MÄLARDALEN UNIVERSITY

Division of Applied Mathematics

Swaption Pricing under Hull–White Model using Finite Difference Method with Extension to Cancellable Swaps

Master Thesis in Financial Engineering

Author: Xinyan Lin
Supervisors: Jan Röman
Co-supervisor: Anatoliy Malyarenko

January 18, 2015

Contents

List of Figures

List of Tables

Declaration

I declare that this thesis was composed by me, except where explicitly stated in the text. I am responsible for any questions regarding to the works in this study.

Xinyan Lin
October, 2014

Acknowledgement

To my supervisors Jan Röman who is a lecturer in division of Applied Mathematics and Anatoliy Malyarenko who a professor in division of Applied Mathematics. Your altruistic attitude in disseminating your knowledge and experience have given us better chance understanding the world, you made us a brighter future.
To my lovely wife Kaili Chen, without your continuous encouragement, passion, love, I would not have come so far in my studies. You are a star in the sky, every time when I feel perplexed, you always direct me to the right places. I am grateful for what you have sacrificed for me.
Specific thanks to Torsten Dilot who is a specialist in Nuclear Safety Analysis, Torsten has provided valuable advice during my studies as well as in my study life in Sweden. He has treated me and my wife as his part of family.

Abstract

This thesis mainly focuses on analyzing and pricing European swaption via Crank–Nicolson Finite Difference method. This paper begins with some rather common instruments, definitions and valuations are also provided. MATLAB is the main computer language used throughout this paper, for the numerical examples, the MATLAB codes are also provide in the appendix in order for reader to reproduce the result. Also, the paper extends to price cancellable swap in the end.

Chapter 1

Introduction

Derivative market is is getting more and more attentions as the whole financial market getting more involved globally. Various funds emerged since last decade has contributed to the completeness of the market, which also stimulate sorts of financial instruments to be created. Swaps are particularly useful in the restructuring of risk for big financial institute, for example, hedge. Swaption is one of the financial instruments created to meet certain investors' needs, swaptions also exist as long term investment, taking account of the credit risk, the trading of swaptions are mostly allowed among banks. as swaps and swaptions become more and more popular, it is important to understand the valuations.

The most widely accepted model to value swaptions is the Black–76 model in which the underlying volatility is the most important input. However, there are disadvantages to use Black–76 model, one of which comes from its own assumptions: constant volatility and interest rate. These assumptions make inappropriate to value American or Bermudan options which can be exercised before the maturity.

There are a lot studies on models other than Black–76 model that try to incorporate the fluctuations of interest rate and volatility into models. For example, Hull–White model for term structures. Whereas, the studies out there are mostly focus on one or two specific area, for instance, one paper may focus on discussing the volatilities, the other may focus on calibrations. This study is the second step to the paper by Ekstrom and Tysk [1], my main goal is still the same that demonstrate the pricing procedures for swaption. As the works have been done in the first paper, this paper focuses more on the calibrations and some adjustments to the first paper.

Assumptions are needed because there is no model that can capture all the

influencing factors from the market. Worth noted that methods and models provided in this paper are not unique; readers could always find alternatives regarding certain financial situation they are analyzing.
This paper is organized as follows: Chapter 2 refreshes some definitions and formulas as well as some methods that are widely used in pricing instruments; Chapter 3 tries to analyze and price the zero-coupon bond under Hull-White model in Finite Difference approach; Chapter 4 price the swaption in Finite Difference approach; Chapter 5 calibrates the parameters in Hull-White model; Chapter 6 an application of the calibrated Hull–White model;Chapter 7 extends the price of the European swaption to European cancellable swaps; Chapter 8 concludes all chapters and provides some limitations and the alternatives valuations.

Chapter 2

A Quick Review on Basic Definitions and Formulas

2.1 Single–factor Hull–White interest rate model

The price of a zero–coupon bond at time t is denoted as $p(t, T)$ which defines at maturity time T guaranteed to pay 1 unit disregard anything happen in the market. At the maturity, $p(T, T) = 1$, can be expressed as:

$$p(t, T) = e^{-r \cdot (T-t)}$$

under the assumption of constant interest rate r.
The forward rate is the interest rate determined at time t for time t_1 to t_2. Under continuously compounding interest rate, the forward rate can be expressed as:

$$f(t, t_1, t_2) = -\frac{\ln[p(t, t_2)] - \ln[p(t, t_1)]}{t_2 - t_1}$$

There exists many models for the dynamics of the short rate. In this paper we focus on the single–factor Hull–White model. The Hull–White model (1990) is a generalization of the Vasicek model with time dependent parameters $\alpha(t)$ and $\sigma(t)$, one of the most appealing properties of the Hull–White model is that it allows the negative interest rates, as can be seen in the Over–Night–Rate in EUR recently are negative:

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

where

$\theta(t)$ is a deterministic function of time[6]:

$$\theta(t) = \frac{df(t,T)}{dt} + \alpha f(t,T) + \frac{(1-e^{-2\alpha t})\sigma^2}{2\alpha}$$

$f(t,T)$ is the forward rate.

2.2 Options

Definition

An option is generally a contract which gives the owner the right but not the obligation, to buy or sell a security at a predetermined price at a specific date. Depending on the type of option, the put gives the owner the right to sell; the call gives the owner the right to buy. The valuations of an option have to take the option style into consideration, in which we refer to European, Bermudan and American options. European option is an option that can only be exercised on expiration, the Bermudan option can only be exercised on some specific datas pre-agreed, whereas the American option can be exercised on any trading day on or before the expiration date.

Valuation

The value of a call option at maturity is given by:

$$C_T = [S_T - X, 0]^+ \tag{2.2.1}$$

$[\cdot]^+$ means that the value in the bracket is greater or equal to zero, the symbol is used elsewhere in this paper. Under risk-neutral meassure Q, the value at time t can be expressed by the discounted value of the expectation value:

$$C_t = e^{-r\cdot(T-t)} \cdot \mathbb{E}^Q [C_T]$$

where
C = Price of the call option
T = The maturity of the option
t = the valuation time
r = Risk free interest rate
X = Strike price

S = Current underlying price

There are generally four methods for solving Equation (2.2.1). They are Black–Scholes model, Lattice models, Finite Difference Models and Monte Carlo Simulation method.

2.2.1 Black–Scholes Formula

The Black–Scholes formula for a European call option on an non–dividend–paying underlying asset price S is given by:

$$C(S,t) = SN(d_1) - KN(d_2)e^{-r(T-t)}$$

$$d_1 = \frac{ln\frac{S}{K} + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

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

where
$C(S,t)$ =the value of a call option on underlying asset S at time t
S =the underlying price
K =the strike price
$T-t$ =time to maturity
r =annualized risk free rate
σ =the volatility of the underlying asset
$N(\cdot)$ is the cumulative distribution of the standard normal distribution

2.2.2 Lattice Models

In a lattice model as the binomial model [5], we need an upward and a downward movement u and d respectively. When they and the risk-free interest rate are known, we can calculate the risk-neutral probabilities of moving upward q_u and downward q_d, which are given as:

$$q_u = \frac{e^{r\cdot(T-t)} - d}{u - d}$$

$$q_d = 1 - q_u$$

Doing the same procedure for every node all the way up to the maturity, we will create a so called binomial tree. At the maturity, the option price at each node is computed from Equation (2.2.1). To obtain the price of the option, we discount back the payoff back in time.

In the case of trinomial tree, we need to add one more underlying movement m, the corresponding probabilities for non-dividend-paying underlying are given as [4]:

$$p_u = \left(\frac{e^{r\cdot(T-t)/2} - e^{-\sigma\cdot\sqrt{(T-t)/2}}}{e^{\sigma\cdot\sqrt{(T-t)/2}} - e^{-\sigma\cdot\sqrt{(T-t)/2}}} \right)^2$$

$$p_d = \left(\frac{e^{\sigma\cdot\sqrt{(T-t)/2}} - e^{r\cdot(T-t)/2}}{e^{\sigma\cdot\sqrt{(T-t)/2}} - e^{-\sigma\cdot\sqrt{(T-t)/2}}} \right)^2$$

$$p_m = 1 - p_u - p_d$$

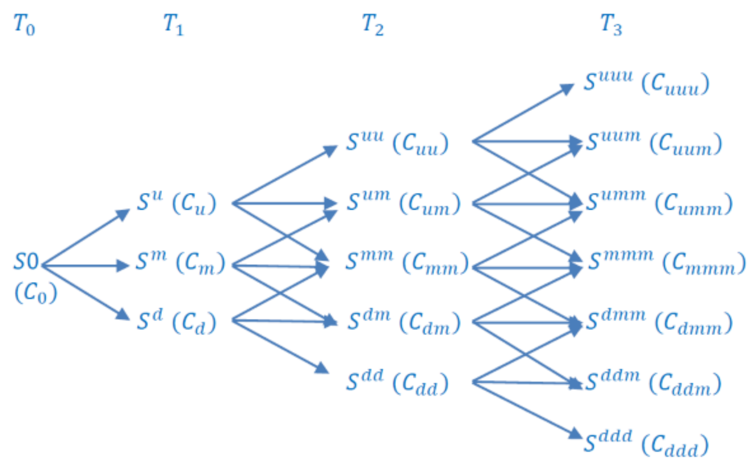


Figure 2.1: Trinomial Tree

The trees look very similar to Figure 2.1, where the subscripts for the underlying S represents the nodes and the subscripts to the C represents the corresponding option price.

2.2.3 Finite Difference Models

In finite difference models, the solution to a partial differential equation is approximated by approximating the differential equation over the domains of independent variables. The option price is expressed in partial differential equations [5], which describes how the option price evolves over time:

$$\begin{cases} \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} - rC = 0 \\ C_T = [S_T - K, 0]^+ \end{cases} \tag{2.2.3.1}$$

Here C represents European call option value, C_T stands for the value of the option at maturity. In the case of put option, the terminal value will be $\max[K - S_T, 0]$. The notations r, σ, S and t are the same as in the previous sections. With the equation (2.2.3.1) in hand, explicit method, implicit method or Crank–Nicolson method can be applied to discrete the derivatives. Particularly, Crank–Nicolson has raised great attention to the option pricing because its unconditional stability advantage [7] in which case we discretise the above Equation (2.2.3.1)as:

$$\frac{\partial C(S_i, t_j)}{dt} = \frac{C_{i,j+1} - C_{i,j}}{\delta t}$$

$$\frac{\partial C(S_i, t_j)}{dS} = \frac{1}{2}\frac{(C_{i+1,j+1} - C_{i-1,j+1}) + (C_{i+1,j} - C_{i-1,j})}{2\delta s}$$

$$\frac{\partial^2 C(S_i, t_j)}{\delta s^2} = \frac{1}{2}\frac{(C_{i+1,j+1} - 2C_{i,j+1} + C_{i-1,j+1}) + (C_{i+1,j} - 2C_{i,j} + C_{i-1,j})}{\delta s^2}$$

By doing such discretization, the PDE will end up as:

$$-P^u C_{i+1,j} - (P^m - 2)C_{i,j} - P^d C_{i-1,j} = P^u C_{i+1,j+1} + P^m C_{i,j+1} + P^d C_{i-1,j+1}$$

$$P^u = -\frac{1}{4}\delta t\left(\frac{\sigma^2}{\delta s^2} + \frac{(r - \sigma^2/2)}{\delta s}\right)$$

$$P^m = \delta t\left(\frac{\sigma^2}{2\delta s^2} + \delta t\frac{r}{2} + 1\right)$$

$$P^d = -\frac{1}{4}\delta t\left(\frac{\sigma^2}{\delta s^2} - \frac{(r - \sigma^2/2)}{\delta s}\right)$$

Here $C_{i,j}$ represents the option value at time j and underlying node i. As can be obviously seen from the above, the PDE ended up with a tridiagonal matrix which can be solved by Thomas algorithm. With suitable boundary conditions [5], the option value can be approximated backward in time.
The Figures 2.2, 2.3 and 2.4 give general ideas of how the different finite difference method approximates the solution.

