Diplomarbeit

Implementation of Hull-White’s No-Arbitrage Term Structure Model

eingereicht von

Eugen V. Puschkarski

zur Erlangung des Grades

Magister rerum socialium oeconomicarumque

Sozial- und Wirtschaftswissenschaftliche Fakultät
der Universität Wien
Studienrichtung: Internationale Betriebswirtschaft

eingereicht bei

o.Univ.-Prof. Dr. Josef Zechner

Wien, im Mai 1998

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, daß die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Wien, am 20.5.1998

Eugen Puschkarski

Contents

Glossary of Notation

a:	Reversion rate of interest rate process
$B(\cdot)$:	Value of a money market account
c:	Price of European call option
C:	Coupon payment
d:	Proportional down movement in a binomial model
dr:	Discrete change of the short rate
dt:	Discrete change in time
$E(\cdot)$:	Expected value of a variable
f:	Price of an interest rate derivative
j_{max}:	Maximum number of up branches
j_{min}:	$-j_{max}$
k:	Tenor of reference rate
L:	Face value of a bond, swap...
m:	Instantaneous drift
N:	Number of time steps
$O(i,j)$:	Option value at time step i and state j
p:	Probability of an up movement in a binomial model
$P(t,\tau)$:	Value of a discount bond at time t with maturity τ face value 1
P_s:	State price, Arrow-debreu price in a binomial model
p_u:	Up-branching probability in Hull-White model
p_m:	Middle-branching probability in Hull-White model
p_d:	Down-branching probability in Hull-White model
Q:	State price in Hull-White tree
r:	Risk-free rate, short rate
$\ddot{r}$:	Average short rate
$R(t,\tau)$:	Spot rate at time t with maturity τ
RcX:	Strike rate of a derivative
s,σ:	Maturity of a bond, instantaneous standard deviation of the short rate
T:	Maturity of a derivative
u:	Proportional up movement in a binomial model
X:	Strike price of a derivative

z:	Wiener process or Call/Put indicator
Δ:	Delta of a derivative
Γ:	Gamma of a derivative
λ:	Market price of risk of r
τ:	Payout intervall

1 Introduction

As the number of interest rate derivatives has increased in a dramatic way over the last decade, models describing the way interest rates can evolve have become vital for pricing and hedging them. In the subsequent thesis we will show how to implement the Hull-White term structure model to price and calculate hedge parameters for different interest rate derivatives.

First we will present some preliminary definitions and show theoretical relations between them. Then a broad overview of term structure models is given discussing the individual advantages and disadvantages. This overview should help the reader in understanding the possibilities and limitations of the Hull-White model.

Then the tree building procedure for the numerical solution of the Hull-White model is presented and applied in several pricing procedures. Emphasize has been put on presenting how to price exotic derivatives and modified standard instruments.

Additionally the convergence behavior of the numerical solution is analyzed. Moreover it will be demonstrated how to calibrate the model to market data and how to calculate several hedge parameters.

Finally some concluding remarks summarize the situation at the time of writing.

2 Some preliminaries

In the subsequent analysis several specific terms will be necessary to use and will be presented in this section.[1]

In valuing contingent claims a method called *risk-neutral pricing* has become familiar and proves to be highly efficient by computational means. To illustrate the idea of risk-neutral pricing we will denote the value of a discount bond at time *t* with maturity τ and face value 1 by $P(t, \tau)$. We assume that the price of this bond can either move up by *u* to

$$P(t+1, \tau, u) = P(t, \tau)u \qquad (1a)$$

or down by *d* to

$$P(t+1, \tau, d) = P(t, \tau)d \qquad (1b)$$

Additionally we assume we can also invest in a money market account $B(t,\tau)$ which grows at a risk-free rate *r* to $B(t+1,\tau)$ with probability 1.

To rule out arbitrage we have to make sure that no combination of an investment in $B(t,\tau)$ and $P(t,\tau)$ dominates each other. This no-arbitrage requirement, which is at the heart of all valuation models, is in our small example equal to

$$u > r > d \qquad (3)$$

this inequality implies that there exists a unique, strictly positive number less than 1, which we will denote p, so that

$$r = pu + (1-p)d \qquad (4a)$$

or

$$p = \frac{r-d}{u-d} \qquad (4b)$$

[1] see Jarrow (1996), Chpt 6

As a result *p* can be interpreted as a probability measure which is also called *risk-neutral probability* since all investments offer the same expected return namely the risk-free rate *r* to an investor who is neutral to risk and faces the probability *p*.

In order to show the implications for risk-neutral pricing we will transform (4) to

$$P(t,\tau) = p\frac{P(t+1,\tau,u)}{r} + (1-p)\frac{P(t+1,\tau,d)}{r} \qquad (5)$$

by multiplying with *P(t,τ),* dividing by *r* and using (1).

From expression (5) follows that we can value the bond *P(t,τ)* simply by taking the expectation of the discounted future bond price or equivalently to discount the expected future value by the risk-free rate.

If we divide (5) by *B(t,τ),* realizing that *r=B(t+1,τ)/B(t,τ)* and taking the expectation under the risk-neutral probability *p* to get

$$\frac{P(t,\tau)}{B(t,\tau)} = E\left(\frac{P(t+1,\tau)}{B(t+1,\tau)}\right) \qquad (6)$$

we see that *P(t,τ)/B(t,τ)* follows a *martingale* because its expected future value equals its current value. That is why *p* is also called *equivalent martingale measure* or pseudo probability.

