Swaptions pricing under the single factor Hull-White Model through the Analytical formula and Finite Difference Methods

Victor Lopez Lopez[1] Jan Röman[2]

MÄLARDALEN UNIVERSITY
SWEDEN

[1] Corresponding author, student of the Master of Science in Mathematics with focus in Financial Engineering at Mälardalen University.

[2] Division of Applied Mathematics, Mälardalen University.

Abstract

Due to the interesting financial moment we are living, my motivations to write this Master thesis has mostly been the behavior of interest rates and models that can be used predict them. Thus, in this dissertation I have presented the Hull-White model and the way to calibrate it against market data so it can be used to price interest rate derivatives. The reader can find both theoretical and practical presentations and examples along with the code to program them by him/herself.

I would like to take advantage of the situation and thank Jan Röman as I have relied through all this work on his lessons at Mälardalens University. Thus, I would like to refer the reader to his work in the references [1] and [7] when seeking for any deeper knowledge in this topic.

Contents

1 Introduction 4

2 Interest rate derivatives 5

2.1 Bonds, FRN and FRA . . . 6

2.2 Swaps . . . 9

2.3 Swaptions . . . 12

2.4 Interest rate derivatives pricing . . . 12

2.5 Martingale modeling . . . 13

2.5.1 Affine Term structure . . . 13

2.5.2 ATS models . . . 14

3 Hull-White Model 15

3.1 Solving the Hull-White PDE to price Zero-Coupon bonds through the Finite Difference Method . . . 16

3.1.1 Boundary conditions . . . 19

3.2 Practical example of Zero-Coupon bonds pricing under Hull-White by Crank-Nicolson Finite Difference Method using Python . . . 19

3.3 Pricing European Swaptions under the Hull-White model . . . 19

3.3.1 Practical example of Pricing European Swaptions under Hull-White analytical formula using Python . . . 21

4 Black-76 model 23

4.1 Pricing European Swaptions under Black-76 . . . 23

4.1.1 Black-Normal model . . . 24

4.1.2 Practical example of Pricing European Swaptions under Normal-Black using Visual Basic Application on Microsoft Excel . . . 25

5 Calibration of the Hull-White model and results 26

5.0.1 Calibration of the Hull-White model. Practical example using Python . . . 26

5.1 Risk measures. The Greeks . . . 27

5.1.1 Results . . . 28

6 Conclusion 32

7 Fulfillment of thesis objectives 33

8 References 35

9 Appendix 1: Hull-White PDE solution by Finite Difference Method (Crank-Nicolson) to price Zero-Coupon bonds. Python code 36

10 Appendix 2: Pricing Swaptions under Hull-White using the Analytical formula. Python code 41

11 Appendix 3: Pricing Swaptions under Black-76 and Normal-Black. VBA code 47

12 Appendix 4: Calibration of Hull-White Swaption prices. Python code 53

13 Appendix 5: Figures 61

1 Introduction

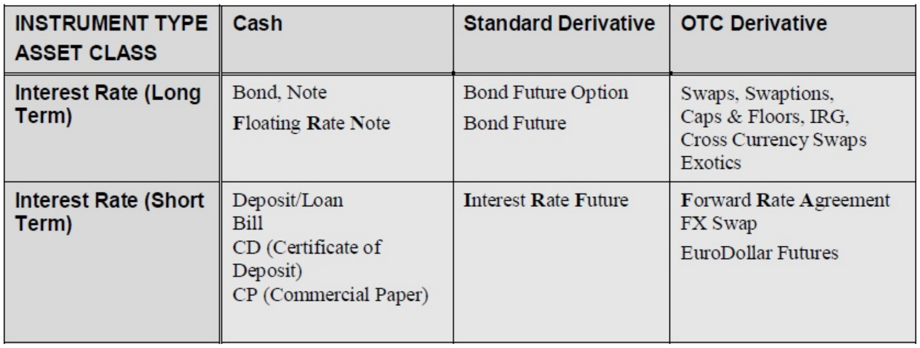
Since the beginning of the crisis that started with the crash of Lehman-Brothers in 2008 the financial world has started to change very sharply. The credit market collapsed and central banks all over the world decided to lower interest rates levels and power up their printing machine engines to try to fix the situation. This fact forced the rates down even more until we got to a situation no one could have imagined before, interest rates reached bellow-zero levels. That means someone will lend money to another person expecting to get a smaller amount back. A financial non-sense. From this moment on, the situation has widened and negative levels are deeper every month and are spreading to almost all developed economies.

Because negative rates where unthinkable, many models to estimate them were made according to the fact that they did not exist. They were based upon log-normal distribution, which makes negative rates impossible to handle. Therefore, nowadays they do not work adjusted to reality anymore. However, there exists some models (usually Gaussian-models) that let rates reach negative levels because they rely on normal distribution. Hull-White's Vacicek extension is one of them and this dissertation will try to explain how to utilize it to price Swaptions as these instruments can be used to calibrate the model against market data so it can price interest rate derivatives according to reality.

Thus, interest rates and their derivatives will be introduced in the first part of this work, followed by the explanation of the Hull-White model and two different ways to find its solution (analytically and through numerical methods). Afterwards, the Black-76 model will be shown as it can be used to calibrate Hull-White. Finally the calibration method and the results will be presented. Practical examples will be given in every section so the reader can see the usefulness of this model by him/herself.

2 Interest rate derivatives

Interest rates are probably the most important financial instrument as they determine the life of millions of people around the globe. Therefore, we can find a bunch of different interest rate derivatives in the market. In Figure 1 we find a classification of the most important fixed income securities. The table has been taken from Röman, J. 2015[1].



INSTRUMENT TYPE ASSET CLASS	Cash	Standard Derivative	OTC Derivative
Interest Rate (Long Term)	Bond, Note Floating Rate Note	Bond Future Option Bond Future	Swaps, Swaptions, Caps & Floors, IRG, Cross Currency Swaps Exotics
Interest Rate (Short Term)	Deposit/Loan Bill CD (Certificate of Deposit) CP (Commercial Paper)	Interest Rate Future	Forward Rate Agreement FX Swap EuroDollar Futures

Figure 1: Classification of interest rates derivatives. Röman, J. 2015[1]

Because interest rates usually mean periodic payments, to price them we have to study their cash flows. Hence, the first thing to keep in mind when valuing cash flows is the moment in time in which they are treated, since we can analyze both their future and present value. To evaluate the present value of a cash flow that will happen in the future, we need to discount it first. The most important methods to discount interest rates are:

- Simple compounding:

$$p(t) = \frac{1}{1 + tr_{simple}(t)}$$

- Periodically compounding:

$$p(t) = \frac{1}{\left(1 + \frac{r_f(t)}{f}\right)^{ft}}$$

- Continuous compounding:

$$p(t) = e^{-rt}$$

Where $p(t)$ is the prize of a zero coupon at time t, f is the frequency of payments per year and r is the interest rate. If we have enough instruments from the market with different maturities we can create a discount curve. A discount curve is a function of time in which we have the future value of a product over time and can be used to discount cash flows. The discount curve is created out of the most liquid interest rate instruments that we could find in the market. Usually these are deposits from 1 day to 3 months, FRA (Forward Rate Agreement) from 3 months to 2 years and Swaps from 2 years on. The formulas to discount these 3 types of products are as follows:

- Over-Night Deposit:
$$D_{O/N} = \frac{1}{1 + r_i^{O/N} \frac{d_{O/N}}{360}}$$
- Tomorrow-Night Deposit:
$$D_{T/N} = \frac{D_{O/N}}{1 + r_i^{T/N} \frac{d_{T/N}}{360}}$$
- Deposits:
$$D_i = \frac{D_{T/N}}{1 + r_i^{par} \frac{d_i}{360}}$$
- FRA:
$$D_i^{FRA} = \frac{D_{i-1}^{FRA}}{1 + r_i^{FRA} \frac{d_i^{FRA}}{360}}$$
- Swaps:
$$D_T = \frac{D_{T/N} - r_T^{par} \sum_{t=1}^{T-1} Y_t D_t}{1 + Y_T r_T^{par}}$$

With D being the discount factors of each product, d the length of the product in days, and r_{par} the Swap rate.

The discount curve for the data in Figure 24 of the Appendix 5 is shown in Figure 21 of the Appendix 5.

The process of creating an interest rate (yield) curve is called bootstrapping. Taking the logarithm of the discount factors with negative sign and dividing it by the amount of days that the instrument lasts we can get the Zero rates. Thus, we can construct an interest rate curve as the one shown in Figure 2, where we have the Zero rates, Swap rates and Forward rates for the data given in Figure 24 of the Appendix 5.

2.1 Bonds, FRN and FRA

Bonds are the most common fixed-income security. They are defined as a contract in which the bond buyer agrees to lend an amount of money to the issuer in exchange to receive interest payments periodically (known as coupons) and the principal at maturity. The price of a bond as a function to the yield-to-maturity is shown as:
$$P = \frac{N}{(1+y)^T} + \sum_{i=1}^{n} \frac{C}{(1+y)^{t_i}}$$

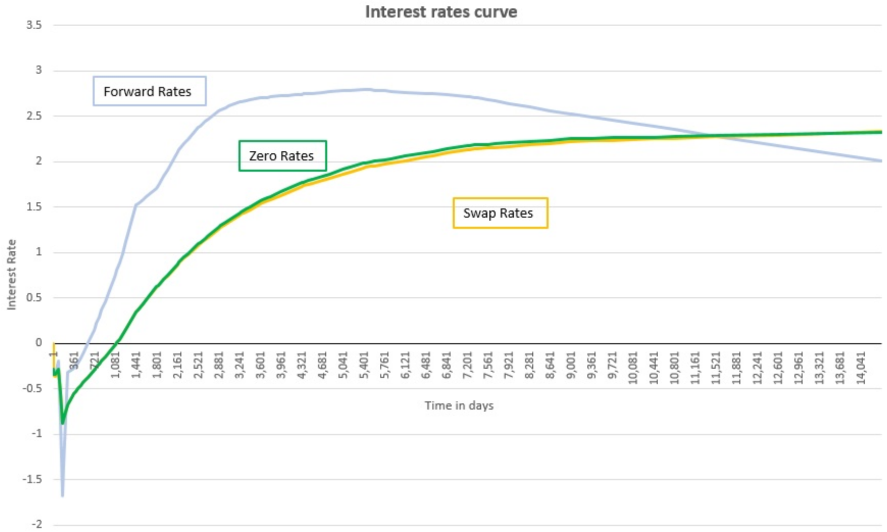


Figure 2: Forward rates, Zero rates and Swap rates curve of the data given in Figure 24 of Appendix 5

where N is the nominal, y is the yield-to-maturity and C is the amount of the coupon ($C = cN$). The present value of a bond is described by the following equation:

$$PV(y,f) = \left(1+\frac{y}{f}\right)^{-Tf}\left\{N+\frac{C}{y}\left[\left(1+\frac{y}{f}\right)^{M}-1\right]\right\}$$

with f equal to the frequency of the payment per year and M being the number of remaining cash flows and T the time to maturity.

There exist many different kinds of bonds like callable/putable (options on bonds), convertible (issued by companies, can be converted into stock), bullet bonds, etc. However, as most of interest rates derivatives can be seen as a series of Zero-Coupons bond, Swaps among them, in this dissertation we will focus on the study of this kind of bonds.

A Zero-Coupon bond is a bond that has just one single payment at maturity made out of the principal plus (or minus) the interest rates, none intermediate coupons will happen. They are of great importance as they can be used to discount and price interest rates derivatives. The idea under Zero-Coupon pricing is taking all individual cash flows as if they where Zero-Coupon bonds. The evaluation of these cash flows is made using a yield curve or the discount function. This way of valuing interest rates derivatives is very popular especially when pricing OTC (Over The Counter) instruments. In Figure 3 we can see how a bond with several coupons can be seen as a sum of Zero-Coupon bonds.

Furthermore, another important interest rates instruments are notes. A

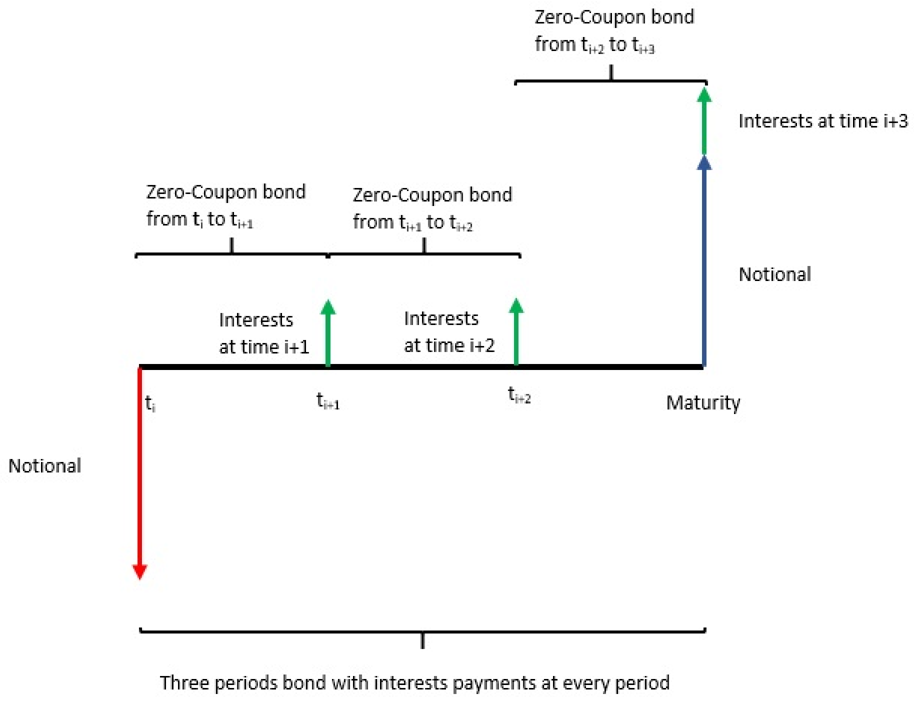


Figure 3: Bond scheme with several coupons vs Zero-Coupon bonds

note is similar to a bond with the difference that it has a shorter lifetime. Notes lifetime is usually between one and ten years, while bonds life is normally longer. Within notes, we can find the so-called Floating Rate Notes. The FRN are a very used financial instrument. They are quoted as a percentage of a notional, like bonds, however, FRN have reset dates. Every coupon date, the rate is reset in line with a money market reference, typically *LIBOR*, plus a fixed spread. The credit quality of the issuer is of great importance when pricing FRN, since if we wanted to trade it in the secondary market, the fixed spread should be shifted in case the rating of the issuer drops. Thus, when discounting, we have to keep in mind the issuer discount curve $p(0, i)$, where i is the current time.

$$P = \sum_{i=1}^{N} [L(i-1, i) + S] p(0, i) + p(0, N)$$

will be the price of an FRN, where we discount the sum of *LIBOR* $L(i-1)$ at the previous coupon payment plus the spread S by the issuer discount to next coupon date i. We can estimate $p(0, i)$ by the following formula:

$$p(0, i) = \frac{p(0, i-1)}{1 + L(i-1, i) + Q}$$

where Q is a measure of the credit quality of the issuer known as *par floater spread* which changes over time. For the case where $Q = S$ we have that the FRN is traded at par. The spread over the *LIBOR* paid in order to make the FRN price at par is known as *Discount Margin* or *Effective LIBOR spread* and is calculated by iterations looking for the value of M in:

$$p = \frac{(L^{next} + S) + \sum_{i-1}^{N} \frac{L+S}{(1+L+M)^i} + \frac{1}{(1+L+M)^N}}{1 + L^* + M}$$

where p is the bond price, L^* is the stub *LIBOR* coupon to next coupon date, L^{next} is the next *LIBOR* payment and M is the discount margin which value we are looking for.

The last instruments we will study in this section are the Forward Rate Agreements. FRA are interest rate derivatives starting in the future without exchanging the principal. Thus, just differences on interest rates are traded when the instrument matures. They are an agreement to borrow (or lend) a notional for a period of time at an agreed interest rate. This interest rate can be lower or higher than the market rate set by the contract and so, it can be used to hedge or speculate. For example, if the rate of the FRA is set at 5% in 6 months on *LIBOR* for 3 months and *LIBOR* stays at a lower level, lets say 4%, the issuer of the FRA must pay the difference to the buyer for a 3 months period. They are very similar to Swaps (presented in Section 2.2) with the contrast that FRA pay the difference of interest rates only once, at maturity. The value of an FRA is calculated as:

$$P = N\tau \frac{L(t,T) - K}{1 + \tau L(t,T)}$$

where t and T are start of the FRA and maturity respectively, K is the strike to be paid at t_2, and $\tau = T - t$ in days (taking a year as 360 days). FRA at par ($K = L$) are standardized nowadays because of its importance and are set as three month contracts between the famous IMM-days (third Wednesday of March, June, September and December).