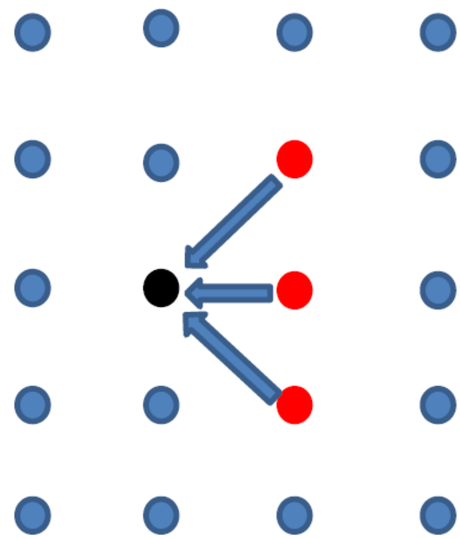


Figure 2.2: Explicit Finite Difference Method

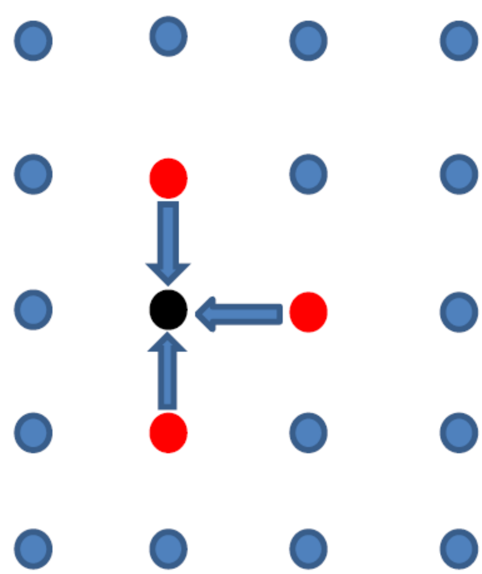


Figure 2.3: Implicit Finite Difference Method

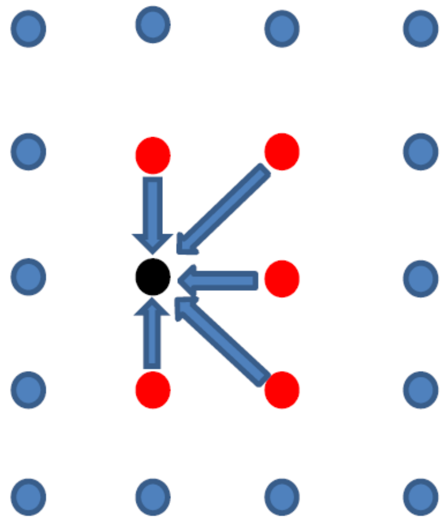


Figure 2.4: Crank–Nicolson Finite Difference Method

2.2.4 Monte Carlo Simulation

Due to the complexity of the financial instruments which makes the computational inefficiency, the Monte Carlo Simulation Method is probably the least appealing method, however, by incorporating more realistic underlying processes, the newly created instruments which have no analytical formulas available can be easily simulated. Therefore, the Monte Carlo Simulation is more often used as comparison to the others.

2.3 Swaps

2.3.1 Single Currency Swaps

Definition

Swap is an over–the–counter contract in which two parties agree to exchange streams of cash flows for some specific periods. Interest–rate swaps are probably the most popular traded, in which one party makes fixed rate payments on a specific principle, and the other party makes floating rate payment on the same principle, which is also known as plain vanilla swap.

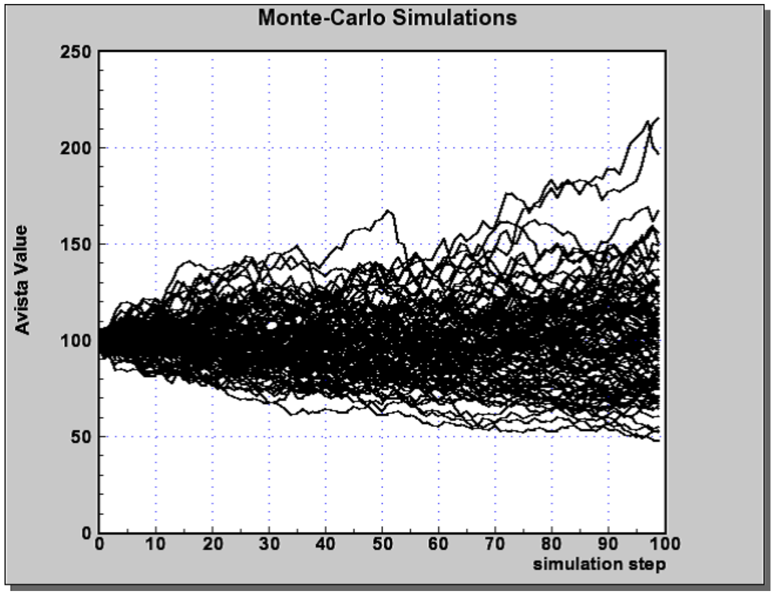
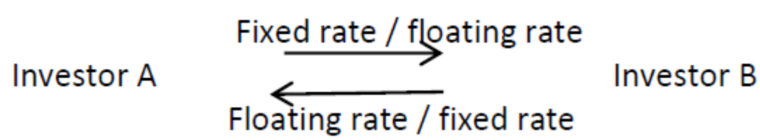


Figure 2.5: Monte Carlo Simulation

Hence, interest swaps are commonly used as a method to hedge their risk of funding. The floating rates are are normally based on specific spread over a LIBOR rate.



In reality, there are a bunch of variants. For instance, investor can enter into float–to–float swaps, or the principle can be specified to be two different currencies in which we call such swaps as dual currency swap.

Valuation

The valuation of the fixed–to–floating interest rate swap is quite obvious theoretically, based on the same principle, we only need to calculate the present value of each leg; the resulting difference is actually the price to enter the swap. However, when entering into a swap, the swap value is to

be considered zero, therefore, we only concern about finding the swap rate in which making the two present values equal, the resulting swap rate is the fixed rate that applies to the fixed payments.
The present values of the fixed leg and the floating leg are expressed as below:

$$PV_{Fixed} = N \cdot [\Sigma_{i=1}^{n} \tau_i p(0, T_i) \cdot x + p(0, T)]$$

$$PV_{Floating} = N \cdot [\Sigma_{j=1}^{n} \tau_j p(0, T_j) \cdot f(0, T_j, T_{j+1}) + p(0, T)]$$

where
N = the notional amount
$p(0, T_i)$ = zero coupon bond prices
$f(0, T_j, T_{j+1})$ = one period forward rate, it represents the rate from time T_j to T_{j+1} contracted at time 0
x =swap rate
τ = the time between two interest payments
By making the above expressions equal, the swap rate x can be solved:

$$x = \frac{\Sigma_{j=1}^{n} \tau_j p(0, T_j) \cdot f(0, T_j, T_{j+1})}{\Sigma_{i=1}^{n} \tau_i p(0, T_i)} \tag{2.3.1.1}$$

So that in the inception of the swap, there is no advantage to either party.
The variation of equation (2.3.1.1) can be seen as:

$$x = \frac{1 - p(0, T)}{\Sigma_{i=1}^{n} \tau_i p(0, T_i)} \tag{2.3.1.2}$$

Because

$$p(0, T_i) = \frac{p(0, T_{i-1})}{1 + \tau_i \cdot f_{i-1,i}}$$

$$f_{i-1,i} \cdot \tau_i \cdot p(0, T_i) = p(0, T_{i-1}) - p(0, T_i)$$

The value of the floating leg in the swap becomes:

$$\Sigma_{i=1}^{T} f_{i-1,i} \cdot \tau_i \cdot p(0, T_i) = \Sigma_{i=1}^{T} p(0, T_{i-1}) - p(0, T_i) = 1 - p(0, T)$$

Noted that the floating rate is reset during the life of the swap, typically, the floating rate is determined at the current payment date and applied to the next payment date. Therefore, the forward rate $f(0, T_j, T_{j+1})$ is used, as the time goes on, T_j becomes further from today, however, the forward only accounts for one period as explained the reset rate is only applied for one period.

2.3.2 Cross Currency Swap

Cross Currency Swap is an agreement between two parties for exchanging interest payments and principals based on two different currencies, it is also known as cross currency interest rate swap. The cross currency swap is widely used for taking favourable advantages. For example, a Swedish company is looking to acquire some U.S. dollars, and a U.S. company is looking for acquire Swedish Krona, in such context, the two companies could make a swap because either Swedish company or U.S. company is much likely to have advantage accessing to domestic debt markets than a foreign company.

2.4 Swaptions

Definition

Swaption, as the name suggests, it is a combination of the swap and option, therefore, it has features of both instruments. A swaption gives the holder the right, but not the obligation to enter a swap at maturity of the option. However, swaps in the swaption are always referred to interest rate swaps. The holder of such a call swaption has the right, but not the obligation to receive fixed in exchange for variable interest rate. Therefore, this option is also known as Payer Swaption. The holder of the put swaption has the right, but not the obligation to pay interest at a fixed rate (Receiver Swaption) and pay variable.
Due to the features from options, swaptions have the same styles as options: European swaption, Bermudian swaption. Features from swap give rise to payer and receiver swaptions which specify the owner pay and receive fixed payments respectively. The owner of a receiver swaption will exercise if the market fixed–rate is less than the strike rate; in contrast, the owner of a payer swaption will exercise if the market fixed–rate is greater than the strike rate.

Swaption example

Suppose the speculator selects to buy a 1–year European payer swaption on a 5-year, 7% Libor swap with a Notional of SEK $10,000,000$.
In this case, we have 1×5 payer swaption with an exercise rate 8%. The underlying swap is 5–year, 7% Libor with Notional SEK $10,000,000$. And

the option premium is SEK $50,000$

On the exercise date:

- If the fixed rate on a 5–year swap were greater than the exercise rate of 7%, then the speculator would exercise to pay the fixed rate 7% which is below the market rate.

- If the fixed rate on a 5–year swap were less than 7%, then the payer swaption would worthless, he loses the premium he paid.

The Figures 2.6 and 2.7 show the payoff diagrams of payer and receiver swaption
Readers can recall the payoff diagrams from call and put option on interest rate, the payoff diagram on payer swaption is equivalent to call option on interest rate, as they both benefit from interest rate rises; on the other hand, the receiver swaption is equivalent to put option on interest rate.

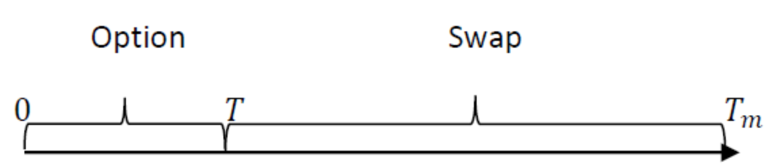
Valuation

The valuation of swaptions is theoretically obvious, again, due to the features from option and swap, evaluating a swaption can be decomposed into two parts:

- Compute the present values of a series of payments based on variable rates and fixed rates. However, the variable rates are not known until the specific time comes, we could only approximate the rates base on the zero coupon bond yield curve provided by the market.

- Evaluate the option part.

The following figure illustrates the basic structure which provides an overall picture to view swaptions, it is helpful for evaluating swaptions.



