Carlo

Monte Carlo methods are statistical tools to estimate a deterministic quantity. The oldest acknowledged use of Monte Carlo methods is traced back to the famous Buffon's needle problem to estimate π. More recent applications are attributed to the period between 1930 and 1950 and come from the field of physics. Nowadays, the use of Monte Carlo is diffused in many field, including financial engineering. They are based on the mathematical intuition that the probability of an event happening is represented by the volume of the possible outcomes that make the event happen. Monte Carlo methods start from the volume of a set of outcomes to estimate the volume of the probability. In simple words, it randomly samples outcomes and picks up only the ones of interest (i.e.: a given set of outcomes), to interpret their volume as the probability of that set of outcomes. Considering an interval $[i; i + \Delta]$ and a function f, the actual value of the volume of this function over the integral is represented by the following integral (Glasserman, 2003):

$$\alpha = \int_{i}^{i+\Delta} f(x)dx$$

Since the volume represents the probability, α is equal to the expectation of X, $\mathbb{E}[f(X)]$, where X is uniformly distributed over the mentioned interval. What Monte Carlo sampling does is simulating random points of X and then computing the value of the function at those points that fall in the interval $[i; i + \Delta]$ (Tezuka, 1998). It is then possible to characterize an estimate of the expectation α by taking an average of the value of f as follows:

$$\widehat{\alpha_n} = \frac{1}{n}\sum_{i=1}^{n} f(X_i)$$

By the law of large number when $n \rightarrow \infty$ then $\widehat{\alpha_n} \rightarrow \alpha$. It is possible to show that the error of the estimate is $\varepsilon \sim N\left(0; \frac{\sigma_f^2}{n}\right)$ where σ_f is the standard deviation of the function over the specific interval and can be estimated with the sample standard deviation (Glasserman, 2003).
It is easy to see from this example that the application of Monte Carlo in derivatives pricing opens many doors.
There is a subtle difference between the terms Monte Carlo sampling and simulation. The former includes all those cases in which is not necessary to simulate overtime, whereas the latter provides for sample paths that might be particularly useful in derivatives pricing where the final payoffs depends on the price path (e.g.: path-dependent exotic options) or in the case of American style derivatives.
The simulation of path can be of three types: discrete time models, continuous time models and discrete time event. For discrete time models the interval of time is divided in discrete time steps, that usually have the same length. The transition between states happens at every step based on the dynamics of the models.
In continuous time models the dynamics are described by a differential equation. The simulation is used when the differential equation is not only composed by a deterministic part, but also by a stochastic one (Brandimarte, 2006). To simulate continuous time models a discretization method is necessary. Since discretization bias plays its role in the simulation estimate, the choice of a discretization method must be made wisely. The two most widespread methods are Euler and Milstein schemes (Frey, 2008).
Considering a generic Stochastic Differential Equation for any variable *X*, as in formula 4.1. If one wants to find a solution for the distribution of the variable a Monte Carlo simulation might be needed, and to simulate the variable in continuous time we need to *estimate* an approximation $\hat{X}_T$ of it through a discretization scheme.

$$dX_t = m(X_{t,}t)dt + s(X_t, t)dW_t$$

The starting point of Euler scheme is $\hat{x}_0 = x_0$, then it is possible to proceed by time steps Δt of constant width. The discrete approximation $\{X_{1i}, X_{2i}, \dots, X_{mi}\}$ will run on the interval needed $[0; T]$, where *m* is the number of total steps and i denotes the length of each step so that $mi = T$. The approximation is the following.

$$\hat{X}(t_{i+1}) = \hat{X}(t_i) + m(\hat{X}(t_i); t_i)i + s(\hat{X}(t_i); t_i)\sqrt{i}W(t_{i+1})$$

Milstein scheme also takes into account Ito's Lemma and Taylor expansion, in fact Euler scheme does not consider any term of subsequent order. Milstein scheme also considers next order terms eliminating the inconsistency as follows (Brandimarte, 2014).

$$\hat{X}(t_{i+1}) = \hat{X}(t_i) + m\big(\hat{X}(t_i); t_i\big)i + s\big(\hat{X}(t_i); t_i\big)\sqrt{i}W(t_{i+1})$$
$$+\frac{1}{2}s'\big(\hat{X}(t_i); t_i\big)s\big(\hat{X}(t_i); t_i\big)i(W^2(t_{i+1}) - 1)$$

For the purposes of this analysis the simple Euler scheme will be used.

The third category of models, discrete event, are substantially continuous time models where the time step is not constant, but the state changes only when a certain event happens.

To conclude this introduction, it is important to underline how Monte Carlo simulation is an extremely powerful tool in finance and other fields, but it has some inefficiencies and technical issues related to sampling, variance of the estimator and computational time (Brandimarte, 2014).

2. Monte Carlo Integration

Monte Carlo methods become particularly useful when it comes to numerical integration. At the beginning of this chapter, it was given an example about the computation of an expected value. Recalling this example and knowing the frequency with which the expectation operator is used in financial problems, the usefulness of Monte Carlo in financial engineering is evident. It is important to underline that sometimes integrals can be solved in analytical or semi-analytical way, that may need complex computational derivation such as the stochastic averaging method, but on the other side often there are no analytical solutions (Dostal & Kreuzer, 2016). Taking as an example the Black Scholes Merton formula, the density function $\Phi(z)$ has a solution that does not have any analytical expression but there is still no need to go through complex numerical integration since there are sufficiently accurate ways to approximate the expression. These type of solutions are called semi-analytical (Judd, 1998). However, in other cases it is needed to go through numerical procedures such as Classical Quadrature formulas like The rectangle rule, the Interpolatory Quadrature formulas or the Gaussian Quadrature (Davis & Rabinowitz, 1984). Those deterministic approaches consist in approximating the value of the integral through interpolation based on polynomials or more sophisticated orthogonal polynomials. The above mentioned method are the most straightforward approaches to use when handling low dimensional integration problems. However, integration through Monte Carlo becomes useful or even necessary when considering high dimensional problems as illustrated in the following example (Lapeyre et al., 1998). Supposing having a vector random variable X with support Ξ, and the following integral at the core of the problem to solve:

$$\mathbb{E}[g(x)] = \int_{\Xi} g(x)f(x)dx$$

Quadrature formulas in this multidimensional case require an high number of points, and Monte Carlo integration is the most efficient solution. On the other end, sometimes Monte Carlo integration might require a significant sample size to reach a certain level of precision. This is the reason why Monte Carlo integration does not always result to be the best solution, but it is important to have in mind the link between these Methods and numerical integration that is now shortly illustrated.

Defining the random variable X, defined over the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the state space that is a set comprising all the possible outcomes of the random variable, $\mathcal{F}$ is the set of measurable events and $\mathbb{P}$ is the probability measure (Munk, 2013).

$$X: \Omega \rightarrow \mathbb{R}^d$$

The set of possible outcomes ω belonging to the state space has a correspondent X$(\omega) \in \mathbb{R}^d$, and X$(\omega)$ must be $\mathcal{F}$- measurable, namely it should be possible to assign a probability to the event X$(\omega) = x_i$. The total mass of the probability is equal to 1. The random variable is assigned with a probability distribution and a density function that allow to compute the expected value of X. In the continuous setting, as already emphasized, the expected value follows the formula:

$$\mathbb{E}(X) = \int_{\Omega} X(\omega)dP(\omega)$$

It is possible to define in the same way the expected value of g(X):

$$\mathbb{E}(g(X)) = \int_{\Omega} g(X(\omega)dP(\omega)$$

The two integrals above are with respect to the probability measure. Calling h_X the distribution of X under P it is possible to switch measure of the integral and write:

$$\mathbb{E}(g(X)) = \int_{\mathcal{F}} g(X(\omega)dh_X(\mathrm{x})$$

Monte Carlo integration starts from the volume of a set of outcomes to estimate the volume of the probability. Considering a vector of variables $x = u_1, u_2, \dots u_d$ and the function of it $f(x) = f(u_1, u_2, \dots u_d)$, square-integrable over the hypercube $[0;1]^d$(Brandimarte, 2006).

$$I = \int_{[0;1]^d} f(x)dx = \int_{[0;1]^d} f(u_1, u_2, \dots u_d)du_1 \dots du_d$$

Monte Carlo estimate of the above integral would be obtained by collecting an independent and identically distributed sample of points of x over the unit hypercube and computing $\hat{I}_n$:

$$\hat{I}_n = \frac{Vol([0;1]^d)}{n} \sum_{i=1}^{n} f(x_i)$$

The volume of the unit hypercube is equal to one, consequently taking the limit of the estimator for $n \to \infty$ gives the following result:

$$\lim_{n\to\infty} \left(\frac{1}{n} \sum_{i=1}^{n} f(x_i) \right) = I$$

When the number of simulation goes to infinity the estimator converges to the true value, the estimator is therefore unbiased. In the next paragraphs the criteria of choice between unbiased estimators will be presented, and additionally variance reduction techniques will be compared.

3. The efficiency of the estimator

In the previous example the limit shows that the estimator is unbiased and converges to the exact value of *I*. The sample points were independent and identically distributed. Applying the central limit theorem to the estimator, as *n* increases the standardized estimator converges to the standard normal distribution.

$$\frac{\hat{I}_n - I}{\sigma_I / \sqrt{n}} \Rightarrow N(0; 1)$$

Consequently, the error of the estimator $\varepsilon = \hat{I}_n - I$ has a distribution that can be computed by a simple manipulation of the above convergence.

$$\varepsilon \sim N(0; \frac{\sigma_I}{\sqrt{n}})$$

The convergence rate is then $O\left(\frac{1}{\sqrt{n}}\right)$, and it is possible to observe that the rate does not depend on the dimension of the integral. This feature is the central advantage of Monte Carlo methods. Other than comparing the bias of different estimators to evaluate their efficiency, estimators are also compared with respect to their computational time and variance. To choose between two unbiased estimators a criterion can be derived. Assuming that per each replication I_i it takes a computing time τ and the and s is the computational budget. The number of replication that are allowed by our budget will be s/τ . When the computational budget tends to infinity the following convergence is defined:

$$\sqrt{s/\tau}\,[\hat{I}_{\frac{s}{\tau}} - I] \Rightarrow N(0, \sigma_I^2)$$

And consequently,

$$\sqrt{s}\,[\hat{I}_{\frac{s}{\tau}} - I] \Rightarrow N(0, \sigma_I^2)$$

Therefore, the estimator will be also normally distributed with variance $\sigma_I^2\tau/s$. Now consider two estimators, both unbiased, with computing time per replication respectively equal to τ_1 and τ_2 and variance per replication equal to σ_1 and σ_2. The best estimator among the two will be the one that has the lowest value of $\tau_i\sigma_i$, because it will give a more accurate estimate and a smaller confidence interval (Glasserman, 2003).

4. Generation of random numbers

At this point, it should be clear that the central issue of the simulation is the generation of uniformly distributed random variables. The simulation and the sample are treated as they were completely random, but it is necessary to notice that the algorithms that generate random numbers on different programs are completely deterministic. However, their ability to simulate pseudorandom variables is good enough to treat the variables as authentically random. A generator of authentically random variables is a process to generate random variables which present two properties:

- each variable is uniformly distributed between 0 and 1;
- the variables are independent with respect to each other.

The second property implies that any *i*-th variable should be impossible to predict from the previous random variables. Accordingly, one should not be able to distinguish any trend among the variables. Some characteristics of a good generator are related to:

- *period length*, as generators with longer periods are better (i.e. generators that repeat themselves after a long number of steps);
- *replicability*, as it might be required to run a simulation with identical inputs;
- *speed*, as an high number of simulations is often needed the generator must be fast;
- *portability*, it should generate the same values on different computing platforms;
- *randomness* both from the theoretical properties perspective and statistical test perspective (Glasserman, 2003).

The simplest category of random number generators is the Linear congruential generators. Their logic is captured by the pure linear form below:

$$x_{i+1} = ax_i mod(m)$$
$$u_{i+1} = x_{i+1}/m$$

They might also take a semi-linear form:

$$x_{i+1} = (ax_i + c)mod(m)$$

Where a, m and c are integer and constants, and the random number generated is determined by these parameters given an x_0 that is the initial value, also called *seed*. With some constraints to these numbers the generator is ensured to have full period (i.e.: it start repeating its sequence only after m-1 steps). The conditions imply that c and m should not have any common divisor except 1, every prime number that divides m should also divide $a-1$, and if m is divisible by 4 also $a-1$ should be (Lehmer, 1949).

Specifically, if $c = 0$ the generator has full period for $x_0 \neq 0$, if $a^{m-1} - 1$ is a multiple of m, and $a^j - 1$ is not a multiple of m for any $i = 1, 2 \dots m - 2$ (Niederreiter, 1992). In this case a is a primitive root of m, and it is ensured that the structure will not return x_0 until $a^{m-1}x_0$. Moreover, when a is a primitive root of m it is possible to demonstrate that if the seed is different from zero also all the generated numbers will be different from zero, and this is important because if one of the generated number is zero all the subsequent numbers will be zero. The semi-linear form of the linear congruential generator, when c is different from zero, is demonstrated to add few generality to the model and slow the model down (Marsaglia, 1972).

When plotting consecutive random numbers generated by the linear congruential generator, it is possible to observe the so called lattice structure that distinguish them from authentically random numbers (Figure 1).

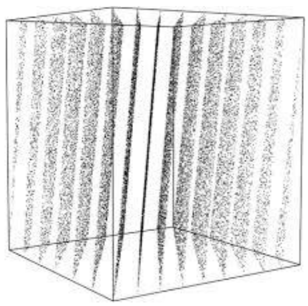


Figure 1: Lattice Structure Hyperplanes (Aljahdali & Mascagni, 2017)

In fact, consecutive outputs of the generator lies on parallel hyperplanes of the d-dimensional unit cube (Marsaglia, 1968). This structure can be used to compare outputs and choose parameters of the generator. The spectral test is one of the most famous way to analyze if the points in the lattice are equally distributed, specifically, it takes into account the distance of adjacent hyperplanes and take the maximum of this measure over all the parallel hyperplanes, providing a measure of how much the points are uniformly distributed.

Another way to obtain random numbers is to combine different linear congruential generator by summing them. This technique allows to keep the qualitative characteristics of linear congruential generators by slightly reducing the lattice structure.
Considering a number J of linear congruential generators and their sets:

$$x_{j,i+1} = a_j x_{j,i} mod(m_j), \qquad u_{j,i+1} = x_{j,i+1}/m_j$$

For Wichmann and Hill combined generator the combined set is obtained by summing the linear generators $u_{i+1} = u_{1,i+1} + u_{2,i+1} + \cdots u_{J,i+1}$.Alternatively, L'Ecruyer method firstly develops the sum and then obtain the combined set, as follows:

$$x_{i+1} = \sum_{j=1}^{J} -1^{j-1} x_{j,i+1}\, mod(m_1 - 1),$$

$$u_{i+1} = \begin{cases} \frac{x_i}{m_1}, & x_{i+1} > 0 \\ (m_1 - 1)/m_1, & x_{i+1} = 0 \end{cases}$$

Alternative methods to generate random numbers may include linear recursion. Assuming to have available an ideal sequence of random variables a simulation can use this as an input to generate sample stochastic paths, transforming the available sample uniformly distributed into a sample that has another distribution. The *inverse-transform* method and the *acceptance rejection* methods are among the most widespread techniques. The aim of the former is to generate random variables that follow a given cumulative distribution function $F(x)$. The function is non-negative, non-decreasing and continuous between $[0; 1]$. It is possible therefore to compute its inverse function. Starting from a percentile level this methods can bring to the associated value of the random variable until it is possible to compute an explicit formula for the inverse function.
The *acceptance rejection* method firstly generates random variables that do not follow the target cumulative distribution, but uses a more convenient distribution to do so. Afterwards variables belonging to a subsample of the obtained set is rejected. The rejection criteria is built so that the remaining sample follows the target. This method is highly applicable (Glasserman, 2003).