2.2 Swaps

Swaps are among the most popular financial products that can be found in the market nowadays and are the most traded ones by far when talking about fixed-income derivatives. A Swap is an agreement by which two counter-parties exchange financial instruments periodically, mainly cash flows on a notional. In this case, one party agrees to pay the other party a cash flow equal to a fix rate on the notional and the other party pays floating rates cash flows (usually based on an interest rate index like *LIBOR* or *EURIBOR*) on the same notional in exchange. It is easy to see how one can use it either to cover against possible fluctuations on interest rates or to speculate under the idea of predicting these fluctuations. Generally these agreements are taken by companies or financial institutions with the idea of hedging possible losses from rates.

The present value of a Swap can be easily calculated discounting the payments and since commonly the notional is the same for both counter-parties, it will not be exchanged. These payments will be done periodically at previously determined dates called *reset dates* (monthly, every semester, annually, etc). To be able to price them, we have to calculate both the discounted fixed payments and the floating ones. However, as we can not determine the floating payments at the start of the contract, we have to estimate them calculating the forward rate for each respective payment date of bonds traded in the market with similar maturities.

Each of these cash flows is discounted by the Zero-Coupon rate for the date of the payment given by the yield curve from the market. Therefore, we can treat the interest rate Swap as a series of Zero-Coupon bonds from a reset date to the following one. The sum of all these Zero-Coupons will be the present value of the floating leg of the Swap.

Furthermore, the Swap fixed rate will be set at a level that will make the fixed payments have the same present value as that of the floating leg. Thus, at the time of the agreement, there will be no advantage for any of the parties. However, during the life of the Swap, the PV of the floating flows will vary as the interest rates fluctuate and both legs will not have the same value when the Swap matures. The structure of a Swap with 2 payment dates starting in the future is shown in Figure 4.

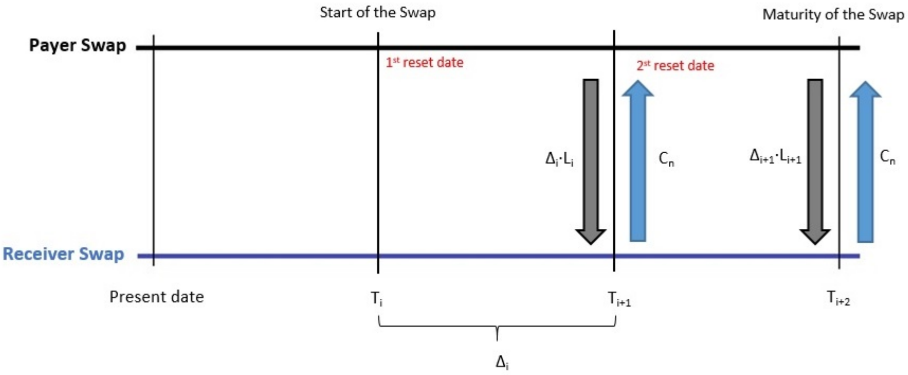


Figure 4: Payment structure of a 2 periods Swap.

Letting Δ_i be the length of the period between T_{i-1} and T_i in days and D_{T_i} be the discounting factor at time T_i that is calculated as follows

$$D(T_i) = \frac{D(T_{i-1})}{1 + \Delta_i F_i}$$

the forward rate of a single currency Swap for the period $[T_{i-1}, T_i]$ will be:

$$F_i = \frac{D(T_{i-1})/D(T_i) - 1}{\Delta_i}$$

where T_i represent the reset days of the Swap. Now, letting L_i be the market rate (*LIBOR*) we can see that the value today of the floating cash flow $\Delta_i L_i$ is:

$$\Delta_i F_i D(T_i) = \Delta_i \frac{D(T_{i-1}) - D(T_i)}{\Delta_i D(T_i)} D(T_i) = D(T_{i-1}) - D(T_i)$$

Thus, if we add up the values for all the life of the Swap we will get the whole floating leg as:

$$\sum_{i=1}^{n} \Delta_i F_i D(T_i) = \sum_{i=1}^{n} D(T_{i-1}) - D(T_i) = D(T_0) - D(T_n) = 1 - D(\bar{T}_n)$$

And the value of the fixed leg with rate C is

$$C \sum_{i=1}^{n} \bar{\Delta}_i D(\bar{T}_i)$$

Thus, to get the fair Swap rate at maturity $\bar{T}_n$, we must find the value C_n for the following equality

$$C_n \sum_{i=1}^{n} \bar{\Delta}_i D(\bar{T}_i) = 1 - D(\bar{T}_n)$$

This rate that equals both legs at the time when the contract is made is known as *par rate*.

Lets now study the case when we use bonds for discounting. Thus, we will be using the forward rate

$$f(t_{i-1}, t_i) = \frac{p(0, t_i) - p(0, t_{i-1})}{\Delta_i p(0, t_i)}$$

and following the steps given above the whole value of the floating leg is as follows:

$$\sum_{i=n+1}^{N} \Delta_i f(t, T) p(t, T_{i-1}) = \sum_{i=n}^{N-1} [p(t, t_i) - p(t, t_{i+1})] = p_n(t) - p_N(t)$$

On the other hand the fixed leg will be equal to

$$C \sum_{i-n+1}^{N} \Delta_i p_i(t)$$

Hence, the total value of the payer Swap is

$$p_n(t) - p_N(t) - C \sum_{i=n+1}^{N} \Delta_i p_i(t)$$

that must be equal to 0 in order to be at par. Therefore, the forward Swap rate $R_n^N(t)$ is equal to

$$R_n^N(t) = C = \frac{p_n(t) - p_N(t)}{\sum_{i=n+1}^{N} \Delta_i p_i(t)} \tag{1}$$

The process $S_n^k = \sum_{i=n+1}^{N} \Delta_i p_i(t)$ is known as the *accrual factor or value of a basis point* (can be found under the acronyms *DV01* and *PV01*).

2.3 Swaptions

A Swaption is an option to enter into a Swap. Similar to stock options, there exists both call and put possibilities, however, here they are defined as payer and receiver Swaptions. A payer Swaption is the right but not the obligation of entering into a Swap in a given date, paying the fixed rate (Strike) and receiving the floating rate. On the other hand, a receiver Swaption is the opposite, the right but not the obligation to enter into a Swap receiving the fixed rate and paying the floating rate in a given date. There exists also both European and American Swaptions. Like in the European options, a Swaption can only be exercised at maturity. However, in contrast to options, an American Swaption can just be exercised on cash flow dates, not continuously. There also exists a hybrid of both American and European Swaptions that can only be exercised at reset dates, these are called mid-Atlantic or Bermudan.

We have then that a payer Swaption is a put option on a fixed rate bond with strike price equal to the face value of the bond. On the other hand, a receiver Swaption is a call option on a fixed rate bond with strike price equal to the face value of the bond.

2.4 Interest rate derivatives pricing

To understand the mathematical procedure of pricing interest rate derivatives, we have to understand that the short rate process is typically set under the P probability. The short rate process is defined as follows:

$$dr(t) = \mu(t, r(t))dt + \sigma(t, r(t))dW(t) \tag{2}$$

where dW is a Wiener process. dW will be used every time we refer to a Wiener process from here on.

We will find that (2) is determined by the General Term Structure equation (3), which defines the short rates behavior during time.

$$\begin{cases} \dfrac{\partial F^T}{\partial t} + \{\mu(t,r) - \lambda(t,r)\sigma(t)\}\dfrac{\partial F^T}{\partial r} + \dfrac{1}{2}\sigma^2\dfrac{\partial^2 F^T}{\partial r^2} - r(t)F^T = 0 \\ \\ F(r,T,T) = \Phi(r) \end{cases} \tag{3}$$

In this partial differential equation we find that the drift of the short rate ($\mu - \lambda\sigma$) is defined under the martingale measure Q. Under martingale probability measures, if we have a stochastic process X with information known up to time t, the expected future value of $X(t+s)$ will be equal to $X(t)$. Therefore, we will need to specify the r-dynamics (2) (that are set under P) on this martingale measure Q. This method is called martingale modeling and we can find a bunch of different models of this kind.

2.5 Martingale modeling

When martingale modeling, it happens that some of these models have a particular term structure that will eliminate most of the mathematical difficulty. These models are said to have an Affine Term structure. Therefore, the PDEs we will have to front when dealing with ATS models are easier to solve.

2.5.1 Affine Term structure

First, we will start describing what an Affine Term structure is:

If the term structure $\{p(t,T); 0 \leq t \leq T; T > 0\}$ has the form $p(t,T) = F(r(t),t,T)$ where

$$F(r,t,T) = e^{A(t,T)-B(t,T)r} \tag{4}$$

and A and B are deterministic functions, then the model possess an Affine Term structure. A model is said to have an ATS if the continuously compounded short rate $R(t,T)$ is an ATS of the short rate $r(t)$. This means that the process is linear plus a constant like:

$$R(t,T) = \alpha(t,T) + \beta(t,T)r(t)$$

In this model α and β are deterministic functions of time.

$$\begin{cases} \alpha(t,T) = \dfrac{A(t,T)}{T-t} \\[2ex] \beta(t,T) = \dfrac{B(t,T)}{T-t} \end{cases}$$

Since having an ATS is so important when martingale modeling, we will describe how to know if we have an Affine Term structure. Assuming that we have (2) under the Q probability measure, the process to check if our model posses an ATS for the case of a Zero-Coupon bond is as follows (demonstration of the calculations can be found in Röman 2015 pp[257-286]):

Substituting (4) in (3), we get

$$\frac{\partial A(t,T)}{\partial t} - \left\{1 + \frac{\partial B(t,T)}{\partial t}\right\} r(t) - \mu(t,r(t))B(t,T) + \frac{1}{2}\sigma^2(t,r(t))B^2(t,T) = 0$$

Furthermore, applying the Zero-Coupon bond boundary condition $F(r,T,T) = 1$ in (4) we get the boundary condition for the equation above as $1 = e^{A(T,T)-B(T,T)r}$. Thus $A(T,T) - B(T,T)r$ must be 0 and then

$$A(T,T) = B(T,T) = 0$$

We can already realize how the calculations are simplified when using ATS. Furthermore, analyzing that result, we see that if μ and σ^2 are both affine, the equation becomes a Separable Differential Equation. Thus, we can assume

$$\begin{cases} \mu(t,r) = a(t)r + b(t) \\ \\ \sigma^2(t,r) = c(t)r + d(t) \end{cases}$$

and so, substituting above, we have:

$$\frac{\partial A(t,T)}{\partial t} - \left\{1 + \frac{\partial B(t,T)}{\partial t}\right\} r(t) - \{a(t)r + b(t)\}B(t,T) + \frac{1}{2}\{c(t)r + d(t)\}B^2(t,T) = 0$$

that can be separated as

$$\begin{cases} \dfrac{\partial A(t,T)}{\partial t} - b(t)B(t,T) + \dfrac{1}{2}d(t)B^2(t,T) = 0 \\ \dfrac{\partial B(t,T)}{\partial t} + a(t)B(t,T) - \dfrac{1}{2}c(t)B^2(t,T) = -1 \\ A(T,T) = B(T,T) = 0 \end{cases} \tag{5}$$

Solving this Separable Differential Equation we can analyze the Q-dynamics on $dr(t)$ and therefore, find the derivatives prices.

2.5.2 ATS models

We can find a bunch of different ATS models. Among the most important ones we can find, along with their short rate process, the followings:

- Vasicek: $dr = (b - ar)dt + \sigma dW$
- Cox-Ingeroll-Ross (CIR): $dr = a(b - r)dr + \sigma\sqrt{r}dW$
- Black-Derman-Toy: $dr = a(t)rdt + \sigma(t)rdW$
- Ho-Lee: $dr = a(t)dt + \sigma(t)dW$
- Hull-White: $dr = (b(t) - a(t)r)dt + \sigma(t)dW(t)$

In this dissertation we will focus on the study of the latter.

3 Hull-White Model

This model was developed by John Hull and Allan White in 1990 in their article *"Pricing Interest-Rate Derivative Securities"*[3]. As they defined it, it is an extension of the Vasicek model. However, in H-W the parameters are time-dependent, what increases the possibility of calibrating them with respect to market data. The Hull-White model is growing in importance lately due to the reason that it allows the interest rates to be negative while we will not find this feature in many other models. In the past, when the people surrounding the financial world thought of negative interest rates as something impossible, it was criticized for the same reason. However, nowadays reality is quite different and negative rates are all over the market. Furthermore, many specialists on the matter say they are here to stay.

The explanation of why negative interest rates are possible in this model is that the short rate in Hull-White is normally distributed (it is a so-called Gaussian model). On the other hand, for example, Black-76 (the most commonly utilized model) has a log-normal distribution. Hence, in these models negative values for the rates are not allowed since the logarithm of a non-positive number is not defined. We find that in Black's model volatility is defined as:

$$dF = \sigma_B F dW \qquad F(0) = f$$

where σ_B is the implied Black volatility. As we have log-normality we will need to calculate $\log(F/K)$ for d_1 and d_2 in the Black formula for pricing interest rates derivatives than can be found in Black, F. 1976 [5]. Therefore, the model is unusable for non-positive values of F. Nevertheless, in Gaussian models:

$$dF = \sigma_N F dW \qquad F(0) = f$$

where σ_N is the normal volatility. Furthermore, another remarkable feature of the Hull-White model is that it allows us to use trinomial trees (lattice), as well.

The Hull-White stochastic process is defined as

$$dr = (\theta(t) - a(t)r)dt + \sigma(t)dW(t)$$

It is easy to see that it follows the dynamics presented in (2). Furthermore, it is an special case which includes mean-reversion. Here θ is a deterministic function of time and will be calibrated against the theoretical yield curve. On the other hand, a and σ should be chosen to reflect the current and future volatilities of the short-term-interest rate. So like in Vasicek model they can be fitted to the current term structure of interest rates and the current term structure of interest-rate volatility respectively. a is the mean reverting parameter, and is defined in a way in which the drift of the process will be negative for values of r greater than θ/a and vice-versa. We can appreciate it better recalling the Term Structure equation with the Zero Coupon boundary condition and the Hull-White r-dynamics:

$$\begin{cases} \dfrac{\partial F^T}{\partial t} + \{\theta(t) - a(t)r(t)\}\dfrac{\partial F^T}{\partial r} + \dfrac{1}{2}\sigma^2 \dfrac{\partial^2 F^T}{\partial r^2} - r(t)F^T = 0 \\ \\ F(r,T,T) = 1 \end{cases} \tag{6}$$

We see that $\mu(t,r) - \lambda(t,r)\sigma(t) = \theta(t) - a(t)r(t)$ and so, the drift includes the market price of risk and the volatility that should be calibrated against real data. The analytical solution for the short rates in this model is as follows:

$$r(t) = r(s)e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\theta(u)du + \sigma \int_s^t e^{-a(t-u)}dW(u)$$

where

$$\theta(t) = f(0,T) + af(0,T) + \frac{\sigma^2}{2a}(1 - e^{-aT})$$

and the Zero-Coupon bond price:

$$p(t,T) = A(t,T)e^{B(t,T)r(t)} \tag{7}$$

$$A(t,T) = \frac{p(0,T)}{p(0,t)} exp\left\{B(t,T)f(0,t) - \frac{\sigma^2}{4a}B(t,T)^2(1 - e^{-2at})\right\}$$

$$B(t,T) = \frac{1}{a}(1 - e^{-a(T-t)})$$

with $f(0,t)$ being the forward rate and $p(0,t)$ the price of a bond starting today with maturity at time t. Calculations to these solutions can be found in Röman 2015 [1] p.309-316.

3.1 Solving the Hull-White PDE to price Zero-Coupon bonds through the Finite Difference Method

In this dissertation we will show the procedure to price Zero-Coupon bonds solving the Hull-White PDE through the Finite Difference Method, in particular, Crank-Nicolson.

There are three types of finite difference methods to solve PDE's. The explicit difference method, that follows backward differences to approximate the derivatives; the implicit difference method, that follows forward differences; and the Crank-Nicolson (mixed) method that combines both of them as can be seen in Figure 5 taken from Röman, J. 2014[4]. Therefore, in the mixed method, we use information both from t and $t - \Delta t$ and furthermore, the accuracy is greater.