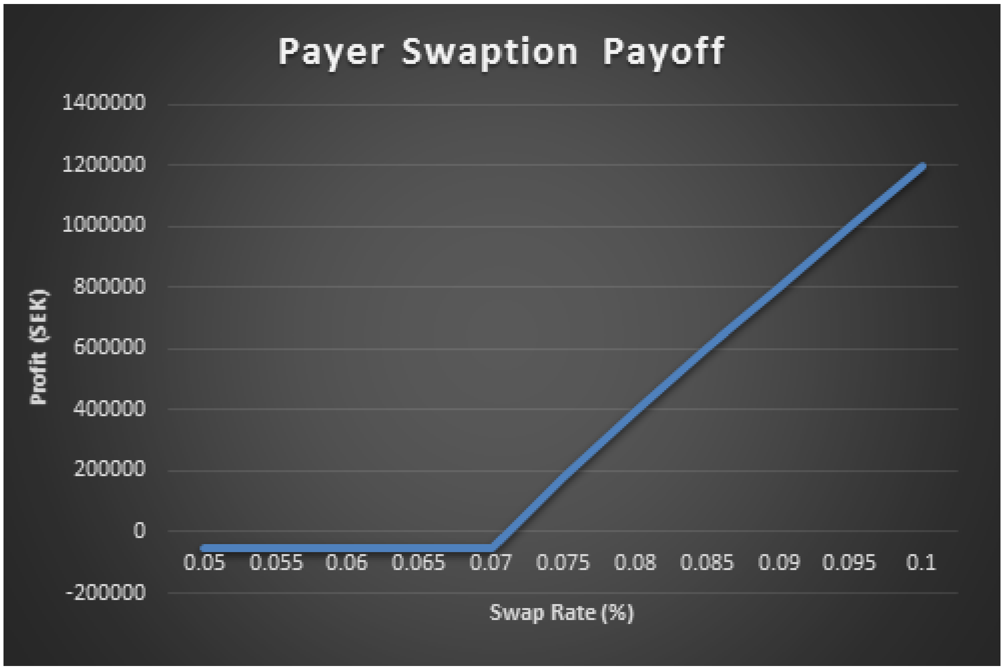


Figure 2.6: Payer Swaption Payoff Diagram at Maturity

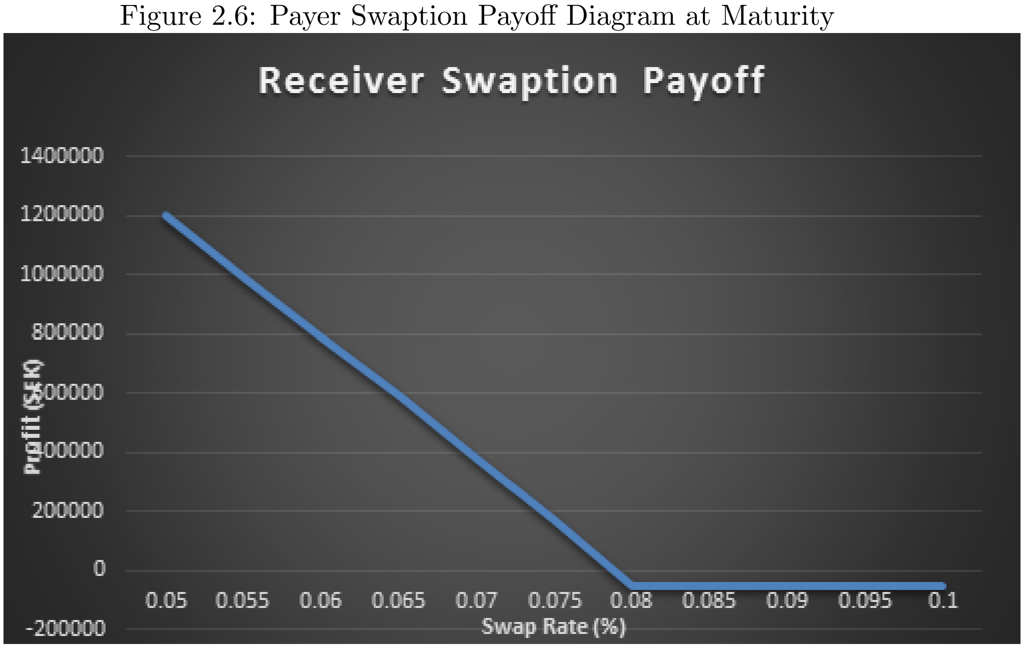


Figure 2.7: Receiver Swaption Payoff Diagram at Maturity

Noted that although evaluating the swaption is theoretically straightforward, in the real world, things are getting rather complex. Interest rates, Libor rates, zero coupon rates, forward rates, counterparty credit, macro economy situations, and underlying volatilities are getting evolved, this is why the models step in. Researches and studies have been done, bootstrapping, assumptions, forecasts have been used to develop formulas in order to catch certain factors which have been concerned to affect the price the most. For instance, Black–Scholes formula. It is developed for pricing options. The major factors are considered are: time(T), underlying volatility(σ), interest rate(r), current underlying price(R), exercise price(K).

European swaption valuation

Cash flows made to the buyer of a payer swaption at time t is:

$$\Sigma_{i=1}^{T} N \cdot e^{-r\cdot(t_i-t)} \cdot (r_F - r_S)(t_i - t_{i-1}) \tag{2.4.1}$$

Where r_F represents the swap rate at market, r_s represents the strike rate of the swaption, r represents current interest rate and t is the starting date of the swap. $0 \leq t \leq t_i$.
The value at today equals:

$$e^{-r\cdot t}\cdot\Sigma_{i=1}^{T} N \cdot e^{-r\cdot(t_i-t)} \cdot (r_F - r_S)(t_i - t_{i-1}) = N\cdot\Sigma_{i=1}^{T} e^{-r\cdot t_i} \cdot (r_F - r_S)(t_i - t_{i-1}) \tag{2.4.2}$$

The receiver swaption can be derived by the same logic:

$$\Sigma_{i=1}^{T} N \cdot e^{-r\cdot(t_i-t)} \cdot (r_S - r_F)(t_i - t_{i-1}) \tag{2.4.3}$$

Analytical Formula for swaptions

For a European receiver swaption (an investor receives fixed payments while pays floating) which starts floating and fixed payments from T to T_m, the payoff of the swaption at time t ($t \leq T \leq T_m$) is given by:

$$Pswaption(t) = [K \cdot \Sigma_{i=1}^{m} \tau_i \cdot p(t, T_i) - (p(t, T) - p(t, T_m))]^+ \tag{2.4.4}$$

where
t = the inception date of the swaption
T_m = the maturity of the swap
T = the inception date of the swap or the maturity of the swaption
Since the swaption is also an option, the above formula can be exploited to be equivalent to:

$$Pswaption(t) = \Sigma_{i=1}^{m} \tau_i \cdot p(t, T_i) \left[K - \frac{p(t,T) - p(t,T_m)}{\Sigma_{i=1}^{m} \tau_i \cdot p(t, T_i)} \right]^+ \tag{2.4.5}$$

$\Sigma_{i=1}^{m} \tau_i \cdot p(t, T_i)$ represents an annuity.
$\frac{p(t,T)-p(t,T_m)}{\Sigma_{i=1}^{m} \tau_i \cdot p(t,T_i)}$ is the forward swap rate, which is the rate at time t an investor wants to enter into a swap starts t_0 and lasts to T
The price of a receiver swaption can be interpreted as a European call option on a face value 1 coupon bond with coupon rate K as they both benefit from interest rate falls.

2.5 Cancellable Swap

Definition

A cancellable swap is a swap that has options embedded which enable the holder to terminate a swap entered before maturity with losing any premium. Regarding to fixed rate payer or receiver has the right, the cancellable swap can be divided into callable or putable swap. For example, an investor wants to enter into a payer swap at time 0, which matures at time T. However, he is afraid of the swap rate goes down at time $t, t < T$. In such case, one of the alternatives that the investor could take is entering into a cancellable swap which has the same maturity as the swap, but with an option embedded that allows the investor to cancel the swap at time t_i, $t < t_i < T$. Typically, the cancellable swap specifies one cancellable date in its life, which is also known as European cancellable swap. However, if the cancellable swap has many cancellable dates, it is known as Bermudan cancellable swap. An American cancellable swap that allows the holder to cancel a swap in a time interval.

Valuation

Such a variation of swap has been popular for some investors. We can valuate it by replicate it with other instruments which provides the same price. For simplicity, in this paper, I only analyze the European cancellable swap which has only one cancellable date. For a European payer cancellable swap (callable swap), the investor pays fixed rate and receives floating rate, and also has a right to cancel the swap at time t. cancelling a swap is equivalent to enter into an opposite swap, in this case, the investor enters into a receiver swap. Therefore, a European payer cancellable swap is equivalent to a payer

swap plus a receiver swaption with strike rate equals to the callable swap rate.
For example, a 10 years callable swap can be called at year 5 is quoted at 6% is equivalent to a 10 years plain vanilla swap say at 5% plus a 5 years into 5 years receiver swaption where the strike rate is 5%.
According to the above analysis, the price of a callable swap (fixed payer) is equivalent to the plain vanilla swap plus a long position in receiver swaption. By the same logic, the price of a putable swap (fixed receiver) is equivalent to a plain vanilla swap minus a payer swaption. The plain vanilla swap which has no cash changes hands in the inception, therefore, the price of a swaption is incorporated into the swap rate respectively making the cancellable swap paying higher or lower swap rate. In other words, the callable swap has higher fixed rate and the putable swap has lower swap rate compare to regular interest rate swap.

Chapter 3

Pricing Zero–Coupon bond under Hull-White model

In this paper, I introduce Crank–Nicolson Method [7] for price zero–coupon bond based on the famous Hull–White model. First of all, the partial differential equation (PDE) under Hull–White model [2] can be found as following:

$$\frac{\partial p(r,t)}{\partial t} + \frac{1}{2}\sigma(t)^2\frac{\partial^2 p(r,t)}{\partial r^2} + (\theta(t) - \sigma(t)r)\frac{\partial p(r,t)}{\partial r} - rp(r,t) = 0$$

To be consistent with the notation in the early chapters, $p(r,t)$ represent the value of the zero–coupon. Interested readers can find the detailed derivation on Appendix A.

In order for applying the Crank–Nicolson Method, we need to define the domain of our approximation to $[0,T] \times [r_{\text{min}}, r_{\text{max}}]$ and then discretise the time T into N equally spaced units with δt step size, and interest rate r into M equally spaced units with δr step size:

$$\delta t = \frac{T}{N}; \delta r = \frac{r_{\text{max}} - r_{\text{min}}}{M}$$

$p(r_i, t_j) \approx p(i\delta r, j\delta t)$, where $i = 1...M, j = 1...N$

3.1 Numerical Solution of the PDE

A solution of a partial differential equations generally is not unique unless we have boundary conditions which specify where the solution is defined. The detailed and formal investigation on boundary conditions are found in

[1]. Here in this paper, I choose $\frac{\partial^2 p}{\partial r^2} = 0$ to find the first and final points. The discretization of $\frac{\partial^2 p}{\partial r^2}$ is presented below.
Intuitively, $p(r_{\max,t_j}) = 0, t_j > 0$ represents the value of a zero–bond will have to drop to 0 when the interest rate is extremely high, say 50%; $p(r_i, T) = 1, 0 < r_i < r_{\max}$ represents the bond pays the face value of 1 in the maturity.
The term structure equation under the Hull-White model is given by:

$$\begin{cases} \frac{\partial p(r,t)}{\partial t} + \frac{1}{2}\sigma(t)^2 \frac{\partial^2 p(r,t)}{\partial r^2} + (\theta(t) - \alpha r)\frac{\partial p(r,t)}{\partial r} - rp(r,t) = 0 \\ p(r,T) = 1 \end{cases} \tag{3.1.1}$$

where
$\alpha(t)$ is the mean reversion parameter, $\sigma(t)$ is the volatility of the short rate. $\theta(t)$ is a deterministic function calculated from initial term structure, where it is defined:

$$\theta(t) = \frac{\partial f(t,T)}{\partial t} + \alpha f(t,T) + \frac{(1 - e^{-2\alpha t})\sigma^2}{2\alpha}$$

where
$f(t,T) = r(t,T) + t\frac{\partial r(t,T)}{\partial t}$
$\frac{\partial f(t,T)}{\partial t} = 2\frac{\partial r(t,T)}{\partial t} + \frac{\partial r^2}{\partial t^2}$

$f(t,T)$ is the forward rate, $r(t,T)$ is the continuously compounded interest rate given by the yield curve. For instance, Interest rate 6%, continuous compounding is: $e^{0.06} - 1 \approx 6.18\%$
Under Crank–Nicholson, let $p_{i,j} \approx p(r_i, t_j)$ be the numerical solution of the terminal value problem at time t, the derivatives can be approximated as:

$$\frac{\partial p(r_i, t_j)}{\partial t} = \frac{p_{i,j+1} - p_{i,j}}{\delta t}$$

$$\frac{\partial p(r_i, t_j)}{\partial r} = \frac{1}{2}\frac{(p_{i+1,j+1} - p_{i-1,j+1}) + (p_{i+1,j} - p_{i-1,j})}{2\delta r}$$

$$\frac{\partial^2 p(r_i, t_j)}{\partial r^2} = \frac{1}{2}\frac{(p_{i+1,j+1} - 2p_{i,j+1} + p_{i-1,j+1}) + (p_{i+1,j} - 2p_{i,j} + p_{i-1,j})}{\delta r^2}$$

by putting them to the PDE and do some easy algebra, we then have:

$$-P^u \cdot p_{i+1,j} - (P^m - 2) \cdot p_{i,j} - P^d \cdot p_{i-1,j} = P^u \cdot p_{i+1,j+1} + P^m \cdot p_{i,j+1} + P^d \cdot p_{i-1,j+1}$$

$$P^u = -\frac{1}{4}\delta t\left(\frac{\sigma^2}{\delta r^2} + \frac{\theta(t_j) - \alpha r_i}{\delta r}\right)$$

$$P^m = \delta t\left(\frac{\sigma^2}{2\delta r^2} + \delta t\frac{r_i}{2} + 1\right)$$

$$P^u = -\frac{1}{4}\delta t\left(\frac{\sigma^2}{\delta r^2} - \frac{\theta(t_j) - \alpha r_i}{\delta r}\right)$$

Noted that the detailed transformation is located in the Appendix A. When the boundary condition 3.1.1 is applied, the P^u and P^m change slightly:

$$P^u = -\frac{1}{4}\delta t\frac{\theta(t_j) - \alpha r_{\min}}{\delta r}$$

$$P^m = 1 + \delta t\frac{r_{\min}}{2}$$

In order to price the zero–coupon bond in the Hull–White Model, we have to calculate the $\alpha(t)$, which is done by bootstrapping the zero curve then to obtain the forward rates through interpolation.