The valuation technique outlined above will be used in the following chapters and is only possible to perform if certain assumptions are made.

1. The market is frictionless in view of no taxes or transaction costs.
2. All securities are perfectly divisible.
3. The bond market is complete which means that there exists a discount bond for each maturity.

The third assumption is necessary to assure no-arbitrage and to be able to find a unique price for each contingent claim either by replicating that claim with a combination of other claims or by risk-neutral pricing.

To complete our discussion of risk-neutral pricing in a discrete time- discrete state framework it is interesting to put risk-neutral probabilities in another relation namely to *state prices* or Arrow-Debreu prices.

A state price is the current value of a security which pays 1 unit in a certain state and 0 in all other possible states, therefore it can be regarded as the most basic contingent claim. It is obvious that almost any contingent claim can be decomposed into several state prices.

If we consider (5) with $P(t,\tau)$ substituted by a state price $P_s(t,\tau)$ then $P_s(t+1,\tau,u)=1$ and $P_s(t+1,\tau,d)=0$ which gives

$$P_s(t,\tau) = \frac{p}{r} \qquad (7)$$

and shows that a state price is the discounted risk-neutral probability. A more rigorous proof is presented in Duffie (1996).

3 Overview of Interest Rate Models

Term Structure Models have been constructed in three different ways in the last decades. Either by assuming (modeling) a stochastic process for discount bond prices, instantaneous forward rates or the internal rate of return of a discount bond with infinitesimally short period of time to maturity called the *short rate r*.

$$r(t) = \lim_{\tau \to 0} R(t,\tau) \qquad (8)$$

where $R(t,\tau)$ is the spot rate at time t which applies to the period τ-t.

In the most general form a stochastic model for the short rate can be described by an Ito process

$$dr = m(r,t)dt + s(r,t)dz \qquad (9)$$

where *m* denotes the instantaneous drift, *s* the instantaneous standard deviation and *z* follows a Wiener process. By specifying different drifts and standard deviations several term structure models have emerged.

3.1 Equilibrium Term Structure Models

In the seventies and early eighties numerous papers have been introduced with the aim to build a model which can produce the term structure of interest rates as an output given some assumptions about how an overall economic equilibrium is achieved.

3.1.1 Rendleman and Bartter

One of the most basic models has been introduced by Rendleman and Bartter (1980) which builds on Cox, Ross and Rubinstein's binomial representation of a stock price following geometric Brownian motion. Similar to stocks, Rendleman and Bartter specify the process for the short rate in (9) with

$$dr = mrdt + srdz \qquad (10)$$

The only advantages of this model is that it does not allow negative interest rates and is easy to implement. On the other hand it has some serious drawbacks which will become apparent as we study its competitors.

3.1.2 Vasicek

In his 1977 paper Vasicek built the theoretical background for interest rate derivatives valuation. He assumes that the short rate follows a Markov process and that the bond price *P* at time *t* with maturity τ is determined by the short rate process in this time interval.

$$P(t,\tau) = E\left[e^{-\bar{r}(\tau - t)} \right] \qquad (11)$$

$\bar{r}$ is the average of the short rate in the time interval τ-*t*. Since the price of the bond is only dependent on one stochastic variable *r* these kind of models are called *one factor models*.

Now that *P* is only dependent on one variable we can proceed in the usual fashion in valuing derivative securities.[2]

By applying Itos Lemma, Vasicek derives the differential equation for any discount bond *P* as

$$\frac{\partial P}{\partial t} + r\frac{\partial P}{\partial r}(m + \lambda s) + \frac{1}{2}s^2 r^2 \frac{\partial^2 P}{\partial r^2} = rP \qquad (12)$$

where λ is called the *market price of risk* of *r* and is assumed to be constant and equal to

$$\lambda = \frac{\mu - r}{\sigma} \qquad (13)$$

with μ and σ as the expected return and volatility of *P*. Equation (12) can be solved subject to the boundary condition $P(\tau,\tau)$=1 like Black and Scholes (1973) did for stocks.

In a second step the term structure of interest rates is completely defined by

$$R(t,\tau) = -\frac{1}{\tau - t}\ln P(t,\tau) \tag{14}$$

if we regard *R(t,τ)* as the continuously compounded interest rate at time *t* for the time interval *τ-t*.

Additionally Vasicek proposed a certain specification for *m* and *s* in (9)

$$dr = a(b - r)dt + sdz \tag{15}$$

where *a*, *b* and *s* are constants. This specification for the short rate forces it to revert to its long-run mean *b* at rate *a*. This is called *mean reversion* and has become a popular feature in several term structure models because there is evidence that interest rates behave in a similar way in the real world.

By solving (12) with regard to (15) and subject to the boundary condition given above, Vasicek presents the price of a discount bond as[3]

$$\lambda = \frac{\mu - r}{\sigma} \tag{16}$$

where

$$B(t,T) = \frac{1 - e^{-a(T-t)}}{a} \tag{17}$$

and

$$A(t,T) = \exp\left[\frac{(B(t,T) - T + t)(a^2 b - s^2/2)}{a^2} - \frac{s^2 B(t,T)^2}{4a}\right] \tag{18}$$

---

[2]see Hull (1997) Chpt 13
[3]see Hull (1997) Chpt 17.4

which gives in conjunction with (14) the entire term structure. One drawback of this model is that the short rate is normally distributed so that negative interest rates can occur even though by mean reversion the probability for negative rates should be small.