- Forward difference: $\dfrac{\partial f(x,y)}{\partial y} \cong \dfrac{f(x, y+\delta_y) - f(x,y)}{\delta_y}$
  with rate of convergence $O(\Delta x + \Delta y)$

- Backward difference: $\dfrac{\partial f(x,y)}{\partial y} \cong \dfrac{f(x,y) - f(x, y-\delta_y)}{\delta_y}$
  with rate of convergence $O(\Delta x^2 + \Delta y)$

- Central difference: $\dfrac{\partial f(x,y)}{\partial y} \cong \dfrac{f(x,y+\delta_y)-f(x,y-\delta_y)}{2\delta_y}$
  with rate of convergence $O(\Delta x^2 + \Delta(y/2)^2)$
- Second central difference: $\dfrac{\partial^2 f(x,y)}{\partial y^2} \cong \dfrac{f(x,y+\delta_y)-2f(x,y)+f(x,y-\delta_y)}{\delta_y^2}$
  with rate of convergence $O(\Delta y^4)$

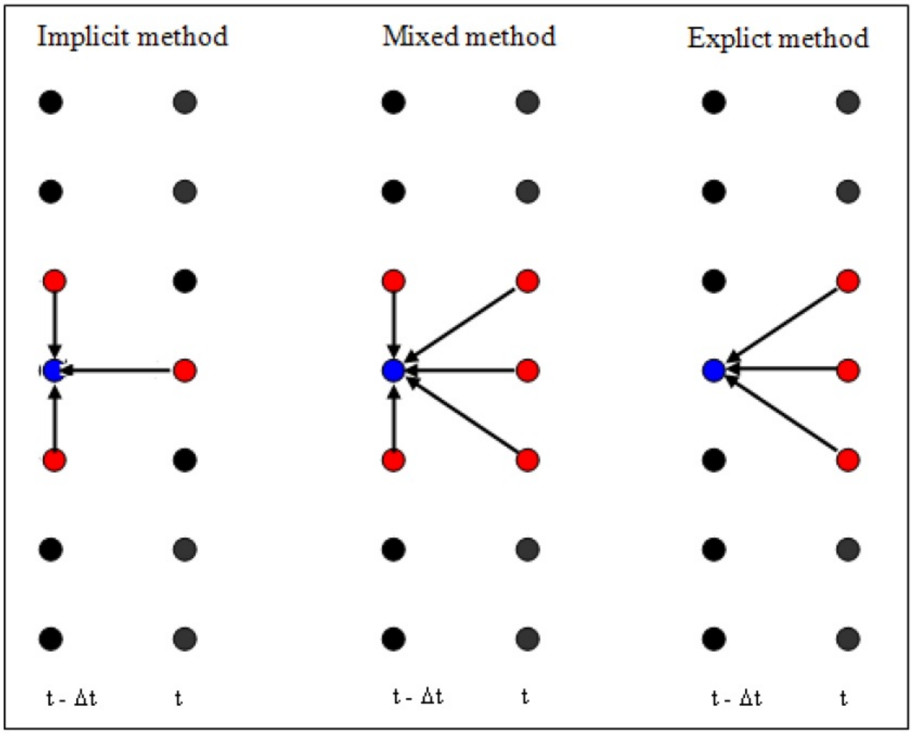


Figure 5: Finite difference methods information schema, Röman J. 2015, [4].

To solve the PDE in (6) through Crank-Nicolson we then need to define a grid with the rates as rows and time as columns. The idea behind this grid is to discretise these two variables in N pieces of h length equal to dr and T pieces of k length equal to dt. Therefore, applying the central and second central differences to both variables we get

$$u_{ij} = u(r_i, t_j)$$

$$\frac{\partial u(r_i,t_j)}{\partial t} = \frac{u_{ij+1} - u_{ij-1}}{2k}$$
$$\frac{\partial u(r_i,t_j)}{\partial r} = \frac{u_{i+1j} - u_{i-1j}}{2h}$$

$$\frac{\partial^2 u(r_i, t_j)}{\partial r^2} = \frac{u_{i+1j} + u_{i-1j} - 2u_{ij}}{h^2}$$

Now, applying these Crank-Nicolson equalities to the Hull-White term-structure equation (6) we have:

$$\frac{\partial u_{ij}}{\partial t} = r_i u_{ij} - \{\theta(t) - a(t)r(t)\}\frac{u_{i+1j} - u_{i-1j}}{2h} - \frac{1}{2}\sigma^2 \frac{u_{i+1j} + u_{i-1j} - 2u_{ij}}{h^2}$$

that can be realigned to give the multipliers of each u_{ij}

$$\frac{\partial u_{ij}}{\partial t} = \left(-\frac{\sigma^2}{2h^2} + \frac{\{\theta(t) - a(t)r(t)\}}{2h}\right)u_{i-1j} + \left(\frac{\sigma^2}{h^2} + r_i\right)u_{ij} - \left(\frac{\sigma^2}{2h^2} + \frac{\{\theta(t) - a(t)r(t)\}}{2h}\right)u_{i+1j}$$

Thus, letting:

$$\begin{aligned} P_d &= -\frac{\sigma^2}{2h^2} + \frac{\{\theta(t) - a(t)r(t)\}}{2h} \\ P_m &= \frac{\sigma^2}{h^2} + r_i \\ P_u &= -\frac{\sigma^2}{2h^2} - \frac{\{\theta(t) - a(t)r(t)\}}{2h} \end{aligned} \tag{8}$$

we can perform the following matrix where $t_j = jk; j = 0, 1, 2, ..., T$ and $r_i = ih; i = 0, 1, 2, ..., N$

$$A = \begin{pmatrix} P_m(r_0, t_j) & P_u(r_0, t_j) & 0 & 0 & 0 & .... & 0 \\ P_d(r_1, t_j) & P_m(r_1, t_j) & P_u(r_1, t_j) & 0 & 0 & .... & \vdots \\ 0 & P_d(r_2, t_j) & P_m(r_2, t_j) & P_u(r_2, t_j) & 0 & .... & \vdots \\ 0 & 0 & P_d(r_3, t_j) & P_m(r_3, t_j) & P_u(r_3, t_j) & .... & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & .... & 0 & P_d(r_N, t_j) & P_m(r_N, t_j) \end{pmatrix}$$

that will define the behavior of the rates every time-step k passes. Letting a vector $\mathbf{u}$ represent the rates at time j, following Crank-Nicolson, we will have the functions that define increments in rates from j to $j+1$ as:

$$\frac{\mathbf{u}_{j+1} - \mathbf{u}_j}{k} = \frac{\mathbf{A}(t_j)\mathbf{u}_j + \mathbf{A}(t_{j+1})\mathbf{u}_{j+1}}{2}$$

that will be solved as:

$$\mathbf{u}_j = \left(\mathbf{I} + \frac{k}{2}\mathbf{A}(t_j)\right)^{-1}\left(\mathbf{I} - \frac{k}{2}\mathbf{A}(t_{j+1})\right)u_{j+1}$$

Reprocessing this calculations starting from maturity backwards for all time-steps and organizing $\bar{u}$ in a matrix we will get the solution for the PDE. Note that in the case of Zero-Coupon bonds we should start with the boundary condition $\bar{u}_T = 1$ for all i since the value of a Zero-coupon bond at maturity will equal 1. The code of a *Python* application to compute this FDM process can be found in the Appendix 1.

3.1.1 Boundary conditions

As we are pricing Zero-Coupon bonds, at maturity our derivatives will be worth 1. Therefore, the first boundary condition for this system of equations will be $u_T = 1$ for all rates.

On the other hand, $\dfrac{\partial^2 u(r_i, t_j)}{\partial r^2} = 0$ is used as boundary conditions at the first step of the grid ($P_d(r_0, t_j)$, $P_m(r_0, t_j)$ and $P_u(r_0, t_j)$).

3.2 Practical example of Zero-Coupon bonds pricing under Hull-White by Crank-Nicolson Finite Difference Method using Python

In this section we will show a practical example of the process to solve the Hull-White PDE (6) by the Crank-Nicolson Finite Difference Method in *Python*. The whole code to compute this application can be found in Appendix 1. The set values for this practical example are $\sigma = 0.02$, $\theta = 0.02$, $a = 0.01$ and the maximum and minimum values to define the C-N grid of r are -0.015 and 0.3 respectively. The grid has been created as it has a size of 63x333 in which we plot the rate level on the vertical axis and the time to maturity on the horizontal axis. To define the pieces in which the grid will be divided we have used the following formulas $M = \dfrac{r_{max} - r_{min}}{dr}$ and $N = \dfrac{1}{dt}$ where dr and dt are the size of each piece for rates and time and M and N are the number of pieces in which the axis are divided. After defining the boundary conditions given in Subsection 3.1.1 we can create the matrix $A(T-1)$ in Section 3.1 applying (8) for each time-step. The first rows and columns of this matrix are shown in Figure 22 of the Appendix 5. Once this step is performed, we set $A(T-2)$ to be able to calculate u_{T-2}, that will be the second column of our Crank-Nicholson grid (the first column is set equal to 1 as the boundary condition at maturity). Performing this process from maturity time to the first time-step of the bond life we complete the Crank-Nicholson grid (Figures 6 and 7). This grid represents the different values for Zero-Coupon bonds for rates range $(-0.015, 0.3)$ from maturity to bond start. Each vector of the grid can be used as a discounting function to discount financial instruments for its respective risk-free rate.

3.3 Pricing European Swaptions under the Hull-White model

Following Jamshidian [6] if we have *"a term structure that is completely determined by the value of the instantaneous interest rate $r(t)$ that follows a mean-reverting Gaussian process, [...] an option on a portfolio of pure discount bonds decomposes into a portfolio of options on the individual discount bonds in the portfolio."* Hence, we can think of the European Swaption as a sum of European options on Zero-Coupon bonds. This method is called *Jamshidian's decomposi-*

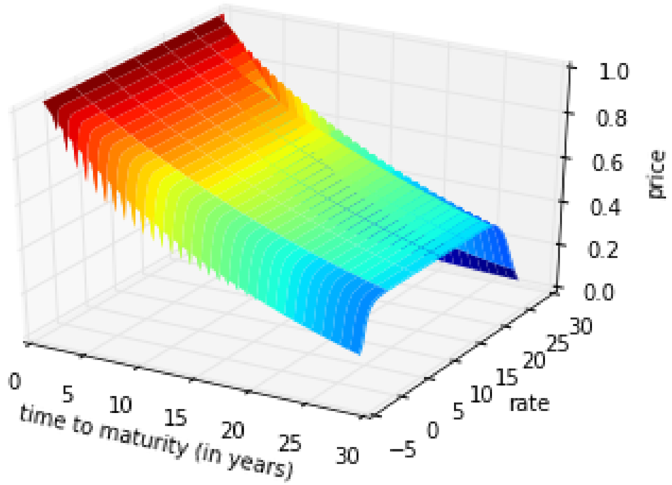


Figure 6: Crank-Nicolson grid for Zero-Coupon bonds

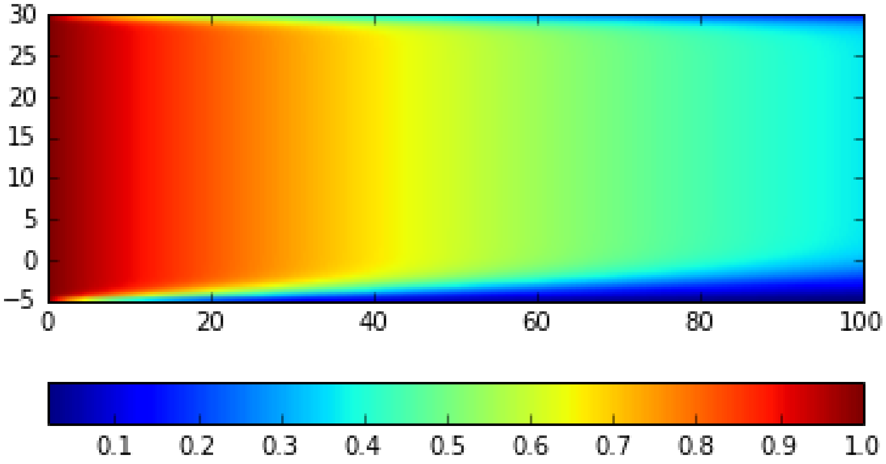


Figure 7: Crank-Nicolson heat map for Zero-Coupon bonds

tion in his honor, and is mathematically explained as follows:

$$\max\left[\sum_{i=1}^{n} C_i p(r,t,s_i) - K, 0\right] = \sum_{i=1}^{n} C_i \max\left[p(r,t,s_i) - K_i, 0\right] \tag{9}$$

The method is the following:
Consider a European Swaption with strike rate K, maturity T and nominal L with payment dates T_i for $i = 1, ..., n$. Since the value of a floating rate bond is worth its face value, we can take it as an option on a bond paying $C_i = K_{s_i}$ with strike price L. This option will be exercised when $r(T) < r*$ where r^* is the solution to

$$L = \sum_{i=1}^{n} C_i p(r^*, T, T_i)$$

This means that L is taken as a sum of discounted flows C_i for a short rate r^* at time T. To find the value of r^* we have to discount all the Swap cash-flows, sum them up and let the sum be equal to the notional. We have to use the Newton-Raphson algorithm to follow this calculation r^*, that will be the correspondent level of r for that equality to happen. The Newton-Raphson algorithm has been taken from the imported Python library *scypi.optimization.* Once r^* is found, we have to calculate the discounted cash-flows substituting it in the discounting factor $e^{-r\Delta t}$ and let them be the strike prices of our call options on Swaps. Afterwards, we calculate the payoff of the call options on the cash-flows with the strike price mentioned and sum them up. The payoff of this sum of options will thus be

$$\max \left[\sum_{i=1}^{n} C_i p(r, T, T_i) - L, 0 \right]$$

which using (10) we find is equivalent to

$$\sum_{i=1}^{n} C_i \max \left[p(r, T, T_i) - L_i, 0 \right]$$

where $L_i = p(r^*, T, T_i)$. Therefore, the Swaption is calculated as the sum of n options on discounted bonds with the exercise price of the i^{th} option equal to L_i.

After the price of a European Swaption under the Hull-White model is calculated, we have to calibrate it against the market data. Usually this calibration method is carried out taking the target prices to match the ones given by the Black-76 or Normal-Black models and trying to get as close as possible to them. Further information on this method can be found in sections 4 and 5.

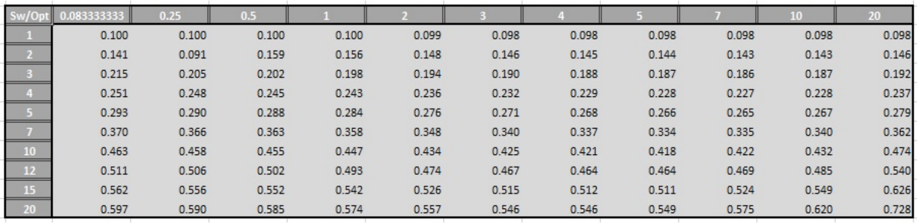
3.3.1 Practical example of Pricing European Swaptions under Hull-White analytical formula using Python

A practical example on pricing European payer Swaptions under Hull-White will using the analytical formula be given in this section. The whole code of the *Python* application can be found in the Appendix 2. The inputs have been set as $r = 0.01\%$, $\sigma = 1\%$, $a = 0.05$, Swap tenors=$\{1, 2, 3, 4, 5, 7, 10, 12, 15, 20\}$ and Swaption maturities=$\{1/12, 4/12, 6/12, 1, 2, 3, 4, 5, 7, 10, 20\}$ in years. Frequency has been let as one per year to simplify calculations.

As it has been presented above, the Hull-White analytical solution is given by (7). The first step is therefore, to calculate the discount factors applying it to the Zero bonds that are given in the Figure 24 of the Appendix 5. Once we have the discount factors we can calculate the floating and fixed legs of the Swaps and the Swap rates as it is shown in Subsection 2.2. Afterwards, we can find the forward Swap rates calculating

$$R_n^N(t) = \frac{p_n(t) - p_N(t)}{\sum_{i-n+1}^{N} \Delta_i p_i(t)}$$

At this point we have to apply Jamshidian as shown in the previous section. To do so, we calculate the cash-flows given by the discounted fixed rate payments and make them be equal to the strike rate. In this practical example, the strike rate has been set 1 basis point bellow the correspondent forward Swap rate. Then, applying the Newton-Raphson algorithm we get r^* and we can use it to get the value of the strike rate of our options in Swaps. The Swaption valuation is straightforward then, and it is done by summing up all the options. In Figure 8 we can find the European payer Swaption prices in percentage for the Swap tenors in rows and Swaption maturities in columns given above.