5. Variance reduction techniques

As already mentioned, to evaluate the efficiency of a model one must take into account both the variance and the computational time per replication of the estimator. Therefore, multiple methods were developed to reduce the variance of the estimators in order to obtain a higher level of efficiency. The most straightforward method to reduce the variance of the estimator is surely by increasing the number of simulations. However this approach is not reasonable considering the computational efficiency of the algorithm. Variance reduction techniques can be classified in two main categories: the first branch tries to exploit tractable characteristic of the model to adjust the output; while the second one tries to curtail the volatility of the inputs. In this paragraph I will go through the most common ones, that are Control Variates method, Antithetic Variates method and Stratified Sampling method.

The method of *Control Variates* falls under the first category. It consists in using information available of a simpler variable to reduce the variance of the estimator. Considering $Y_1, Y_2, Y_3, Y_4 \dots Y_n$ as outputs of n simulations, and another type of output $X_1, X_2, X_3, X_4 \dots X_n$, that has known expectation. Supposing that the pair made of the two variables are independent and identically distributed, the method exploits the deterministic information available of the other variable to reduce the variance. Constructing the i-th Y as follows:

$$Y_i(b) = Y_i - b(X_i - E[X])$$

The known error of X will serve as "control" of the variable Y. The estimate of $Y(b)$ is still unbiased with variance equal to:

$$Var[Y_i(b)] = Var[Y_i - b(X_i - E[X])] = \sigma_Y^2 - 2b\sigma_Y\sigma_X\rho_{XY} + b^2\sigma_X^2$$

Therefore if $b^2\sigma_X < 2b\sigma_Y\rho_{XY}$ the variance of this estimator will be lower. The parameter b can be chosen to minimize the variance of this new estimator, the optimal choice is given by:

$$b^* = \frac{\sigma_Y}{\sigma_X}\rho_{XY}$$

Substituting this optimal value it is possible to find that the ratio of $\frac{Var[Y_i - b(X_i - E[X])]}{Var[Y_i(b)]}$ is equal to $1 - \rho_{XY}^2$. Hence, it is possible to conclude that the effectiveness of the reduction depends on the correlation between the output variables X and Y. Yet, in practice the covariance between the variables is rarely known and is estimated through a least-squares regression. A substantial improvement to Control Variates is given by non-linear Control Variates, especially for small sample (Perninge et al., 2008).

Another widespread method is the *Antithetic Variates* one. It is based on the assumption that if the output variable U is uniformly distributed between $[0; 1]$, also $1 - U$ will be uniformly distributed in the same interval. In particular the variance reduction is implemented by pairing sequences of Brownian increments $Z_1, Z_2, Z_3 \dots\ Z_n$ and increments of the path's reflection with respect to the origin $-Z_1, -\ Z_2, -\ Z_3 \dots -\ Z_n$. If the objective of the simulation is to obtain an estimate of $E[Y]$ and pairs of antithetic observations $(Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2) \dots\ (Y_n, \tilde{Y}_n)$, where each pair is independent and identically distributed and each Y_i has the same distribution of its antithetic. The antithetic estimator will simply be the average of all the variables.

$$\hat{Y}_{AV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i + \tilde{Y}_i}{2}\right)$$

Assuming that the computational time needed to generate the antithetic random variables is exactly doubled it can be concluded that the method is effective in reducing the variance when

$$Var[\hat{Y}_{AV}] < Var\left[\frac{1}{2n}\sum_{i=1}^{2n} Y_i\right].$$

This means that a negative dependence of each $Y_i\ and\ \tilde{Y}_i$ is needed so that $Cov(Y_i, \tilde{Y}_i) < 0$ (Glasserman, 2003).

The last type of approach is stratified sampling, that selects subsets of the input sample space to make them more regular. The sample space is divided in *m* partitions $\alpha_1, \alpha_1 \dots\ \alpha_m$ called strata. Afterwards, a number *m* Monte Carlo simulations are performed, constraining the random variable X to be on $\alpha_1, \alpha_1 \dots\ \alpha_m$. Neyman allocation can be used to choose the variables over each partition to minimize the variance (Perninge et al., 2008). The multidimensional extension of this sampling method is called Latin Hypercube Sampling, although in principle the multidimensional application is feasible, it is in practice often inapplicable. The optimal choice of the appropriate technique to reduce the variance is crucial, and it depends on the specific application. Furthermore, combining different techniques might be the optimal choice in some cases.

6. Monte Carlo simulation to price Swaptions

In conclusion, Monte Carlo methods have an high level of flexibility, and their use in finance is necessary when considering more complex derivatives. Specifically, when considering fixed income derivatives the use of the simulation allows to describe the true dynamics of interest rates incorporating also the instantaneous correlation between the multiple rates that might be needed to price a more complex derivative. A correct calibration of the model in use will also give dynamics that are coherent with some observed market data. Monte Carlo standard error can be reduced with the application of variance reduction techniques.

The next chapter will be fully dedicated to explain the implementation of the pricing algorithm for swaptions. Under the T-Forward-neutral measure I will generate m different realization of the k different Forward rates needed to compute each payoff of the option, that is the Swap expected future price. The following forward rates are simulated to obtain the value of the corresponding forward rates observed at the exercise date T_O, that is equal to the maturity of the forward rate chosen as numeraire T_α.

$$F_{\alpha+1}(T_\alpha), F_{\alpha+2}(T_\alpha), \ldots, F_\beta(T_\alpha)$$

The Forward rates trajectories will follow the stochastic process implied by the LMM. Since the mentioned dynamics do not have a distribution that is known, it is necessary to discretize the process. The latter, with some algebraic manipulations, will yield an equation with deterministic diffusion coefficient for which Euler discretization scheme correspond to Milstein one. In this case The simulated forward rates will be used as input for the computation of the payoff. Each one of the m simulation will lead to a different future Swap value, and the average of their discounted prices will be the Monte Carlo estimate (Brigo & Mercurio, 2001b).

Chapter 3

Swaptions pricing

1. European Swaptions and their use

In this paragraph the structure and the use of a swaption will be explained in detail. A swaption contract is a contract that gives its buyer the right to enter the underlying Swap with a given strike swap rate at the exercise date. The underlying contract can be a payer or a receiver Swap, that is a swap that pays or receive the fixed swap rate against a floating rate determined at the previous payment date depending on the reset frequency. The floating rate is determined in the contract and can be indexed to one of the most famous benchmark rates, such as LIBOR, EURIBOR and EONIA rates. Due to recent European regulation, a transition to new risk free indices is taking place. The new risk free benchmark to be used for the Eurozone is defined in Regulation (EU) 2016/1011 and it is called ESTER, European Short Term Rate. Therefore, the old rates are no more compliant with new regulations and all the entities exposed to those rate must transition to the new ones (Zaegel et al., 2019).

Swaption contracts are useful instruments of insurance and hedging against any interest rates rise or fall (Akume et al., 2003). For instance, the issuer of a floating rate bond might want to be protected against a rise in interest rates in the future. She can buy a payer swaption with strike rate r_K, that will be the swap rate to be payed if the option is exercised. If at the exercise date the interest rates are higher and, consequently, the swap strike rate is lower than the current swap rate implied by current market condition, then the option can be exercised. The gain will be equal to the difference between the two swap rates: the strike and the current swap rate.

In the same manner a swaption can be useful to an investor of a callable security. In fact, with this type of instrument in case of a fall in interest rates the issuer will be able to obtain funding at cheaper rates, and for this reason will call back the security. The investor would receive her capital earlier than maturity with the possibility to reinvest the capital at the current market

conditions, namely lower rates. To hedge this risk the investor can buy a receiver swaption to exercise in case interest rates drop. Thus, the investor will receive the fixed strike rate and pay a floating rate, that can be offset by reinvesting the capital received at the floating rate.
The pricing process for the swaption implies the computation of all the payoffs at maturity, that are themselves based on the pricing of the Swap at each final node. To price a Swap at a reset date, the key observation is that at every reset date the floating leg will be at par, therefore it is sufficient to compute the price of the bond corresponding to the fixed leg and subtract the notional. The setting becomes more complicated when pricing a Swap contract at a future date, because the different future discount factors must be computed.
The option that will be priced in this analysis is a receiver swaption with floating rate equal to the LIBOR rate. The valuation date is the 1st of September 2020, the exercise date is in one year, the underlying Swap tenor, i.e. the length of the swap contract, is four years and the contract has a semi-annual reset frequency. The analysis will be conducted on different parameters of the strike swap rate. For the sake of pricing the number of forward rates to simulate is eight. The next paragraph will highlight the reason behind the choice of the model used to simulate and the difference between the use of different models.

2. The choice of the model

The LIBOR Market Model was introduced at the end of the first chapter, and it is built around the change in the probability measure. The use of the T-Forward rate as numeraire allows to characterize the process of forward LIBOR rates. The Swap Market Model instead uses a different numeraire, that is equal to the sum of multiple discount factors, so that it is possible to characterize the process of forward Swap rates (Veronesi, 2005a). To understand the importance of the introduction of Market Models, and which one should be chosen to price the swaption introduced in the previous paragraph, it is useful to remind that until those models were introduced Short Rates model were the main choice to price interest rates derivatives. Short Rates Models introduced in the first chapter describe the instantaneous interest rate through the definition of a diffusion process, the latter stochastic process explains the evolution of the yield curve and gives as outcome an analytical formula for forward LIBOR rates. The interest rates derivatives can be priced using these inputs and computing a risk neutral expectation, assuming a deterministic discount factor equal to de ZCB price. This last assumption generates inconsistencies in the pricing process. Those inconsistencies can be solved with the use of a different probability measure instead of the traditional risk neutral one.

The main advantage of Market Models introduction is that, unlike Short Rate models, they are compatible with the widespread Black's market formula. In fact, LMM can correctly price caps and SMM can correctly price swaptions consistently with the corresponding Black's formulas. While Short Rate models cannot be compatible with the distribution implied by the Black's formula for any θ_t (Formula 4.0). The only way to calibrate short term models is to choose the parameters that match both initial term structure and observed cap prices. This calibrating process however is way more complex than choosing the LIBOR Market Model and focusing on the instantaneous volatility parameter. Nonetheless, this model is only compatible with a part of interest rates derivatives market, that includes caps, while SMM is compatible with the other part, that is made by swaptions. The incompatibility of the two models forces to choose one of the two for the entire market of interest rates derivatives, because a joint lognormal assumption would be inconsistent. Forward rates are considered to be more explanatory about the yield curve, and taking into account the fact that the LIBOR Market model is easier to handle from a mathematical point of view, it is more smooth to use the numeraire implied by the LMM also for swaption market instead of doing the opposite (A. Brace et al., 1998). Therefore, the simulation to price the swaption will be based on the LIBOR Market Model.
The complexity of swaptions pricing with the use of the latter relies in the fact that the future swap price depends on multiple forward rates, specifically, for the option considered in this implementation there are eight different forward rates involved in the computation. When taking expectation of the payoff, unlike caps, those forward rates are not independent from each other and their joint distribution influences the final price. Different instantaneous volatility structure and consequently different correlation structure can be used. In the next paragraph the process used in the simulation will be illustrated and It will be shown how previous forward rates enter in the drift of subsequent ones, determining their path to an extent determined by their correlation (Brigo & Mercurio, 2001b). The stochastic process derived for this implementation relies on a basic assumption regarding the structure of instantaneous correlations on which the model is calibrated, that allows to simplify the more generic process of the LIBOR Market Model.

3. Model Calibration

The first step in the direction of running the entire simulation in order to obtain the actual price is to determine the input data and calibrate the model.

The specific discretized process that will be used to simulate the trajectories of the forward rates will be explained in details in the next paragraph, while now I will focus solely on the steps needed to be able to simulate those trajectories.

First and foremost, as clarified in the previous paragraph, the model matches the Black's formula for caps, and the implied volatilities in the market prices of those caps will be the input of the model[1]. Therefore, the calibration to caps and floors volatility surface is straightforward. The quotes semi-annual for at-the-money caps on LIBOR with six months to six years maturity were extracted from Thomson Reuters in terms of implied volatilities. To obtain the corresponding dollar price it was used as an input the quoted implied volatility in the Black's closed formula for caps. Additionally, the latter formula requires LIBOR based discount factors to be implemented.

Therefore, LIBOR rates corresponding to discrete points in time were retrieved from Bloomberg, and the transition to continuous time LIBOR based yield curve required a Cubic Spline Interpolation, that is one of the most widespread method. The result obtained is showed in Figure 2.

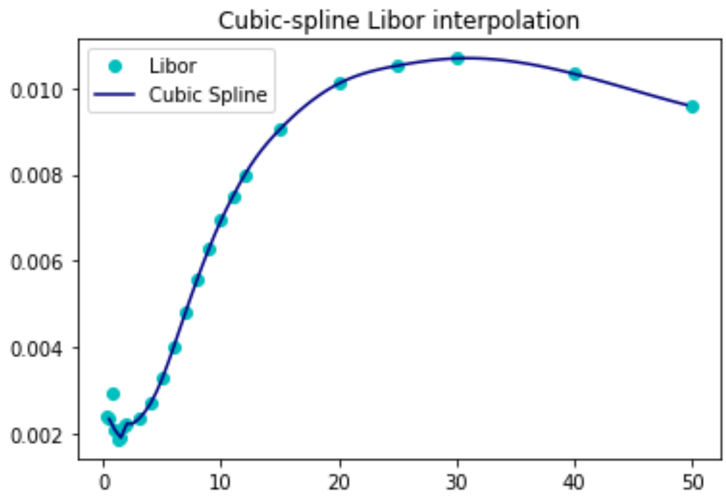


Figure 2: Cubic Spline LIBOR Interpolation

Once those input for the Black formula were defined, a function to extract dollar prices of at the money caps was constructed in Python and with the use of a stripping algorithm the forward volatility structure was extracted. The forward volatility structure can be used to compute the volatility structure that is the input of this model, as it was explained at the end of chapter 1.