3.1.3 Cox, Ingersoll and Ross

Cox, Ingersoll and Ross developed an intertemporal general equilibrium asset pricing model in assuming stochastic production functions in that model economy and other stochastic factors influencing the equilibrium interest rate.

In a second paper Cox, Ingersoll and Ross specify only one factor determining the interest rates namely the short rate which follows

$$dr = a(b-r)dt + s\sqrt{r}dz \qquad (19)$$

which is similar to Vasicek's specification with the exception that the volatility of *r* is proportional to $\sqrt{r}$ which prevents *r* to become negative as everytime *r* gets close to zero the decreasing volatility narrows the bounds *r* can fluctuate in. Additionally the model incorporates mean reversion.

Given the process in (19) Cox, Ingersoll and Ross derive closed form solutions for bond prices similar to (16) but with different *A* and *B* functions.

It is interesting to note that Cox, Ingersoll and Ross' approach differs from Vasicek's in the way that they start their analysis by giving a general valuation formula contingent on the driving factors which follow stochastic processes. One special case of this model is if we only choose the short rate as one factor. Vasicek on the other hand starts by modeling this short rate process and values bond as simple derivatives of the short rate.

3.1.4 Two Factor Models

To give equilibrium models more flexibility one factor interest rate models have been extended by introducing a second factor which determines the term structure. This factor allows the term structure not only to shift up and down but also twist.

3.1.4.1 Brennan and Schwartz

Brennan and Schwartz (1982) define the second factor to be the long rate *l*, which is the yield of a perpetual discount bond and which follows a stochastic process.

$$dr = (a_1 + b_1(l - r))dt + \sigma_1 r dz \qquad (20a)$$

$$dl = l(a_2 + b_2 r + c_2 l)dt + \sigma_2 l dw \qquad (20b)$$

where $dwdz = \rho dt$. σ_1 and σ_2 are the standard deviations of the short rate and long rate, respectively, and ρ is the correlation coefficient of the two rates.

3.1.4.2 Longstaff and Schwartz

Longstaff and Schwartz, on the other hand, use the general equilibrium framework of Cox, Ingersoll and Ross and extend it by assuming two independent unspecified state variables (factors) which follow stochastic processes of the form

$$dx = a_x(b_x - x)dt + \sigma_x \sqrt{x} dz_x \qquad (21a)$$

$$dy = a_y(b_y - y)dt + \sigma_y \sqrt{y} dz_y \qquad (21b)$$

Both factors are assumed to affect the mean of the instantaneous rate of return of the production process in the model economy, but only the second factor is assumed to affect the instantaneous variance. Therefore risk is only priced for the second factor.

Using the fundamental partial differential equation for interest rate contingent claims dependent on two factors developed by Cox, Ingersoll and Ross it is possible to determine the short rate *r* and its instantaneous variance *V* as part of the equilibrium

$$r = x + y \qquad (22a)$$

$$V = \sigma_x^2 x + \sigma_y^2 y \qquad (22b)$$

Now it is possible to change variables and express the valuation equation in terms of the new and observable variables *r* and *V*.

3.1.5 Three-Factor Models

In recent years researchers have come up with some yield-based term structure models which specify three factors driving the future term structure.[4] These factors are assumed to follow stochastic processes which can take on different forms. For example these factors can be

$$dr = \kappa(\theta - r)dt + \sqrt{V}\,dz \qquad (23a)$$

$$d\theta = \alpha(\beta - \theta)dt + \eta dw \qquad (23b)$$

$$dV = a(b - V)dt + \phi\sqrt{V}\,dy \qquad (23c)$$

where *r* is the short rate, θ denotes the long run mean of *r* and *V* is the variance of the short rate. This set of factors are designed to give the term structure evolution more flexibility in that it allows not only for parallel shifts like One-Factor Term Structure Models, but also for twists and not perfectly correlated bond prices. This advantage comes at the price of higher computational demands and theoretical sophistication.

3.2 No-Arbitrage Term Structure Models

---

[4]see Balduzzi et al. (1996), Chen (1995), Kraus (1993), Singh (1995)

Unlike equilibrium term structure models which take the different stochastic factors as an input and give the term structure (and implicitly bond prices) as an output, no-arbitrage term structure models take the initial term structure as an input by using time-varying parameters. This procedure of adjusting parameters so that the initial term structure is exactly matched is generally called *calibrating*.

3.2.1 Ho and Lee

Ho and Lee (1986) model the discrete evolution of bond prices by specifying perturbation functions. Since there is a relation between bond prices and interest rates it has been shown that the continuous time version of the Ho and Lee model can be represented by a model of the short rate[5]

$$dr = \theta(t)dt + \sigma dz \tag{24a}$$

with time varying mean of *dr*

$$\theta(t) = F_t(0,t) + \sigma^2 t \tag{24b}$$

where $F_t(0,t)$ denotes the partial derivative of the initial instantaneous forward rate with respect to t.

Because $r(t)$ is Markov which is equivalent to a constant volatility of forward rates[6], the short rate r can be represented by a recombining binomial tree so that the sequence of up and down movements is irrelevant. Additionally the Ho and Lee model incorporates no mean reversion but is still a major breakthrough in term structure modeling using all the information in the observable initial term structure by matching it exactly.