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	0.100	0.100	0.100	0.100	0.099	0.098	0.098	0.098	0.098	0.098	0.098
2	0.141	0.091	0.159	0.156	0.148	0.146	0.145	0.144	0.143	0.143	0.146
3	0.215	0.205	0.202	0.198	0.194	0.190	0.188	0.187	0.186	0.187	0.192
4	0.251	0.248	0.245	0.243	0.236	0.232	0.229	0.228	0.227	0.228	0.237
5	0.293	0.290	0.288	0.284	0.276	0.271	0.268	0.266	0.265	0.267	0.279
7	0.370	0.366	0.363	0.358	0.348	0.340	0.337	0.334	0.335	0.340	0.362
10	0.463	0.458	0.455	0.447	0.434	0.425	0.421	0.418	0.422	0.432	0.474
12	0.511	0.506	0.502	0.493	0.474	0.467	0.464	0.464	0.469	0.485	0.540
15	0.562	0.556	0.552	0.542	0.526	0.515	0.512	0.511	0.524	0.549	0.626
20	0.597	0.590	0.585	0.574	0.557	0.546	0.546	0.549	0.575	0.620	0.728

Figure 8: Swaption prices under Hull-White model in %

4 Black-76 model

The Black-76 model is a modification of the original Black-Scholes formula made by Black in 1976 [5] (hence its name) for valuing interest rate derivatives. It is among the most used models nowadays to price this kind of instruments due to its simplicity in comparison with more complicated models like the ones presented in Subsection 2.3. The European call and put prices under Black-76 are as follows:

$$C = e^{-rT}[FN(d_1) - KN(d_2)]$$
$$P = e^{-rT}[KN(-d_2) - FN(-d_1)]$$

where

$$d_1 = \frac{\ln(F/K) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

F refers to the underlying of the derivative and K to the strike price. It is easy to see that for the case of $F(T) = F_0 e^{rT}$ Black-76 becomes Black-Scholes, so we can think of it as a generalization of the original B-S formula.

However, this model has its weaknesses. As in Black-Scholes, it assumes that the underlying follows a log-normal distribution, which makes negative rates impossible to analyze.

4.1 Pricing European Swaptions under Black-76

Recalling the explanation given in Subsection 2.3 and the European call formula above, we can define a payer European Swaption as the contract given by:

$$P_n^N(t, R_n^N(T_n), K) = S_n^N(T_n)\max\{R_n^N(T_n) - K, 0\}$$

Letting S_n^N be the numeraire, a payer Swaption will be a call option on R_n^N with strike rate K. Under the Black-76 formula, the value of this contract is as follows:

$$P_n^N(t) = S_n^N(t)\{R_n^N(t)N(d_1) - KN(d_2)\} \tag{10}$$

where

$$d_1 = \frac{\ln\{\frac{R_n^N(t)}{K}\} + \frac{1}{2}\sigma_{n,N}^2(T_n - t)}{\sigma_{n,N}\sqrt{T_n - t}}$$
$$d_2 = d_1 - \sigma_{n,N}\sqrt{T_n - t}$$

with $\sigma_{n,N}$ being the *Black volatility.* As it is the volatility implied by the Black formula, it is also known as *implied Black volatility.*

Now, using (1) we can transform (10) to

$$P_n^N(t) = \frac{p_n(t) - p_N(t)}{R_n^N(t)}\{R_n^N(t)N(d_1) - KN(d_2)\}$$

and doing some algebra we can calculate the annuity as

$$\frac{p_n(t) - p_N(t)}{R_n^N(t)} = \frac{p(t,t_n)[1 - p(t_n,t_N)]}{R_n^N(t)} = \frac{p(t,t_n)\left\{1 - \dfrac{1}{(1 + f(t_n,t_N))^{t_N - t_n}}\right\}}{R_n^N(t)}$$

Thus, we can finally get the Black-76 formulas for payer (PS) and receiver Swaption (RS) as:

$$PS = \frac{1 - \dfrac{1}{(1 + C/m)^{\tau m}}}{C} e^{-rT}[CN(d_1) - KN(d_2)]$$

$$RS = \frac{1 - \dfrac{1}{(1 + C/m)^{\tau m}}}{C} e^{-rT}[KN(-d_2) - CN(-d_1)]$$

where we let C be the forward Swap rate between n and N, m be the amount of reset days per year, τ be the tenor of Swap (time between Swaption and Swap maturities),

$$d_1 = \frac{\ln\{\frac{C}{K}\} + \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}}$$

and

$$d_2 = d_1 - \sigma\sqrt{T}$$

4.1.1 Black-Normal model

As presented above, in a world with negative rates a log-normal model has its disadvantages. Therefore, we need to apply some transformations to the Black-76 model in order to be used with normality. This model is called Black-Normal and is basically Black-76 with Normal volatility. Prices for Swaptions under the Black-Normal model are calculated as bellow:

$$PS = \frac{1 - \dfrac{1}{(1 + C/m)^{\tau m}}}{C} e^{-rT}\left[(C - K)N(d) + \frac{\sigma\sqrt{T}}{\sqrt{2\pi}} e^{-d^2/2}\right] \tag{11}$$

$$RS = \frac{1 - \dfrac{1}{(1 + C/m)^{\tau m}}}{C} e^{-rT}\left[(K - C)N(d) + \frac{\sigma\sqrt{T}}{\sqrt{2\pi}} e^{-d^2/2}\right] \tag{12}$$

with

$$d = \frac{C - K}{\sigma\sqrt{T}}$$

and the transformation between Normal volatility and Black volatility is given by the next equality

$$\sigma_N = C\sqrt{\frac{2\pi}{T}}\left[2N\left(\frac{\sigma_B\sqrt{T}}{2}\right) - 1\right]$$

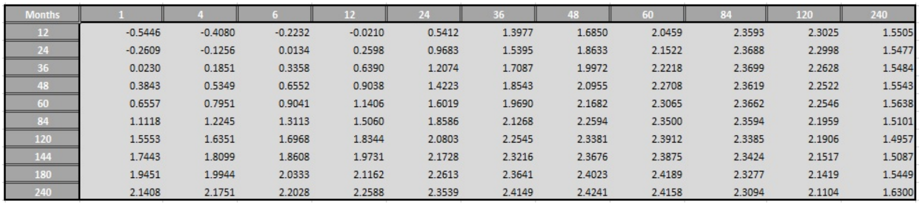
where σ_N represents the Normal volatility and σ_B the Black volatility. In case the reader is interested in going deeper in these equivalences they are referenced to Röman,J. 2015 [1] p.111 and p.444-446.

4.1.2 Practical example of Pricing European Swaptions under Normal-Black using Visual Basic Application on Microsoft Excel

In this section, a basic example of the theoretical explanation given above of how to value a European payer Swaption under Normal-Black is presented. For the calculations VBA on Ms Excel has been used and the full code to the model can be found in the Appendix 3.

In Figure 23 of the Appendix 5 we have the Quoted Normal Volatilities given by the market in Term/Term Swap/Swaption form up to 20 years. This information has been provided by Jan Röman and corresponds to the market data gathered by Swedbank at 2015-07-07. However, it can be found on financial information providers like *Bloomberg* or *Reuters* everyday.

Now we need to calculate the at-the-money forward rates to let them be our Swap rates. To do so we have applied (1) on the bootstraped Swap rates that have been shown in Figure 2 in Section 2.1. Following this process we have created the Swap rates matrix given in Figure 9, where we can find each and every forward Swap rate at the same periods of time as we have the volatilities with columns representing Swaption maturity and rows being Swap tenors.

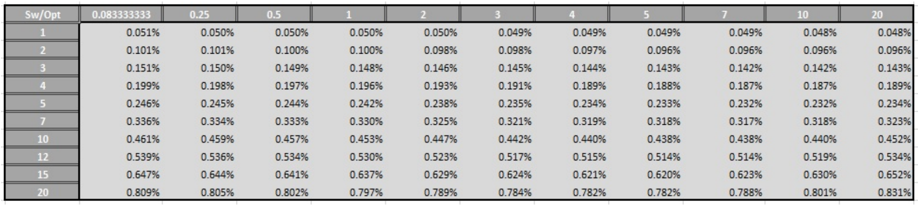


Months	1	4	6	12	24	36	48	60	84	120	240
12	-0.5446	-0.4080	-0.2232	-0.0210	0.5412	1.3977	1.6850	2.0459	2.3593	2.3025	1.5505
24	-0.2609	-0.1256	0.0134	0.2598	0.9683	1.5395	1.8633	2.1522	2.3688	2.2998	1.5477
36	0.0230	0.1851	0.3358	0.6390	1.2074	1.7087	1.9972	2.2218	2.3699	2.2628	1.5484
48	0.3843	0.5349	0.6552	0.9038	1.4223	1.8543	2.0955	2.2708	2.3619	2.2522	1.5543
60	0.6557	0.7951	0.9041	1.1406	1.6019	1.9690	2.1682	2.3065	2.3662	2.2546	1.5638
84	1.1118	1.2245	1.3113	1.5060	1.8586	2.1268	2.2594	2.3500	2.3594	2.1959	1.5101
120	1.5553	1.6351	1.6968	1.8344	2.0803	2.2545	2.3381	2.3912	2.3385	2.1906	1.4957
144	1.7443	1.8099	1.8608	1.9731	2.1728	2.3216	2.3676	2.3875	2.3424	2.1517	1.5087
180	1.9451	1.9944	2.0333	2.1162	2.2613	2.3641	2.4023	2.4189	2.3277	2.1419	1.5449
240	2.1408	2.1751	2.2028	2.2588	2.3539	2.4149	2.4241	2.4158	2.3094	2.1104	1.6300

Figure 9: Forward Swap rates

Once we get to this point, we can apply the formula (11) for payer Swaption to our data (or (12) in case we wanted to calculate the prices of the receiver Swaption). For this practical example we have applied the formula defined in Appendix 3 *NormalSwaptionPricing* (11) to an annual Swaption with risk-free rate $r = 0.1\%$. The volatilities for each period are gotten from Figure 15 and Swap rates are provided in Figure 9. The strike rates have been set 0.01 points below the Swap rates by default for these calculations. The results can be found in Figure 10, which represent the prices in percentage for Swaptions in Term/Term Swap/Swaption.

After we have calculated the Normal prices of European payer Swaptions for all the combinations of Swap Tenor and Swaption maturity that can be seen in the matrix on Figure 10. This prices will be the ones we will use to calibrate our Hull-White model.



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	0.051%	0.050%	0.050%	0.050%	0.050%	0.049%	0.049%	0.049%	0.049%	0.048%	0.048%
2	0.101%	0.101%	0.100%	0.100%	0.098%	0.098%	0.097%	0.096%	0.096%	0.096%	0.096%
3	0.151%	0.150%	0.149%	0.148%	0.146%	0.145%	0.144%	0.143%	0.142%	0.142%	0.143%
4	0.199%	0.198%	0.197%	0.196%	0.193%	0.191%	0.189%	0.188%	0.187%	0.187%	0.189%
5	0.246%	0.245%	0.244%	0.242%	0.238%	0.235%	0.234%	0.233%	0.232%	0.232%	0.234%
7	0.336%	0.334%	0.333%	0.330%	0.325%	0.321%	0.319%	0.318%	0.317%	0.318%	0.323%
10	0.461%	0.459%	0.457%	0.453%	0.447%	0.442%	0.440%	0.438%	0.438%	0.440%	0.452%
12	0.539%	0.536%	0.534%	0.530%	0.523%	0.517%	0.515%	0.514%	0.514%	0.519%	0.534%
15	0.647%	0.644%	0.641%	0.637%	0.629%	0.624%	0.621%	0.620%	0.623%	0.630%	0.652%
20	0.809%	0.805%	0.802%	0.797%	0.789%	0.784%	0.782%	0.782%	0.788%	0.801%	0.831%

Figure 10: Payer Swaption prices

5 Calibration of the Hull-White model and results

Probably the most important part when pricing financial derivatives is the calibration of the model we are managing. As we are setting random values for some important parameters like the volatility and the mean-reverting factor, we have to trace the market and look for these parameters' real values. Therefore, whether we want to handle a model that estimates the prices accurately or simply gives random results will be defined in this step. This procedure is performed trying to minimize the sum of the Squared differences between the calculated and the objective prices (usually Black prices).

$$\min \sqrt{\sum_{i=1}^{n} \left(\frac{ModelPrice_i - MarketPrice_i}{MarketPrice_i} \right)^2} \tag{13}$$

In the Hull-White model we have to try different values for both σ and a until we find the ones that get the minimum value for the sum of squared errors. This is normally done using root-finding algorithms like Newton-Raphson.

5.0.1 Calibration of the Hull-White model. Practical example using Python

In this Subsection a practical example of the calibration process of the Hull-White model against Black-Normal using Python will be given, in case the reader is interested in the program, the whole code can be found in the Appendix 4. $r = 0.1\%$ is set as risk-free rate parameter in this practical example.

For the calibration, a root-finding algorithm similar to Newton-Raphson has been created. This algorithm repeats the Swaption pricing program and compares the output prices with the Black-Normal presented in Subsection 4.1.2 for a range of values of σ and a and calculates (13) for each combination of them. To simplify calculations, constant values are supposed for these two variables, however, the model can be developed to find time dependent values for each time-step. In this case, the chosen ranges for the parameters are $\sigma = (0.1\% - 10\%)$ and $a = (0.001 - 0.1)$. Furthermore, the results given are shown in Figure 11.

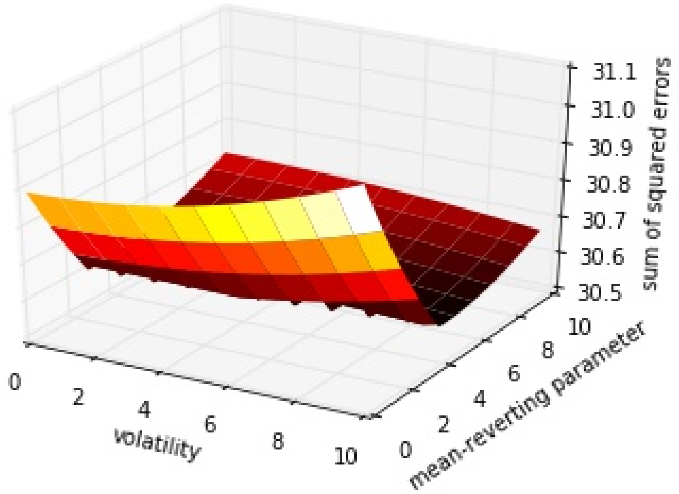
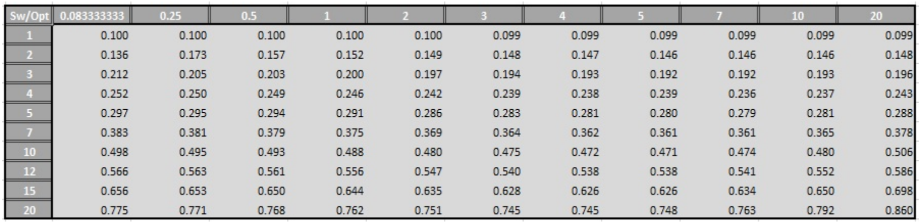


Figure 11: Sum of squared errors for the range of values given for volatility and mean-reverting parameter

As it can be seen in Figure 11, the minimum error is reached at $\sigma = 6.4\%$ and $a = 0.041$. The calibrated prices for the Swaptions in Subsection 3.3.1 are shown in Figure 12.