[1] Caps quotes are found in term of implied volatilities. However, considering the current transitional market situation to the new risk-free rate ESTER, some maturities may be characterized by temporarily illiquidity. Therefore, for the less liquid maturities the implied volatility might correspond to older market status quo. Due to this temporal misalignment, for one value that was incoherent with the whole volatility structure, linear interpolation was used to balance this bias. (Brigo & Mercurio, 2001a)

The instantaneous volatility surface at the date of the calibration, September 1 2020, is illustrated in figure 3

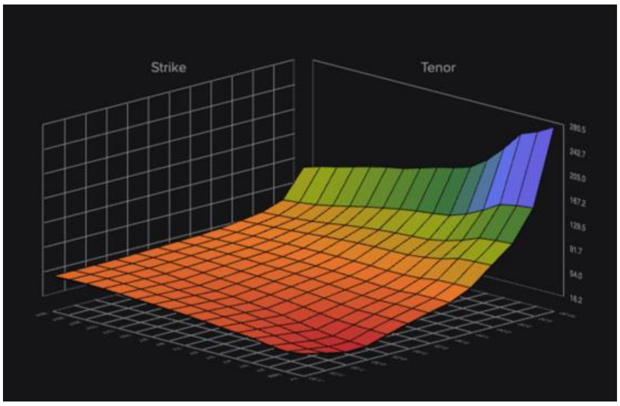


Figure 3: Market Volatility Surface (Bloomberg)

Figure 4 shows a comparison between the three different volatility structure obtained, flat volatilities extracted from at-the-money caps, forward volatilities and volatilities of the forward rates. When the flat volatility is upward shaped the forward volatility curve lies above the former, and in the same way the semi-linear volatilities S will be above the forward volatility curve.

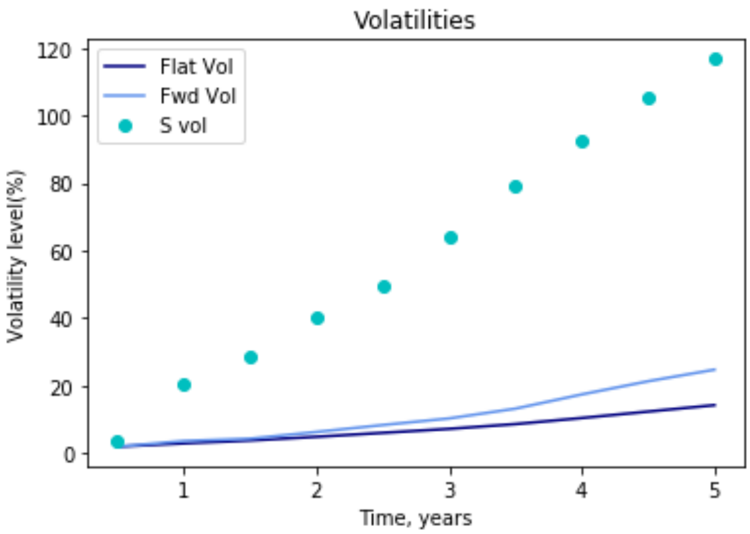


Figure 4: Volatility Structures

Considering this calibration the next step is to simulate forward rates trajectories, that will be explained in the next paragraph.

4. Derivation of the specific stochastic process used in Monte Carlo simulation

Under the forward measure, solving the stochastic differential equation yields the process that can be used to run a Monte Carlo simulation, as introduced in the first chapter. Nonetheless, there is a specific assumption to be made to be able to simulate forward rates without the use of an instantaneous correlation matrix. In fact, from Brigo & Mercurio, the generic stochastic process obtained in the LIBOR Market Model is the following for any $T_i < T_k$, where T_k is the option exercise date.

$$\frac{df_n(t,T_i,T_{i+1})}{f_n(t,T_i,T_{i+1})} = \left(\sum_{j=\bar{\imath}}^{i} \frac{\Delta f_n(t,T_j,T_{j+1})\sigma_f^{i+1}(t)\sigma_f^{j+1}(t)\rho_{i+1,j+1}}{1+\Delta f_n(t,T_j,T_{j+1})}\right)dt - \frac{\sigma_f^{i+1}(t)^2}{2} + \sigma_f^{i+1}(t)dX_i(t)$$

Where $\rho_{i+1,j+1}$ is the correlation element, for which it would be needed a *correlation structure* between the forward rates.

By assuming that $dX_i(t) = dX_t$, we assume that all forward rates have the exact same Brownian Increment, or namely, that all the forward rates for different maturities are perfectly correlated. This assumption, whilst simplistic and not coherent with true market dynamics is necessary to make the implementation feasible, and still allows to compute a representative result of the swaption price.

The derivation of the stochastic process under the forward probability measure and with perfect correlation between the different rates is now shown.

Considering the dynamics of a variable Y induced by $Z(0,T_{i+1})$ and $Z(0,T_{\bar{\imath}})$, under the forward probability measure.

$$dY_t = \left(m^*(Y,t) + \sigma_{Z,T_{i+1}}(t)s(Y,t)\right)dt + s(Y,t)dX_t \qquad 3.1$$

$$dY_t = \left(m^*(Y,t) + \sigma_{Z,T_{\bar{\imath}}}(t)s(Y,t)\right)dt + s(Y,t)dX_t \qquad 3.2$$

Where $\sigma_{Z,T_{\bar{\imath}}}$ and $\sigma_{Z,T_{i+1}}$ are the diffusion terms of the securities, which themselves follow the risk neutral process below.

$$\frac{dZ(t,T)}{Z(t,T)} = rdt + \sigma_{Z,T}(t)dX_t$$

Taking the difference between equations 3.2 and 3.1 it is possible to derive the change in drift showed below, which is the change ΔdY_t that should be taken into account when moving from the dynamics implied by the first security to the dynamics implied by the second security.

$$\Delta dY_t = (\sigma_{Z,T_{\bar{\iota}}} - \sigma_{Z,T_{i+1}})$$

The variable we are considering in this model is the forward rate, whose dynamics are implied by the numeraire $Z(T_i, T_{i+1})$.

$$df_n(t, T_i, T_{i+1}) = \sigma_f^{i+1} f_n(t, T_i, T_{i+1}) dX_t$$

Considering that it is possible to write:

$$\frac{Z(0, T_{i+1})}{Z(0, T_{\bar{\iota}})} = \frac{1}{1 + \Delta f_n(t, T_{\bar{\iota}}, T_{\bar{\iota}+1})} \cdot \frac{1}{1 + \Delta f_n(t, T_{\bar{\iota}+1}, T_{\bar{\iota}+2})} \cdot \cdots \frac{1}{1 + \Delta f_n(t, T_i, T_{i+1})}$$

And consequently, taking logs we obtain the following equation.

$$\log\left(\frac{Z(0, T_{i+1})}{Z(0, T_{\bar{\iota}})}\right) = -\sum_{j=\bar{\iota}}^{i} log(1 + \Delta f_n\big(t, T_j, T_{j+1}\big))$$

Applying Ito’s lemma allows to see that the diffusion term of the right hand side of the equation above is equal to:

$$\sum_{j=\bar{\iota}}^{i} \frac{1}{1 + \Delta f\big(t, T_j, T_{j+1}\big)} \Delta\sigma^{j+1}(t) f\big(t, T_j, T_{j+1}\big)$$

Since this must also be the diffusion term of the left hand side of the equation, we obtain:

$$\big(\sigma_{Z,T_{\bar{\iota}}} - \sigma_{Z,T_{i+1}}\big) = \sum_{j=\bar{\iota}}^{i} \frac{1}{1 + \Delta f\big(t, T_j, T_{j+1}\big)} \Delta\sigma^{j+1}(t) f\big(t, T_j, T_{j+1}\big)$$

To conclude, the change in drift of the process of the forward rate should be $\big(\sigma_{Z,T_{\bar{\iota}}} - \sigma_{Z,T_{i+1}}\big) s(Y, t)$ which is equal to:

$$\left(\sum_{j=\bar{\iota}}^{i} \frac{1}{1 + \Delta f\big(t, T_j, T_{j+1}\big)} \Delta\sigma^{j+1}(t) f\big(t, T_j, T_{j+1}\big)\right) \sigma_f^{i+1} f_n(t, T_i, T_{i+1})$$

Which gives the equation below to use in the simulation of the forward rates. (Veronesi, 2005b)

$$\frac{df_n(t, T_i, T_{i+1})}{f_n(t, T_i, T_{i+1})} = \left(\sum_{j=\bar{\iota}}^{i} \frac{\Delta f_n(t, T_j, T_{j+1}) \sigma_f^{i+1}(t) \sigma_f^{j+1}(t)}{1 + \Delta f_n(t, T_j, T_{j+1})}\right) dt + \sigma_f^{i+1}(t) dX_t \qquad 3.3$$

5. Model implementation and results interpretation

To implement the pricing algorithm I started with equation 3.3, deriving the discretized process to which the simulation should be applied. The key step of the algorithm is to simulate random shocks, $\varepsilon_t \sim N(0,1)$ to be used in place of the Brownian increment of the stochastic process. The following scheme, derived with a logarithmic transformation, is the discretized implementation of equation 3.3.

$$f_n^m(t+\delta, T_i, T_{i+1}) = f_n^m(t, T_i, T_{i+1})e^{\mu_{i+1}^m(t)\delta + S(T_{i-1}-t)\sqrt{\delta}\varepsilon_t}$$

Where the drift μ is equal to:

$$\mu_{i+1}^m(t) = \sum_{j=\bar{\iota}}^{i} \frac{\Delta f_n^m(t, T_j, T_{j+1})S(T_{i+1}-t)S(T_{j+1}-t)}{1+\Delta f_n^m(t, T_j, T_{j+1})} - \frac{1}{2}S(T_{i+1}-t)^2$$

Where m indicates the specific simulation number of which the equations refer. I simulated ten thousand trajectories of the eight forward rates needed to price a swaption with exercise date the 1st of September 2021, which gives the right to enter in a receiver swap with 4 years tenor. The simulated forward rates will then be used to compute the corresponding discount factors with the following formula, where M is the maturity date of the underlying swap and $O = \bar{\iota}$ is the exercise date.

$$Z(T_o, T_M) = \frac{1}{1+\Delta r_n(T_{\bar{\iota}}, T_{\bar{\iota}+1})} \cdot \frac{1}{1+\Delta f_n(T_{\bar{\iota}}, T_{\bar{\iota}+1}, T_{\bar{\iota}+2})} \cdot \dots \frac{1}{1+\Delta f_n(T_{\bar{\iota}}, T_{\bar{\iota}+m-1}, T_{\bar{\iota}m})}$$

Each simulation was run through 252 steps, which is the notation for actual days in one year. The drift μ changes every step and for each one of the eight forward rates, retrieving a 252 rows and 8 columns matrix of drifts per each of the ten thousand simulations. The following sample graph shows the 888-th simultaneous simulation of the forward rates.

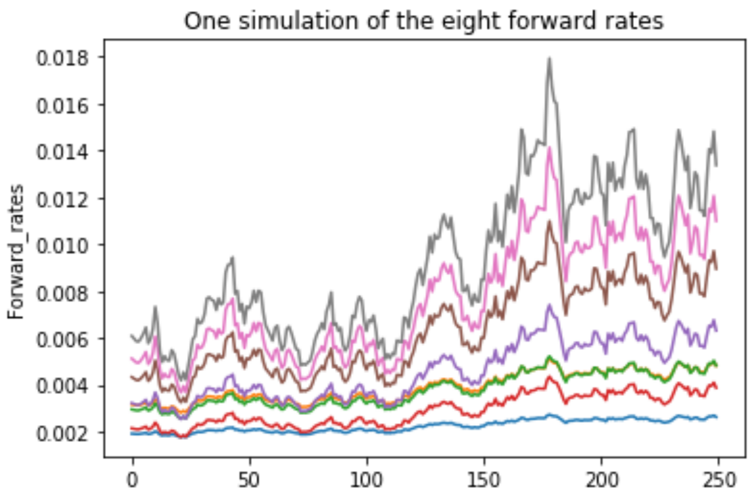


Figure 5: One Simulation of The Eight Forward Rates

The assumption of perfect correlation can be easily spotted observing the movements of the trajectories of the forward rates.

Figure 6 shows 10 sample simulations of the first forward rate. The simulation shows the dynamics of the forward $f(0{,}1y, 1.5y)$. The starting point is the forward rate observed the 1st of September 2020. Each step represents the value of the same forward rate observed one day later, until the last step at which the forward rate will converge to the future spot rate.

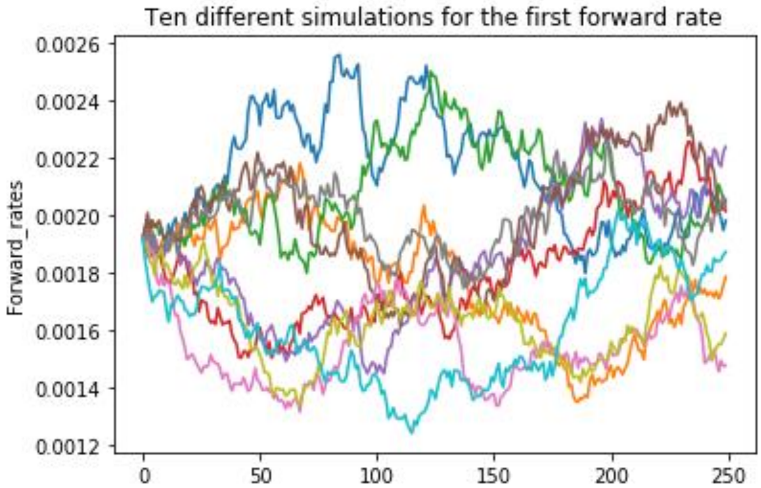


Figure 6: Ten Different MC Simulations for the First Forward Rate

Once the forward rates trajectories are estimated, the future swap price can be computed. Afterwards, is the maximum between the price and zero is estimated computing ten thousand different payoff. The last step consists in computing Monte Carlo estimator by discounting and computing an average of all the different payoffs, as follows.

$$\hat{V}^{Swaption} = \frac{1}{10000} \sum_{i=1}^{10000} Z(0{,}1) Payoff_{1\,year}^{i}$$

Where,

$$Payoff_{1year}^{i} = \max\left(\sum_{j=2}^{m} \Delta r_K Z(1{,}0.5(j+1)) + Z(1, m) - 1; 0 \right)$$

The prices of the European Swaptions with maturity 1 year were computed for different strike levels and different tenors, as it is illustrated Figure 7.

The result indicates that to higher levels of strikes correspond higher prices, and that for higher tenors, which means higher maturities of the underlying swap, the prices are higher for all the strike levels. This is because higher level of volatility correspond to higher tenors, and price and volatility are directly related.

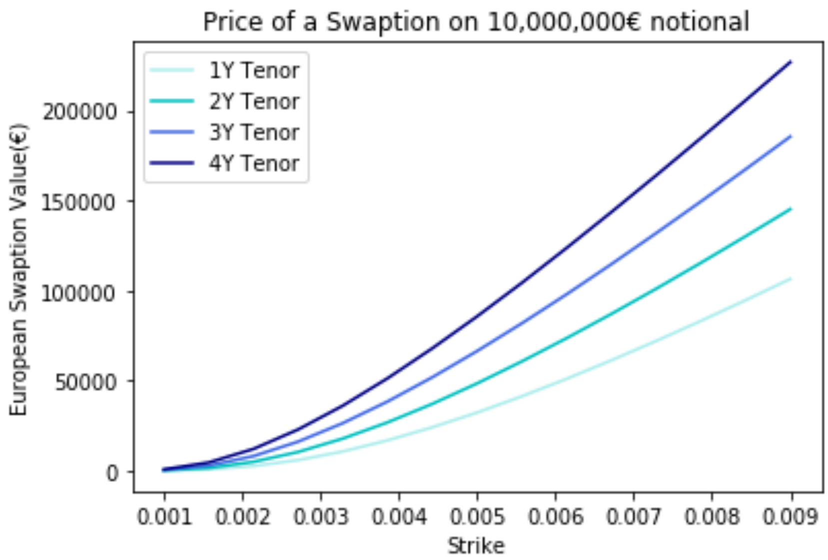


Figure 7: Swaption Prices Plot

The table below illustrates the obtained prices to whom Figure 7 refers, for different tenors (columns) and strike rates (rows).