3.2.2 Hull and White

[5] see Hull (1990) and (1993a)
[6] see Heath, Jarrow, Morton (1990) and Hull, White (1993a)

In 1990 Hull and White developed a series of term structure models which can be regarded as extensions of such established models like Ho and Lee or Vasicek proposed. It is inasmuch an extension of Ho and Lee as it allows mean reversion and of Vasicek as it is a no-arbitrage model. The version which will be treated in this paper is

$$dr = \left(\theta(t) - ar\right)dt + \sigma dz \qquad (25)$$

where $\theta(t)$ is chosen to match the current term structure of interest rates. This model has the advantage over other similar models that it incorporates mean reversion and is analytically tractable. At any time *t* the short rate *r* reverts to $\theta(t)/a$ at rate *a*. It is easy to see that when *a=0* the Hull and White model reduces to the Ho and Lee model. Because of its analytic tractability θ(t) can be expressed by

$$\theta(t) = F_t(0,t) + aF(0,t) + \frac{\sigma^2}{2a}(1 - e^{-2at}) \qquad (26)$$

Hull and White show that bond prices can be calculated at any time *t* from

$$P(t,T) = A(t,T)e^{-B(t,T)r(t)} \qquad (27a)$$

where

$$B(t,T) = \frac{1 - e^{-a(T-t)}}{a} \qquad (27b)$$

and

$$\ln A(t,T) = \ln\frac{P(0,T)}{P(0,t)} - B(t,T)\frac{\partial \ln P(0,t)}{\partial t} - \frac{1}{4a^3}\sigma^2\left(e^{-aT} - e^{-at}\right)^2\left(e^{2at} - 1\right) \qquad (27c)$$

This has been a short introduction to the Hull-White model and in the subsequent paper its implementation and use in risk management will be presented.

3.2.3 Heath, Jarrow, and Morton

Heath, Jarrow, and Morton built a term structure model of a general type by modeling the instantaneous forward rate. They showed in their 1992 paper that there exists a link between the drift and standard deviation of this instantaneous forward rate, so that it is sufficient to know the standard deviations to construct a term structure model. Their general result can be summarized by[7]

$$dF(t,T) = m(t,T)dt + \sum_k s_k(t,T)dz_k \qquad (28)$$

allowing for *k* factors to influence the instantaneous forward rate *F(t,T)*. The link between drift *m(t,T)* and standard deviations $s_k(t,T)$ is given by

$$m(t,T) = \sum_k s_k(t,T)\int_t^T s_k(t,T)d\tau \qquad (29)$$

One specific version of (28) with two factors has become known as the Heath, Jarrow, and Morton model

$$dF(t,T) = m(t,T)dt + \sigma_1(t,T)F(t,T)dz_1 + \sigma_2(t,T)F(t,T)dz_2 \qquad (30)$$

which is a lognormal model and allows for parallel shifts and twists in the term structure. On the other hand equation (28) does generally not allow to be represented in a recombining tree as the forward rate and therefore the short rate are non-Markov. Only by restricting the volatility of bond prices not to be stochastic Markov models for the short rate can be constructed.

Since the Ho and Lee model assumes a bond price volatility of

$$v(t,T) = (T-t)\sigma \qquad (31)$$

[7] see Hull (1997), 426-427

and Hull and White of

$$v(t,T) = \frac{\sigma}{a}\left[1 - e^{-a(T-t)}\right] \qquad (32)$$

with σ as the standard deviation of the change in the short rate, these models can be constructed by a recombining tree. This is a strong advantage as it is computationally easier and faster to implement a recombining tree than a path-dependent tree which requires in most cases the use of Monte Carlo simulation.

3.2.4 Matching the volatility term structure of interest rates

With the no-arbitrage term structure models presented so far it is only possible to match the term structure of interest rates but not the term structure of interest rate volatilities which often can be observed in the market.

For this reason additionally to $\theta(t)$ in a process for the short rate like in (25) other time-varying parameters have been added to the short rate process in order to give it enough degrees of freedom to match also the term structure of volatilities exactly. Unfortunately introducing more time-dependent parameters has the drawback that the resulting future volatility term structure is often quite different from the initially observed one. At least two popular models which have the feature of matching the volatility term structure of interest rates are presented below.

3.2.4.1 Black, Derman, and Toy

In 1990 Black, Derman, and Toy introduced an interest rate model which can not only match the initial term structure of interest rates but also of interest rate volatilities. They use a binomial tree with constant local probabilities ($p_u=p_d=0.5$) and time steps to match the twofold term structure by a trial and error procedure. Obviously this procedure is computationally inefficient as Jamshidian (1991) points out. Therefore forward induction or the analytic approximation tree building procedure of Bjerksund and Stensland (1996) are to be preferred.

It can be shown that the continuous process of the short rate is

$$d\ln r = \left[\theta(t) + \frac{\frac{\partial\sigma(t)}{\partial t}}{\sigma(t)}\ln r\right]dt + \sigma(t)dz \qquad (33)$$

where the speed of mean reversion $\frac{\frac{\partial\sigma(t)}{\partial t}}{\sigma(t)}$ and the short rate volatility $\sigma(t)$ are time-dependent. Short rates are lognormally distributed.