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	0.100	0.100	0.100	0.100	0.100	0.099	0.099	0.099	0.099	0.099	0.099
2	0.136	0.173	0.157	0.152	0.149	0.148	0.147	0.146	0.146	0.146	0.148
3	0.212	0.205	0.203	0.200	0.197	0.194	0.193	0.192	0.192	0.193	0.196
4	0.252	0.250	0.249	0.246	0.242	0.239	0.238	0.239	0.236	0.237	0.243
5	0.297	0.295	0.294	0.291	0.286	0.283	0.281	0.280	0.279	0.281	0.288
7	0.383	0.381	0.379	0.375	0.369	0.364	0.362	0.361	0.361	0.365	0.378
10	0.498	0.495	0.493	0.488	0.480	0.475	0.472	0.471	0.474	0.480	0.506
12	0.566	0.563	0.561	0.556	0.547	0.540	0.538	0.538	0.541	0.552	0.586
15	0.656	0.653	0.650	0.644	0.635	0.628	0.626	0.626	0.634	0.650	0.698
20	0.775	0.771	0.768	0.762	0.751	0.745	0.745	0.748	0.763	0.792	0.860

Figure 12: Calibrated Swaption prices under Hull-White model in %

5.1 Risk measures. The Greeks

After calibration is done, we should measure risks from both models to compare them. This process is done using the Greeks. The Greeks are different measures of how much an instrument price varies when one of its variables changes. Usually, the shift used in the Greeks calculations is 1/1000 of a basis point and the result is then scaled to a one basis point shift (letting $scale = 1000$).The most important Greeks are:

- Delta price: Shows the change in the theoretical price given a variation of a unit in the price of the underlying.

$$\Delta_{price} = \frac{\partial(PV)}{\partial U} = \frac{PV(U+h) - PV(U)}{h} * scale$$

- Delta yield: Shows the change in the present value, given a shift of one basis point in the yield curve. Normally it is done adding 0.00001 to the prices of the Zero-Coupon curve and giving *scale* the value 1000 i.e. shifting it by 1 basis point.

$$\Delta_{yield} = \big[PV(r+h) - PV(r)\big] * scale$$

- Gamma price: Shows the variation in Delta price, given a unit change in the underlying.

$$\Gamma_{price} = \frac{PV(U+2h) - 2PV(U+h) + PV(U)}{h^2} * scale$$

- Gamma yield: Shows the change in Delta yield given a basis point change in the yield curve. As in the Delta yield, 0.00001 is added to each node of the Zero-Coupon curve. However, in this case, as it is similar to taking the square of the Delta yield, the *scale* will be 1000^2

$$\Gamma_{yield} = \big[PV(y+2h) - 2PV(y+h) + PV(y)\big] * scale$$

- Rho: Shows the change in the present value, given a shift of one basis point in the repo curve.

$$\rho = \big[PV(r_{repo}+h) - PV(r_{repo})\big] * scale$$

- Theta: Shows the change in the present value given the valuation date until the following calendar date.

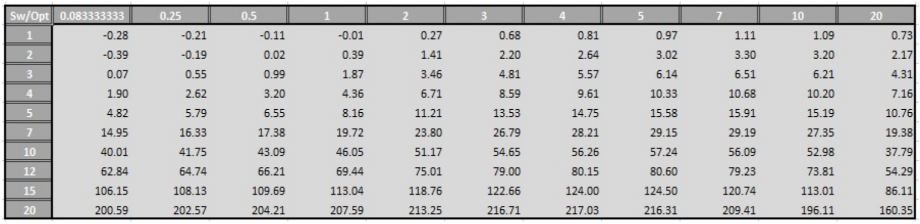
$$\Theta = PV(date) - PV(date+1)$$

- Vega: Shows the change in the present value given an shift in volatility of 1%.

$$\nu = \big[PV(\sigma+0.01) - PV(\sigma)\big]$$

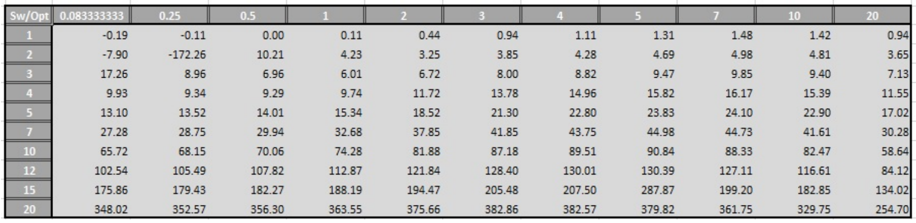
5.1.1 Results

Therefore, to compare the risks between both Normal-Black and Hull-White models, we have performed the Delta yield, Gamma yield and Vega to our products. However, as the prices of the Swaptions are given in percentage and the Greeks will take small values, a notional of 10.000.000 SEK has been set,



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	-0.28	-0.21	-0.11	-0.01	0.27	0.68	0.81	0.97	1.11	1.09	0.73
2	-0.39	-0.19	0.02	0.39	1.41	2.20	2.64	3.02	3.30	3.20	2.17
3	0.07	0.55	0.99	1.87	3.46	4.81	5.57	6.14	6.51	6.21	4.31
4	1.90	2.62	3.20	4.36	6.71	8.59	9.61	10.33	10.68	10.20	7.16
5	4.82	5.79	6.55	8.16	11.21	13.53	14.75	15.58	15.91	15.19	10.76
7	14.95	16.33	17.38	19.72	23.80	26.79	28.21	29.15	29.19	27.35	19.38
10	40.01	41.75	43.09	46.05	51.17	54.65	56.26	57.24	56.09	52.98	37.79
12	62.84	64.74	66.21	69.44	75.01	79.00	80.15	80.60	79.23	73.81	54.29
15	106.15	108.13	109.69	113.04	118.76	122.66	124.00	124.50	120.74	113.01	86.11
20	200.59	202.57	204.21	207.59	213.25	216.71	217.03	216.31	209.41	196.11	160.35

Figure 13: Black-Delta yield



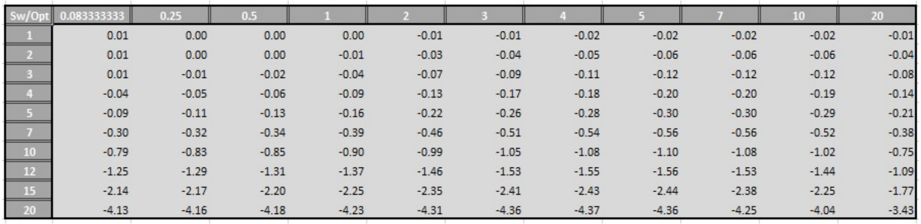
Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	-0.19	-0.11	0.00	0.11	0.44	0.94	1.11	1.31	1.48	1.42	0.94
2	-7.90	-172.26	10.21	4.23	3.25	3.85	4.28	4.69	4.98	4.81	3.65
3	17.26	8.96	6.96	6.01	6.72	8.00	8.82	9.47	9.85	9.40	7.13
4	9.93	9.34	9.29	9.74	11.72	13.78	14.96	15.82	16.17	15.39	11.55
5	13.10	13.52	14.01	15.34	18.52	21.30	22.80	23.83	24.10	22.90	17.02
7	27.28	28.75	29.94	32.68	37.85	41.85	43.75	44.98	44.73	41.61	30.28
10	65.72	68.15	70.06	74.28	81.88	87.18	89.51	90.84	88.33	82.47	58.64
12	102.54	105.49	107.82	112.87	121.84	128.40	130.01	130.39	127.11	116.61	84.12
15	175.86	179.43	182.27	188.19	194.47	205.48	207.50	287.87	199.20	182.85	134.02
20	348.02	352.57	356.30	363.55	375.66	382.86	382.57	379.82	361.75	329.75	254.70

Figure 14: HW-Delta yield

since it is a common amount for a Swaption contract. Doing so, the results will be more understandable.

From Figures 13 and 14 we can affirm that for a Swap of tenor 20 years with a Swaption of 1 year, if the Zero-Coupons were shifted by 1 basis point, the price of the contract would change 207 SEK according to the Normal-Black model and 352 SEK in the Hull-White program we have developed. Hence, the difference in risks between the two models is very small (145 SEK) what is an acceptable amount when dealing with a contract of 10 million SEK.

Furthermore, traders would check the Gamma yield risk to be sure of how the Delta yield would change given the same shift in the Zero-Coupon curve. In Figures 15 and 16 we have the values for this Greek.

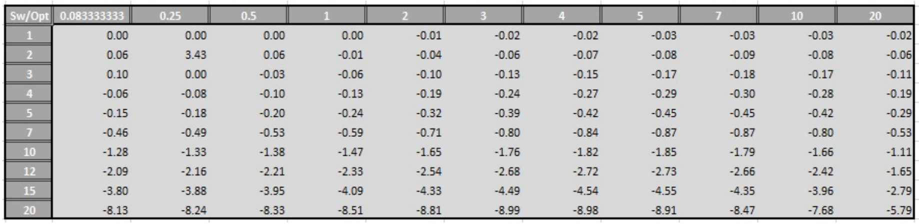


Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	0.01	0.00	0.00	0.00	-0.01	-0.01	-0.02	-0.02	-0.02	-0.02	-0.01
2	0.01	0.00	0.00	-0.01	-0.03	-0.04	-0.05	-0.06	-0.06	-0.06	-0.04
3	0.01	-0.01	-0.02	-0.04	-0.07	-0.09	-0.11	-0.12	-0.12	-0.12	-0.08
4	-0.04	-0.05	-0.06	-0.09	-0.13	-0.17	-0.18	-0.20	-0.20	-0.19	-0.14
5	-0.09	-0.11	-0.13	-0.16	-0.22	-0.26	-0.28	-0.30	-0.30	-0.29	-0.21
7	-0.30	-0.32	-0.34	-0.39	-0.46	-0.51	-0.54	-0.56	-0.56	-0.52	-0.38
10	-0.79	-0.83	-0.85	-0.90	-0.99	-1.05	-1.08	-1.10	-1.08	-1.02	-0.75
12	-1.25	-1.29	-1.31	-1.37	-1.46	-1.53	-1.55	-1.56	-1.53	-1.44	-1.09
15	-2.14	-2.17	-2.20	-2.25	-2.35	-2.41	-2.43	-2.44	-2.38	-2.25	-1.77
20	-4.13	-4.16	-4.18	-4.23	-4.31	-4.36	-4.37	-4.36	-4.25	-4.04	-3.43

Figure 15: Black-Gamma yield

The Gamma yield is thus, equal to -4.23 in the Normal-Black model and -8.51 in Hull-White, also very similar one to each other.

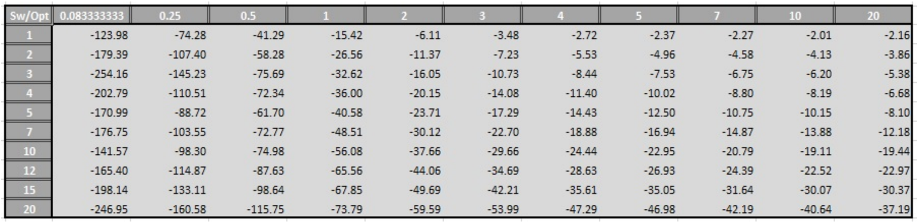
Finally, we will calculate Vega as the traders would analyze the possible risks



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	0.00	0.00	0.00	0.00	-0.01	-0.02	-0.02	-0.03	-0.03	-0.03	-0.02
2	0.06	3.43	0.06	-0.01	-0.04	-0.06	-0.07	-0.08	-0.09	-0.08	-0.06
3	0.10	0.00	-0.03	-0.06	-0.10	-0.13	-0.15	-0.17	-0.18	-0.17	-0.11
4	-0.06	-0.08	-0.10	-0.13	-0.19	-0.24	-0.27	-0.29	-0.30	-0.28	-0.19
5	-0.15	-0.18	-0.20	-0.24	-0.32	-0.39	-0.42	-0.45	-0.45	-0.42	-0.29
7	-0.46	-0.49	-0.53	-0.59	-0.71	-0.80	-0.84	-0.87	-0.87	-0.80	-0.53
10	-1.28	-1.33	-1.38	-1.47	-1.65	-1.76	-1.82	-1.85	-1.79	-1.66	-1.11
12	-2.09	-2.16	-2.21	-2.33	-2.54	-2.68	-2.72	-2.73	-2.66	-2.42	-1.65
15	-3.80	-3.88	-3.95	-4.09	-4.33	-4.49	-4.54	-4.55	-4.35	-3.96	-2.79
20	-8.13	-8.24	-8.33	-8.51	-8.81	-8.99	-8.98	-8.91	-8.47	-7.68	-5.79

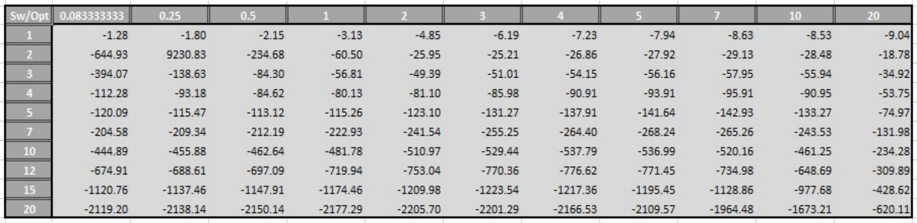
Figure 16: HW-Gamma yield

coming from changes in volatility. The results are shown in Figures 17 and 18.



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	-123.98	-74.28	-41.29	-15.42	-6.11	-3.48	-2.72	-2.37	-2.27	-2.01	-2.16
2	-179.39	-107.40	-58.28	-26.56	-11.37	-7.23	-5.53	-4.96	-4.58	-4.13	-3.86
3	-254.16	-145.23	-75.69	-32.62	-16.05	-10.73	-8.44	-7.53	-6.75	-6.20	-5.38
4	-202.79	-110.51	-72.34	-36.00	-20.15	-14.08	-11.40	-10.02	-8.80	-8.19	-6.68
5	-170.99	-88.72	-61.70	-40.58	-23.71	-17.29	-14.43	-12.50	-10.75	-10.15	-8.10
7	-176.75	-103.55	-72.77	-48.51	-30.12	-22.70	-18.88	-16.94	-14.87	-13.88	-12.18
10	-141.57	-98.30	-74.98	-56.08	-37.66	-29.66	-24.44	-22.95	-20.79	-19.11	-19.44
12	-165.40	-114.87	-87.63	-65.56	-44.06	-34.69	-28.63	-26.93	-24.39	-22.52	-22.97
15	-198.14	-133.11	-98.64	-67.85	-49.69	-42.21	-35.61	-35.05	-31.64	-30.07	-30.37
20	-246.95	-160.58	-115.75	-73.79	-59.59	-53.99	-47.29	-46.98	-42.19	-40.64	-37.19

Figure 17: Black-Vega

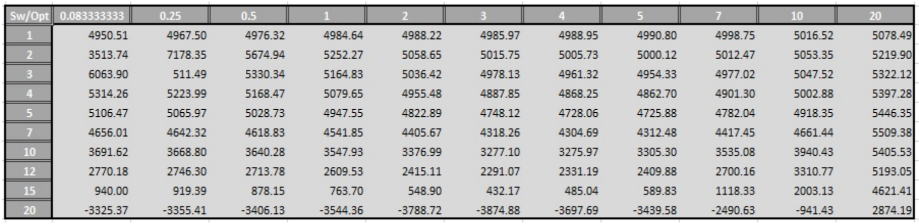


Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	-1.28	-1.80	-2.15	-3.13	-4.85	-6.19	-7.23	-7.94	-8.63	-8.53	-9.04
2	-644.93	9230.83	-234.68	-60.50	-25.95	-25.21	-26.86	-27.92	-29.13	-28.48	-18.78
3	-394.07	-138.63	-84.30	-56.81	-49.39	-51.01	-54.15	-56.16	-57.95	-55.94	-34.92
4	-112.28	-93.18	-84.62	-80.13	-81.10	-85.98	-90.91	-93.91	-95.91	-90.95	-53.75
5	-120.09	-115.47	-113.12	-115.26	-123.10	-131.27	-137.91	-141.64	-142.93	-133.27	-74.97
7	-204.58	-209.34	-212.19	-222.93	-241.54	-255.25	-264.40	-268.24	-265.26	-243.53	-131.98
10	-444.89	-455.88	-462.64	-481.78	-510.97	-529.44	-537.79	-536.99	-520.16	-461.25	-234.28
12	-674.91	-688.61	-697.09	-719.94	-753.04	-770.36	-776.62	-771.45	-734.98	-648.69	-309.89
15	-1120.76	-1137.46	-1147.91	-1174.46	-1209.98	-1223.54	-1217.36	-1195.45	-1128.86	-977.68	-428.62
20	-2119.20	-2138.14	-2150.14	-2177.29	-2205.70	-2201.29	-2166.53	-2109.57	-1964.48	-1673.21	-620.11

Figure 18: HW-Vega

Vega shows that the price of the Black-Normal model will be less sensitive against variations in the volatility than the Hull-White model. For instance, on the same 1 into 20 Swaption, a change in 1 basis point in volatility would signify a drop of 73 SEK in the Normal-Black model, while it would mean a drop of 2177 SEK in Hull-White.

As a last test for our calibration, we will show the differences in prices between both models in Figure 19. It can be seen that the differences in all the instruments are on the level of 5000 SEK. Once again, a small value for contracts of an amount of 10 million SEK, what gives us the confirmation that our Hull-White model is well calibrated. In Figure 20 the heat-map of these differences is shown. It can be appreciated that the best estimations are for Swaps of 15 year tenor and the model gets more accurate as Swap tenor increases.