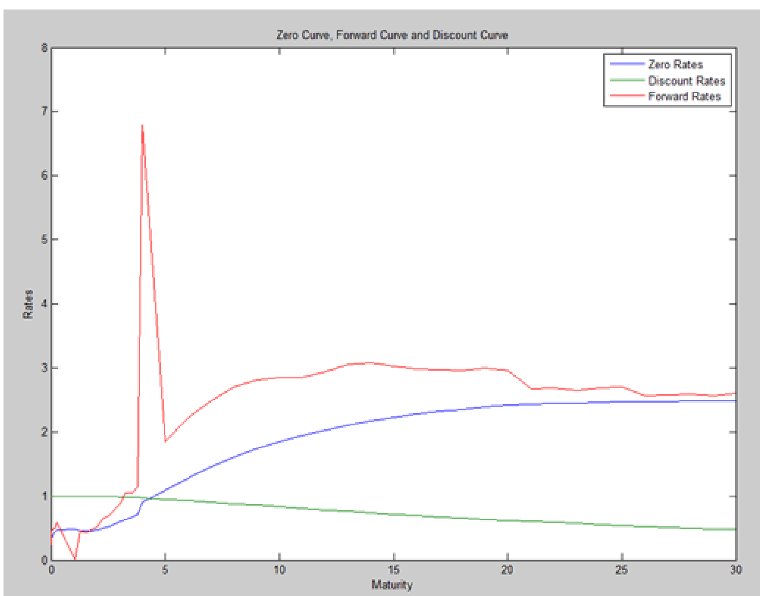


Figure 3.1: Display of Zero Curve, Forward Curve and Discount Curve

3.2 Numerical results

With $0 \leq r \leq 0.5$, $T = 1$, $dr = 0.05$, $dt = 0.02$, $M = 50$, $N = 10$ and the model parameters $\sigma = 0.1$, $\alpha = 0.05$, $\theta = 0.025$, although the Hull–White interest model allows the negative interest rate, in my numerical example, I try to avoid the negative rate by setting the minimum rate to zero. The zero bond price under Hull–White model gives:

$Maturity = 1$	$FaceValue = 1$	$\theta = 0.025$
$\sigma = 0.1$	$\alpha = 0.05$	$r_{min} = 0$
$r_{max} = 0.5$	$dt = 0.02$	$dr = 0.01$

Table 3.1: Model Inputs

r = 0.01 – 0.25	Zero Bond Price	T = 0.26–0.5	Zero Bond Price
0.01	0.999867445304551	0.26	0.760444763999349
0.02	0.989395624364098	0.27	0.749694635536810
0.03	0.979207384402766	0.28	0.738266670793813
0.04	0.969256539943234	0.29	0.726022225470504
0.05	0.959505575154329	0.30	0.712809026641745
0.06	0.949924326410486	0.31	0.698462768212305
0.07	0.940488763458410	0.32	0.682809545974898
0.08	0.931179871632850	0.33	0.665669186869713
0.09	0.921982632181426	0.34	0.646859482773974
0.10	0.912885092871526	0.35	0.626201284782278
0.11	0.903877516710007	0.36	0.603524351877765
0.12	0.894951592859077	0.37	0.578673781721497
0.13	0.886099690741718	0.38	0.551516785623299
0.14	0.877314135984150	0.39	0.521949509982681
0.15	0.868586485361802	0.40	0.489903558332692
0.16	0.859906777461915	0.41	0.455351837141375
0.17	0.851262736556270	0.42	0.418313339533068
0.18	0.842638909436110	0.43	0.378856497561283
0.19	0.834015718965725	0.44	0.337100777227718
0.20	0.825368424125663	0.45	0.293216260508440
0.21	0.816665984564314	0.46	0.247421052229894
0.22	0.807869838289532	0.47	0.199976461407255
0.23	0.798932614093130	0.48	0.151180029266845
0.24	0.789796815381836	0.49	0.101356600807474
0.25	0.780393528790090	0.50	0.050847753880045

Table 3.2: Zero–Coupon Bond Price under the assumed inputs

As can be seen from the figures above, the one year zero–coupon bond price drops down quickly when the interest rate goes up beyond 20%. When the interest rate is within the reasonable range, say 5%, the prices stay well above 0.8 and decrease slowly. This observation proves that Crank–Nicolson method gives a good approximation.

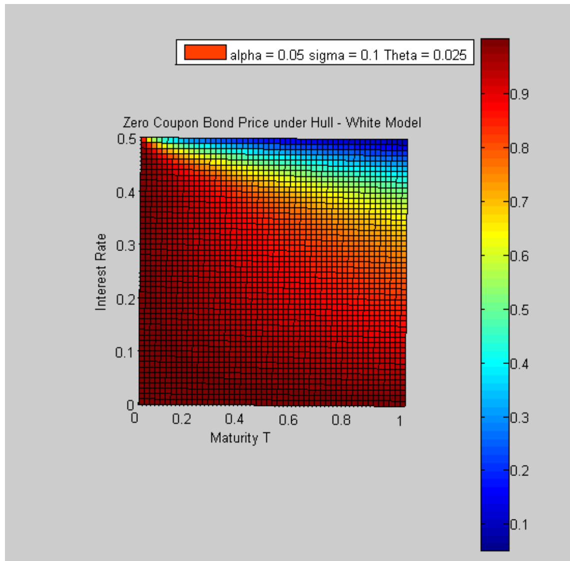


Figure 3.2: Heat Map of the zero–coupon bond price under Hull–White model with parameter stated above

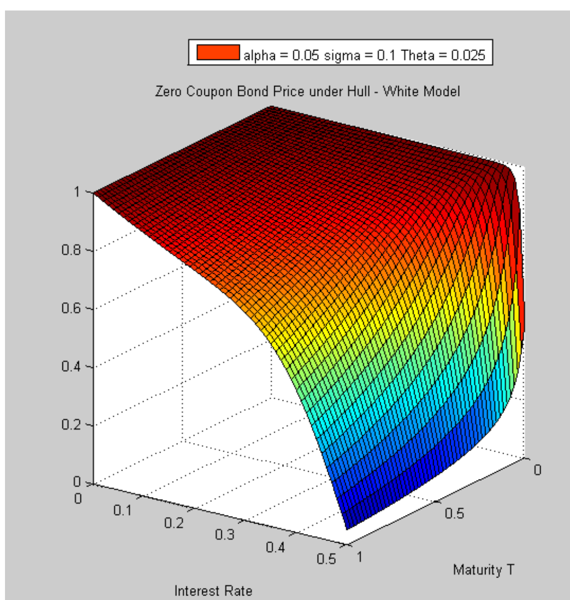


Figure 3.3: Visualizing Zero–Coupon bond under Hull–White Model

Chapter 4

Pricing Swaption under Hull–White Model through Finite Difference Method

The worldwide accepted pricing swaption model is probably the Black–76 model, as it was demonstrated in section 2.4, the analytical formula (2.4.5) for swaptions looks very much the same as for options, which means that pricing swaption can be done in the same manner as in pricing options in Black model. The only input needed to implement the Black model is the implied volatility which is quoted by the market. While, because the market does not quote every possible maturities, for example 2×6 is not quoted by the market.

This paper tries to develop an alternative way to address the limitation in pricing swaptions. Pricing swaption under Hull–White interest rate model has actually the same Partial Differential Equation as in zero–bond case, however, in the zero-bond case, the initial boundary condition at maturity is set to the face value; in the swaption case, the initial boundary condition should be changed in order to represent the option property. The PDE and the boundary condition for European payer swaption is given:

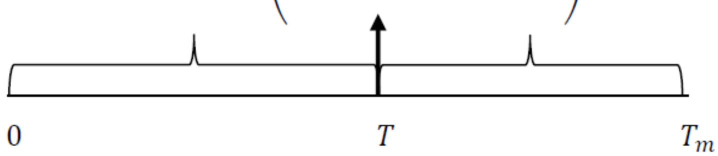
$$\begin{cases} \frac{\partial V(r,t)}{\partial t} + \frac{1}{2}\sigma(t)^2 \frac{\partial^2 V(r,t)}{\partial r^2} + (\theta(t) - \alpha r)\frac{\partial V(r,t)}{\partial r} - rV(r,t) = 0 \\ V(r,T) = \max(r^* - K, 0) \end{cases}$$

where
K=strike rate
$V(r,t)$=swaption value
r^*=market swap rate at maturity or the forward swap rate

4.1 Formulation in Finite Difference Method

In order to be consistent with the zero–coupon bond pricing, the swaption pricing in this section is also under Crank–Nicolson Scheme. The solution of a European swaption $V(r_i, T)$ approximated at the maturity is expressed as:

$$V(r_i, T) = \begin{pmatrix} [max(K - r^*_{min}), \ 0] \\ \vdots \\ \vdots \\ [max(K - r^*_{max}), \ 0] \end{pmatrix}$$



The rest part is equivalent to put option on stock under Finite Difference method, the interest readers are referred to [2] for more details.
However, when the option price is found at time 0, the price has to be multiplied by an annuity factor. The reason is that, the option price only represents the time interval between t_0 and T, while the swap is exchanging cash flows during the lifetime of the swap. To take this fact into account, an annuity factor is multiplied.

4.2 Numerical Result

As in the analytical formula for swaption (2.4.5), we need to calculate the forward swap rate r^*, in order to approximate the value of the swaption. Letting T_m to be the maturity of the swap and T to be the inception of the swap, for example a swaption with 4×6 structure means $T = 4$, $T_m = 10$ and the swaption has 4 years maturity to decide whether it is worth to enter into the 6 year swap.

$\theta = 0.025$	$\sigma = 0.1$	$\alpha = 0.05$	$K = 0.01878$	4×6
$r_{\text{min}} = 0$	$r_{\text{max}} = 0.5$	$dt = 0.02$	$dr = 0.01$	$r^* = 0.02864$

Table 4.1: Model Inputs

With the given input, the swaption price is equal to 0.0335 for notional 1. However, the Hull–White single factor interest model needs to be calibrated

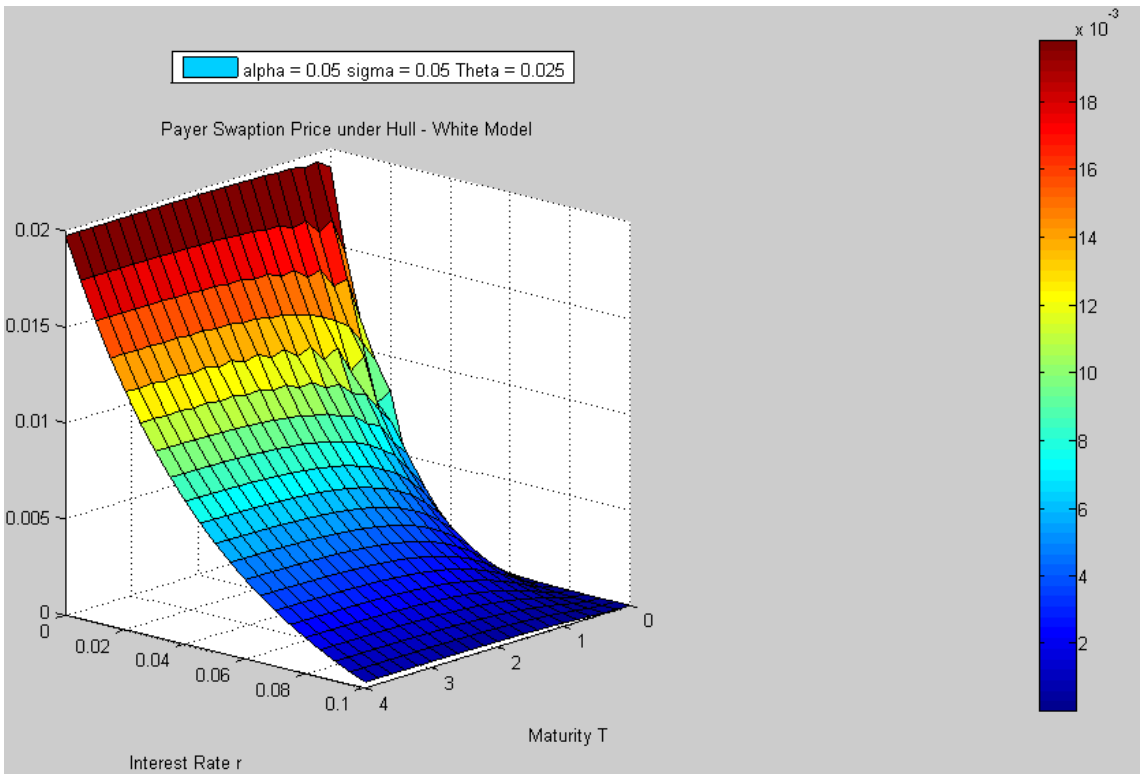


Figure 4.1: Visualize Swaption Price under Hull–White interest rate Model

to the market data in order to be used for pricing the inactive instruments, which lead us to the Chapter 5.

Chapter 5

Calibration

The purpose of calibration is to compute prices of not so liquid derivatives instruments, or more complex instruments. For example, the market only quote swaptions at certain maturity, however, the investors may encounter a swaption with more frequent swap payments. Calibration is to estimate the parameters of the model fitting a given observed market data (cap or swaption implied volatility surface). The calibration is performed by minimizing the sum of the squares of the differences between model and market prices.

$$Min\Sigma_i^n(Marketprice_i - Modelprice_i)^2 \tag{5.1}$$

Calibrating a model is a rather complex process, to be able to accomplish this task, we have to use some Optimization Algorithms, for example, Quasi–Newton techniques, which produces the best estimate of the parameters that minimize the objective function (5.1).

The Hull–White interest rate model is always calibrated by:

- Calibration to cap volatility.

  At–the–money Euro cap–volatility

- Calibration to swaption volatility.

  Swaption−volatility matrix

5.1 Calibrate Hull–White to Swaption Volatility

As stated in Chapter 3, $\theta(t)$ can be determined from initial term structure where only two parameters α and σ discard to be calibrated. However, the

$\theta(t)$ itself is a function of α and σ , therefore, in this paper, I decided to calibrate the three parameters at the same time.
In Table 5.1, the columns represent the option maturities, on the other hand the rows are the maturities of the underlying swap. For instance, the combination of third row and fourth column correspond to 1 year option on a 3 year swap.