3.2.4.2 Black and Karasinski

Black and Karasinski (1991) provide a more general model by making the speed of mean reversion independent of the volatility of interest rates but preserving its time-dependent feature. They present the implementation of their model by a similar procedure to Black, Derman, and Toy's with the difference that they use time steps of varying lengths. The continuous lognormal process for the short rate which they suggest is

$$d\ln r = \left[\theta(t) - \phi(t)\ln r\right]dt + \sigma(t)dz \qquad (35)$$

This process involves three different time-varying parameters giving the model enough degrees of freedom to match the given term structures. Also note that (35) is the lognormal Hull-White model of (25) with a time dependent *a* and σ.

4 A Tree-Building Procedure for Implementing One-Factor Term Structure Models

Although for the Hull-White model in equation (25) exist analytical solutions for discount bonds and plain-vanilla options, it is sometimes necessary to represent term structure models by numerical approximations. This is the case if we have to deal with path-dependency or more complex payoff functions like in exotic options.

Therefore we will present in this section a tree-building procedure which is computationally effective for one -factor term structure models. We will show explicitly how to use it for the Hull-White model. This procedure was first presented by Hull-White in 1994a and is a more efficient version of a similar procedure presented in 1993a.

The procedure starts by assuming that the short rate *r* changes over a time period Δt by $k\Delta r$ where *k* is a positive or negative integer. Hull and White choose to represent the evolution of *r* by a trinomial tree because it gives the model enough freedom to match the expected value and variance of Δr. By defining *k* to take values between -2 and 2 tree branching can be of any form shown in Figure 1

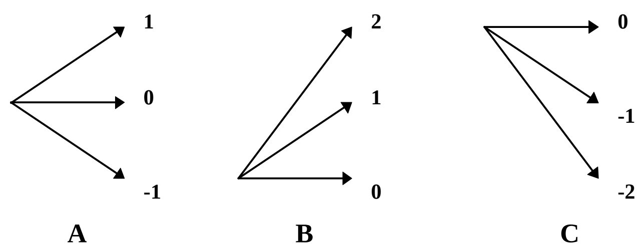

Figure 1

Branching of type B and C is added to standard branching of type A to facilitate implementing the mean reversion feature of the Hull-White model.

The procedure outlined here consists of two steps: First an auxiliary tree for the process

$$dr = -ardt + \sigma dz \qquad (34)$$

is constructed and in a second stage a certain α is calculated to match the given initial term structure.

The process in (34) determines *dr* to be normally distributed since *dz* is as usual a Wiener process. To calculate the probabilities of each branch in the tree we consider to match the expected change and variance in *r* over a time interval Δt. Additionally the probabilities must also sum to unity which gives us three equations in the three probabilities p_u, p_m and p_d where the subscript indicates the highest, middle and lowest branch of a node.

Since the time step Δt and the interest rate step Δr can be chosen discretionary it is now possible to construct a trinomial tree. However Hull-White suggest to choose Δr to be

$$\Delta r = \sqrt{3V} \qquad (35)$$

with *V* indicating the variance of (34) which equals

$$Var[dr] = V = \frac{\sigma^2\left(1 - e^{-2a\Delta t}\right)}{2a} \qquad (36)$$

in order to minimize errors in numerical procedures.

If we denote *Mr* to be the expected change in *r* which equals

$$E[dr] = Mr = \left(e^{-a\Delta t} - 1\right)r \qquad (36)$$

the branching probabilities at node *(i,j)* indicating the position in the time *(t=iΔt)*—state *(r=jΔr)* space for type A are given by

$$p_u = \frac{1}{6} + \frac{j^2 M^2 + jM}{2}$$
$$p_m = \frac{2}{3} - j^2 M^2$$
$$p_d = \frac{1}{6} + \frac{j^2 M^2 - jM}{2}$$

for type B by

$$p_u = \frac{1}{6} + \frac{j^2 M^2 - jM}{2}$$
$$p_m = -\frac{1}{3} - j^2 M^2 + 2jM$$
$$p_d = \frac{7}{6} + \frac{j^2 M^2 - 3jM}{2}$$

and for type C by

$$p_u = \frac{7}{6} + \frac{j^2 M^2 + 3jM}{2}$$
$$p_m = -\frac{1}{3} - j^2 M^2 - 2jM$$
$$p_d = \frac{1}{6} + \frac{j^2 M^2 + jM}{2}$$

Additionally if a>0 it is necessary to choose a maximum and minimum level for j which we will denote j_{max} and j_{min} to prevent the probabilities to become negative. It can be shown that j_{max} has to be chosen to be an integer between -0.184/M and -0.816/M, and j_{min} respectively to be an integer between 0.184/M and 0.816/M. Hull and White recommend to set j_{max} equal to the smallest integer greater than -0.184/M and j_{min} equal to -j_{max}. Figure 2 depicts the notation in this trinomial tree and its evolution for six time steps and j_{max}=3.