Sw/Opt	0.083333333	0.25	0.5	1	2	3	4	5	7	10	20
1	4950.51	4967.50	4976.32	4984.64	4988.22	4985.97	4988.95	4990.80	4998.75	5016.52	5078.49
2	3513.74	7178.35	5674.94	5252.27	5058.65	5015.75	5005.73	5000.12	5012.47	5053.35	5219.90
3	6063.90	511.49	5330.34	5164.83	5036.42	4978.13	4961.32	4954.33	4977.02	5047.52	5322.12
4	5314.26	5223.99	5168.47	5079.65	4955.48	4887.85	4868.25	4862.70	4901.30	5002.88	5397.28
5	5106.47	5065.97	5028.73	4947.55	4822.89	4748.12	4728.06	4725.88	4782.04	4918.35	5446.35
7	4656.01	4642.32	4618.83	4541.85	4405.67	4318.26	4304.69	4312.48	4417.45	4661.44	5509.38
10	3691.62	3668.80	3640.28	3547.93	3376.99	3277.10	3275.97	3305.30	3535.08	3940.43	5405.53
12	2770.18	2746.30	2713.78	2609.53	2415.11	2291.07	2331.19	2409.88	2700.16	3310.77	5193.05
15	940.00	919.39	878.15	763.70	548.90	432.17	485.04	589.83	1118.33	2003.13	4621.41
20	-3325.37	-3355.41	-3406.13	-3544.36	-3788.72	-3874.88	-3697.69	-3439.58	-2490.63	-941.43	2874.19

Figure 19: Differences in Swaption prices between the two models

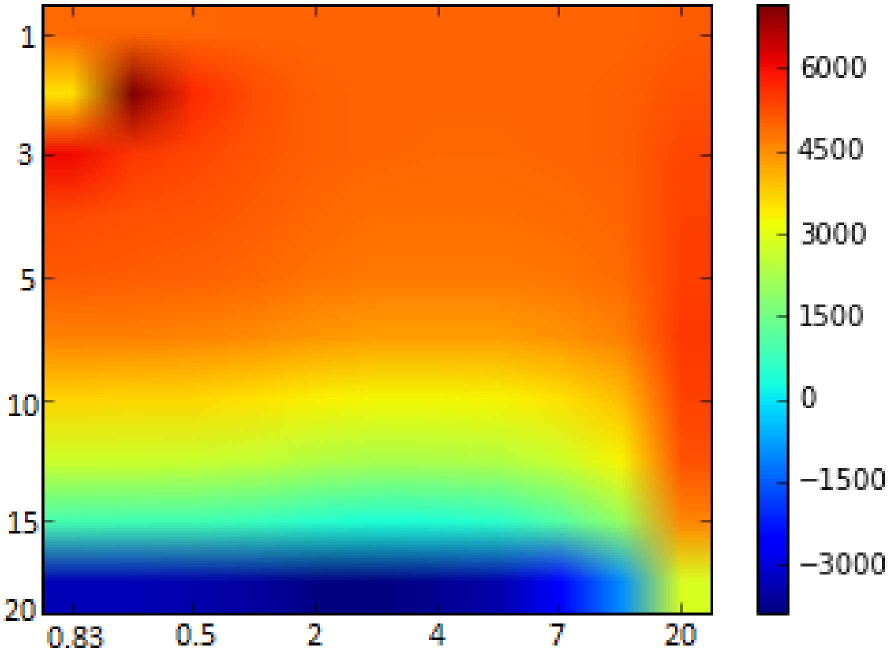


Figure 20: Heat-map of the Swaption prices differences

To summarize, the values of these risk measures and the small differences in prices against market data show that the model performed in this work behaves well, in levels similar to Black-Normal. Therefore, we can conclude that it calibrated and can be used to price interest rates derivatives.

6 Conclusion

Throughout this dissertation we have presented the Hull-White model and used it to price Swaptions. We have shown its solution through the analytical formula and Finite Difference Methods. Due to the complexity of the Crank-Nicolson method, the analytical formula would simplify calculations and processing time. However, my opinion is that the Crank-Nicolson solution is more useful as one can split time in as many pieces as he/she wants and study a "closer to continuity" time elapsing. Hull-White, like other martingale models, is hard to compute and Jamshidian decomposition could be sometimes difficult to program because the root-finding algorithm that has to be applied. Anyway, I find it the best model to use when dealing with interest rate derivatives because of its accuracy and flexibility, as we can use it to price every product that can be divided into Zero-Coupons.

Afterwards, the Black-76 and Black-Normal models have been shown, making clear that Black will always be the easiest to compute model of them all. Furthermore, we can find Black and Normal volatilities for as many products as we want in market information machines like *Bloomberg*. Thus, I would say that even though we could have problems dealing with negative rates, Black-76 is the model to choose for those analysts that look for simplicity and time saving. Hence the reason why is the most utilized among traders.

In the last part of this dissertation the calibration of Hull-White against Black-Normal has been developed. This process is easy to apply for the case of constant mean-reverting, risk-free rate and volatility parameters. However, if the reader wants to go deeper in this method, I would recommend to apply time-dependent parameters as this behavior is closer to reality. The worse part of the calibration application is the processing time a computer needs, since it will repeat the program for each input we want to try and sometimes can take long time to find the solution (even more for time-dependency).

Finally, the results have been presented. The risk measures show that the differences in risk between Hull-White and Black are small, as well as the prices of the products analyzed. Therefore, we could affirm that while calibrated to reality, Hull-White performs very well to price interest rate instruments.

To conclude, due to its difficulty, I would like to thank and show my respects to every person that has studied this topic before me, including all the authors in the references and specially to Jan Röman, who has developed [1] and [7] and has guided me as coordinator during the process of writing this Master thesis.

7 Fulfillment of thesis objectives

As required by the Swedish National Agency for Higher Education to Master theses (2 years), the objectives and criteria for a master degree in mathematics, mathematical statistics, financial mathematics and actuarial science and its fulfillment are as follows:

Objective 1: *For Master degree, student should demonstrate knowledge and understanding in the major field of study, including both broad knowledge in the field and substantially deeper knowledge of certain parts of the area as well as insight into current research and development.*

Due to the difficulty of the field a broad knowledge is needed. Since without the understanding of the advanced models presented and the mathematics used to solve them the development of this thesis would be impossible. Furthermore, the understanding of complex financial derivatives (as Swaptions are) implies a very deep knowledge in finance and the markets behavior.

Objective 2: *For Master Degree, student should demonstrate deeper methodological knowledge in the major field of study.*

As can be seen along the dissertation, an advanced methodological knowledge is used. The Crank-Nicolson method, the algorithm programed in Python for the calibration and the use of Visual Basic to find the Black prices are clear examples of it.

Objective 3: *For Master degree, student should demonstrate the ability to critically and systematically integrate knowledge and to analyse, assess and deal with complex phenomena, issues and situations even with limited information.*

The process of finding the information needed to solve some of the problems presented in this dissertation has been tough sometimes. The algorithm used to program Crank-Nicolson is, for instance, fully developed by myself. The process of transforming this finite difference method mathematical explanation to programing code is difficult and needs of great imagination ability. The same situation happens when solving the Jamshidian method to price Swaptions.

Objective 4: *For Master degree, student should demonstrate the ability to critically, independently and creatively identify and formulate issues and to plan and carry out advanced tasks within specified time frames, thereby contributing to the development of knowledge and to evaluate this work.*

The process of programing the Crank-Nicolson method, the Hull-White theoretical formula and the calibration in Python needs of a great creativity and capacity of transforming mathematical concepts into algorithms.

Objective 5: *For Master degree, student should demonstrate ability in both national and international contexts, orally and in writing to present and discuss their conclusions and the knowledge and arguments behind them, in dialogue with different groups.*

The international context is clear when looking at the nationality of the references used along this dissertation. Furthermore, due to my nationality (Spanish) taking a master program in English language in Sweden with classmates from more than ten different nationalities demonstrates the ability to present and discuss conclusions and dialog with different groups.

Objective 6: *For Master degree, student should demonstrate ability in the major field of study make judgments taking into account relevant scientific, social and ethical aspects, and demonstrate an awareness of ethical issues in research and development.*

The sources are cited all over the dissertation as required by scientific ethic in order to honor the developer and creator of each research. Furthermore, the principal objective of this work is to help to future researchers and to enlighten the possible lack of understanding about this topic to every person with curiosity to learn about it. And over all, to make the mathematical and financial human knowledge a little broader.

8 References

1 Röman, J. Lecture notes in Analytical Finance II, Mälardalen University, Sweden, 2015.

2 Kohn, R. PDE for Finance Notes, Spring 2011, Courant Institute of Mathematical Sciences, 2003.

3 Hull, J. & White, A. Pricing Interest-Rate Derivative Securities, University of Toronto, 1990.

4 Hull, J. Options, Futures and Other Derivatives, 7th edition, Pearson, U.S., 2009.

5 Black, F. The Pricing of Commodity Contracts, Journal of Financial Economics, vol 3 pp [167-179], M.I.T, Mass., U.S., 1976.

6 Jamshidian, F. An Exact Bond Option Formula, The Journal of Finance, vol 44 pp [205-209], U.S., 1989.

7 Röman, J. Lecture notes in Analytical Finance I, Mälardalen University, Sweden, 2014.

9 Appendix 1: Hull-White PDE solution by Finite Difference Method (Crank-Nicolson) to price Zero-Coupon bonds. Python code

```
”””
European Swaption Valuation Under Hull−White using Crank−Nicolson

Created on Sun Mar 27 13:55:07 2016

@author: Victor
”””
import numpy as np
import matplotlib.pyplot as plt
import math
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D

# inputs

rmin = −0.015
rmax = 0.3
dr = 0.005
dt = 0.003
T = 1
M = int ((rmax−rmin)/dr)
N = int (T/dt)
sigma = 0.02
theta = 0.02
a = 0.01
```

```
# Create Matrices

A = np.zeros((M,M))
B = np.zeros((M,M))

# Create factor vectors for A

PAu = np.zeros((M,1))
PAm = np.zeros((M))
PAd = np.zeros((M,1))

# Define factors for A. I+dt/2*A(t_i)

#Boundary conditions for A, let second derivative with respect to r be 0

PAu[0] = -dt/4*(theta-a*rmin)/dr
PAm[0] = 1+rmin*dt/2
PAd[0] = -dt/4*(-theta+a*rmin)/dr

for i in range (1,M):
        PAu[i] = -dt/4*(math.pow(sigma,2)/(math.pow(dr,2))-(theta-a*PAd[i-1])/dr)
        PAd[i] = -dt/4*(math.pow(sigma,2)/(math.pow(dr,2))+(theta-a*PAd[i-1])/dr)

        for j in range (1,M):
                PAm[j] = 1+dt/2 *(math.pow(sigma,2)/math.pow(dr,2)+PAm[j-1])

# Define Matrix A
```

```
    A = np.diagflat([PAu[i] for i in range(M-1)],-1)+\
    np.diagflat([PAm[i] for i in range(M)])+\
    np.diagflat([PAd[i] for i in range(M-1)],+1)

# Create factor vectors for B

PBu = np.zeros((M,1))
PBm = np.zeros((M,1))
PBd = np.zeros((M,1))

# Define factors for B. I-dt/2*B(t_i+1)

# Define boundary conditions for B, let second derivative with respect to r be 0

PBu[0] = dt/4 * (theta-a*rmin)/dr
PBm[0] = 1-rmin* dt/2
PBd[0] = dt/4*(-theta+a*rmin)/dr

for i in range (1,M):
    PBu[i] = dt/4*(math.pow(sigma,2)/(math.pow(dr,2))-(theta-a*PBd[i-1])/dr)
    PBd[i] = dt/4*(math.pow(sigma,2)/(math.pow(dr,2))+(theta-a*PBd[i-1])/dr)

    for j in range (1,M):
        PBm[j] = 1-dt/2*(math.pow(sigma,2)/math.pow(dr,2)+PBm[j-1])

# Define Matrix B
```

```
        B = np.diagflat([PBu[i] for i in range(M-1)],-1)+\
        np.diagflat([PBm[i] for i in range(M)])+\
        np.diagflat([PBd[i] for i in range(M-1)],+1)

# Define matrix U with vectors equal to interest Zero Bond Prices for each time-step

U = np.zeros((M,N))

for i in range (0,M):
        U[i,0] = 1

for i in range (0,N-1):
        U[:,i+1] = np.dot(np.linalg.inv(A),np.dot(B,U[:,i]))

# Printing the grid

X = np.arange(0, 333)
Y = np.arange(-3, 60)
X, Y = np.meshgrid(X/333, Y/2)
Z = U
fig = plt.figure()
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, Z, linewidth=0, cmap=cm.jet)
ax.set_xlabel('time to maturity (%)')
ax.set_ylabel('rate')
ax.set_zlabel('price')
ax.set_zlim(0, 1)
fig.colorbar(surf, shrink=1, aspect=5)
```

```
plt.show()

im = plt.imshow(Z, cmap=cm.jet, extent=[0,100,-5,30])
plt.colorbar(im, orientation='horizontal')
plt.show()
```

10 Appendix 2: Pricing Swaptions under Hull-White using the Analytical formula. Python code

```
”””
European Swaption Valuation Under Hull−White using analytical formula

Created on Tue Apr 19 20:44:04 2016

@author: Victor
”””

import pandas as pd
import numpy as np
import scipy.optimize as sp
import math

# import the rates data from an excel file

Location = ”..\Black_and_Bootstrap.xlsm”
rates = pd.read_excel(Location, 1)
table_rates=pd.DataFrame({’years’: rates[’T (days)’]/360, ’discounts’: rates[’Discount Factors’],
        ’forwards’: rates[’Forward Rates’]/100})

# inputs

r = 0.001
sigma = 0.01
a = 0.05
```

```
# define vectors to be filled with discounting factors
p_now = np.zeros((table_rates['discounts'].size,1))
p_now[0]=table_rates['discounts'][0]

# loop to fill the vector with the HW analytical formula for Zero Coupon bonds prices
for i in range (1,p_now.size):
        t=table_rates['years'][0]
        mat=table_rates['years'][i]
        p_mat=table_rates['discounts'][i]
        p_t=table_rates['discounts'][0]
        forwardrate=table_rates['forwards'][0]

funcB1= (1-math.exp(-a*(mat-t)))/a
funcA1= p_mat/p_t*math.exp(funcB1*forwardrate-math.pow(sigma,2)/(4*a)*math.pow(funcB1,2)*(1-math.exp(-2*a*t)))
p_now[i]=funcA1*math.exp(funcB1*r)

# Forward Swap rates matrix

forward_swap_matrix = np.zeros((10,11))
swap_tenor = np.zeros((10,1))
swaption_maturity = np.zeros((10,11))
swaption_swap_maturity = np.zeros((10,11))

swap_tenor=([1.0,2.0,3.0,4.0,5.0,7.0,10.0,12.0,15.0,20.0])
swaption_maturity=([1/12,1/3,1/2,1.0,2.0,3.0,4.0,5.0,7.0,10.0,20.0])

# Legs of the Swap and Swap rates
```

```
fixed_leg = np.zeros((p_now.size,1))
fixed_leg[0]=p_now[0]*table_rates['years'][0]
floating_leg = np.zeros((p_now.size,1))
floating_leg[0]=0
swap_rates=np.zeros((p_now.size,2))
swap_rates[:,0]=table_rates['years']

for i in range (0,p_now.size):
        floating_leg[i] = p_now[0]-p_now[i]

for i in range (1,p_now.size):
        fixed_leg[i] = fixed_leg[i-1]+p_now[i]*(table_rates['years'][i]-table_rates['years'][i-1])
        swap_rates[i,1] = floating_leg[i]/fixed_leg[i]

for i in range (0,10):

        for j in range (0,11):
                swaption_swap_maturity[i,j]=swap_tenor[i]+swaption_maturity[j]

                for k in range (0,p_now.size):
                        if table_rates['years'][k]==swaption_swap_maturity[i,j]:

                                for l in range (0,p_now.size):
                        if table_rates['years'][l]==swap_tenor[i]:

                                for m in range (0,p_now.size):
                                        if table_rates['years'][m]==swaption_maturity[j]:
```

```
                        forward_swap_matrix[i,j]=(p_now[m]-p_now[k])/fixed_leg[l]


# Jamshidian decomposition to value Swaptions

#defining all cash flow discounted payments
bond_t=np.zeros((11,1))
bonds_matrix_division=np.zeros((10,11))
forwardrate_vector=np.zeros((11,1))
r_level=np.zeros((10,11))
payments_function=np.zeros((10,11))
swaption_prices_matrix=np.zeros((10,11))

for i in range (0,10):
    cashflows=np.zeros((swap_tenor[i],1))
    sum_of_cashflows=np.zeros((swap_tenor[i],1))
    cashflows_r_star=np.zeros((swap_tenor[i],1))
    sum_of_options=np.zeros((swap_tenor[i],1))
    options_in_swaps=np.zeros((swap_tenor[i],1))

    for j in range (0,11):

        for k in range (0,p_now.size):
            #defining a vector with the bonds we need (=tenors)
            if table_rates['years'][k]==swaption_maturity[j]:
                #defining a vector with the forward rates at the start of each Swap
                forwardrate_vector[j]=table_rates['forwards'][k]
                bond_t[j]=table_rates['discounts'][k]
        #defining a matrix with the bonds we need to discount (maturity/tenors)
        if table_rates['years'][k]==swaption_swap_maturity[i,j]:
                bonds_matrix_division[i,j]=table_rates['discounts'][k]/bond_t[j]

#H-W factors
funcB= (1-math.exp(-a*(swaption_swap_maturity[i,j]-swaption_maturity[j])))/a
funcA= bonds_matrix_division[i,j]*math.exp(funcB*forwardrate_vector[j]-math.pow(sigma,2)/
(4*a)*math.pow(funcB,2)*(1-math.exp(-2*a*swaption_maturity[j])))
#vector created with all discounted cash flows from interest rate payments with HW factors
for l in range (0,cashflows[0].size-1):
        cashflows[l]= lambda r_star: forward_swap_matrix[i,j]*
        funcA*math.exp(-r_star*(swap_tenor[i])*funcB)

sum_of_cashflows[0]=cashflows[0]
for m in range (1,cashflows[0].size):
        sum_of_cashflows[m]=sum_of_cashflows[m-1]+cashflows[m]
#cash flows equal to strike rate
payments_function = lambda r_star:  sum_of_cashflows[sum_of_cashflows.size-1] +
forward_swap_matrix[i,j]*math.exp(-r_star*swap_tenor[i])-forward_swap_matrix[i,j]+0.001
#r^* level for which cash flows are equal to the notional (strike)
#calculated using Newton-Raphson
r_level[i,j]=sp.newton(payments_function, forward_swap_matrix[i,j], maxiter=1000000)
#substituting r^* level in every discounted cash flow, the result will be the strike
# of each option
for u in range (0,cashflows_r_star.size):
        cashflows_r_star[u]= forward_swap_matrix[i,j]*math.exp(-r_level[i,j]*(swap_tenor[i]-u))
        options_in_swaps[u]= max(forward_swap_matrix[i,j]*math.exp(-r*(swap_tenor[i]-u))-
        cashflows_r_star[u],0)

sum_of_options[0]=options_in_swaps[0]
for v in range (1,cashflows_r_star.size):
        sum_of_options[v]=sum_of_options[v−1]+options_in_swaps[v]

swaption_prices_matrix[i,j]=sum_of_options[sum_of_options.size −1]
```