5.2 Calibration Process

In this paper, the calibration is done by Matlab, there are a couple of optimization tools in matlab that can be used to find the minimum of the objective function (5.1). Yet, some authors may use the percentage change in market price and model price as their objective function to be minimized. Here I chose the function lsqcurvefit to optimize the α and σ.

The calibratioin process is done as following:

1. Produce the market price from the quoted market swaption volatilities. With the volatility matrix, the market prices of swaptions are available by putting the volatilities into the Black formula. From the swaption volatility matrix in Table 5.1, the Black price is produced and presented in Table 5.2 :

Opt/swap	1	2	3	4	5	6	7	8	9	10
1	31.1	33	33.6	32.5	31.7	30	28.4	27.3	26.4	25.7
2	33.8	32.6	30.6	29.5	28.8	28	27	26.5	26	25.5
3	30.8	28.8	27.9	27.3	26.9	26.5	26	25.8	25.5	25.3
4	28.4	27.5	26.8	26.1	25.5	25.4	25.2	25.1	25	25
5	27.3	26	25.2	24.7	24.4	24.3	24.3	24.4	24.5	24.6
6	26	25.3	24.7	24.1	23.8	23.8	23.7	23.7	24	24.1
7	24.5	24.6	24.3	23.8	23.3	23.3	23.3	23.4	23.5	23.5
8	24	23.8	23.8	23.2	22.8	22.8	22.8	23	23	23
9	23.5	23	23.4	22.6	22.3	22.3	22.4	22.5	22.5	22.5
10	22.9	22.2	22	21.9	21.8	21.8	21.9	21.9	22	22

Table 5.1: European Swaption volatility matrix from Swedbank on 6th August, 2013

On the Table 5.1 The rows represent the option maturities, on the other hand the columns are the maturities of the underlying swap. For

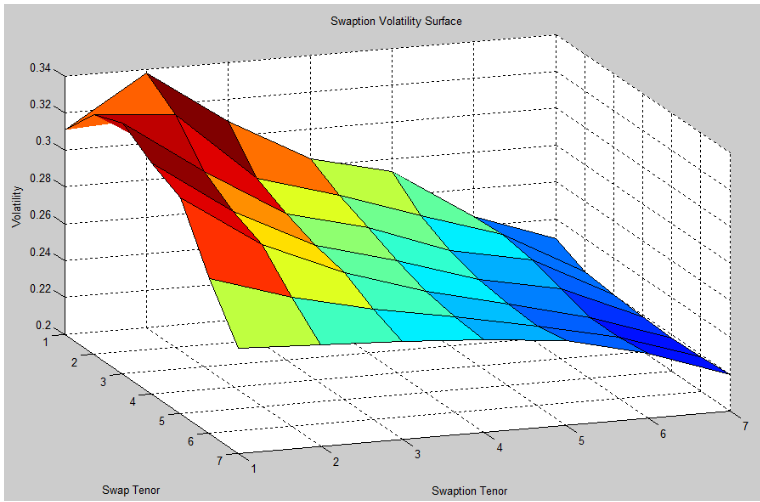


Figure 5.1: The surface of Swaption Volatility

instance, the combination of third row and fourth column correspond to the volatility of a 3 year option on a 4 year swap. The original data has only $1, 2, 3, 4, 5, 7$ and 10 years maturity, the year 6 volatility was interpolated from year 5 and 7, the year 8 and 9 volatilities are interpolated from year 7 and 10.

2. Write a function that calculates the model price by using the swaption structure as input variable.

3. The optimization tool $lsqcurvefit$ needs an initial guess of the parameters to start from. Here I simply write a initial guess $x0 = [0.01, 0.01, 0.01]$

4. Finally, $x = lsqcurvefit(@calibration, x0, xdata, ydata)$ calls the function $lsqcurvefit$ and returns the best estimate of θ, α and σ from the initial guess.

5.3 Numerical Result

In this section, I try to calibrate to a European Receiver swaption. Since the optimation process takes a rather long time in the Finite Difference Model, here I chose only to calibrate the model parameter from 1×1 swaption up to 5×5 swaption, another reason is that we are missing the data for 6 years

Opt/swap	1	2	3	4	5	6	7	8	9	10
1	.193674	.467438	.789733	1.099541	1.413741	1.666638	1.883476	2.106596	2.294825	2.486264
2	.378584	.795249	1.197246	1.603247	2.011119	2.377233	2.692181	3.024673	3.325513	3.613104
3	.495465	.980736	1.468201	1.952457	2.422581	2.860284	3.260358	3.672135	4.054491	4.435994
4	.591835	1.160482	1.713527	2.22098	2.700147	3.204031	3.66552	4.136344	4.585038	5.039164
5	.661662	1.260535	1.821768	2.357796	2.875324	3.391889	3.915312	4.436617	4.945546	5.445553
6	.707069	1.345042	1.937734	2.480369	3.35781	3.69883	4.07669	4.596576	5.166051	5.663454
7	.68607	1.358655	1.98237	2.551977	3.084793	3.654082	4.205001	4.761777	5.287446	5.775776
8	.703653	1.365862	2.008073	2.587372	3.138419	3.713337	4.268403	4.832824	5.342684	5.836222
9	.698076	1.341171	2.033603	2.590064	3.158228	3.732881	4.293736	4.840198	5.351249	5.851137
10	.695281	1.344412	1.971263	2.577827	3.157739	3.715775	4.273107	4.796688	5.332117	5.818351

Table 5.2: The European swaption price from Black model

Opt/swap	1	2	3	4	5	6	7	8	9	10
1	1.61265	1.84696	2.07475	2.28063	2.44275	2.57187	2.67431	2.75373	2.8147	2.86121
2	2.07998	2.30581	2.50468	2.65145	2.76496	2.85234	2.91725	2.96535	3.00053	3.04016
3	2.53415	2.7191	2.8437	2.93727	3.00793	3.05766	3.09263	3.116	3.14734	3.17066
4	2.90366	2.99849	3.07098	3.12593	3.16217	3.18547	3.19914	3.22396	3.24123	3.25524
5	3.09228	3.15421	3.19983	3.22663	3.24159	3.24842	3.26972	3.28343	3.29425	3.30199
6	3.21376	3.2525	3.27072	3.27837	3.27896	3.29888	3.31044	3.31926	3.32534	3.32174
7	3.29196	3.29977	3.30044	3.29536	3.31562	3.32629	3.33412	3.33917	3.33365	3.32817
8	3.30699	3.30442	3.29603	3.32164	3.33323	3.3412	3.34595	3.33889	3.33223	3.3268
9	3.30228	3.29046	3.32653	3.33981	3.34807	3.35239	3.34341	3.33535	3.32894	3.32388
10	3.27993	3.3389	3.35247	3.35964	3.36251	3.35035	3.34014	3.33233	3.32634	3.32029

Table 5.3: The Corresponding Forward Swap rate or Strike rate

$dt = 0.02$	$dr = 0.0001$	$r_{min} = 0$	$r_{max} = 0.15$

maturity, such choice would make my calibration process less subject to the interpolation.
The calibrated parameters are:
$\sigma = 0.0065846, \alpha = 0.058089, \theta = 0.00072754$
The best parameter estimations are found under the MATLAB default setting that the final change in the sum of squares relative to its initial value is less than the default value of the function tolerance.
The Figures 5.4, 5.5, 5.6 and 5.7 give the fitted price from different angle which can lead to an easy conclusion that the parameters are fairly well calibrated which reproduced the market price with a squared 2-norm of the residual $4.1425e-06$.
The calibrated prices matrix together with the selected market prices matrix are shown in Appendix C.
There are some opinions on calibration methods. Some authors suggest taking an estimation of one parameter from historical data, then calibrate the other. Such method is appealing since it saves times in the calibration

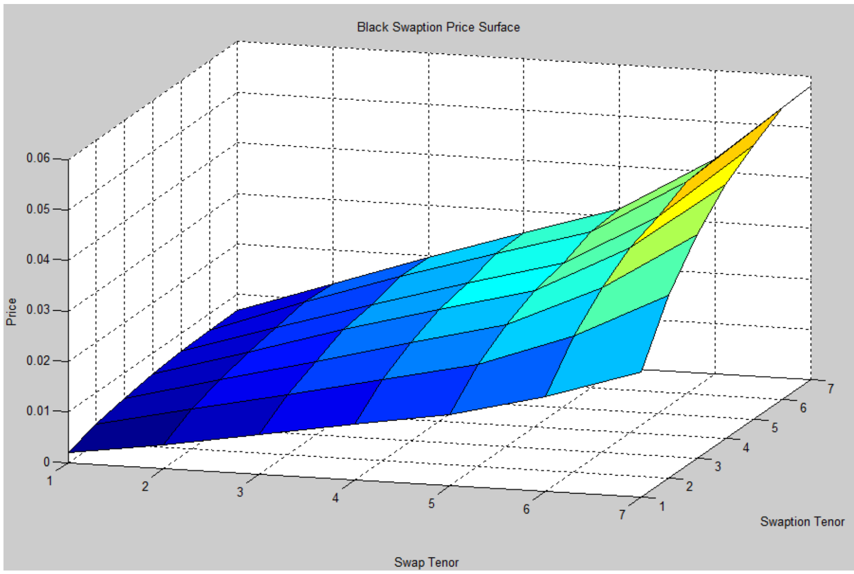


Figure 5.2: The surface of Black Swaption Price

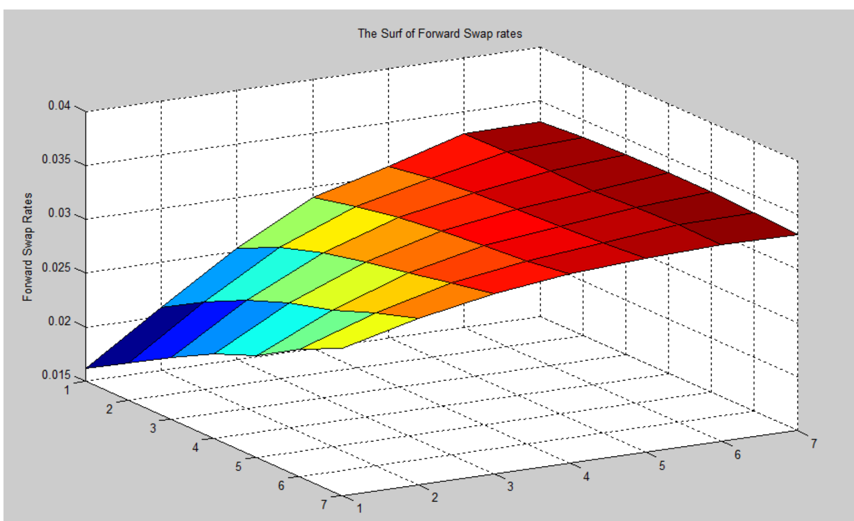


Figure 5.3: The Surface of the Corresponding Forward Swaption Rates

process, however, the calibration is a forward process, getting a backward historical estimation involved may not be necessary. I have also made a calibration in such way, while the calibrated prices do not fit market prices

as good as it should be.
The MATLAB code for calibration is located in Appendix D.