Strike/Tenor	1Y	2Y	3Y	4Y
0.001000	€68,20	€186,09	€408,08	€765,76
0.001571	€705,38	€1.565,81	€2.855,28	€4.585,94
0.002143	€2.592,69	€4.954,48	€8.124,18	€12.104,65
0.002714	€5.926,76	€10.430,61	€16.165,86	€22.915,11
0.003286	€10.715,34	€17.879,88	€26.440,31	€36.148,70
0.003857	€16.868,27	€26.866,33	€38.455,09	€51.347,18
0.004429	€241.333,26	€37.126,48	€51.904,67	€67.977,89
0.005000	€32.299,64	€48.479,00	€66.442,30	€85.685,92
0.005571	€41.318,38	€60.682,09	€81.821,71	€104.280,37
0.006143	€51.009,71	€73.570,23	€97.921,11	€123.640,47
0.006714	€61.268,54	€87.021,17	€114.624,80	€143.535,88
0.007286	€719.999,87	€100.965,14	€131.823,50	€163.866,81
0.007857	€83.124,84	€115.361,20	€149.394,01	€184.556,36
0.008429	€94.621,74	€130.107,92	€167.281,44	€205.534,14
0.009000	€106.420,62	€145.132,59	€185.435,90	€226.757,99

Figure 8: Swaption Prices Table

To analyze in deep the obtained estimator I computed its variance, using the sample variance formula. The result confirm that the variance of the estimator is significantly high, especially for high tenor levels and high strike levels. In order to reduce the variance it is possible to apply one of the variance reduction techniques mentioned in paragraph 3.5. Estimator variances had been plotted in a 3-D graph (Figure 3.8) to show the relation with the parameters considered, which are tenor and strike. Variance values in the graph are scaled.

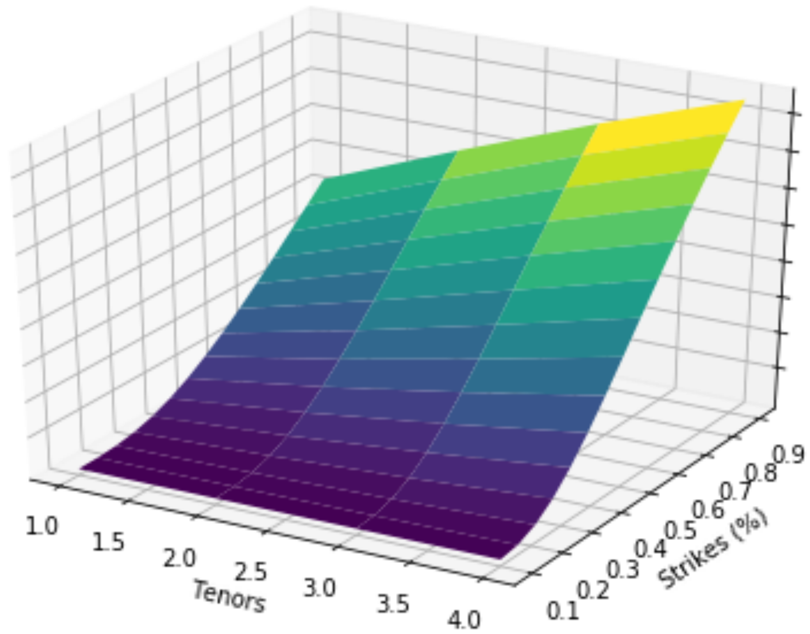


Figure 9: Monte Carlo Estimators Variances

Last but not least, to validate the solution given by the model I compared the values of ATM swaptions quoted on the market with the corresponding swaptions priced by the implemented model. Since swaptions are quoted on the market through Black's implied volatilities I extracted the dollar price by applying the Black formula (Monoyios & Hambly, 2017). The obtained results of the comparison are shown in Figure 10. The exercise date of the option is one year from the valuation date which is 1st of September 2020.

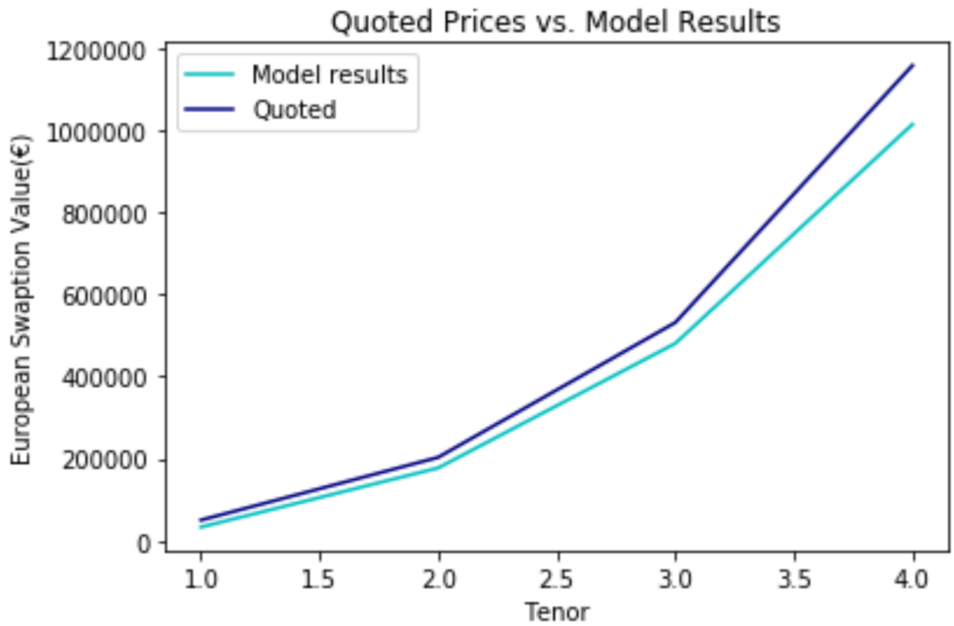


Figure 10: Monte Carlo Estimator Validation

The graph above shows the level of accuracy of the model. It is possible to see that for higher tenors the distance between the values increases, which means that the precision of the model decreases. The root mean squared error of the model is equal to 77613.19€. It must be underlined that the model tends to underprice the derivative for every tenor, and this might be due to a lack of convergence between the LMM and the SMM, since the model implemented is calibrated only on Cap volatilities.

6. Modern pricing techniques in the global markets' landscape

Structured products started to be transacted in the UK in the early 90s, and at the time those banks who had the chance to develop the most sophisticated mathematical models for pricing and structuring those products were one step ahead in the industry (Walker & Keohane, 2020). Derivatives had been used for hedging and speculation purposes and since then this market grew 24% per year on average, reaching €457 trillion of notional outstanding in 2008 (Mai, 2008). The crash of 2008 moved the markets in one negative direction, creating a feeling of distrust towards the derivatives and structured instruments markets. After the crisis, the spread and use of data science and machine learning techniques has given a significant turn to the reality of the financial markets, bringing the trading of these products towards increasingly sophisticated and automated processes (Dizard, 2019). In this context market models and the use of Monte Carlo simulation represented one of the most important steps given their ability to take into account current market variables such as the term structure and the implied volatilities quoted on the market for some specific instruments. It is known that the slope of the volatility and term structures are together good indicators of macroeconomic conditions and compensation for volatilities at specific maturities. The implementation of the model was showed and its accuracy was demonstrated in the previous paragraph. With some extensions the results obtained from the model can be used in multiple ways. In fact, fixed income derivatives by definition are highly connected with interest rates and consequently strongly influenced by monetary policy. This characteristic makes them good candidates to be used to interpret market sentiment and expectations with respect to macroeconomic news and business cycles (Fang et al., 2007). Swaptions together with Caps are among the most liquid derivatives of this category, and they are available on a broad range of interest rates.

Now there are new challenges for fixed income derivatives' pricing models, which are related to recent market changes. The first will be the transition to new European risk free benchmarks (Jones & Stafford, 2020). In fact, as new regulation is imposing new characteristics on these rates . Secondly, with the coronavirus outbreak many central banks implemented or started to consider a negative interest rates policy (Kochkodin, 2020). NIRP is known to be a good stimulus for the economy allowing for the propagation of monetary accommodation through the whole yield curve and avoiding further downward pressure on the term premium (Schnabel, 2020). However, from the perspective of fixed income derivative prices negative interest rates policy means that Black formula, used to quote securities in term of implied volatility, is no longer applicable. The model implemented in this research does not allow either changes in the sign of forward rates in the simulation. In the next paragraph I will briefly go through direction in which the research of this paper can be extended.

7. Suggestions for future researches: Bermudian Swaptions, Longstaff-Schwartz algorithm and Least Squares Monte Carlo

Further steps forward in this research could be done considering a Bermudian swaption, that by definition can be exercised earlier than the option maturity at pre-determined dates. Since Monte Carlo simulation by construction is an algorithm that only goes forward in time while usually Bermudian and American derivatives are priced backwards and, additionally, it is needed to simultaneously price the swaption at different points in time and compute its continuation value and compare them in order to define an optimal stopping frontier. The early exercise possibility adds challenges to the pricing and there are multiples approaches to address these issues, such as the Stochastic Tree Method, the Stochastic Mesh Method and the Longstaff-Schwartz algorithm (LSM). In this paragraph the last method will be briefly introduced as it represents a possible extension of the implementation carried in this paper.
Least Squares Monte Carlo method simply estimates the continuation using cross-sectional information from the simulation with a least squares regression. The algorithm works backwards and at each exercise date it compares the continuation value with the swaption value, and in the case the former is smaller than the latter the option is exercised immediately. The unbiased estimate of the continuation value is obtained as a function of a number k of basis functions, which are usually quadratic, and depend themselves on the underlying variable. Starting from the last step of the simulation the algorithm begins a recursion scheme that works backwards and where each continuation value is estimated by the fitted values from the regression. The recursion breaks when the first exercise date possible is reached (Longstaff & Schwartz, 2001). The convergence and the robustness of the algorithm were addressed by a

series of more recent studies that show their dependence on the number of basis functions used and the specific type of derivative instrument taken into account (Frey, 2008).

Conclusions

The objective of this research was to implement an algorithm to price European Swaption taking into account current values of volatility and term structures. Thanks to the change of the probability measure it was possible to overcome the problems related to the correlation between payoffs and the discount factors, and the inconsistent assumption of a deterministic discount factor under the traditional risk-neutral measure. The adoption of the T-forward measure opens a range of possibilities concerning the use of Market Models. The latter models, unlike Short Rate models, are compatible with the distribution implied by Black's formula, and can therefore be calibrated both on the term structure and on caps and swaptions prices quoted on the market. LIBOR Market Model and Swap Market Model and the inconsistency of their simultaneous use forced us to choose one model for the whole interest rates derivatives market. The evidence extracted from previous studies showed the convenience of adopting the LIBOR Market Model because of its mathematical tractability and the intuition in using forward rates to explain the term structure.

Furthermore, thanks to the adoption of Monte Carlo methods, the true dynamics of interest rates can be simulated including an instantaneous correlation structure between the multiple rates, that is needed when pricing more complex derivatives like swaptions.

After an appropriate calibration of the input parameters on the current market conditions of term and volatility structures, the implementation of the algorithm generated ten thousand trajectories per each simulation of the different forward rates, needed to price swaptions with different tenors. The swaption analyzed had a maturity of one year and the price was computed for different values of the tenor, with the need to generate up to 8 rates simultaneously.

The swaption price estimator was computed for different strike rates and tenors, and the variance of the estimator was investigated, observing an increase for higher values of tenor and strike rate.

To validate the model the obtained prices for at-the-money swaptions were compared with quoted prices of the corresponding swaptions. The validation showed the accuracy of the model and its implementation, and the Mean Squared Error was computed giving a satisfactory result that is illustrated. From the comparison of obtained and quoted prices, it can be observed that the model tend to slightly underprice the swaption for every tenor, and when the tenor increases the amount of the underpricing increases as well. This might be due to a lack of convergence between the LIBOR Market Model and the Swap Market Model, a matter of internal coherence which finds its roots in the calibration on caps' volatilities.

Moreover, the model presents some limitations related to the assumption of a perfect correlation between the different forward rates. In fact, each simulation was performed using the same random numbers in place of the Brownian increments, which implies a perfect correlation between the different rates, that are subject to the same random shocks. This simplification was fundamental to avoid the need to derive an instantaneous correlation structure on which the model should have been further calibrated. The complexity in defining such a structure represents one of the different directions in which the implementation can be extended.

For further exploration of the topic, the analysis could be also carried out for a Bermudian or American swaption with the possibility of early exercise, using the Longstaff-Schwartz algorithm and Least Squares Monte Carlo method.

The current landscape of global markets introduces new challenges for pricing algorithms concerning the input term structure, which is globally going towards strong reductions of interest rates, that are very close to zero or negative. Furthermore, the transition to the new European risk-free benchmarks represents another challenge for interest-rate derivatives market. Given the strong connection, explained in the last chapter, between those instruments and interest rate risks, monetary policy, and business cycles, the centrality of modern pricing models that are able to adapt to rapid changes is undoubted.