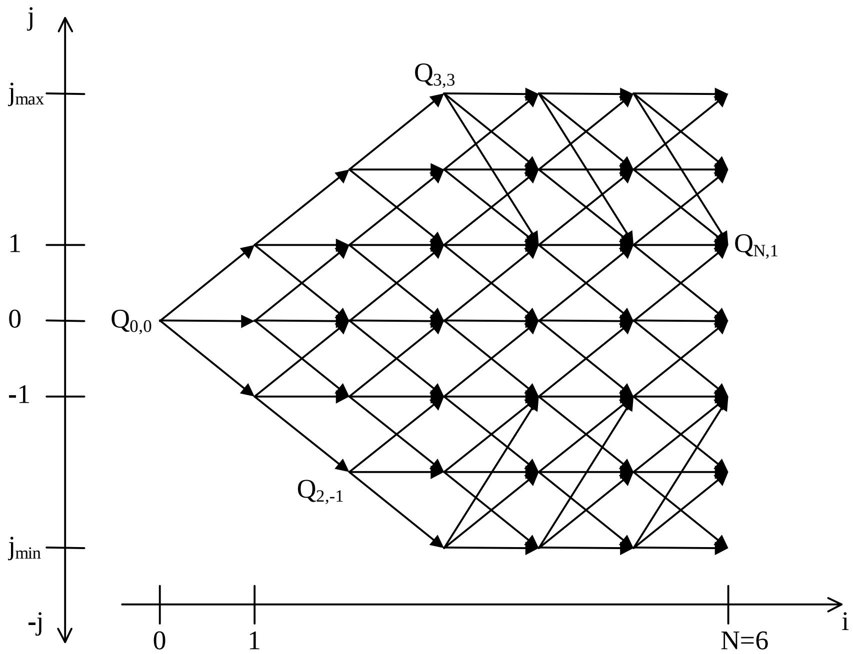

Figure 2

Now that the initial tree is built, it is necessary to displace the nodes at time $i\Delta t$ by α_i which is calculated to produce bond prices consistent with the initial term structure.

$$\alpha(t) = r(t) - r_{init}(t) \tag{37}$$

r_{init} stands for the interest rate which follows the process in (34) and which has been determined at each node using $r=j\Delta r$. Now it is possible for the Hull-White model to calculate $\alpha(t)$ analytically

$$\alpha(t) = F(0,t) + \frac{\sigma^2}{2a^2}\left(1 - e^{-at}\right)^2 \tag{38}$$

However as this analytic solution gives correct bond prices only in the limit as Δt approaches zero, an iterative procedure using *forward induction*[8] is necessary to match the initial term structure exactly.

To illustrate how forward induction works let's suppose $Q_{i,j}$ is a state price like introduced in a previous section. This means $Q_{i,j}$ is the present value of a security which pays 1 unit if node *(i,j)* is reached and zero in all other states. It follows that $Q_{0,0}$ is 1. We can calculate α_0 by applying (37) and noting that $r_{init}=j\Delta r$. Therefore α_0 is set equal to the initial Δt period rate. Now we have to use α_0 to calculate $Q_{1,1}$, $Q_{1,0}$, and $Q_{1,-1}$. Since we know the interest rate and the probabilities for the first period we simply apply risk neutral pricing in multiplying each state price with its probability and discount at the risk free rate.

$$\begin{aligned} Q_{1,1} &= p_u e^{-(\alpha_0+0\Delta r)\Delta t} \\ Q_{1,0} &= p_m e^{-(\alpha_0+0\Delta r)\Delta t} \\ Q_{1,-1} &= p_d e^{-(\alpha_0+0\Delta r)\Delta t} \end{aligned}$$

To determine α_1 we use the previously calculated Q's. The price of a discount bond maturing at time step 2 can be calculated from the initial term structure. It should be equal to $e^{-r(2)*2\Delta t}$. On the other hand the price of the bond at node (1,1) should be $e^{-(\alpha_1+1\Delta r)\Delta t}$, at node (1,0) it should be $e^{-(\alpha_1+0\Delta r)\Delta t}$, and at node (1,-1) it should be $e^{-(\alpha_1-1\Delta r)\Delta t}$. It follows that at the initial node (0,0) the price of the bond should be

$$P(0,2) = Q_{1,1}e^{-(\alpha_1+\Delta r)\Delta t} + Q_{1,0}e^{-(\alpha_1)\Delta t} + Q_{1,-1}e^{-(\alpha_1-\Delta r)\Delta t} = e^{-r(2)2\Delta t}$$

to avoid arbitrage. By rearranging it is possible to calculate α_1.

$$\alpha_1 = \left[\frac{\ln(Q_{1,1}e^{-\Delta r\Delta t} + Q_{1,0} + Q_{1,-1}e^{\Delta r\Delta t}) - \ln(e^{-r(2)2\Delta t})}{\Delta t} \right] \quad (39)$$

The next step is to calculate the Q's for the second time step. For example $Q_{2,1}$ can only be reached from node (1,1) with a probability p_m or from node (1,0) with a probability p_u. Since we know the interest rate which applies to the period one to two

[8]see Jamshidian(1991)

at node (1,1) to be equal to $(\alpha_1+1\Delta r)\Delta t$ and at node (1,0) to be at equal to $(\alpha_1+0\Delta r)\Delta t$ it follows that

$$Q_{2,1} = p_m e^{-(\alpha_1+\Delta r)\Delta t} Q_{1,1} + p_u e^{-(\alpha_1)\Delta t} Q_{1,0} \tag{40}$$

The other Q's for the second time step are calculated in a similar way. Then α_2 is calculated similar to (39) and the $Q_{3,j}$'s similar to (40) and so on.