11 Appendix 3: Pricing Swaptions under Black-76 and Normal-Black. VBA code

```
'_____________________________
' INTERPOLATION
'_____________________________

Public Function interpolation(T1 As Double, T2 As Double, r1 As Double, r2 As Double, _
T As Double) As Double

interpolation = r1 + (r2 - r1) * (T - T1) / (T2 - T1)

End Function


'_____________________________
' FORWARD RATE
'_____________________________

Public Function forwardRate(T1 As Double, T2 As Double, r1 As Double, _
r2 As Double) As Double

forwardRate = 100 * (((1 + r2 / 100) ^ T2 / (1 + r1 / 100) ^ T1) ^ (1 / (T2 - T1)) - 1)

End Function

'______________________________
```

```
' NORMAL DIST, code provided by Jan R man
'______________________________

Public Function NormalDist(ByRef X As Double) As Double
Dim sign As Long

Dim x2, q0, q1, q2 As Double

If (X < 0) Then
X = -X
sign = -1
ElseIf (X > 0) Then
sign = 1
Else ' (x = 0.0)
NormalDist = 0.5
Exit Function
End If

If (X > 20) Then
If (sign < 0) Then
NormalDist = 0
Else
NormalDist = 1
End If
Exit Function
End If

X = X * 0.707106781186547
```

```
x2 = X * X
If (X < 0.46875) Then
q1 = 3209.37758913847 + x2 * (377.485237685302 + x2 * (113.86415415105 _
+ x2 * (3.16112374387057 + x2 * 0.185777706184603)))

q2 = 2844.23683343917 + x2 * (1282.61652607737 + x2 * (244.024637934444 _
+ x2 * (23.6012909523441 + x2)))
NormalDist = 0.5 * (1 + sign * X * q1 / q2)
ElseIf (X < 4) Then
q1 = X * (8.88314979438838 + X * (0.56418849698867 + X * 2.15311535474404E-08))
q1 = X * (881.952221241769 + X * (298.6351381974 + X * (66.1191906371416 + q1)))
q1 = 1230.339354798 + X * (2051.07837782607 + X * (1712.04761263407 + q1))

q2 = X * (117.693950891312 + X * (15.7449261107098 + X))
q2 = X * (3290.79923573346 + X * (1621.38957456669 + X * (537.18110186201 + q2)))
q2 = 1230.33935480375 + X * (3439.36767414372 + X * (4362.61909014325 + q2))

NormalDist = 0.5 * (1 + sign * (1 - Exp(-x2) * q1 / q2))
Else
q0 = 1 / x2
q1 = 6.58749161529838E-04 + q0 * (1.60837851487423E-02 + q0 * (0.125781726111229 _
+ q0 * (0.360344899949804 + q0 * (0.305326634961232 + q0 * 1.63153871373021E-02))))

q2 = 2.33520497626869E-03 + q0 * (6.05183413124413E-02 + q0 * (0.527905102951428 _
+ q0 * (1.87295284992346 + q0 * (2.56852019228982 + q0))))
NormalDist = 0.5 * (1 + sign * (1 - Exp(-x2) / X * (0.564189583547756 - q0 * q1 / q2)))
End If
End Function
```

```
'______________________________
 ' NORMAL–BLACK SWAPTION PRICING
'______________________________

Function NormalSwaptionPricing(SwapRate As Double, StrikeRate As Double, _
SwaptionMaturity As Double, Tau As Double, NormalVolatility As Double) As Double
Dim Frequency As Double
Dim r As Double
Dim d As Double
Dim pi As Double

Frequency = 1
r = 0.001
d = (SwapRate - StrikeRate) / (NormalVolatility * Sqr(SwaptionMaturity))
pi = 3.14159265359

NormalSwaptionPricing = (1 - 1 / ((1 + SwapRate / Frequency) ^ (Tau * Frequency))) / _
SwapRate * Exp(-r * SwaptionMaturity) * _((SwapRate - StrikeRate) * NormalDist(d) + _
vol * Sqr(T) / Sqr(2 * pi) * Exp(-d ^ 2 / 2))


End Function

'______________________________
 ' BLACK VOLATILITY
'______________________________
```

```
Function BlackVolatility(SwapRate As Double, SwapMaturity As Double, _
NormalVolatility As Double) As Double

Dim pi As Double

pi = 3.14159265359

BlackVolatility = 2 / Sqr(SwapMaturity) * 1 / (NormalDist(NormalVolatility / _
(2 * SwapRate) * Sqr(SwapMaturity / (2 * pi)) + 0.5))


End Function


'____________________________
' BLACK-76 SWAPTION PRICING
'____________________________

Function SwaptionPricing(SwapRate As Double, StrikeRate As Double, _
SwaptionMaturity As Double, Tau As Double, BlackVolatility As Double) As Double
Dim Frequency As Double
Dim r As Double
Dim d1 As Double
Dim d2 As Double

Frequency = 1
```

```
r = 0.001
d1 = (Log(SwapRate / StrikeRate) + BlackVolatility ^ 2 / 2 * SwaptionMaturity) / _
(BlackVolatility * Sqr(SwaptionMaturity))
d2 = (d1 - (BlackVolatility * Sqr(SwaptionMaturity)))

SwaptionPricing = (1 - 1 / ((1 + SwapRate / Frequency) ^ (Tau * Frequency))) / _
SwapRate * Exp(-r * SwaptionMaturity) * (SwapRate * NormalDist(d1) - _
StrikeRate * NormalDist(d2))


End Function
```

12 Appendix 4: Calibration of Hull-White Swaption prices. Python code

```
”””
Hull−White model calibration against Normal−Black

Created on Sun May  1 19:24:55 2016

@author: Victor
”””

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import scipy.optimize as sp

# import the rates data from an excel file

Location = ”..\Black_and_Bootstrap.xlsm”
rates = pd.read_excel(Location, 1)
table_rates=pd.DataFrame({’years’: rates[’T (days)’]/360, ’discounts’: rates[’Discount Factors’],
’forwards’: rates[’Forward Rates’]/100})

# import the normal−black prices data from an excel file

swaptions = pd.read_excel(Location,2)
black_prices = np.zeros((10,11))
black_prices[:,0]=swaptions[’Unnamed: 3’][35:45]
```

```
black_prices[:,1]=swaptions['Unnamed: 4'][35:45]
black_prices[:,2]=swaptions['Unnamed: 5'][35:45]
black_prices[:,3]=swaptions['Unnamed: 6'][35:45]
black_prices[:,4]=swaptions['Unnamed: 7'][35:45]
black_prices[:,5]=swaptions['Unnamed: 8'][35:45]
black_prices[:,6]=swaptions['Unnamed: 9'][35:45]
black_prices[:,7]=swaptions['Unnamed: 10'][35:45]
black_prices[:,8]=swaptions['Unnamed: 11'][35:45]
black_prices[:,9]=swaptions['Unnamed: 12'][35:45]
black_prices[:,10]=swaptions['Unnamed: 13'][35:45]

#root finding algorithm setting inputs

volatility_min=1 #integer, number per thousand
volatility_max=100 #integer, number per thousand
mean_reverting_min=1 #integer, number per thousand
mean_reverting_max=100 #integer, number per thousand
iterations_vol=volatility_max-volatility_min #each iteration will sum 1/1000
#to volatility and mean reverting coefitient
iterations_mean=mean_reverting_max-mean_reverting_min #each iteration will sum 1/1000 to
#volatility and mean reverting coefitient
sum_squared_errors=np.zeros((iterations_mean,iterations_vol))

for z in range (volatility_min,volatility_max):
        for w in range (mean_reverting_min,mean_reverting_max):
                # inputs

                r = 0.001
```

```
sigma = z/1000
a = w/1000

# define vectors to be filled with discounting factors
p_now = np.zeros((table_rates['discounts'].size,1))
p_now[0]=table_rates['discounts'][0]

# loop to fill the vector with the HW analytical formula for Zero Coupon bonds prices
for i in range (1,p_now.size):
        t=table_rates['years'][0]
        mat=table_rates['years'][i]
        p_mat=table_rates['discounts'][i]
        p_t=table_rates['discounts'][0]
        forwardrate=table_rates['forwards'][0]

        funcB1= (1-math.exp(-a*(mat-t)))/a
        funcA1= p_mat/p_t*math.exp(funcB1*forwardrate-math.pow(sigma,2)/(4*a)*
        math.pow(funcB1,2)*(1-math.exp(-2*a*t)))
        p_now[i]=funcA1*math.exp(funcB1*r)

# Forward Swap rates matrix

forward_swap_matrix = np.zeros((10,11))
swap_tenor = np.zeros((10,1))
swaption_maturity = np.zeros((10,11))
swaption_swap_maturity = np.zeros((10,11))

swap_tenor=([1.0,2.0,3.0,4.0,5.0,7.0,10.0,12.0,15.0,20.0])
```

```
swaption_maturity=([1/12,1/3,1/2,1.0,2.0,3.0,4.0,5.0,7.0,10.0,20.0])

# Legs of the Swap and Swap rates

fixed_leg = np.zeros((p_now.size,1))
fixed_leg[0]=p_now[0]*table_rates['years'][0]
floating_leg = np.zeros((p_now.size,1))
floating_leg[0]=0
swap_rates=np.zeros((p_now.size,2))
swap_rates[:,0]=table_rates['years']

for i in range (0,p_now.size):
        floating_leg[i] = p_now[0]-p_now[i]

for i in range (1,p_now.size):
        fixed_leg[i] = fixed_leg[i-1]+p_now[i]*(table_rates['years'][i]-
        table_rates['years'][i-1])
        swap_rates[i,1] = floating_leg[i]/fixed_leg[i]

for i in range (0,10):

        for j in range (0,11):
                swaption_swap_maturity[i,j]=swap_tenor[i]+swaption_maturity[j]

                for k in range (0,p_now.size):
                        if table_rates['years'][k]==swaption_swap_maturity[i,j]:

                                for l in range (0,p_now.size):
                                        if table_rates['years'][l]==swap_tenor[i]:

                                                for m in range (0,p_now.size):
                                                        if table_rates['years'][m]==
                                                        swaption_maturity[j]:
                                                                forward_swap_matrix[i,j]=
                                                                (p_now[m]-p_now[k])/fixed_leg[l]


# Jamshidian decomposition to value Swaptions

#defining all cash flow discounted payments
bond_t=np.zeros((11,1))
bonds_matrix_division=np.zeros((10,11))
forwardrate_vector=np.zeros((11,1))
r_level=np.zeros((10,11))
payments_function=np.zeros((10,11))
swaption_prices_matrix=np.zeros((10,11))

for i in range (0,10):
        cashflows=np.zeros((swap_tenor[i],1))
        sum_of_cashflows=np.zeros((swap_tenor[i],1))
        cashflows_r_star=np.zeros((swap_tenor[i],1))
        sum_of_options=np.zeros((swap_tenor[i],1))
        options_in_swaps=np.zeros((swap_tenor[i],1))

        for j in range (0,11):
```

```
for k in range (0,p_now.size):
        #defining a vector with the bonds we need (=tenors)
        if table_rates['years'][k]==swaption_maturity[j]:
                #defining a vector with the forward rates at the start
                #of each Swap
                forwardrate_vector[j]=table_rates['forwards'][k]
                bond_t[j]=table_rates['discounts'][k]

        #defining a matrix with the bonds we need to discount (maturity/tenors)
        if table_rates['years'][k]==swaption_swap_maturity[i,j]:
                bonds_matrix_division[i,j]=table_rates['discounts'][k]/bond_t[j]

#H−W factors
funcB= (1−math.exp(−a*(swaption_swap_maturity[i,j]−swaption_maturity[j])))/a
funcA= bonds_matrix_division[i,j]*math.exp(funcB*forwardrate_vector[j]−
math.pow(sigma,2)/(4*a)*math.pow(funcB,2)*
(1−math.exp(−2*a*swaption_maturity[j])))
#vector created with all discounted cash flows from interest rate payments
#with HW factors
for l in range (0,cashflows[0].size−1):
        cashflows[l]= lambda r_star: forward_swap_matrix[i,j]*funcA*
        math.exp(−r_star*(swap_tenor[i])*funcB)

sum_of_cashflows[0]=cashflows[0]
for m in range (1,cashflows[0].size):
        sum_of_cashflows[m]=sum_of_cashflows[m−1]+cashflows[m]
#cash flows equal to strike rate
payments_function = lambda r_star:  sum_of_cashflows[sum_of_cashflows.size−1] +
        forward_swap_matrix[i,j]*math.exp(-r_star*swap_tenor[i])-
        forward_swap_matrix[i,j]+0.001
        #r^* level for which cash flows are equal to the notional (strike)
        #calculated using Newton-Raphson
        r_level[i,j]=sp.newton(payments_function, forward_swap_matrix[i,j],
        maxiter=1000000)
        #substituting r^* level in every discounted cash flow, the result will be the
        #strike of each option
        for u in range (0,cashflows_r_star.size):
            cashflows_r_star[u]= forward_swap_matrix[i,j]*
            math.exp(-r_level[i,j]*(swap_tenor[i]-u))
            options_in_swaps[u]= max(forward_swap_matrix[i,j]*
            math.exp(-r*(swap_tenor[i]-u))-cashflows_r_star[u],0)

        sum_of_options[0]=options_in_swaps[0]
        for v in range (1,cashflows_r_star.size):
            sum_of_options[v]=sum_of_options[v-1]+options_in_swaps[v]

        swaption_prices_matrix[i,j]=sum_of_options[sum_of_options.size -1]

# Least-Squared method

square_errors=np.zeros((10,11))

for i in range (0,10):
    for j in range (0,11):
        square_errors[i,j]=math.sqrt(math.pow((swaption_prices_matrix[i,j]-
        black_prices[i,j])/black_prices[i,j],2))
```

```
                sum_squared_errors[w-mean_reverting_min,z-volatility_min]=
                sum_squared_errors[w-mean_reverting_min,z-volatility_min]+square_errors[i,j]
if  sum_squared_errors[w-mean_reverting_min,z-volatility_min]<1.0:
        break
```