Appendix – Python code

```
"""
Created: Thu Aug 20 17:57:31 2020
Last Modified: Thu Sep 24 2020

This code contains an algorithm to price, at valuation date 1st of
September 2020,
a Swaption with maturity one year for different tenors. The file is
divided in
different parts, firs the data are imported from excel. Some necessary
pricing
functions based on the Black's formula are created and tested. Afterwards
the data
are cleaned manipulated and adjusted. The Forward volatility structures is
derived through a bootstrapping procedure. Random numbers trajectories are
generated
to be used in the Monte Carlo Simulation for forward rates. Discount rates
are derived
and the swaptions are priced and plotted. The model is validated with
quoted swaptions
prices.

@author: Laura Bruno
"""
import numpy as np
import pandas as pd
import scipy.stats
from scipy.optimize import minimize
import matplotlib.pyplot as plt
import scipy
import math
import sympy as sp
import pdb

##########################################################################
"Import excel file"
DF1=pd.read_excel('/Users/Laura/Desktop/Data
thesis/LIBOR_and_VOLATILITY.xlsx',
sheet_name = "Volatility")
DF1=pd.DataFrame(DF1)
DF2=pd.read_excel('/Users/Laura/Desktop/Data
thesis/LIBOR_and_VOLATILITY.xlsx',
sheet_name = "Libor")

##########################################################################
```

```
"Create arrays with volatilities, strikes, discount factors"
VOLdf=DF1.iloc[0:11,0:3]
Cap_Strikes=np.asarray(VOLdf.drop(columns=["Maturity","Flat
Volatility"]))/100
Flat_vol=np.asarray(VOLdf.drop(columns =["Maturity", "Strike"]))/10000
Maturities=np.asarray(DF2.drop(columns =["Scad", "Tasso mercato",
"Sconto", "Spos", "Ts spostato", "Tasso zero"]))
Spot_Libor=np.asarray(DF2.drop(columns =["Scad", "maturity (YR)",
"Sconto", "Spos", "Ts spostato", "Tasso zero"]))/100
Discount=np.asarray(DF2.drop(columns =["Scad", "Tasso mercato", "maturity
(YR)", "Spos", "Ts spostato", "Tasso zero"]))
#print(Maturities)
#print(Discount)
#print(Spot_Libor)
#print(Cap_Strikes)
#print(Flat_vol)

##########################################################################
"Libor and Discount factors interpolation, periodical Volatility"

f=scipy.interpolate.splrep(Maturities, Spot_Libor)
Maturities_new=np.arange(0.5,6, 0.5)
Libor_new= np.asarray(scipy.interpolate.splev(Maturities_new, f, der=0))
#print(Libor_new)

"""
#Plot the libor interpolation
plt.figure()
plt.plot(Maturities, Spot_Libor, 'co', Maturities_new, Libor_new, "navy")
plt.legend(['Libor', 'Cubic Spline', 'True'])
plt.title('Cubic-spline Libor interpolation')
plt.show()
"""
f1=scipy.interpolate.splrep(Maturities, Discount)
Maturities_new=np.arange(0.5,6.5, 0.5) #Nb: I need an extra discount
factor
Discount_new= np.asarray(scipy.interpolate.splev(Maturities_new, f1,
der=0))

#VOLATILITY *SQRT(DAYS)
Flat_volP=np.empty([11,1]) #Volatility relative to the period
for index in range(11):
    V=Flat_vol[index]
    PeriodV=V*math.sqrt(252*Maturities_new[index])
    print(PeriodV)
    Flat_volP[index]=PeriodV
```

```
##########################################################################
"Function for Caplet and Cap dollar price using black formula"
#Creation of pricing functions for Caplets and Caps

def Caplet_fun(Ti,sigma, rK, delta_step, Notional, discount0_Ti,
discount0_Ti1):
    forward_rateTiTi1=discount0_Ti1/discount0_Ti #forward rate between Ti
and Ti+1
    forward_rateTiTi1=(1/delta_step)*((1/forward_rateTiTi1)-1) #Correct
compounding fwd rate

d1=math.log(forward_rateTiTi1/rK)/(sigma*math.sqrt(Ti))+0.5*sigma*math.sqr
t(Ti)
    d2=d1-sigma*math.sqrt(Ti)
    Nd1=scipy.stats.norm(0,1).cdf(d1) #density function of d1 and d2
    Nd2=scipy.stats.norm(0,1).cdf(d2)
    dollarprice=
(Notional*delta_step*discount0_Ti1)*(forward_rateTiTi1*Nd1-rK*Nd2)
    return dollarprice

def CAP(Tn, sigma, rK, delta_step, Notional, Discount_vector):
    Cap=0
    t=0
    for i in np.arange(int(Tn/delta_step)):
        t=delta_step*(i+1)
        Cap+= Caplet_fun(t, sigma, rK, delta_step, Notional,
Discount_vector[i], Discount_vector[i+1])
    return Cap

def ATMCaplet_fun(Ti,sigma, rK, delta_step, Notional, discount0_Ti1):
    forward_rateTiTi1 = rK

d1=math.log(forward_rateTiTi1/rK)/(sigma*math.sqrt(Ti))+0.5*sigma*math.sqr
t(Ti)
    d2=d1-sigma*math.sqrt(Ti)
    Nd1=scipy.stats.norm(0,1).cdf(d1) #density function of d1 and d2
    Nd2=scipy.stats.norm(0,1).cdf(d2)
    dollarprice=
(Notional*delta_step*discount0_Ti1)*(forward_rateTiTi1*Nd1-rK*Nd2)
    return dollarprice

def ATMCAP(Tn, sigma, rK, delta_step, Notional, Discount_vector):
    Cap=0
    t=0
    for i in np.arange(int(Tn/delta_step)):
```

```
        t=delta_step*(i+1)
        Cap+= ATMCaplet_fun(t, sigma, rK, delta_step, Notional, 
Discount_vector[i+1])
    return Cap

def ATMSwaption_Black(Tn, sigma, rK, delta_step, Notional, 
Discount_vector,OptMatur):
    forward_rateTiTi1 = rK
    
d1=math.log(forward_rateTiTi1/rK)/(sigma*math.sqrt(OptMatur))+0.5*sigma*ma
th.sqrt(OptMatur)
    d2=d1-sigma*math.sqrt(OptMatur)
    Nd1=scipy.stats.norm(0,1).cdf(d1) #density function of d1 and d2
    Nd2=scipy.stats.norm(0,1).cdf(d2)
    A=0
    for index in range(1,Tn*2-1):
        A+=delta_step*Discount_vector[index]
    dollarprice= (Notional*A)*(forward_rateTiTi1*Nd1-rK*Nd2)
    return dollarprice

def Swaption_Black(Tn, sigma, rK, forward,delta_step, Notional, 
Discount_vector,OptMatur):
    forward_rateTiTi1 = forward
    
d1=math.log(forward_rateTiTi1/rK)/(sigma*math.sqrt(OptMatur))+0.5*sigma*ma
th.sqrt(OptMatur)
    d2=d1-sigma*math.sqrt(OptMatur)
    Nd1=scipy.stats.norm(0,1).cdf(d1) #density function of d1 and d2
    Nd2=scipy.stats.norm(0,1).cdf(d2)
    A=0
    for index in range(1,Tn*2-1):
        A+=delta_step*Discount_vector[index]
    dollarprice= (Notional*A)*(forward_rateTiTi1*Nd1-rK*Nd2)
    return dollarprice

#TEST
"""
discount=[99.4580,98.8510,97.6673]
discount=discount/100

A=Caplet_fun(0.25,0.235, 0.02555,0.25,100,0.994580,0.988510)
print(A)  #Test Caplet function, ok

B=CAP(0.25,0.235, 0.02555,0.25,100,discount)
print(B) #Test CAP function check 1, ok
```

```
C=CAP(0.5,0.235, 0.02555,0.25,100,discount)
print(C) #TEST Cap function check 2, ok

B1=Caplet_fun(0.5,0.235, 0.02555,0.25,100,98.8510,97.6673)
print(B1+A) #Cap functions counter check 2, B1+A must be equal to C, OK
"""


##########################################################################
"Forward Volatilities Bootstrapping"
#Input-> np arrays for ZCB(n+1), Strikes(n), Flat Vol (n) for ATM
instruments
#where n is the number of maturities of interest.
#NB: I need data for one extra maturity: If I want to price a swap that
#has payments
#until 2.5 years I need data until year 3!
#DATA:
Discount_factors6M=Discount_new
Flat_volP=Flat_volP
Cap_Strikes=Cap_Strikes
N_maturities=11 # n in this case is 5, meaning that I enter in a swap with
in 1 year with 3 payment dates
Delta_step=0.5
Notional=10000000 #10M

Cap_Prices=np.empty([N_maturities,1])
for index in range(0,N_maturities): #range must be (1, n+1), index is the
i-th time step
    Cap = ATMCAP(Delta_step*(index+1), Flat_volP[index],
Cap_Strikes[index], Delta_step, Notional, Discount_factors6M)
    #NB ZCB vector must be n+1-dimensional, Flat_vol and Cap_Strikes being
np.array
    Cap_Prices[index] = Cap  #Creates a vector with cap prices per each
maturity
print(Cap_Prices)

#Bootstrapping
Fwd_vol = np.zeros([N_maturities,1])
Fwd_vol[0]=Flat_volP[0]
for maturity in range(1,N_maturities+1):
    SUM=0
    matur=maturity
    for i in range(1,matur):
        SUM+=ATMCaplet_fun(i*Delta_step,Fwd_vol[i-1],Cap_Strikes[matur-
1],Delta_step,Notional,Discount_factors6M[matur-1])
    Final=Cap_Prices[matur-1]-SUM
```

```
    aim = Final
    check = np.zeros((1500,3))
    if maturity == 1:
        sigma_range=np.linspace(Fwd_vol[maturity-1],Fwd_vol[maturity-
1]+0.04,1500)
    else:
        sigma_range=np.linspace(Fwd_vol[maturity-2],Fwd_vol[maturity-
1]*2,1500)

    for i in range(len(sigma_range)):
        try_sigma = sigma_range[i]
        Price = ATMCaplet_fun(matur-1,try_sigma, Cap_Strikes[matur-
1],Delta_step,Notional,Discount_factors6M[matur-1])
        check[i,0] = Price #Cap function check 1, ok
        check[i,1] = Price - aim
        check[i,2] = try_sigma
        minimum = np.min(check[:,1])
        min_list=check[:,1].tolist()
        location=min_list.index(minimum)
        result = check[location,2]
    Fwd_vol[maturity]=result
print(Fwd_vol)
#I obtain n (11) Fwd Volatilities

##########################################################################
"Volatility structure of Forward Rates Bootstrapping"

S_vol = np.empty([N_maturities-1,1])
#NB: for n forward volatilities I can extract n-1 volatilities of fwd
rates
#Look Veronesi book p.722
for index in range(N_maturities-1):
    Sum=0
    if index == 0:
        S_vol[index] = Fwd_vol[index+1]

    else:
        Sum+=S_vol[index-1]*S_vol[index-1]*Delta_step
        S=math.sqrt((Fwd_vol[index+1]*index*Delta_step-Sum)/Delta_step)
        S_vol[index] = S
print(S_vol)
print(Fwd_vol)
print(Flat_volP)
len(S_vol)
#I obtain n-1 (10) volatilities
```

```
"Plot Volatilities"
#plt.figure()
#plt.plot(Maturities_new[0:10], Flat_volP[0:10]*100, 'navy',
Maturities_new[0:10], Fwd_vol[0:10]*100, "cornflowerblue",
Maturities_new[0:10],S_vol[0:10]*100, "co")
#plt.legend(['Flat Vol', 'Fwd Vol', 'S vol'])
#plt.title('Volatilities')
#plt.xlabel("Time, years")
#plt.ylabel("Volatility level(%)")
#plt.show()

##########################################################################
"Random number generation, Monte Carlo"
avg = 0
std_dev = 1
num_reps = 250
num_simulations = 10000
random_num_matrix=np.empty([num_reps,num_simulations])
np.random.seed(1234)
for i in range(num_simulations):
    random_num_matrix[:,i] = np.random.normal(avg, std_dev,
num_reps).round(4)
#I obtain a matrix of 10k trajectories made of 250 steps

#plt.plot(random_num_matrix[:,150])
#plt.show()

##########################################################################
"Forward Rates Simulation"
#Simulation of the eight forward rates, to obtain the forward rates for
the
#same period but observed in one year
#Find start forward rates F(0,1,i):

MC_simulation=np.empty([250,8,num_simulations])
Start_forward_rates=np.empty([N_maturities-3]) #NB I need 8 forward rates

for rate in range(2,10):
    forward_rate1_T=Discount_factors6M[rate]/Discount_factors6M[rate-1]
#forward rate between Ti and Ti+1
    forward_rate1_T=(1/Delta_step)*((1/forward_rate1_T)-1)
    Start_forward_rates[rate-2]=forward_rate1_T

#Check initial forward rates
#print(Start_forward_rates)
```

```
for simulation in range(num_simulations):
    for i in range(8):
        MC_simulation[0,i,simulation]=Start_forward_rates[i]
#Add the forward rates as starting point of the empty matrix in EACH
simulation


delta=1/250
m=0

#Run the simulation for 10k times
for simulation in range(num_simulations):
    for step in range(1,250):
        for column in range(8):
            m_start=(Delta_step*MC_simulation[step-
1,column,simulation]*(S_vol[column+2]**2))/(1+Delta_step*MC_simulation[ste
p-1,column,simulation])-(S_vol[column+2]**2)*0.5
            if column == 0:
                m = m_start
            else:
                m =
(m/S_vol[column+1]+S_vol[column+1]*0.5+(Delta_step*MC_simulation[step-
1,column,simulation]*S_vol[column+2])/(1+Delta_step*MC_simulation[step-
1,column,simulation])-S_vol[column+2]*0.5)*S_vol[column+2]
            forward=MC_simulation[step-
1,column,simulation]*math.exp(m*delta+S_vol[column+2]*math.sqrt(delta)*ran
dom_num_matrix[step,simulation])
            MC_simulation[step,column,simulation] = forward

#Check Simulation
#plt.plot(MC_simulation[:,:,888])
#plt.title('One simulation of the eight forward rates')
#plt.ylabel('Forward_rates')
#plt.show()

#plt.plot(MC_simulation[:,0,700:710])
#plt.title('Ten different simulations for the first forward rate')
#plt.ylabel('Forward_rates')
#plt.show()

"TENOR 5Years"
##########################################################################
"Computation of discount factors Z between To and Tm (pag. 724 Veronesi)"

Z_discount_forward=np.empty([num_simulations,N_maturities-3])
```

```
for simulation in range(num_simulations):
    Z=1
    for rate in range(8):
        Z=Z*1/(1+MC_simulation[249,rate,simulation]*Delta_step)
        Z_discount_forward[simulation,rate] = Z


#print(Z_discount_forward)
"TENOR 5Years"
##########################################################################
"Computation of the final payoff (Swap value in To) in each simulation
(x10k payoffs)"

Swaption_values=np.empty([15,4])
Estimator_variance=np.empty([15,4])
Tenor_difference=np.array([6,5,4,3])
Tenors=np.array([1,2,3,4])

Strike=np.linspace(0.001,0.009,15)
Final_payoff=np.empty([num_simulations])
"""Swap_rates=np.empty([num_simulations])
Swap_rates=MC_simulation[249,0,:]"""

for Tenor in range(len(Tenor_difference)):
    for strike in range(len(Strike)):
        Curr_strike=Strike[strike]
        #NB the eight forward rates in each simulation will be used to
compute one payoff
        for simulation in range(num_simulations):
            Payoff=0
            for maturity in range(N_maturities-Tenor_difference[Tenor]):

Payoff+=Delta_step*Curr_strike*Z_discount_forward[simulation,maturity]
                if maturity == N_maturities-Tenor_difference[Tenor]-1:

Payoff=(Payoff+Z_discount_forward[simulation,N_maturities-4]-1)*Notional
#Subtract notional
            Final_payoff[simulation]=Payoff
        #print(Final_payoff)
        "Receiver European Swaption value"
        European_swaption=0
        for simulation in range(num_simulations):
            European_swaption+=max(Final_payoff[simulation],0)

European_swaption=European_swaption*Discount_factors6M[1]/num_simulations
        Swaption_values[strike,Tenor]=European_swaption
```

```
        variance=0
        for simulation in range(num_simulations):
            variance+=(Swaption_values[strike,Tenor]-
Discount_factors6M[1]*max(Final_payoff[simulation],0))**2
        Estimator_variance[strike,Tenor]=variance/num_simulations

print(Swaption_values)
print(Estimator_variance)

from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D

X=Tenors
Y=Strike*100
X, Y = np.meshgrid(X, Y)
Z=Estimator_variance
fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.viridis)
plt.xlabel("Tenors")
plt.ylabel("Strikes (%)")
plt.show()



plt.plot(Strike,Swaption_values[:,0],"paleturquoise",Strike,Swaption_value
s[:,1],"c",Strike,Swaption_values[:,2],"royalblue",Strike,Swaption_values[
:,3],"navy")
plt.xlabel("Strike")
plt.ylabel("European Swaption Value(€)")
plt.title("Price of a Swaption on 10,000,000€ notional")
plt.legend(['1Y Tenor', '2Y Tenor', '3Y Tenor',"4Y Tenor"])
plt.show()


Data={"Strike/Tenor": Strike, "2Y": Swaption_values[:,0],
"3Y":Swaption_values[:,1], "4Y":Swaption_values[:,2],
"5Y":Swaption_values[:,3]}
Data_frame_swaption=pd.DataFrame(Data, columns =
["Strike/Tenor","2Y","3Y","4Y","5Y"])

print(Data_frame_swaption)

#Price Quoted Swaptions
ATM_Strikes=np.empty([4])
for i in range(2,6):
```

```
    a=Discount_new[i*2-1]/Discount_new[1]
    atmstrike=(1/Delta_step)*((1/a)-1)
    ATM_Strikes[i-2]=atmstrike

Implied_vol=np.array([21.49,27.21,33.84,39.95])

Quoted_swaptions=np.empty([4])
for index in range(4):

Swaption=ATMSwaption_Black(2+index,Implied_vol[index],ATM_Strikes[index],D
elta_step,Notional,Discount_factors6M,1)
    Quoted_swaptions[index]=Swaption


#Price Corresponding model swaptions

Swaption_values=np.empty([4,4])
Tenor_difference=np.array([6,5,4,3])

StrikeATM=ATM_Strikes
Final_payoff=np.empty([num_simulations])
"""Swap_rates=np.empty([num_simulations])
Swap_rates=MC_simulation[249,0,:]"""

for Tenor in range(len(Tenor_difference)):
    for strike in range(len(StrikeATM)):
        Curr_strike=StrikeATM[strike]
        #NB gli otto forward rates ad ogni simulazione serviranno a
calcolare 1 payoff
        for simulation in range(num_simulations):
            Payoff=0
            for maturity in range(N_maturities-Tenor_difference[Tenor]):

Payoff+=Delta_step*Curr_strike*Z_discount_forward[simulation,maturity]
                if maturity == N_maturities-Tenor_difference[Tenor]-1:

Payoff=(Payoff+Z_discount_forward[simulation,N_maturities-4]-1)*Notional
#Add notional
            Final_payoff[simulation]=Payoff
        #print(Final_payoff)
        "Receiver European Swaption value"
        European_swaption=0
        for simulation in range(num_simulations):
            European_swaption+=max(Final_payoff[simulation],0)

European_swaption=European_swaption*Discount_factors6M[1]/num_simulations
```

```
        Swaption_values[strike,Tenor]=European_swaption

print(Swaption_values)

My_values=np.array([Swaption_values[0,0],Swaption_values[1,1],Swaption_val
ues[2,2],Swaption_values[3,3]])
Tenor1=np.array([Quoted_swaptions[0],My_values[0]])
Tenor2=np.array([Quoted_swaptions[1],My_values[1]])
Tenor3=np.array([Quoted_swaptions[2],My_values[2]])
Tenor1=np.array([Quoted_swaptions[3],My_values[3]])
Names=np.array(["Quoted","Model result"])

Tenors=np.array([1,2,3,4])

Data={"Tenor": Names, "2Y": Tenor1, "3Y":Tenor2, "4Y":Tenor3, "5Y":Tenor3}
Data_frame_comparison=pd.DataFrame(Data, columns =
["Tenor","2Y","3Y","4Y","5Y"])
"#Comparison Quotes e prices ATM"

var=0
for i in range(4):
    var+=(Quoted_swaptions[i]-My_values[i])**2
MSE=var/4


plt.plot(Tenors,My_values,"c",Tenors,Quoted_swaptions,"navy")
plt.xlabel("Tenor")
plt.ylabel("European Swaption Value(€)")
plt.title("Quoted Prices vs. Model Results")
plt.legend(["Model results","Quoted"])
plt.show()
#Quoted Swaption prices for all strikes
```