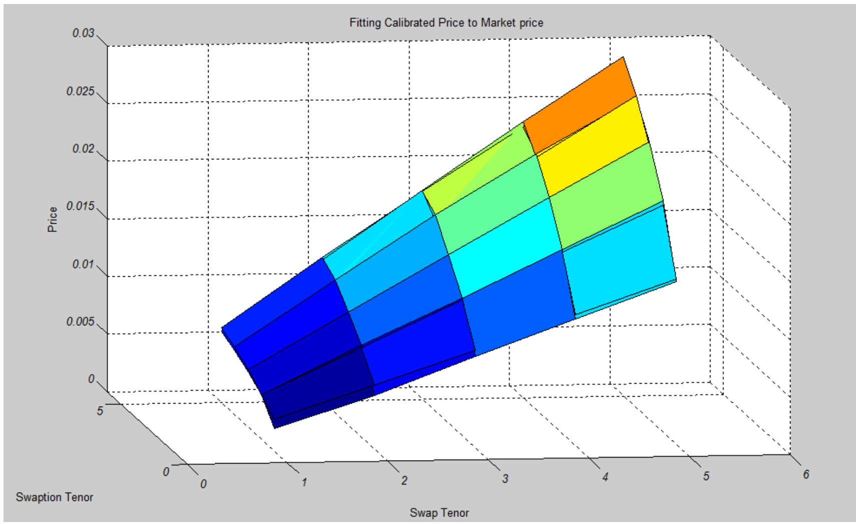


Figure 5.4: Fitting Calibrated Model Price to Market Price (1)

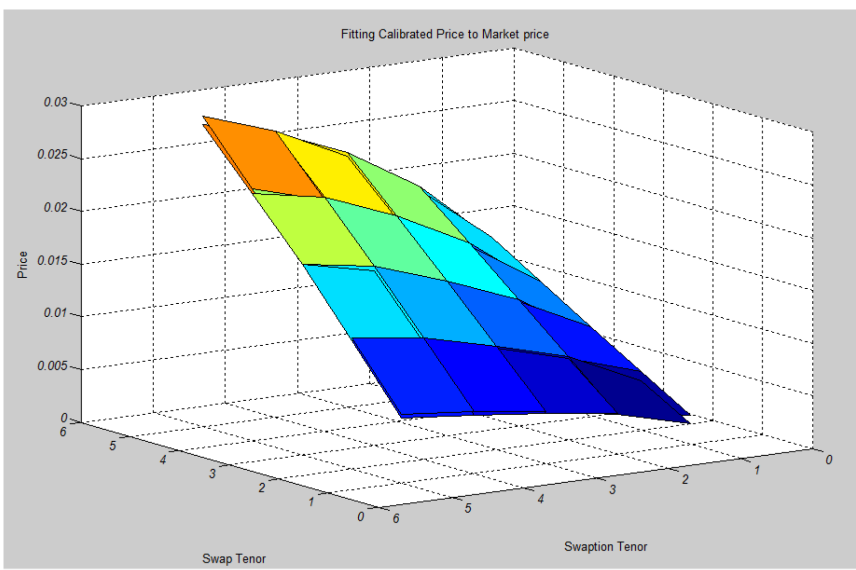


Figure 5.5: Fitting Calibrated Model Price to Market Price (2)

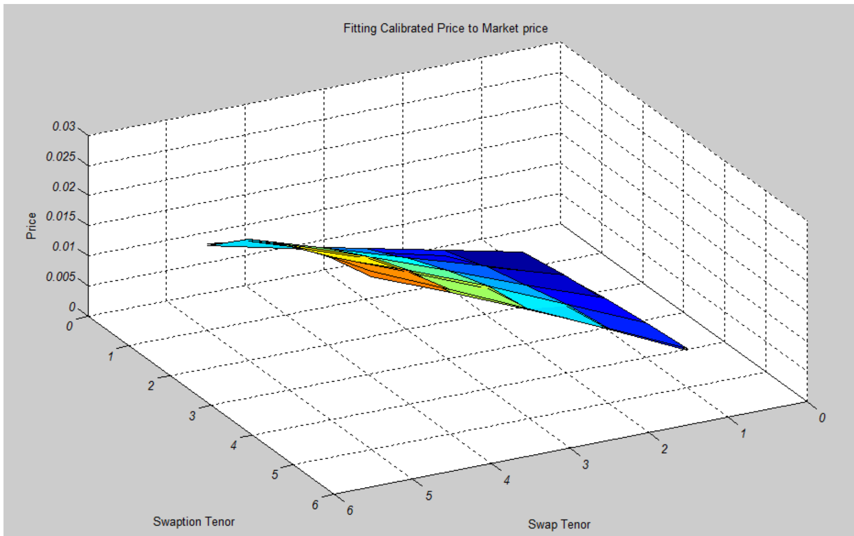


Figure 5.6: Fitting Calibrated Model Price to Market Price (3)

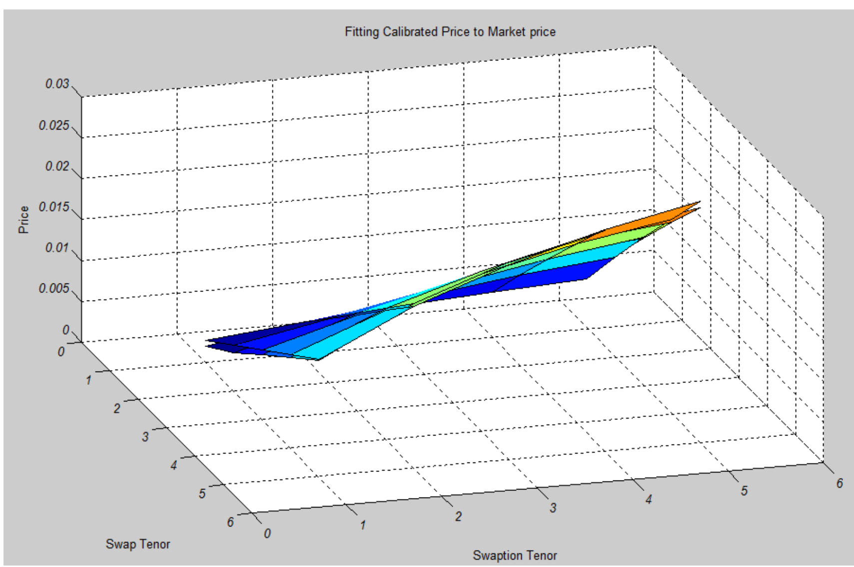


Figure 5.7: Fitting Calibrated Model Price to Market Price (4)

Chapter 6

Pricing Swaption under Calibrated Hull–White Model

6.1 Numerical Example

Under the calibrated Hull-White model:

$dr = (\theta(t) - \alpha(t)r(t))dt + \sigma(t)dW(t)$
where
$\sigma = 0.0065846, \alpha = 0.058089, \theta = 0.00072754$

$dt = 0.02$	$dr = 0.0001$	$r_{\min} = 0$	$r_{\max} = 0.15$

A 4 year to 6 year swaption has a value of 3.24% of its notional compare to 3.21% in Black Model. The 0.03% of the difference could result from the calibration process, however, it is an acceptable error in the pricing.

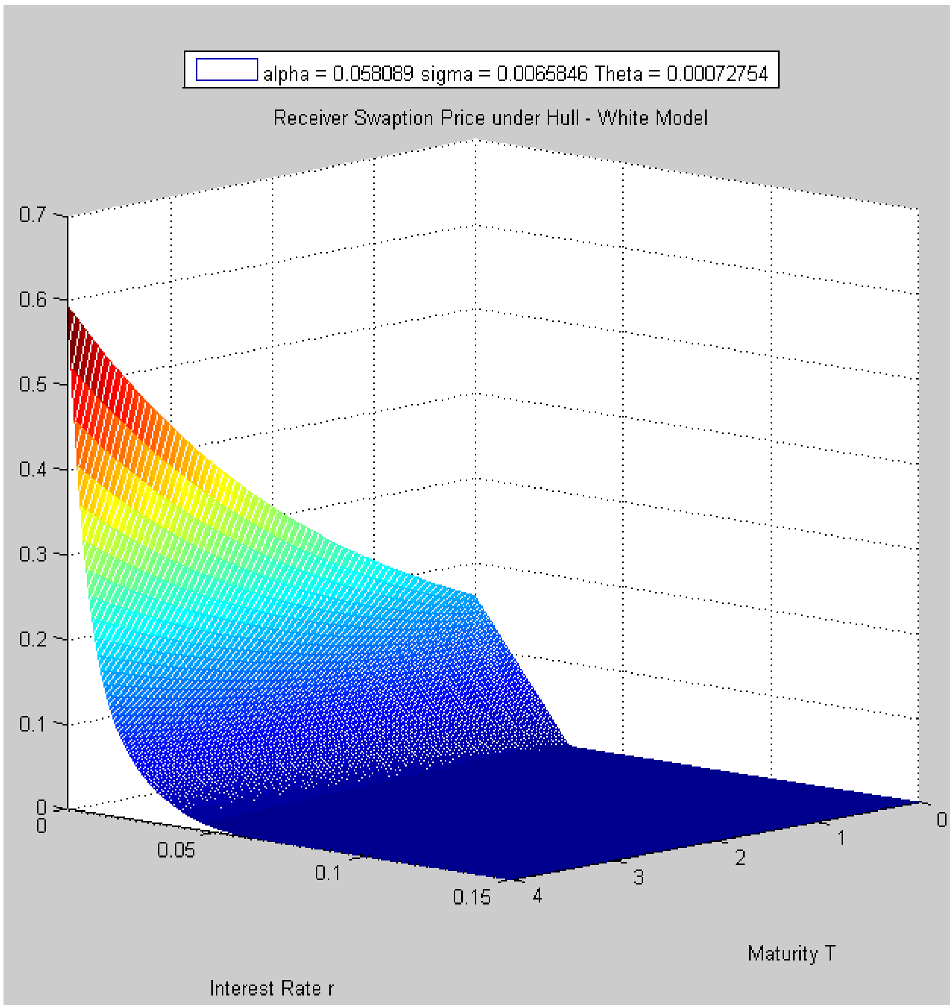


Figure 6.1: Receiver Swaption Price

Chapter 7

Pricing European Callable Swap

As discussed in Section 2.5, entering a European cancellable swap is equivalent to enter a Plain Vanilla swap and at same time enter to a European swap, the same logic applies to Bermudan cancellable swap and American cancellable swap. In this section, I try to price a European callable swap, which is equivalent to price a Vanilla swap plus a European receiver swaption. Having priced the swaption, the callable swap can be derived from discounting the basis point over the plain vanilla swap rate to match the swaption price.

7.1 Numerical Example

Continue with the example in Section 6.1, suppose we enter a European callable swap which is callable at year 4, the market quote on a 10 year Plain Vanilla swap is 2.63607%, having calculated the European 4×6 swaption above 3.24%, discounting the swaption price over 10 year time horizon gives 0.37% which is added to the Plain Vanilla swap 2.63607% gives the price of the European callable swap 3.00607%

Chapter 8

Conclusion and Future work

As demonstrated in the Section 5, the calibrated parameters in Hull-White model do not really seem sensible from financial point of view. However, mathematically, they are the best estimation from the market. Yet, there is no consensus among investors which model always gives the best price for such inactive instruments, especially for swaptions which price dependent largely on the shape of term structure as well as the instantaneous interest rates.

As noted in the paper, the data I used in this paper is based on August 6, 2013. The calibrated parameters may result very much differently from time to time. Investors need to watch closely enough on the market in order to get the most updated data when it comes to calibration as well as pricing interest rate derivatives. For Finite Difference Method to be well implemented, the boundary conditions need to be well organized regarding to different underlying instrument. The boundary conditions need to be set not only for computational sense, but also financial meaning will always need to be taken into account.

Chapter 9

Summary of reflection of objectives in the thesis

9.1 Objective 1 Knowledge and understanding

The chapters in this paper cover a broad knowledge in the field of this research. In chapter 2, I show definitions and formulas in pricing certain financial instruments. As knowing a programming language is a must in solving financial problems, the Matlab was selected to help me complete this research.

9.2 Objective 2 Methodological knowledge

In section 2, I show four different methods for option pricing, while the Crank–Nicolson method is selected for this research in the following chapters. In section 3.2, the Crank–Nicolson method is applied to transform the partial differential equation into systems of equations which is solved by Matlab located in appendix.

9.3 Objective 3 Critically and systematically integrate knowledge

The sources used in this paper are mainly acquired from Mälardalen University's library and websites searched by Google. Learning from public sources has becoming popular, while it still remains challenging since not all the articles are closely related to this paper. Articles with complete understanding or providing hints for this paper are cited.

9.4 Objective 4 Independently and creatively identify and carry out advanced tasks

In section 3.1, the expression for numerical solution of a zero–coupon bond is derived under Crank–Nicolson method, this finding is contributed to the chapter 5, 6, 7 with correct boundary conditions. The Matlab code in the appendix are made in the way that is easily understandable and modifiable.

9.5 Objective 5 Present and discuss conclusions and knowledge

This paper is understandable by reader with knowledge in linear algebra, multivariate calculus, fairly good knowledge in financial world. Readers who have knowledge in programming would have better understanding in some chapters, such as chapter 4, 5, 6. In the conclusion section, I point out some limitations which lead to future researches. As it is always the case that one research paper can not take everything into account.

9.6 Objective 6 Scientific, social and ethical aspects

In this paper, sources are correctly cited to the best of by knowledge. In the chapter 4, by showing the finite difference method in pricing swaption together with the appendix regarding the Matlab code, this paper could be used more intensively later on when it comes more complex financial instruments. Making mathematics more interesting and prevailing and will benefit students, further researchers and society at large.

Bibliography

[1] Erik Ekstrom, Johan Tysk. Boundary condtions for the single-factor term structure equation, The Annals of Applied Probability, Vol 21, No.1, 332-350, 2011.