Assuming we have calculated the $Q_{i,j}$'s for $i \le m$ ($m \ge 0$) α_m can be calculated more generally by

$$\alpha_m = \frac{\ln \sum_{j=-n_m}^{n_m} Q_{m,j} e^{-j\Delta r\Delta t} - \ln P_{m+1}}{\Delta t} \tag{41}$$

where n_m is the number of nodes on each side of the central node at time $m\Delta t$. Then the next $Q_{i,j}$'s can be determined by

$$Q_{m+1,j} = \sum_k Q_{m,k}\, p(k,j) e^{-(\alpha_m+k\Delta r)\Delta t} \tag{42}$$

where $p(k,j)$ denotes the probability of moving from node (m,k) to node $(m+1,j)$. The summation is taken over all values of k for which this is nonzero. This means that to determine the Arrow-Debreu prices of the next time step t_{m+1} we have to consider all of their Arrow-Debreu price predecessors at time t_m and the probabilities connecting them.

By using (41) and (42) consecutively it is possible to construct the whole trinomial tree of the short rate r divided in time intervals Δt up to an arbitrary point in time as long as discount bond prices exist for each time step.

It is important to note that since the variable *dr* is normally distributed, *r* can take on negative values although the probability should be small for that occurrence since mean-reversion tends to pull *r* up if it gets close to zero.

5 Applications

Now that we have set up the theoretical framework of the Hull-White model, we will show how to apply it to the valuation of several financial instruments. Later on we will show how it might be used in managing interest rate risks.

5.1 Discount bond options

The usual and most often used model in pricing discount bond options is Black's 76 model[9] which represents an extension of the famous Black-Scholes model. This model makes three assumptions:

1. that the price of the underlying variable[10] has a lognormal probability distribution at the expiration of the option
2. interest rates are non-stochastic and
3. the standard deviation of the natural logarithm of the underlying's price is the standard deviation of the futures/forward price of the underlying times the square root of the time to the option maturity

Given these assumptions it is possible to find closed form solutions for valuing discount bond options and coupon bond options[11].

However there are some pitfalls in using the Black 76 model. For example when it is used to price a cap the underlying forward rates are assumed to be lognormal and when it is used to price a swaption the swap rate is assumed to be lognormal. This

[9] see Black (1976)
[10] which can be an interest rate, a bond price or whatever
[11] analogous to options on stocks providing dividend payments

shows that Black 76 inherits theoretical inconsistencies because both the forward rate and the swap rate cannot be distributed lognormal simultaneously. On the other hand Black's 76 model fails in providing solutions to all kinds of American options and options with more exotic payout functions.

5.1.1 Analytical solution

Due to the analytic tractability of the Hull-White model there exists a closed form solution for discount bond options. If we denote *c* as the price of a call option at time *t=0* and maturity *T* on a bond with maturity *s* it is

$$c = P(0,s)N(h) - XP(0,T)N\left(h-\sigma_p\right) \qquad (43)$$

with

$$h = \frac{1}{\sigma_p}\ln\frac{LP(0,s)}{P(0,T)X} + \frac{\sigma_p}{2}$$

and

$$\sigma_p = \frac{\sigma}{a}\left[1-e^{-a(s-T)}\right]\sqrt{\frac{1-e^{-2aT}}{2a}}$$

The variable σ_p is the standard deviation of the logarithm of the bond price at time *T*.

5.1.2 Numerical solution

On the other hand it is also possible to use the numerical procedure described in section four to value options on discount bonds. First one calculates the evolution of the short rate up to option maturity, then one is able to apply equation (27a) which gives all possible bond prices at option maturity and consequently the terminal option values. In a second step *backward induction* is used iteratively to calculate the option values at all previous time steps up to time zero.

The backward induction procedure using risk-neutral pricing is diagrammatically illustrated in Figure 3 for one branch in a tree like shown in Figure 2:

$$O(0,0) = \frac{O(1,1)\,p_u + O(1,0)\,p_m + O(1,-1)\,p_d}{e^{rdt}}$$

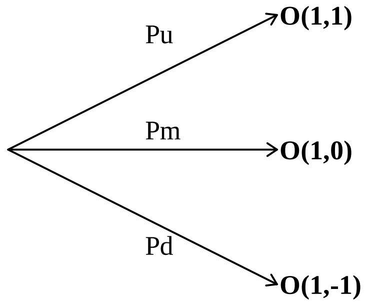

Figure 3

Please note that if the branching switches to non-standard branching we have to make the appropriate adaptations like shown in the computer code in the appendices.

On the other hand backward induction is not the only way to price European options on discount bonds. It is also possible to use the previously calculated state prices *Q(i,j)* because the present value *O(0,0)* of a cash flow occurring at time *t=i*dt* is simply the sum of each possible cash flow multiplied by the corresponding Arrow-Debreu price.

$$O(0,0) = \sum_{j=-Min(i,j\max)}^{Min(i,j\max)} Q(i,j)O(i,j) \qquad (44)$$

For example this would give

$$O(0,0) = Q(1,1)O(1,1) + Q(1,0)O(1,0) + Q(1,-1)O(1,-1)$$

if the cash flow occurred at time step one.

Example 1: To give an example we will calculate a European put bond option on a 9 year zero coupon bond and face value 100 with time to maturity of 3 years. We assume the two volatility parameters to be a=0.1 and σ=0.01. Additionally we face the following term structure

Maturity	Zero Rate
$\frac{days}{360}$	continuously compounded zero rates
0,008219178	0,0501772
0,083333333	0,0498284
0,166666667	0,0497234
0,25	0,0496157
0,5	0,0499058
1	0,0509389
2	0,0579733
3	0,0630595
4	0,0673464
5	0,0694816
6	0,0708807
7	0,0727527
8	0,0730852
9	0,073979
10	0,0749015

Table 1

The strike price is set to the three year forward price of the bond

$$X = 100e^{-r(9)9}e^{r(3)3} = 62{,}08706573 \approx 63$$

In Table 2 the numerical results for various numbers of time steps and the exact analytic value are presented using the computer code given in the appendices.

analytical	dt	Steps	numerical	difference
1,809283	0,3	10	**1,84983763183**	0,04055
	0,1	30	**1,81791386600**	0,00863
	0,06	50	**1,80606152739**	-0,00322
	0,03	100	**1,81297580628**	0,00369
	0,015	200	**1,80917049019**	-0,00011

Table 2

Please note that as the number of time steps increases and therefore the time interval between the steps gets smaller and smaller the numerical solution converges to the analytic one.