Bibliography

Akume, D., Luderer, B., & Weber, G. W. (2003). *Pricing And Hedging Of Swaption*. Supported by DAAD under grant A/001/31321, at Chenmitz University of Technology.

Aljahdali, A., & Mascagni, M. (2017). *Feistel-inspired scrambling improves the quality of linear congruential generators.*

Black, F., Derman, E., & Toy, W. (1990). A One-Factor Model of Interest Rates and Its Application to Treasury Bond Options. *Financial Analysts Journal*.

Brace, A., Dun, T., & Barton, G. (1998). Towards a central interest rate model. *Global Derivatives Conference*.

Brace, Alan, Gatarek, D., & Musiela, M. (1997). The Market Model of Interest Rate Dynamics. *Mathematical Finance*, *7*.

Brandimarte, P. (2006). *Numerical Methods in Finance and Economics*. Wiley.

Brandimarte, P. (2014). *Handbook in Monte Carlo Simulation Applications in Financial Engineering, Risk Management, and Economics*. Wiley.

Brigo, D., & Mercurio, F. (2001a). *Interest Rate Models Theory and Practice*. Springer.

Brigo, D., & Mercurio, F. (2001b). The LIBOR and Swap Martket Models. In *Interest Rate Models Theory and Practice* (pp. 195–310).

Cox, J., Jonathan, I., & Ross, S. (1985). An intertemporal general equilibrium model of asset prices. *Econometrica*.

Davis, P. J., & Rabinowitz, P. (1984). Methods of Numerical Integration. *Academic Press*.

Dizard, J. (2019). Wall Street must recognise challenges posed by machine learning. *Financial Times*.

Dostal, L., & Kreuzer, E. J. (2016). Analytical and Semi-analytical Solutions of Some Fundamental Nonlinear Stochastic Differential Equations. *ScienceDirect*.

Fang, V., Lin, C. T., & Roadcap, L. (2007). *Macroeconomic News Surprises, Business Cycles, and Interest Rate Swap Spreads*. Department of Accounting and Finance, Monash University, Victoria 2 School of Commerce, University of Adelaide, South Australia.

Frey, R. (2008). *Monte Carlo methods : with application to the pricing of interest rate derivatives*.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Springer. https://doi.org/10.1002/9781118445785.scard

Ho, T., & Lee, S.-B. (1986). Term Structure Movements and Pricing Interest Rate Contingent Claims. *The Journal of Finance*.

Hull, J. (2018a). Martingale and measures. In *Options, Futures and Other Derivatives* (pp. 655–673).

Hull, J. (2018b). *Options, Futures, and Other Derivatives* (2018th ed.).

Hull, J., & White, A. (1990). Pricing Interest-Rate-Derivative Securities. *Oxford University Press Inc.*

Jones, C., & Stafford, P. (2020). ECB working group to ask Brussels for year delay to EU benchmark reforms. *Financial Times*.

Judd, K. L. (1998). Numerical Methods in Economics. *MIT Press*.

Kim, D., & Wright, J. (2005). An Aribtrage-Free Three Factor Term Structure Model and the Recent Behavior of Long-Term Yields and Distant-Horizon Forward Rates. *Federal Reserve Board, Working Paper*.

Kochkodin, B. (n.d.). Negative Rates Are Rewriting the Rules of Modern Finance. *2020*.

Lapeyre, B., Pardoux, E., & Sentis, R. (1998). Introduction to Monte-Carlo Methods for Transport and Diffusion Equations. *Oxford University Press Inc.*

Lehmer, D. H. (1949). Mathematical methods in large-scale computing units. *Annals of the*

Computation Laboratory of Harvard University.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing american options by simulation: a simple least-squares approach. *The Review of Financial Studies*.

Mai, S. (2008). *The global derivatives market*. Deutesche Borse AG.

Marsaglia, G. (1968). Random Numbers fall mainly in the planes. *Proceedings of the National Academy of Science*.

Marsaglia, G. (1972). The structure of linear congruential generators. In *Applications of Number Theory to Numerical Analysis* (pp. 249–286). Academic Press.

Monoyios, M., & Hambly, B. (2017). *Fixed Income*.

Munk, C. (2013). *Financial asset pricing theory*.

Musiela, M., & Rutkowski, M. (1997). *Martingale Methods in Financial Modelling*. Springer.

Niederreiter, H. (1992). Random Number Generation and Quasi-Monte Carlo Methods. *Society for Industrial and Applied Mathematics*.

Perninge, M., Amelin, M., & Knazkins, V. (2008). Comparing Variance Reduction Techniques for Monte Carlo Simulation of Trading and Security in a Three-Area Power System. *Transmission and Distribution Conference and Exposition*.

Rebonato, R. (1999). *On the pricing implications of the joint lognormal assumption for the swaption and cap markets*.

Rebonato, R. (2014). Why does the market price of risk depend on the slope of the yield curve? *Journal of Fixed Income*.

Rebonato, R. (2016). Structural Affine Models for Yield Curve Modeling. In *Hanbook of Fixed Income Securities*.

Rebonato, R., Saroka, I., & Putyatin, V. (2014). A principal-component based affine term structure model. *Applied Mathematical Finance*.

Schnabel, I. (2020). *The ECB's monetary policy during the coronavirus crisis – necessary, suitable and proportionate*.

Tezuka, S. (1998). Random and Quasi-Random Point Sets. In *Financial Application of Monte Carlo and Quasi-Monte Carlo methods*.

Vašíček, O. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*.

Veronesi, P. (2005a). *Fixed Income Securities: Valuation, Risk, and Risk Management*.

Veronesi, P. (2005b). Forward Risk Neutral Pricing And The LIBOR Market Model. In *Fixed Income Securities: Valuation, Risk and Risk Management* (pp. 707–740).

Walker, O., & Keohane, D. (2020). SocGen's maths geeks built an empire: do the sums still add up? *Financial Times*.

Zaegel, X., Ramos, S., Martino, P., & Peters, J.-P. (2019). *Transition from EONIA, LIBOR & EURIBOR to alternative risk-free rates*. Deloitte Touche Tohmatsu Limited.

LUISS

MSc. in Corporate Finance

Department of Business and Management

Course of Asset Pricing

Monte Carlo Methods and Market Models for European Swaptions pricing

Thesis Summary

Prof. Paolo Porchia

SUPERVISOR

Prof. Marco Pirra

CO-SUPERVISOR

Laura Bruno

ID: 706961

CANDIDATE

Academic Year 2019/2020

Table of Contents

Introduction

This research is aimed at implementing an algorithm to price European Swaption that can capture current market conditions, which are embedded in macro-financial variables. In the model built in this paper, the calibration primarily takes into account the current term structure and quotes for specific classes of derivatives.

Derivatives started to be traded in the early 90s and, despite the crash related to the 2008 financial crisis, the volume of the notional amount exchanged on OTC market reached 640 trillion (as in June 2019). Those instruments are used from multiple individuals and corporates for speculation and hedging purposes. In particular, fixed income derivatives are mainly used for hedging the risk arising from interest rates, and not surprisingly interest rates derivatives are the most liquid sub-category. Those instruments allow investors that are exposed to interest rates, such as issuers of floating-rate securities or investors of callable securities, to be protected against fluctuations in the term structure. Swaptions are exotic options where the underlying is a swap contract, and the buyer of a receiver/payer swaption will have the right at maturity to enter in a receiver/payer swap at a pre-determined strike swap rate. Notwithstanding the diffusion of these instruments, their pricing process poses multiple challenges in its implementation, that will be examined in this paper.

First of all, since the payoff depends on future interest rates, this type of derivatives requires an appropriate process to understand and predict the term structure. Over time, an extensive literature has developed on term structure modelling, and although there are multiple types of models the most widespread one is the Affine class which will be thoughtfully analyzed.

Furthermore, whilst in equity derivatives, the payoff and the discount factors depend on different variables, in fixed income derivatives both the discount factor and the payoff are built upon the interest rate. This peculiarity causes a non-zero correlation between these two which alters the discounted expected value of future outcomes. Changing the probability measure and obtaining the so-called Equivalent Martingale Measure result, will allow to have a deterministic discount factor outside the expectation operator, and consequently eliminate the problem of correlation. The adoption of a new probability measure, namely the T-forward measure, lays the groundwork for the application of Market Models such as the LIBOR Market Model and the Swap Market Model. This study will analyze and compare short rates models and market models and their ability to price swaptions reflecting current market conditions.

The chosen model, after a discretization procedure, is implemented through the famous statistical tool of Monte Carlo simulation, which allows describing the true dynamics of interest rates, whom advantages for swaptions will be illustrated.

With the appropriate calibration, the resulting pricing algorithm performed generates an estimator of the price of a European receiver swaption with one year of maturity. On the result are performed sensitivity analyses to explain the variation of the price concerning the main variables: strike and tenor of the underlying swap. The variance of Monte Carlo estimators is determined and appears to be higher for greater tenor and strike values. Finally, to validate the pricing algorithm based on Monte Carlo simulation and LIBOR Market Model its outcome is compared with At-the-money swaptions quoted on the market at the day of evaluation. The Mean Squared Error is additionally investigated.

The research structure follows the logical steps of the pricing process. The first section reviews extensively the most famous Affine models for term structure modelling and the different existing volatility structures, examining the mathematical steps needed to change the probability measure and obtain the solution to the stochastic differential equation for pricing. The second section provides the theoretical knowledge about Monte Carlo methods, underlining the importance of these tools in complex multidimensional integration problems and explaining the advantages of using Monte Carlo estimations. This chapter goes also through variance reduction techniques and the convenience in the application of Monte Carlo simulation to swaption pricing. The third chapter is aimed at performing the pricing, explaining the derivation of the specific discretized stochastic process simulated and its calibration on the current term and volatility structure. The results and the sensitivity analyses are presented together with the model validation. The role of modern pricing techniques in the landscape of global markets is explained and future challenges of the latter techniques are identified. Finally, the conclusions are reported and extensions of the model for future implementation are suggested.

Chapter 1: Affine Models for term structure modelling

1. Exotic options and European swaptions

The term "exotic options" includes any option type that is different from Vanilla options, from the perspective of payoffs and cash flows structure. This class of derivatives brings many advantages in term of use, because of their complex structure of payoffs and cash flows they can easily meet different investors' needs. Two main categories of exotic options can be distinguished: path independent are characterized by payoffs that are function only of the underlying asset price at maturity, whereas path dependent's payoffs are functions of the price path of the underlying as whole or in some specific portion. For categorization purposes also the dimension and the order of the option are often taken into account. The former is the number of variables that characterize the payoffs, and the latter is related to the type of function that links those variables and the payoffs. In this paper, I will go through interest rate models and Monte Carlo simulation with the final intent of pricing a type of exotic option on fixed income, that is Bermudian and European Swaption. There exist a considerable body of literature that will be soon introduced, that shows the progress that have been made in term of accuracy to price fixed income derivatives. Notably, several theories have been dedicated to swaptions pricing with some remarkable results during the latest year of the XX century.

2. Structural Affine Models

Structural models for yield curve are aimed not only at characterizing but also at understanding the term structure and its changes, being in this way extremely useful for forecasting purposes. With no doubt, among many types of structural models, the most widespread class is the affine one. An important constraint that is imposed on these models is the no-arbitrage condition, which ensures that prices do not allow no-risk profits. The no-arbitrage condition can also guide statistical estimates in the process of characterization of the yield curve. In this chapter I will analyze briefly the first generation of Affine Models, Vasicek (1977) and CIR (1985) and three important drivers of the yield curve, to focus later on in the chapter on more complex models which rely on the forward neutral probability measure. More specifically the Libor Market Model by Brace, Gatarek and Musiela will be used to simulate the trajectories of future spot Libor rates and price Bermudian Swaptions.