[2] Sebastian Schlenkrich, Evaluating Sensitivities of Bermudan Swaptions, March 20, 2011

[3] Jiang Xue, Pricing Callable Bonds, December 2011

[4] Röman, J., Lecture Notes in Analytical Finance I II, working paper, Mlardalens University 2012

[5] Les Clewlow, Chris Strickland, Implementing Derivatives Models

[6] Damiano Brigo, Fabio Mercurio, Interest Rate Models, Theory and Practice: With Smile, Inflation and Credit, Springer Finance, 64-65, 2006

[7] Fadugba S.E., Edogbanya O.H, Zelibe S. C Crank Nicolson Method for Solving Parabolic Partial Differential Equations, International Journal of Applied Mathematics and Modeling IJA2M, Vol.1, No. 3, 8-23, September, 2013

Appendix A

Crank-Nicolson Finite Difference Scheme

The notations are the same as in section 3.1
Under Crank-Nicholson method, the discretizations for Equation 3.1.1 are presented below:

$p_{i,j} = p(r_i, t_j)$
$\frac{\partial p(r_i,t_j)}{dt} = \frac{p_{i,j+1} - p_{i,j}}{\delta t}$
$\frac{\partial p(r_i,t_j)}{dr} = \frac{1}{2} \cdot \frac{(p_{i+1,j+1} - p_{i-1,j+1}) + (p_{i+1,j} - p_{i-1,j})}{2h}$
$\frac{\partial^2 p(r_i,t_j)}{dr^2} = \frac{1}{2} \cdot \frac{(p_{i+1,j+1} - 2 \cdot p_{i,j+1} + p_{i-1,j+1}) + (p_{i+1,j} - 2 \cdot p_{i,j} + p_{i-1,j})}{\delta r^2}$

By putting them into the Equation 3.1.1, the equation will be:

$$\frac{\partial p_{i,j}}{dt} = r_i \cdot p_{i,j} - (\theta(t_j) - \alpha r_i) \cdot \frac{p_{i+1,j} - p_{i-1,j}}{2\delta r} - \frac{1}{2} \cdot \sigma^2 \cdot \frac{p_{i+1,j} - 2 \cdot p_{i,j} + p_{i-1,j}}{\delta r^2} = \frac{r_i}{2} \cdot (p_{i,j+1} + p_{i,j}) - \frac{\theta(t_j) - \alpha r_i}{4\delta r} \cdot [(p_{i+1,j+1} - p_{i-1,j+1}) + (p_{i+1,j} - p_{i-1,j})] - \frac{\sigma^2}{4\delta r^2}[(p_{i+1,j+1} - 2 \cdot p_{i,j+1} + p_{i-1,j+1}) + (p_{i+1,j} - 2 \cdot p_{i,j} + p_{i-1,j})]$$

After some simple algebra, the equation becomes:

$$-\frac{p_{i,j+1} - p_{i,j}}{\delta t} = (-\frac{\sigma^2}{4\delta r^2} + \frac{\theta(t_j) - \alpha r_i}{4\delta r}) \cdot p_{i-1,j} + (\frac{\sigma^2}{2\delta r^2} + \frac{r_i}{2}) \cdot p_{i,j} + (-\frac{\sigma^2}{4\delta r^2} - \frac{\theta(t_j) - \alpha r_i}{4\delta r}) \cdot p_{i+1,j} + (-\frac{\sigma^2}{4\delta r^2} + \frac{\theta(t_j) - \alpha r_i}{4\delta r}) \cdot p_{i-1,j+1} + (\frac{\sigma^2}{2\delta r^2} + \frac{r_i}{2}) \cdot p_{i,j+1} + (-\frac{\sigma^2}{4\delta r^2} - \frac{\theta(t_j) - \alpha r_i}{4\delta r}) \cdot p_{i+1,j+1}$$

Then, by grouping the similar terms, the equation ends up in a nice form below:

$(\frac{\delta t\sigma^2}{4\delta r^2}+\frac{\delta t(\theta(t_j)-\alpha r_i)}{4\delta r})\cdot p_{i+1,j}+(1-\frac{\delta t\sigma^2}{2\delta r^2}-\frac{\delta tr_i}{2})\cdot p_{i,j}+(\frac{\delta t\sigma^2}{4\delta r^2}-\frac{\delta t(\theta(t_j)-\alpha r_i)}{4\delta r})\cdot p_{i-1,j} = (-\frac{\delta t\sigma^2}{4\delta r^2}-\frac{\delta t(\theta(t_j)-\alpha r_i)}{4\delta r})\cdot p_{i+1,j+1}+(\frac{\delta t\sigma^2}{2\delta r^2}+\frac{\delta tr_i}{2}+1)\cdot p_{i,j+1}+(-\frac{\delta t\sigma^2}{4\delta r^2}+\frac{\delta t(\theta(t_j)-\alpha r_i)}{4\delta r})\cdot p_{i-1,j+1}$

Since we work backward in time, the pricing problem of zero-coupon bond becomes:

$$-P^u\cdot p_{i+1,j}-(P^m-2)\cdot p_{i,j}-P^d\cdot p_{i-1,j} = P^u\cdot p_{i+1,j+1}+P^m\cdot p_{i,j+1}+P^d\cdot p_{i-1,j+1}$$

where
$P^u = -\frac{1}{4}\delta t(\frac{\sigma^2}{\delta r^2}+\frac{\theta(t_j)-\alpha r_i}{\delta r})$
$P^m = (\frac{\delta\sigma^2}{2\delta r^2}+\frac{\delta tr_i}{2}+1)$
$P^d = -\frac{1}{4}\delta t(\frac{\sigma^2}{\delta r^2}-\frac{\theta(t_j)-\alpha r_i}{\delta r})$

Appendix B

Working Data

The following data was taken from Swedbank 6th August, 2013 which was plotted in Figure 3.1

	Time	ZeroRates	DiscountingFactors	ForwardRates
O/N	0.002778	0.2433	0.9999933	0.2433
T/N	0.005556	0.2682	0.9999853	0.2879800432
1W	0.019444	0.2985	0.9999264	0.4241258681
1M	0.075	0.3874	0.9996604	0.4788951121
2M	0.158333	0.4334	0.9992522	0.4901084381
3M	0.236111	0.4728	0.9987962	0.5868567041
Sep-14	1.030556	0.4738	0.9987936	0.0003276673
Dec-14	1.269444	0.455	0.9977092	0.4547317613
Mar-15	1.525	0.4482	0.996629	0.4238862423
Jun-15	1.775	0.4533	0.9954653	0.4673273277
Sep-15	2.030556	0.4692	0.9941428	0.5202021204
Dec-15	2.269444	0.4933	0.9926216	0.6410260715
Mar-16	2.522222	0.524	0.9908684	0.6993463009
Jun-16	2.794444	0.5612	0.9887319	0.7929254002
Sep-16	3.047222	0.5977	0.9865331	0.8807471803
Dec-16	3.263889	0.6331	0.9843053	1.0434282594
Mar-17	3.536111	0.6755	0.9814805	1.0557439106
Jun-17	3.788889	0.716	0.9786351	1.1485573079
4Y	4.002778	0.9014	0.9645151	6.7948130919
5Y	5.002778	1.091	0.9468255	1.8510677899
6Y	6.002778	1.2791	0.9259965	2.2244355162
7Y	7.011111	1.454	0.9030818	2.4850241631
8Y	8.005556	1.6061	0.8791892	2.6962795455
9Y	9.005556	1.7404	0.8548151	2.8114930879
10Y	10.008333	1.8512	0.8307911	2.8427864681
11Y	11.011111	1.9432	0.8073346	2.8560832989
12Y	12.013889	2.0269	0.7838782	2.9402872077
13Y	13.011111	2.1054	0.7603376	3.0576046220
14Y	14.008333	2.1731	0.7373807	3.0743638585
15Y	15.013889	2.2313	0.7152963	3.0239439644
16Y	16.011111	2.2778	0.6943147	2.9854482258
17Y	17.011111	2.3185	0.673995	2.9702623457
18Y	18.019444	2.3548	0.6542167	2.9537910935
19Y	19.016667	2.3873	0.6349671	2.9948622939
20Y	20.013889	2.4161	0.616471	2.9644256513
21Y	21.013889	2.4284	0.6002334	2.6692702182
22Y	22.013889	2.4397	0.5843378	2.6839339312
23Y	23.025	2.45	0.568905	2.6471653116
24Y	24.019444	2.4595	0.5538797	2.6915487906
25Y	25.016667	2.4682	0.5391721	2.6987644906
26Y	26.019444	2.4725	0.5254765	2.5658092411
27Y	27.022222	2.4764	0.5120925	2.5728722825
28Y	28.019444	2.4801	0.4990494	2.5872056509
29Y	29.025	2.4836	0.4863386	2.5657446941
30Y	30.025	2.4868	0.473851	2.6012162844

Appendix C

Compare Model Prices and Market Prices

The following presents the comparison of the calibrated model prices to the market prices:
Calibrated Model Prices =

0.0027413	0.0055984	0.008472	0.011373	0.014321
0.0038156	0.0077514	0.011764	0.015765	0.019698
0.0047377	0.009584	0.014425	0.019193	0.023834
0.0055432	0.011073	0.016542	0.021889	0.027081
0.006136	0.012199	0.018121	0.023888	0.029457

Swaption Black Prices =

0.0019	0.0047	0.0079	0.0110	0.0141
0.0038	0.0080	0.0119	0.0160	0.0201
0.0050	0.0098	0.0146	0.0195	0.0241
0.0059	0.0115	0.0170	0.0222	0.0270
0.0065	0.0124	0.0180	0.0235	0.0286

Appendix D

Matlab Codes

1. Matlab code for zero curve, forward curve and discount curve

   - input rates
   - calculate the forward rates and plot

```
clear all
```

input rates

```
inputrates=...
[ 0.002778          0.2433          0.9999933
  0.005556          0.2682          0.9999853
  0.019444          0.2985          0.9999264
  0.075000          0.3874          0.9996604
  0.158333          0.4334          0.9992522
  0.236111          0.4728          0.9987962
  1.030556          0.4738          0.9987936
  1.269444          0.4550          0.9977092
  1.525000          0.4482          0.9966290
  1.775000          0.4533          0.9954653
  2.030556          0.4692          0.9941428
  2.269444          0.4933          0.9926216
  2.522222          0.5240          0.9908684
  2.794444          0.5612          0.9887319
  3.047222          0.5977          0.9865331
  3.263889          0.6331          0.9843053
  3.536111          0.6755          0.9814805
```

```
  3.788889           0.7160               0.9786351
  4.002778           0.9014               0.9645151
  5.002778           1.0910               0.9468255
  6.002778           1.2791               0.9259965
  7.011111           1.4540               0.9030818
  8.005556           1.6061               0.8791892
  9.005556           1.7404               0.8548151
 10.008333           1.8512               0.8307911
 11.011111           1.9432               0.8073346
 12.013889           2.0269               0.7838782
 13.011111           2.1054               0.7603376
 14.008333           2.1731               0.7373807
 15.013889           2.2313               0.7152963
 16.011111           2.2778               0.6943147
 17.011111           2.3185               0.6739950
 18.019444           2.3548               0.6542167
 19.016667           2.3873               0.6349671
 20.013889           2.4161               0.6164710
 21.013889           2.4284               0.6002334
 22.013889           2.4397               0.5843378
 23.025000           2.4500               0.5689050
 24.019444           2.4595               0.5538797
 25.016667           2.4682               0.5391721
 26.019444           2.4725               0.5254765
 27.022222           2.4764               0.5120925
 28.019444           2.4801               0.4990494
 29.025000           2.4836               0.4863386
 30.025000           2.4868               0.4738510]

dataset({inputrates 'Time','ZeroRates','DiscountingFactor'},...
     'obsnames', {'O/N',' T/N','1W','1M','2M','3M','sep-14','dec-14','mar-15',...
     'jun-15','sep-15','dec-15','mar-16','jun-16','sep-16','dec-16',...
     'mar-17','jun-17','4Y','5Y','6Y','7Y','8Y','9Y','10Y','11Y','12Y','13Y',...
     '14Y','15Y','16Y','17Y','18Y','19Y','20Y','21Y','22Y','23Y',...
     '24Y','25Y','26Y','27Y','28Y','29Y','30Y'})
```

calculate the forward rates and plot

```
forwardrates = -100*log(inputrates(2:end,3)./inputrates(1:end-1,3))./...
(inputrates(2:end,1)-inputrates(1:end-1,1))
```

```
inputrates(2:end,4) = forwardrates
inputrates(1,4) = inputrates(1,2)
dataset({inputrates 'Time','ZeroRates','DiscountingFactor','forwardrates'},...
    'obsnames', {'O/N',' T/N','1W','1M','2M','3M','sep-14','dec-14','mar-15',...
    'jun-15','sep-15','dec-15','mar-16','jun-16','sep-16','dec-16','mar-17',...
     'jun-17','4Y','5Y','6Y','7Y','8Y','9Y','10Y','11Y','12Y','13Y','14Y',...
     '15Y','16Y','17Y','18Y','19Y','20Y','21Y','22Y','23Y','24Y','25Y',...
     '26Y','27Y','28Y','29Y','30Y'})

plot(inputrates(:,1), inputrates(:,2),inputrates(:,1), inputrates(:,3),...
inputrates(:,1), inputrates(:,4))
title('Zero Curve, Forward Curve and Discount Curve')
xlabel('Maturity')
xlim([0 30])
ylabel('Rates')
ylim([0 8])
legend('Zero Rates','Discount Rates','Forward Rates','location','northeast')
```

- defining inputs
- defining coefficients
- forming matrix A and B
- boundary condition
- Solve System
- plot price