13 Appendix 5: Figures

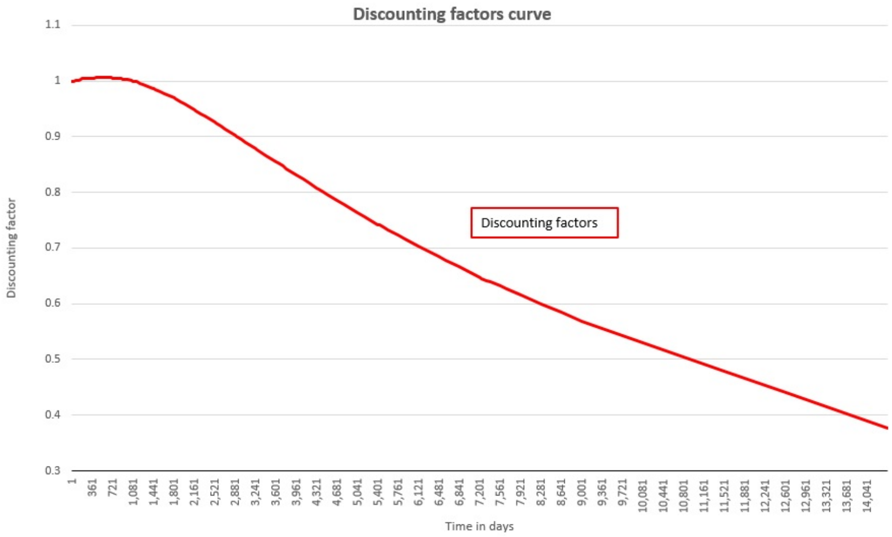
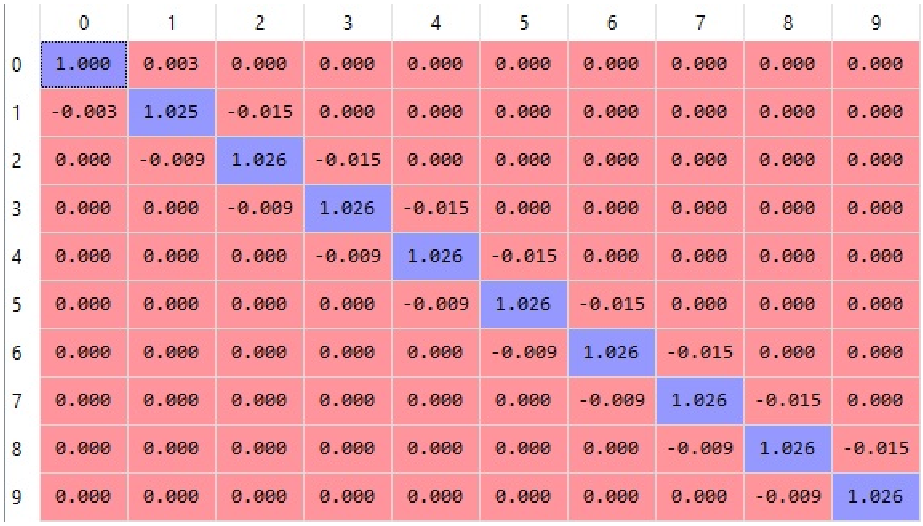
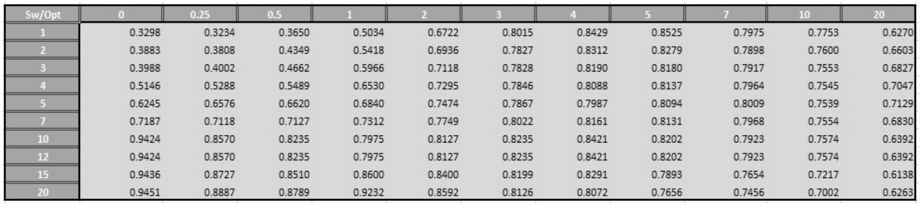


Figure 21: Discounting curve



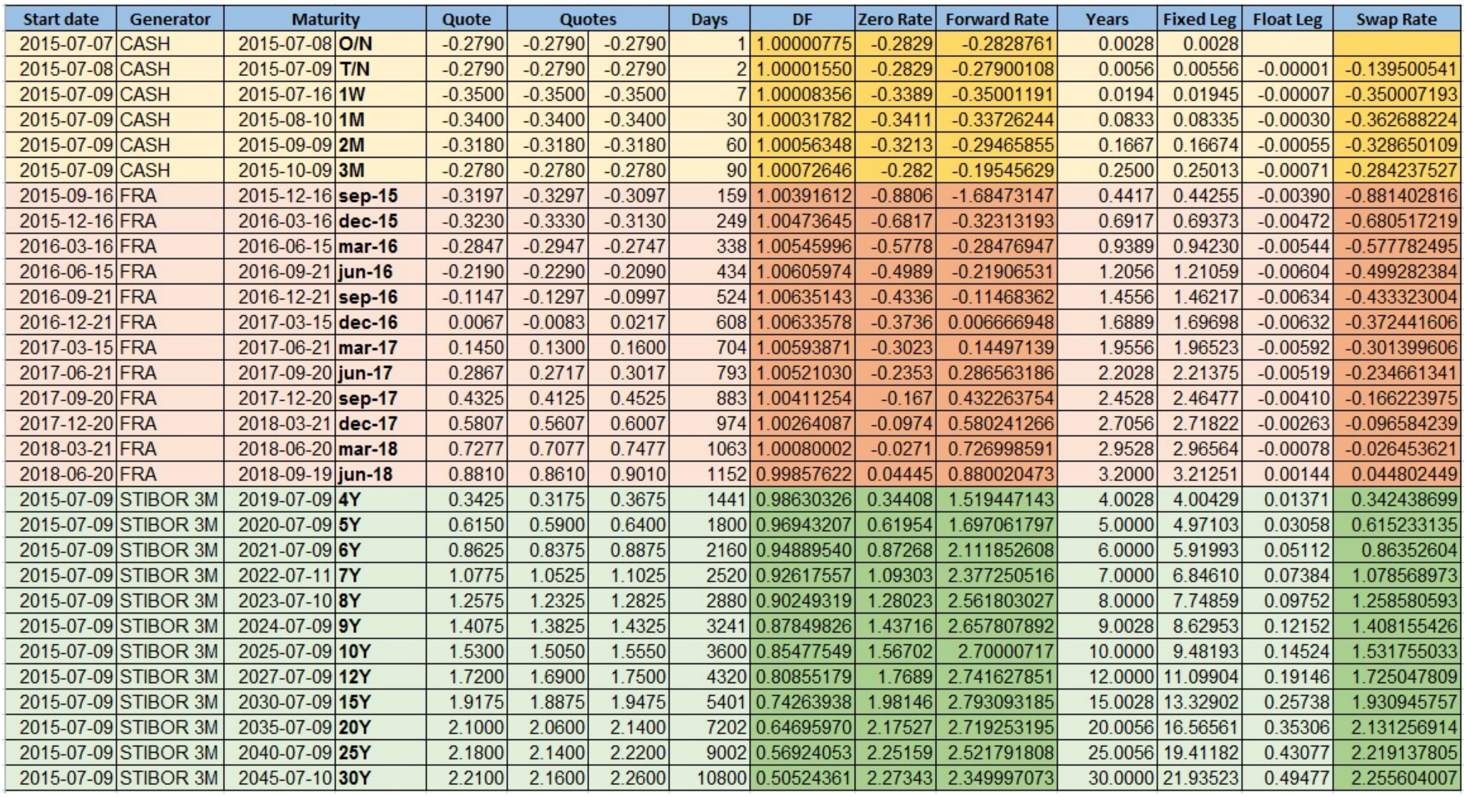
	0	1	2	3	4	5	6	7	8	9
0	1.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	-0.003	1.025	-0.015	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	-0.009	1.026	-0.015	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	-0.009	1.026	-0.015	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	-0.009	1.026	-0.015	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	-0.009	1.026	-0.015	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	-0.009	1.026	-0.015	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	-0.009	1.026	-0.015	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.009	1.026	-0.015
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.009	1.026

Figure 22: First rows and columns of the C-N A matrix



Sw/Opt	0	0.25	0.5	1	2	3	4	5	7	10	20
1	0.3298	0.3234	0.3650	0.5034	0.6722	0.8015	0.8429	0.8525	0.7975	0.7753	0.6270
2	0.3883	0.3808	0.4349	0.5418	0.6936	0.7827	0.8312	0.8279	0.7898	0.7600	0.6603
3	0.3988	0.4002	0.4662	0.5966	0.7118	0.7828	0.8190	0.8180	0.7917	0.7553	0.6827
4	0.5146	0.5288	0.5489	0.6530	0.7295	0.7846	0.8088	0.8137	0.7964	0.7545	0.7047
5	0.6245	0.6576	0.6620	0.6840	0.7474	0.7867	0.7987	0.8094	0.8009	0.7539	0.7129
7	0.7187	0.7118	0.7127	0.7312	0.7749	0.8022	0.8161	0.8131	0.7968	0.7554	0.6830
10	0.9424	0.8570	0.8235	0.7975	0.8127	0.8235	0.8421	0.8202	0.7923	0.7574	0.6392
12	0.9424	0.8570	0.8235	0.7975	0.8127	0.8235	0.8421	0.8202	0.7923	0.7574	0.6392
15	0.9436	0.8727	0.8510	0.8600	0.8400	0.8199	0.8291	0.7893	0.7654	0.7217	0.6138
20	0.9451	0.8887	0.8789	0.9232	0.8592	0.8126	0.8072	0.7656	0.7456	0.7002	0.6263

Figure 23: Normal Volatilities at 2015-07-07, data provided by Röman, J. Swedbank



Start date	Generator	Maturity		Quote	Quotes		Days	DF	Zero Rate	Forward Rate	Years	Fixed Leg	Float Leg	Swap Rate
2015-07-07	CASH	2015-07-08	**O/N**	-0.2790	-0.2790	-0.2790	1	1.00000775	-0.2829	-0.2828761	0.0028	0.0028		
2015-07-08	CASH	2015-07-09	**T/N**	-0.2790	-0.2790	-0.2790	2	1.00001550	-0.2829	-0.27900108	0.0056	0.00556	-0.00001	-0.139500541
2015-07-09	CASH	2015-07-16	**1W**	-0.3500	-0.3500	-0.3500	7	1.00008356	-0.3389	-0.35001191	0.0194	0.01945	-0.00007	-0.350007193
2015-07-09	CASH	2015-08-10	**1M**	-0.3400	-0.3400	-0.3400	30	1.00031782	-0.3411	-0.33726244	0.0833	0.08335	-0.00030	-0.362688224
2015-07-09	CASH	2015-09-09	**2M**	-0.3180	-0.3180	-0.3180	60	1.00056348	-0.3213	-0.29465855	0.1667	0.16674	-0.00055	-0.328650109
2015-07-09	CASH	2015-10-09	**3M**	-0.2780	-0.2780	-0.2780	90	1.00072646	-0.282	-0.19545629	0.2500	0.25013	-0.00071	-0.284237527
2015-09-16	FRA	2015-12-16	**sep-15**	-0.3197	-0.3297	-0.3097	159	1.00391612	-0.8806	-1.68473147	0.4417	0.44255	-0.00390	-0.881402816
2015-12-16	FRA	2016-03-16	**dec-15**	-0.3230	-0.3330	-0.3130	249	1.00473645	-0.6817	-0.32313193	0.6917	0.69373	-0.00472	-0.680517219
2016-03-16	FRA	2016-06-15	**mar-16**	-0.2847	-0.2947	-0.2747	338	1.00545996	-0.5778	-0.28476947	0.9389	0.94230	-0.00544	-0.577782495
2016-06-15	FRA	2016-09-21	**jun-16**	-0.2190	-0.2290	-0.2090	434	1.00605974	-0.4989	-0.21906531	1.2056	1.21059	-0.00604	-0.499282384
2016-09-21	FRA	2016-12-21	**sep-16**	-0.1147	-0.1297	-0.0997	524	1.00635143	-0.4336	-0.11468362	1.4556	1.46217	-0.00634	-0.433323004
2016-12-21	FRA	2017-03-15	**dec-16**	0.0067	-0.0083	0.0217	608	1.00633578	-0.3736	0.006666948	1.6889	1.69698	-0.00632	-0.372441606
2017-03-15	FRA	2017-06-21	**mar-17**	0.1450	0.1300	0.1600	704	1.00593871	-0.3023	0.14497139	1.9556	1.96523	-0.00592	-0.301399606
2017-06-21	FRA	2017-09-20	**jun-17**	0.2867	0.2717	0.3017	793	1.00521030	-0.2353	0.286563186	2.2028	2.21375	-0.00519	-0.234661341
2017-09-20	FRA	2017-12-20	**sep-17**	0.4325	0.4125	0.4525	883	1.00411254	-0.167	0.432263754	2.4528	2.46477	-0.00410	-0.166223975
2017-12-20	FRA	2018-03-21	**dec-17**	0.5807	0.5607	0.6007	974	1.00264087	-0.0974	0.580241266	2.7056	2.71822	-0.00263	-0.096584239
2018-03-21	FRA	2018-06-20	**mar-18**	0.7277	0.7077	0.7477	1063	1.00080002	-0.0271	0.726998591	2.9528	2.96564	-0.00078	-0.026453621
2018-06-20	FRA	2018-09-19	**jun-18**	0.8810	0.8610	0.9010	1152	0.99857622	0.04445	0.880020473	3.2000	3.21251	0.00144	0.044802449
2015-07-09	STIBOR 3M	2019-07-09	**4Y**	0.3425	0.3175	0.3675	1441	0.98630326	0.34408	1.519447143	4.0028	4.00429	0.01371	0.342438699
2015-07-09	STIBOR 3M	2020-07-09	**5Y**	0.6150	0.5900	0.6400	1800	0.96943207	0.61954	1.697061797	5.0000	4.97103	0.03058	0.615233135
2015-07-09	STIBOR 3M	2021-07-09	**6Y**	0.8625	0.8375	0.8875	2160	0.94889540	0.87268	2.111852608	6.0000	5.91993	0.05112	0.86352604
2015-07-09	STIBOR 3M	2022-07-11	**7Y**	1.0775	1.0525	1.1025	2520	0.92617557	1.09303	2.377250516	7.0000	6.84610	0.07384	1.078568973
2015-07-09	STIBOR 3M	2023-07-10	**8Y**	1.2575	1.2325	1.2825	2880	0.90249319	1.28023	2.561803027	8.0000	7.74859	0.09752	1.258580593
2015-07-09	STIBOR 3M	2024-07-09	**9Y**	1.4075	1.3825	1.4325	3241	0.87849826	1.43716	2.657807892	9.0028	8.62953	0.12152	1.408155426
2015-07-09	STIBOR 3M	2025-07-09	**10Y**	1.5300	1.5050	1.5550	3600	0.85477549	1.56702	2.70000717	10.0000	9.48193	0.14524	1.531755033
2015-07-09	STIBOR 3M	2027-07-09	**12Y**	1.7200	1.6900	1.7500	4320	0.80855179	1.7689	2.741627851	12.0000	11.09904	0.19146	1.725047809
2015-07-09	STIBOR 3M	2030-07-09	**15Y**	1.9175	1.8875	1.9475	5401	0.74263938	1.98146	2.793093185	15.0028	13.32902	0.25738	1.930945757
2015-07-09	STIBOR 3M	2035-07-09	**20Y**	2.1000	2.0600	2.1400	7202	0.64695970	2.17527	2.719253195	20.0056	16.56561	0.35306	2.131256914
2015-07-09	STIBOR 3M	2040-07-09	**25Y**	2.1800	2.1400	2.2200	9002	0.56924053	2.25159	2.521791808	25.0056	19.41182	0.43077	2.219137805
2015-07-09	STIBOR 3M	2045-07-10	**30Y**	2.2100	2.1600	2.2600	10800	0.50524361	2.27343	2.349997073	30.0000	21.93523	0.49477	2.255604007

Figure 24: Bootstrapping data. All values from the first seven columns have been provided by Jan Röman