To introduce the Affine setting, Rebonato highlights three factors which determine the structure of the yield curve: Expectation, Risk Premia and Convexity. Expectations are often included in affine models (e.g.: Vasicek) through a mean-reverting component, which is able to capture a long term mean reversion that unfortunately does not fit well the intuition by which expectation might influence the short term part of the yield curve instead of the long term one (Rebonato, 2016).

Vasicek and CIR models represent the first generation of affine models, they both take into account the no-arbitrage condition and the main drivers of the yield curve in a one-factor equation, and the state variable is an affine diffusion under both physical and risk neutral measures. Short rate models, in general, are highly intuitive and flexible for their ability to explore the dynamics of an instantaneous continuously compounded short rate r_t. In the Vasicek model, the short rate increment follows a generic Gaussian Markov process in which the short rate reverts to a long term fixed level γ with a reversion speed of φ (Cox et al., 1985).

The component φ has central importance, precisely when the reversion speed is zero the duration of the security grows linearly with maturity, while at a higher reversion speed the duration grows less with maturity, meaning that for high reversion speeds the security is less sensitive to changes in the yield (Vašíček, 1977).

The risk premia is the excess return required from investors to bear some specific level of risk, and the compensation related to each risk factor per unit risk is determined as $u_j^{t,T} = \frac{\partial P_t^T}{\partial x_j} \sigma_j^t \lambda_j^t$

Where $\frac{\partial P_t^T}{\partial x_j}$ is the price sensibility of the specific bond to the j^{th} risk factor, σ_j^t is the volatility of the risk factor, and λ_j^t is the market price of one unit of that risk. Considering all the risk factors that characterize a security, the expected return at time t can be defined as the sum between the compensations of all the risk factors.

The market price of risk λ_j^t is assumed to be constant in the first generation of Affine Models. A number of questions regarding this assumption remain to be addressed, and even though this approach is particularly straight forward it does not capture the shown trend of positive excess return when the yield curve is steep and zero or negative excess returns when the curve is flat or downward sloping (Rebonato, 2014).

Convexity captures the non-linear relationship between yields and prices and might be responsible for the shape of the term structure for long maturities. The above mentioned non-linear effect in the Vasicek model is observed in the volatility of the yield, which depends quadratically on the volatility of the state variable and on the sensitivity term (Rebonato, 1999).

The no-arbitrage assumption, that represents an important landmark in pricing is translated in the equation below, which in simple words states that the return on a security should equate the sum of the compensation for every source of risk (Kim & Wright, 2005).

At this point it is important to underline that there is a trade-off regarding the type of variables (or factors) on which the model should depend. From one perspective, choosing variables which derive from macroeconomic equilibrium models surely simplifies the intuition behind the model, yet it is important to double-check the robustness of the macroeconomic assumptions behind them. From the other perspective choosing variables that come from statistical and econometric analyses, such as Principal Component analysis (Rebonato et al., 2014), adds opaqueness and introduces the problem of overparameterization (Rebonato, 2016).

Models such as Vasicek and Cox, Ingersoll and Ross are not able to exactly fit the dynamic of the current yield curve, luckily other studies carried by Ho and Lee, Hull and White and Black Derman and Toy proposed new models that were not only able to fit the term structure, but once revisited and extended were also able to fit the observed volatility structure.

Ho Lee model, from 1986, is the most straight forward and the first one able to fit the term structure of interest rates. Despite the intuitive nature of the Ho Lee model, there are some flaws due to its simplicity. In fact, the model allows a positive probability of negative interest rates because of the symmetrical distribution. Furthermore, it uses the empirical volatility, computed from historical interest rates assuming a flat volatility structure (Veronesi, 2005a). Therefore, it tends to overprice low maturity caps, floors and swaption and underprice long maturity ones.

The Hull White model, introduced in 1990, extends the Vasicek model in order to fit the term structure. In 1994 the same model was broadened to a two factor model, to give a better shape of the term structure. Unfortunately, it still allows negative values for the short rate. On the positive side, one can choose the parameters γ^* and σ to best fit the forward volatility structure.

The Black, Derman and Toy (BDT) model, introduced in 1990 as well, applies a transformation to the short rate defining a new variable, $z_t = \ln(r_t)$. The logarithmic transformation gives the variable a distribution with positive skewness that changes the result of the estimate. This procedures gives a zero probability to negative interest rates, however assigning higher probabilities to high levels of interest rates and lower probability to low interest rates.

3. Probability measures and forward neutral models

Regrettably, the Fundamental Pricing Equation is not easily applicable to interest rate derivatives, whose pricing is the aim of this thesis. In fact, in that case both the discount factor and the payoff depend on the interest rate, consequently there is a positive correlation element that should be taken into account.

$$V(r,t;T) = \mathbb{E}^*\left[e^{-\int_t^T r_u du} h_T\right] = \mathbb{E}^*\left[e^{-\int_t^T r_u du}\right] * \mathbb{E}^*[h_T] + cov(e^{-\int_t^T r_u du}, h_T)$$

The change of numeraire allows a simplification of the pricing formula in this situation. The equivalent martingale measure result, which is fully derived in the full text of this paper, shows that for a given numeraire, and a particular choice of the market price of risk, the normalized price $\tilde{V}(r,t;T)$ is a martingale for all the securities (Musiela & Rutkowski, 1997).

The change of measure in the case of constant market price of risk is applied in the stochastic process of interest with a parallel shift of the drift. The important result is that now the discount factor is outside the expectation operator.

A swaption gives the option to enter into a swap contract with swap rate equal to a predetermined strike rate, at the exercise date. If at the exercise date the current swap rate on the market is less convenient (higher/lower swap rate observed on the market in case of a payer/receiver) the option will be exercised. Thus, the payoff value at the exercise date can be computed as the value of the exchange of two coupon bonds: one for the fixed leg characterized by a coupon rate equal to the option strike, and one for the floating leg that is at-par by definition. However to compute the present value of this bond is necessary to have as many discount factors as the number of payment of the bond. It is obvious that the final payoff of this derivative depends on multiple future LIBOR rates. For this reason under the T-forward risk neutral measure there is only one forward rate that is a martingale, while all the other rates are no more log-normally distributed. Therefore only one Forward rate will be driftless, with the other rates having a more complex stochastic process.

By choosing as numeraire the $\bar{T}$-forward rate by which for every i+1 $\bar{T} < T_{i+1}$ is true, that means choosing as numeraire the smallest of the discount factors of interest, the process followed by the increment of the forward rate is (Alan Brace et al., 1997):

$$\frac{df_n(t,T_i,T_{i+1})}{f_n(t,T_i,T_{i+1})} = \left(\sum_{j=\bar{\iota}}^{i} \frac{\Delta f_n(t,T_j,T_{j+1})\sigma_f^{i+1}(t)\sigma_f^{j+1}(t)}{1+\Delta f_n(t,T_j,T_{j+1})}\right) dt + \sigma_f^{i+1}(t) dX_t$$

4. Volatility Structures

Notwithstanding the fact that the above models are able to fit the term structure and consequently correctly price bonds, they still do not correctly price the type of interest rate derivatives that are being analyzed in this paper. This is due to the lack of matching with the volatility implied by market prices of those securities. The implied volatility is indeed the volatility implied by the dollar price, that if applied to the pricing formula gives back the exact same price of the market. Often derivatives quotes are expressed in terms of implied volatility, such as for swaption contracts. If we consider different caps, one for each maturity, starting from their prices we can extract with the use of the Black formula the implied volatility for each cap. The issue related to this specific volatility is that each one is able to price only a single cap because with this process it is of necessity assumed a different flat volatility structure for each cap. However each cap has different caplets at different maturities, and recalling that the price of the cap is equal to the sum of the single caplets' prices, it is inconsistent to price with different volatilities two caplets with identical maturities coming from two different caps having different maturities. Accordingly, it is reasonable to extract from implied volatilities the structure of Forward volatilities so that to each point in time corresponds a volatility that is able to price all the caplets for that maturity. This is implemented through a bootstrapping procedure where the starting point is the cap with the closest maturity, whose flat volatility corresponds to the first step forward volatility. In the simplest application of the model, the instantaneous volatility was considered to be constant in each forward rate, independently of the time t at which that specific forward rate is observed. However, the forward volatility structure extracted from caplets is characterized by a bump which is not coherent with the assumption just explained. Furthermore, in a swaption the payoff is not only function of a single LIBOR, therefore the simplistic assumption previously made is not feasible anymore. Alternatively, in the application of the model that this paper follows there are two other assumptions to be made, for the volatility of the forward rate used in the final the simulation. The first assumption is that the volatility of the forward rates has a dependence only with the time to maturity, through a specific function $S(\cdot)$. The second assumption is that the function is constant in each period. These assumptions give as a result a semi-linear volatility structure with S_i being the volatility of the forward rate $f_n(t,T_i,T_{i+1})$ (Brigo & Mercurio, 2001b). To extract this volatility structure is sufficient to have the forward volatility structure mentioned above and then apply a bootstrapping procedure. The first step of the bootstrapping is similar to the one already presented for Forward volatilities:

$$S_1 = \sigma_f^{Fwd}(i\Delta)$$

While the other steps are obtained with the following formula, assuming that $t = 0$:

$$S_i = \sqrt{\frac{1}{\Delta}\left(\sigma_f^{Fwd}(T_{i+1})T_i - \sum_{j=1}^{i-1} S_j^2\Delta\right)}$$

Yet convenient, this volatility structure has some flaws. In fact, when the Forward volatility declines fast the function S drops to zero by its own nature (Brigo & Mercurio, 2001b).

Chapter 2: Monte Carlo methods

1. Introduction to Monte Carlo methods

The use of Monte Carlo is diffused in many field, including financial engineering. They are based on the mathematical intuition that the probability of an event happening is represented by the volume of the possible outcomes that make the event happen. Monte Carlo methods start from the volume of a set of outcomes to estimate the volume of the probability. In simple words, it randomly samples outcomes and picks up only the ones of interest (i.e.: a given set of outcomes), to interpret their volume as the probability of that set of outcomes. Monte Carlo simulation provides for sample paths that might be particularly useful in derivatives pricing where the final payoffs depends on the price path (e.g.: path-dependent exotic options) or in the case of American style derivatives. In continuous time models the dynamics are described by a differential equation. The simulation is used when the differential equation is not only composed by a deterministic part, but also by a stochastic one (Brandimarte, 2006). To simulate continuous time models a discretization method is necessary. Since discretization bias plays its role in the simulation estimate, the choice of a discretization method must be made wisely. The two most widespread methods are Euler and Milstein schemes (Frey, 2008). For the purposes of this analysis the simple Euler scheme will be used.

Monte Carlo methods become particularly useful when it comes to numerical integration, it becomes useful or even necessary when considering high dimensional problems.

Monte Carlo estimate of the an integral would be obtained by collecting a independent and identically distributed sample of points of x over the unit hypercube and computing $\hat{I}_n$:

$$\hat{I}_n = \frac{Vol([0;1]^d)}{n}\sum_{i=1}^{n} f(x_i)$$

The volume of the unit hypercube is equal to one, consequently taking the limit of the estimator for $n \to \infty$ gives the following result:

$$\lim_{n\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right) = I$$

When the number of simulation goes to infinity the estimator converges to the true value, the estimator is therefore unbiased. In the full text the thesis the criteria of choice between unbiased estimators is be presented, and additionally variance reduction techniques are be compared.

To evaluate the efficiency of a model one must take into account both the variance and the computational time per replication of the estimator. Therefore, multiple methods were developed to reduce the variance of estimators in order to obtain a higher level of efficiency. The most straightforward method to reduce the variance of the estimator is surely by increasing the number of simulations. However this approach is not reasonable considering the computational efficiency of the algorithm. Variance reduction techniques can be classified in two main categories: the first branch tries to exploit tractable characteristic of the model to adjust the output; while the second one tries to curtail the volatility of the inputs. The most common ones, that are Control Variates method, Antithetic Variates method and Stratified Sampling method.

2. Monte Carlo for swaption pricing

In conclusion, Monte Carlo methods have an high level of flexibility, and their use in finance is necessary when considering more complex derivatives. Specifically, when considering fixed income derivatives the use of the simulation allows to describe the true dynamics of interest rates incorporating also the instantaneous correlation between the multiple rates that might be needed to price a more complex derivative. A correct calibration of the model in use will also give dynamics that are coherent with some observed market data. Monte Carlo standard error can be reduced with the application of variance reduction techniques.

The next chapter will be fully dedicated to explain the implementation of the pricing algorithm for swaptions. Under the T-Forward-neutral measure I will generate m different realization of the k different Forward rates needed to compute each payoff of the option, that is the Swap expected future price. The following forward rates are simulated to obtain the value of the corresponding forward rates observed at the exercise date T_O, that is equal to the maturity of the forward rate chosen as numeraire T_α: $F_{\alpha+1}(T_\alpha), F_{\alpha+2}(T_\alpha), \dots, F_\beta(T_\alpha)$.

The Forward rates trajectories will follow the stochastic process implied by the LMM. Since the mentioned dynamics do not have a distribution that is known, it is necessary to discretize the process. The latter, with some algebraic manipulations, will yield an equation with deterministic diffusion coefficient for which Euler discretization scheme correspond to Milstein one. In this case The simulated forward rates will be used as input for the computation of the payoff. Each one of the m

simulation will lead to a different future Swap value, and the average of their discounted prices will be the Monte Carlo estimate (Brigo & Mercurio, 2001b).

Chapter 3: Swaption pricing

1. European Swaptions and their use

A swaption contract is a contract that gives its buyer the right to enter the underlying Swap with a given strike swap rate at the exercise date. The underlying contract can be a payer or a receiver Swap, that is a swap that pays or receive the fixed swap rate against a floating rate determined at the previous payment date depending on the reset frequency. The floating rate is determined in the contract and can be indexed to one of the most famous benchmark rates, such as LIBOR, EURIBOR and EONIA rates. Due to recent European regulation, a transition to new risk free indices is taking place. The new risk free benchmark to be used for the Eurozone is defined in Regulation (EU) 2016/1011 and it is called ESTER, European Short Term Rate. Therefore, the old rates are no more compliant with new regulations and all the entities exposed to those rate must transition to the new ones (Zaegel et al., 2019). Swaption contracts are useful instruments of insurance and hedging against any interest rates rise or fall (Akume et al., 2003). For instance, the issuer of a floating rate bond might want to be protected against a rise in interest rates in the future. She can buy a payer swaption with strike rate r_K, that will be the swap rate to be payed if the option is exercised. If at the exercise date the interest rates are higher and, consequently, the swap strike rate is lower than the current swap rate implied by current market condition, then the option can be exercised. The gain will be equal to the difference between the two swap rates: the strike and the current swap rate.

In the same manner a swaption can be useful to an investor of a callable security. In fact, with this type of instrument in case of a fall in interest rates the issuer will be able to obtain funding at cheaper rates, and for this reason will call back the security. The investor would receive her capital earlier than maturity with the possibility to reinvest the capital at the current market conditions, namely lower rates. To hedge this risk the investor can buy a receiver swaption to exercise in case interest rates drop. Thus, the investor will receive the fixed strike rate and pay a floating rate, that can be offset by reinvesting the capital received at the floating rate.

The option that will be priced in this analysis is a receiver swaption with floating rate equal to the LIBOR rate. The valuation date is the 1st of September 2020, the exercise date is in one year, the underlying Swap tenor, i.e. the length of the swap contract, is four years and the contract has a semi-annual reset frequency. The analysis will be conducted on different parameters of the strike swap rate. For the sake of pricing the number of forward rates to simulate is eight. The next paragraph will highlight the reason behind the choice of the model used to simulate and the difference between the use of different models.