2. Matlab code for zero-coupon bond under Hull White Model

- defining inputs
- defining coefficients
- forming matrix A and B
- boundary condition
- Solve System
- plot price

```
clear all
```

defining inputs

```
dr        = 0.05; % interest rate step
dt        = 0.02; % time step
rmax      = 0.5;  % maximum of interest rate
rmin      = 0;    % minimum of interest rate
r0        = 0.03; % assuming interest rate at current time
theta     = 0.025;% deterministic function, will be calibrated
a         = 0.05; % mean reversion parameter
sig       = 0.1;  % volatility of interest rate
T         = 1;             % time horizon
rvector   = rmin:dr:rmax;
bondprice = 1;
M         = T/dt;           % number of time intervals
N         = (rmax-rmin)/dr; % number of interest rate intervals


%%specifies the initial boundary conditions
% at Maturity
for j = 1:N+1
    f(j,1) = bondprice
end
% r= rmax
for i = 2 : M+1
    f(N,i) = 0
end
```

defining coefficients

```
for j = 1 : N
pu(j)  =  -0.25*dt*((sig/dr)*(sig/dr) +  theta/dr         -  a*(j-1))
pd(j)  =  -0.25*dt*((sig/dr)*(sig/dr) -  theta/dr         +  a*(j-1))
pm(j)  =  0.5*dt*(sig/dr)*(sig/dr)    +  0.5*(j-1)*dr*dt  +  1
pmm(j) =  pm(j )-2;
end
```

forming matrix A and B

```
%form matrix A; A*U(t+1) = B*U(t)
for j = 2 : N+1
A(j, j - 1) = pd(j - 1)
```

```
A(j, j) = pm(j - 1)
end;
for j = 2 : N
A(j, j + 1) = pu(j - 1)
end;

% form matrix B; A*U(t+1) = B*U(t)
for j = 2 : N+1
B(j, j - 1) = -pd(j - 1)
B(j, j) = -pmm(j -1)
end;
for j = 2 : N
B(j, j + 1) = -pu(j - 1)
end;
```

boundary condition

```
A(1, 1) = 1 + dt*rmin/2
A(1, 2) = -0.25*dt*(theta-a*rmin)/dr

B(1, 1) = -(1 + dt*rmin/2-2)
B(1, 2) = (-0.25*dt*(theta-a*rmin)/dr)
```


boundary condition

```
A(1, 1) = 1 + dt*rmin/2
A(1, 2) = -0.25*dt*(theta-a*rmin)/dr

B(1, 1) = -(1 + dt*rmin/2-2)
B(1, 2) = (-0.25*dt*(theta-a*rmin)/dr)
```


Solve System

```
[L, U] = lu(A);
for i = 1 : 1 : M+1
f(:, i + 1) = U\(L\(B*f(:, i)))
end;
zerobondprice = interp1(rvector, f(:,end),r0)
```

plot price

```
surf(0:dt:T,rvector,f(:,2:end))
surf(0:dt:T,rvector,f(:,2:end))
title('Zero Coupon Bond Price under Hull - White Model')
xlabel('Maturity T')
xlim([0 T])
ylabel('Interest Rate')
ylim([rmin rmax])
legend('alpha = 0.05 sigma = 0.1 Theta = 0.025','location','northwest')
```

3. Matlab code for pricing European swaption under uncalibrated Hull-white model

- defining inputs
- discounting factors
- specifies the Initial Boundary at Maturity
- Defining Coefficients
- Forming Matrix A and B
- Boundary Condition
- Solving System of Equations
- Interpolating The Price of The Swaption

```
clear all
```

defining inputs

```
dr        = 0.005;   % interest rate step
dt        = 0.02;    % time step
rmax      = 0.5;     % maximum interest rate
rmin      = 0;       % minimum interest rate
r0        = 0.02864; % assumed current interest rate
theta     = 0.025;   % deterministic function, will be calibrated
a         = 0.05;    % mean reversion parameter
sig       = 0.03;    % interest rate volatility
T         = 4;                 % Inception of swap
t         = [ 4 5 6 7 8 9 10] ; % tenor of the swap 6 years swap
K         = 3.18547 /100       % the strike of the swaption;
fswaprate = K;
rvec      = rmin:dr:rmax;
M         = T/dt;          % number of time intervals
N         = (rmax-rmin)/dr; % number of interest rate intervals
```

discounting factor

```
zerorates=...
    [1       0.012458;
     2       0.014123;
     3       0.016354;
     4       0.018689;
     5       0.020749;
     6       0.022442;
     7       0.023830;
     8       0.024962;
     9       0.025863;
    10       0.026581;
    11       0.027144;
    12       0.027713;
    13       0.028181;
    14       0.028584;
    15       0.028930;
    16       0.029176;
    17       0.029388;
    18       0.029576;
    19       0.029746;
    20       0.029893;]
discountfactors = exp(-zerorates(:,1).*zerorates(:,2))

annuity = sum(discountfactors(T+1:t(end)))
```

specifies the Initial Boundary at Maturity

```
f=zeros(N+1,M+1)
f(:,1) = max( K-rvec',0)*annuity
f(N+1,2:M+1) = 1
```

defining coefficients

```
for j = 1 : N
pu(j)  =  -0.25*dt*((sig/dr)*(sig/dr) +  theta/dr         -  a*(j-1))
pd(j)  =  -0.25*dt*((sig/dr)*(sig/dr) -  theta/dr         +  a*(j-1))
pm(j)  =  0.5*dt*(sig/dr)*(sig/dr)    +  0.5*(j-1)*dr*dt  +  1
pmm(j) =  pm(j )-2;
end
```

forming matrix A and B

```
%form matrix A; A*U(t+1) = B*U(t)
for j = 2 : N+1
A(j, j - 1) = pd(j - 1)
A(j, j) = pm(j - 1)
end;
for j = 2 : N
A(j, j + 1) = pu(j - 1)
end;

% form matrix B; A*U(t+1) = B*U(t)
for j = 2 : N+1
B(j, j - 1) = -pd(j - 1)
B(j, j) = -pmm(j -1)
end;
for j = 2 : N
B(j, j + 1) = -pu(j - 1)
end;
```

boundary condition

```
A(1, 1) = 1 + dt*rmin/2
A(1, 2) = -0.25*dt*(theta-a*rmin)/dr

B(1, 1) = -(1 + dt*rmin/2-2)
B(1, 2) = (-0.25*dt*(theta-a*rmin)/dr)
```

Solving System of Equations

```
[L, U] = lu(A);
for i = 1 : 1 : M+1
f(:, i + 1) = U\(L\(B*f(:, i)))
end;
zerobondprice = interp1(rvector, f(:,end),r0)
```

Interpolating The Price of The Swaption

```
swaptionprice = f(:,end)
swaptionprice = interp1(rvec,f(:,end),K)
mesh(0:dt:T,rvec,f(:,3:end))
```

```
title('recieveer Swaption Price under Hull - White Model')
xlabel('Maturity T')
xlim([0 T])
ylabel('Interest Rate r')
ylim([rmin rmax])
legend('alpha = 0.05 sigma = 0.1 Theta = 0.025','location','northwest')
```

4. Matlab code for Hull-White Model calibration

- xdata is the swaption_matrix which mainly used for specifying the ni, and nj,
- from model
- discount factors
- specifies the Initial Boundary at Maturity
- Defining Coefficients
- Forming Matrix A and B
- Boundary Condition
- Solving System of Equations
- Interpolating The Pricing of The Swaption

```
function F = calibration(x,swaption_matrix)
```

from model

```
  %swaptionprice_matrixmodel =...
   %[0.0092    0.0191    0.0290    0.0386    0.0474    0.0649    0.0881;
   %0.0114    0.0229    0.0342    0.0453    0.0562    0.0757    0.1039;
   %0.0116    0.0231    0.0341    0.0450    0.0547    0.0753    0.1018;
   %0.0120    0.0239    0.0350    0.0459    0.0567    0.0775    0.1045;
   %0.0117    0.0229    0.0335    0.0442    0.0545    0.0742    0.1011;
   %0.0111    0.0219    0.0323    0.0423    0.0520    0.0705    0.0960;
   %0.0102    0.0200    0.0295    0.0387    0.0476    0.0645    0.0878]


 % x=lsqcurvefit(@calibration1,[0.01 0.01],swaption_matrix,swaptionprice_matrix, [0.0

  %dt =0.02 swaption price matrix produced from calibration
  %swaptionprice_matrixmodel2 =...
   %0.0034    0.0078    0.0133    0.0194    0.0257    0.0038    0.0551
   %0.0041    0.0091    0.0147    0.0206    0.0265    0.0379    0.0538
```

```
    %0.0047    0.0101    0.0157    0.0213    0.0268    0.0374    0.0522
    %0.0052    0.0105    0.0159    0.0212    0.0264    0.0361    0.0501
    %0.0052    0.0103    0.0155    0.0205    0.0253    0.0346    0.0476
    %0.0048    0.0095    0.0141    0.0184    0.0228    0.0311    0.0423
    %0.0040    0.0080    0.0118    0.0155    0.0191    0.0257    0.0348

dr    = 0.0001;         % interest rate step
dt    = 0.02;           % time step
rmax  = 0.15;           % maximum interest rate
rmin  = 0;              % minumum interest rate
N     = (rmax-rmin)/dr; % number of interest rate intervals
M     = ni/dt           % number of time intervals
sig   = x(1);           % interst rate volatility
alpha = x(2);           % mean reversion parameter
theta = x(3)            % deterministic function
rvec  = rmin:dr:rmax;
% T = 4;% inception of swap
% t = [4  5  6  7  8  9 10 ] % tenor of the swap 6 years swap
 K = [1.61265  1.8469   2.07475  2.28107  2.44275;
      2.07998  2.30581  2.5043   2.65086  2.76404;
      2.53415  2.71882  2.84327  2.93659  3.00746;
      2.90349  2.99783  3.07037  3.12554  3.16202;
      3.09217  3.15346  3.19936  3.22638  3.24149]/100%
  % the strike of the swaption;
fswaprate = K
```

discount factors

```
zerorates=...
    [1       0.012458;
    2        0.014123;
    3        0.016354;
    4        0.018689;
    5        0.020749;
    6        0.022442;
    7        0.023830;
    8        0.024962;
    9        0.025863;
   10        0.026581;
```

```
   11        0.027144;
   12        0.027713;
   13        0.028181;
   14        0.028584;
   15        0.028930;
   16        0.029176;
   17        0.029388;
   18        0.029576;
   19        0.029746;
   20        0.029893;];
discountfactors = exp(-zerorates(:,1).*zerorates(:,2));

 tao = 1; % make the calibration process easier, I make the tao = 1
          % which means the swap payment occurs once a year
 swaptionprice1=zeros(size(swaption_matrix(:,1:end)))

    for ni=1:size(swaption_matrix,2)
        % = [1 2 3 4 ] column of swaptionprice_matirx option maturity(T)
    for nj=1:size(swaption_matrix,1)
        % = [1 2 3 4 ] row of swaptionprice_matrix  swap maturity (t)

      annuity(ni,nj) = sum(discountfactors(ni+1:ni+nj)) % annuity factor
```

specifies the Initial Boundary at Maturity

```
f = zeros(N+1,M+1)
f(:,1) = max(K(ni,nj) - rvec',0 )
f(N+1,2:M+1) =1

% r= max
for i = 2 : M+1
    f(N,i) = 0;
end
```

Defining Coefficients

```
for j = 1 : N
pu(j)  =  -0.25*dt*((sig/dr)*(sig/dr) +  theta/dr         -  alpha*(j-1));
pd(j)  =  -0.25*dt*((sig/dr)*(sig/dr) -  theta/dr         +  alpha*(j-1));
pm(j)  =  0.5*dt*(sig/dr)*(sig/dr)    +  0.5*(j-1)*dr*dt  +  1;
pmm(j) =  pm(j )-2;
```

```
end
```

Forming Matrix A and B

```
% form matrix A; A*U(t) = B*U(t+1)
for j = 2 : N+1
A(j, j - 1) = pd(j - 1);
A(j, j) = pm(j - 1);
end;
for j = 2 : N
A(j, j + 1) = pu(j - 1);
end;
% form matrix B; A*U(t) = B*U(t+1)
for j = 2 : N+1
B(j, j - 1) = -pd(j - 1);
B(j, j) = -pmm(j -1);
end;
for j = 2 : N
B(j, j + 1) = -pu(j - 1);
end;
```

Boundary Condition

```
A(1, 1) =1 + dt*rmin/2;
A(1, 2) =-0.25*dt*(theta-alpha*rmin)/dr;

B(1, 1) = -(1 + dt*rmin/2-2);
B(1, 2) =-0.25*dt*(theta-alpha*rmin)/dr;
```

Solving System of Equations

```
[L,U] = lu(A)
for i = 1 : 1 : M + 1
f(:, i + 1) = U\(L\(B*f(:, i)));
end;
f=[rvec',f];
```

Interpolating The Pricing of The Swaption

```
swaptionprice = f(:,end)
% swaptionprice = interp1(rvec,f(:,end),K)
```

```
swaptionprice = interp1(rvec,f(:,end),fswaprate(ni,nj));
% the rvec needs to be the forward swap rate which is calculated from the curve

swaptionprice1(ni,nj)= swaptionprice*annuity(ni,nj)

  end

 end
 F = swaptionprice1

end
```