2. The choice of the model

The LIBOR Market Model was introduced at the end of the first chapter, and it is built around the change in the probability measure. The use of the T-Forward rate as numeraire allows to characterize the process of forward LIBOR rates. The Swap Market Model instead uses a different numeraire, that is equal to the sum of multiple discount factors, so that it is possible to characterize the process of forward Swap rates (Veronesi, 2005a). Short Rates Models introduced in the first chapter describe the instantaneous interest rate through the definition of a diffusion process, the latter stochastic process explains the evolution of the yield curve and gives as outcome an analytical formula for forward LIBOR rates. The interest rates derivatives can be priced using these inputs and computing a risk neutral expectation, assuming a deterministic discount factor equal to de ZCB price. This last assumption generates inconsistencies in the pricing process. Those inconsistencies can be solved with the use of a different probability measure instead of the traditional risk neutral one.

The main advantage of Market Models introduction is that, unlike Short Rate models, they are compatible with the widespread Black's market formula. In fact, LMM can correctly price caps and SMM can correctly price swaptions consistently with the corresponding Black's formulas.

However, LMM is only compatible with a part of interest rates derivatives market, that includes caps, while SMM is compatible with the other part, that is made by swaptions. The incompatibility of the two models forces to choose one of the two for the entire market of interest rates derivatives, because a joint lognormal assumption would be inconsistent. Forward rates are considered to be more explanatory about the yield curve, and taking into account the fact that the LIBOR Market model is easier to handle from an mathematical point of view it is more smooth to use the numeraire implied by the LMM also for swaption market instead of doing the opposite (A. Brace et al., 1998). The stochastic process derived for this implementation relies on a basic assumption regarding the structure of instantaneous correlations on which the model is calibrated, that allows to simplify the more generic process of the LIBOR Market Model.

3. Model Calibration

The specific discretized process that will be used to simulate the trajectories of the forward rates was explained in the first chapter of this summary and its derivation is explained in the full text of the thesis, while now I will focus solely on the steps needed to be able to simulate those trajectories.

First and foremost, as clarified in the previous paragraph, the model matches the Black's formula for caps, and the implied volatilities in the market prices of those caps will be the input of the model.

Therefore, the calibration to caps and floors volatility surface is straightforward. The quotes semi-annual for at-the-money caps on LIBOR with six months to six years maturity were extracted from Thomson Reuters in terms of implied volatilities. To obtain the corresponding dollar price it was used as an input the quoted implied volatility in the Black's closed formula for caps. Additionally, the latter formula requires LIBOR based discount factors to be implemented.

Therefore, LIBOR rates corresponding to discrete points in time were retrieved from Bloomberg, and the transition to continuous time LIBOR based yield curve required a Cubic Spline Interpolation, that is one of the most widespread method. The result obtained is showed in the figure 2 of the full text. Once those input for the Black formula were defined, a function to extract dollar prices of at the money caps was constructed in Python and with the use of a stripping algorithm the forward volatility structure was extracted. The forward volatility structure can be used to compute the volatility structure that is the input of this model, that was explained at the end of chapter 1.

In the full text is shown a graphical comparison between the three different volatility structure obtained, flat volatilities extracted from at-the-money caps, forward volatilities and volatilities of the forward rates. When the flat volatility is upward shaped the forward volatility curve lies above the former, and in the same way the semi-linear volatilities S will be above the forward volatility curve.

4. Model implementation and results interpretation

To implement the pricing algorithm I started with the LMM equation, deriving the discretized process to which the simulation should be applied. The key step of the algorithm is to simulate random shocks, $\varepsilon_t \sim N(0,1)$ to be used in place of the Brownian increment of the stochastic process. The following scheme, derived with a logarithmic transformation, is the discretized implementation of equation 3.3.

$$f_n^m(t+\delta, T_i, T_{i+1}) = f_n^m(t, T_i, T_{i+1}) e^{\mu_{i+1}^m(t)\delta + S(T_{i-1}-t)\sqrt{\delta}\varepsilon_t}$$

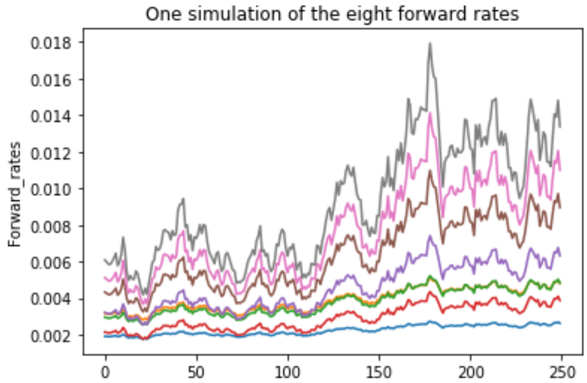
Where the drift μ is equal to:

$$\mu_{i+1}^m(t) = \sum_{j=\bar{\imath}}^{i} \frac{\Delta f_n^m(t, T_j, T_{j+1}) S(T_{i+1}-t) S(T_{j+1}-t)}{1 + \Delta f_n^m(t, T_j, T_{j+1})} - \frac{1}{2} S(T_{i+1}-t)^2$$

Where m indicates the specific simulation number of which the equations refer. I simulated ten thousand trajectories of the eight forward rates needed to price a swaption with exercise date the 1st of September 2021, which gives the right to enter in a receiver swap with 4 years tenor. The simulated forward rates will then be used to compute the corresponding discount factors with the following formula, where M is the maturity date of the underlying swap and $O = \bar{\imath}$ is the exercise date.

$$Z(T_o, T_M) = \frac{1}{1 + \Delta r_n(T_{\bar{\imath}}, T_{\bar{\imath}+1})} \cdot \frac{1}{1 + \Delta f_n(T_{\bar{\imath}}, T_{\bar{\imath}+1}, T_{\bar{\imath}+2})} \cdot \ldots \frac{1}{1 + \Delta f_n(T_{\bar{\imath}}, T_{\bar{\imath}+m-1}, T_{\bar{\imath}m})}$$

Each simulation was run through 252 steps, which is the notation for actual days in one year. The drift μ changes every step and for each one of the eight forward rates, retrieving a 252 rows and 8 columns matrix of drifts per each of the ten thousand simulations. The following sample graph shows the 888-th simultaneous simulation of the forward rates.



The assumption of perfect correlation can be easily spotted observing the movements of the trajectories of the forward rates.

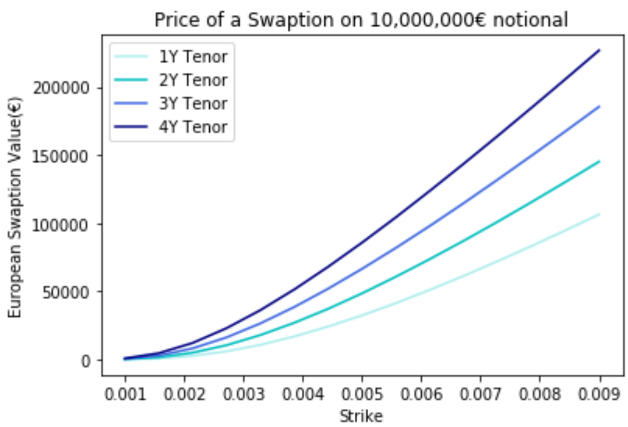
Once the forward rates trajectories are estimated, the future swap price can be computed. Afterwards, is the maximum between the price and zero is estimated computing ten thousand different payoff. The last step consists in computing Monte Carlo estimator by discounting and computing an average of all the different payoffs, as follows.

$$\hat{V}^{Swaption} = \frac{1}{10000} \sum_{i=1}^{10000} Z(0,1) Payoff_{1\,year}^{i}$$

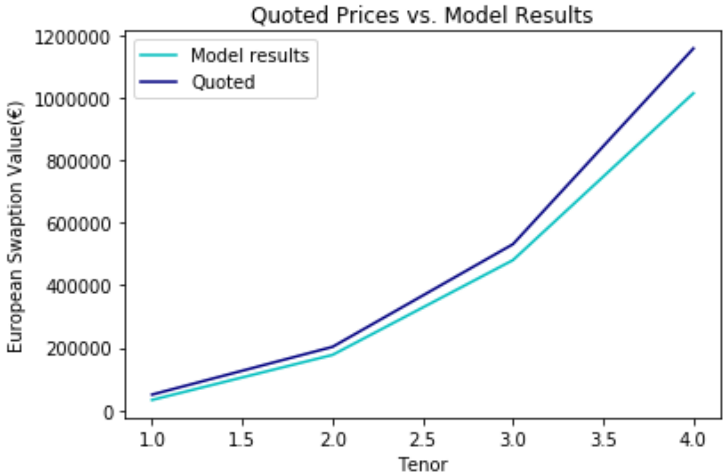
Where,

$$Payoff_{1year}^{i} = \max\left(\sum_{j=2}^{m} \Delta r_K Z(1, 0.5(j+1)) + Z(1,m) - 1; 0 \right)$$

The prices of the European Swaptions with maturity 1 year were computed for different strike levels and different tenors, as it is illustrated below.



To analyze in deep the obtained estimator I computed its variance, using the sample variance formula. The result confirm that the variance of the estimator is significantly high, especially for high tenor levels and high strike levels. In order to reduce the variance it is possible to apply one of the variance reduction techniques mentioned in the previous chapter. Estimator variances had been plotted in a 3-



D graph (in the full text of this thesis) to show the relation with the parameters considered, which are tenor and strike. Last but not least, to validate the solution given by the model I compared the values of ATM swaptions quoted on the market with the corresponding swaptions priced by the implemented model. Since swaptions are quoted on the market through Black's implied volatilities I extracted the dollar price by applying the Black formula (Monoyios & Hambly, 2017). The obtained results of the comparison are shown in Figure 10. The exercise date of the option is one year from the valuation date which is 1st of September 2020. It is possible to see that for higher tenors the distance between the values increases, which means that the precision of the model decreases. The root mean squared error of the model is equal to 77613.19€. It must be underlined that the model tends to underprice the derivative for every tenor, and this might be due to a lack of convergence between the LMM and the SMM, since the model implemented is calibrated on Cap volatilities.

5. Modern pricing techniques in the global markets' landscape

Structured products started to be transacted in the UK in the early 90s, and at the time those banks who had the chance to develop the most sophisticated mathematical models for pricing and structuring those products were one step ahead in the industry (Walker & Keohane, 2020). Derivatives had been used for hedging and speculation purposes and since then this market grew 24% per year on average, reaching €457 trillion of notional outstanding in 2008 (Mai, 2008). The crash of 2008 moved the markets in one negative direction, creating a feeling of distrust towards the derivatives and structured instruments markets. After the crisis, the spread and use of data science and machine learning techniques has given a significant turn to the reality of the financial markets, bringing the trading of these products towards increasingly sophisticated and automated processes (Dizard, 2019). In this context market models and the use of Monte Carlo simulation represented one of the most important steps given their ability to take into account current market variables such as the term structure and the implied volatilities quoted on the market for some specific instruments. It is known that the slope of the volatility and term structures are together good indicators of macroeconomic conditions and compensation for volatilities at specific maturities. The implementation of the model was showed and its accuracy was demonstrated in the previous paragraph. With some extensions the results obtained from the model can be used in multiple ways. In fact, fixed income derivatives by definition are highly connected with interest rates and consequently strongly influenced by monetary policy. This characteristic makes them good candidates to be used to interpret market sentiment and expectations with respect to macroeconomic news and business cycles (Fang et al., 2007). Swaptions together with Caps are among the most liquid derivatives of this category, and they are available on a broad range of interest rates. Now there are new challenges for fixed income derivatives' pricing models, which are related to recent market changes. The first will be the transition to new European risk free benchmarks (Jones & Stafford, 2020). In fact, as new regulation is imposing new characteristics on these rates . Secondly, with the coronavirus outbreak many central banks implemented or started to consider a negative interest rates policy (Kochkodin, 2020). NIRP is known to be a good stimulus for the economy allowing for the propagation of monetary accommodation through the whole yield curve and avoiding further downward pressure on the term premium (Schnabel, 2020). However, from the perspective of fixed income derivative prices negative interest rates policy means that Black formula, used to quote securities in term of implied volatility, is no longer applicable.

Conclusions

The objective of this research was to implement an algorithm to price European Swaption taking into account current values of volatility and term structures. Thanks to the change of the probability measure it was possible to overcome the problems related to the correlation between payoffs and the discount factors, and the inconsistent assumption of a deterministic discount factor under the traditional risk-neutral measure. The adoption of the T-forward measure opens a range of possibilities concerning the use of Market Models. The latter models, unlike Short Rate models, are compatible with the distribution implied by Black's formula, and can therefore be calibrated both on the term structure and on caps and swaptions prices quoted on the market. LIBOR Market Model and Swap Market Model and the inconsistency of their simultaneous use forced us to choose one model for the whole interest rates derivatives market. The evidence extracted from previous studies showed the convenience in adopting the LIBOR Market Model because of its mathematical tractability and the intuition in using forward rates to explain the term structure.

Furthermore, thanks to the adoption of Monte Carlo methods, the true dynamics of interest rates can be simulated including an instantaneous correlation structure between the multiple rates, that is needed when pricing more complex derivatives like swaptions.
After an appropriate calibration of the input parameters on the current market conditions of term and volatility structures, the implementation of the algorithm generated ten thousand trajectories per each simulation of the different forward rates, needed to price swaptions with different tenors. The swaption analyzed had a maturity of one year and the price was computed for different values of the tenor, with the need to generate up to 8 rates simultaneously.
The swaption price estimator was computed for different strike rates and tenors, and the variance of the estimator was investigated, observing an increase for higher values of tenor and strike rate.
To validate the model the obtained prices for at-the-money swaptions were compared with quoted prices of the corresponding swaptions. The validation showed the accuracy of the model and its implementation, and the Mean Squared Error was computed giving a satisfactory result that is illustrated. From the comparison of obtained and quoted prices, it can be observed that the model tend to slightly underprice the swaption for every tenor, and when the tenor increases the amount of the underpricing increases as well. This might be due to a lack of convergence between the LIBOR Market Model and the Swap Market Model, a matter of internal coherence which finds its roots in the calibration on caps' volatilities.
Moreover, the model presents some limitations related to the assumption of a perfect correlation between the different forward rates. In fact, each simulation was performed using the same random numbers in place of the Brownian increments, which implies a perfect correlation between the different rates, that are subject to the same random shocks. This simplification was fundamental to avoid the need to derive an instantaneous correlation structure on which the model should have been further calibrated. The complexity in defining such a structure represents one of the different directions in which the implementation can be extended.
For further exploration of the topic, the analysis could be also carried out for a Bermudian or American swaption with the possibility of early exercise, using the Longstaff-Schwartz algorithm and Least Squares Monte Carlo method.
The current landscape of global markets introduces new challenges for pricing algorithms concerning the input term structure, which is globally going towards strong reductions of interest rates, that are very close to zero or negative. Furthermore, the transition to the new European risk-free benchmarks represents another challenge for interest-rate derivatives market. Given the strong connection, explained in the last chapter, between those instruments and interest rate risks, monetary policy, and business cycles, the centrality of modern pricing models that are able to adapt to rapid changes is undoubted.