5.1.3 Convergence Analysis

If we decide to use a numerical solution it is crucial to know how fast it will converge to the exact value. Since it is possible to calculate bond options either analytically or numerically we will study the convergence behavior of the Hull-White model in the light of this instrument.

Table 3 and Figure 4 summarize the convergence behavior for different strike prices. Figure 5 gives a close-up of in-the-money options. The numbers in Table 3 are percentage deviations of the numerical solutions from the analytical ones. For example for a strike price of 60 and 50 time steps the numerical solution gives a 1.07% higher value than the analytical one.

		Stike						
dt	Steps	50,00	60,00	70,00	80,00	90,00	100,00	110,00
0,3	10	-10,02%	4,31%	0,01%	-0,01%	-0,01%	-0,01%	-0,01%
0,15	20	-9,23%	-1,61%	-0,04%	-0,01%	-0,01%	-0,01%	0,00%
0,1	30	13,80%	0,91%	-0,04%	-0,01%	0,00%	0,00%	0,00%
0,075	40	-1,00%	1,31%	0,01%	-0,01%	0,00%	0,00%	0,00%
0,06	50	-0,62%	1,07%	0,02%	0,00%	0,00%	0,00%	0,00%
0,04	75	5,43%	-0,19%	-0,01%	0,00%	0,00%	0,00%	0,00%
0,03	100	3,41%	0,32%	0,00%	0,00%	0,00%	0,00%	0,00%
0,024	125	3,34%	0,45%	0,01%	0,00%	0,00%	0,00%	0,00%
0,02	150	1,42%	0,24%	0,01%	0,00%	0,00%	0,00%	0,00%
0,015	200	0,98%	0,10%	0,00%	0,00%	0,00%	0,00%	0,00%

Table 3

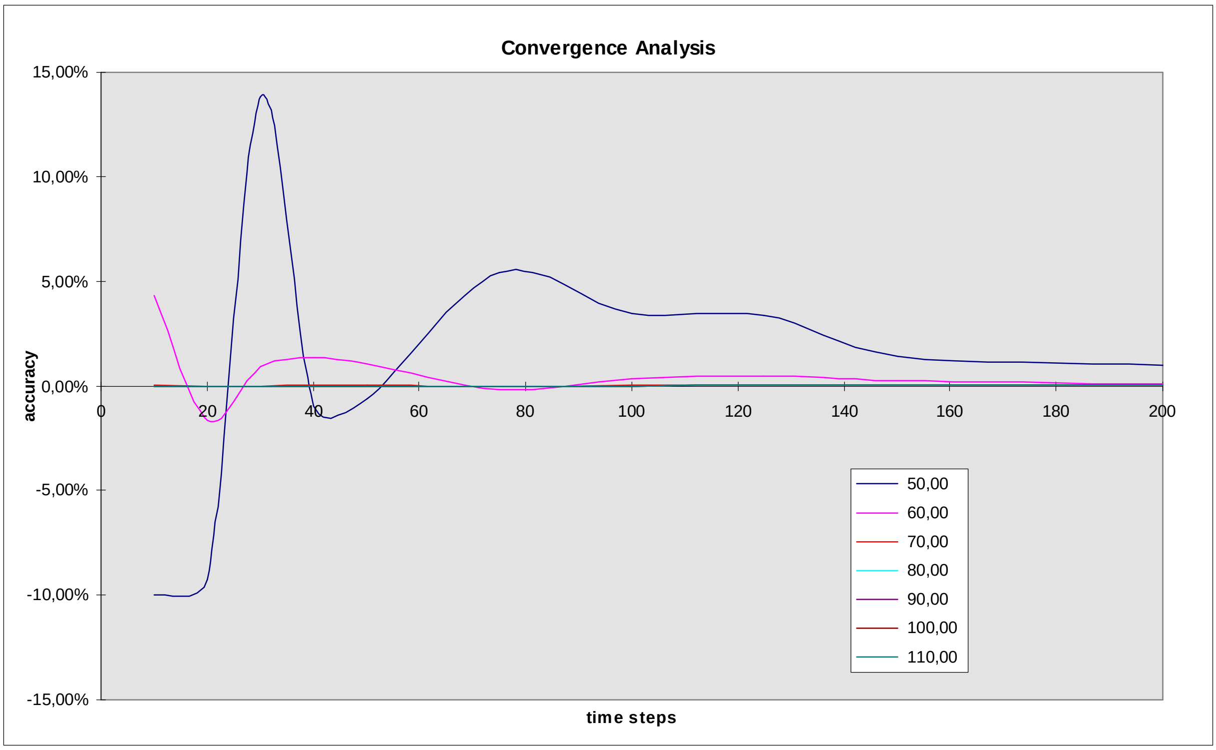

Figure 4

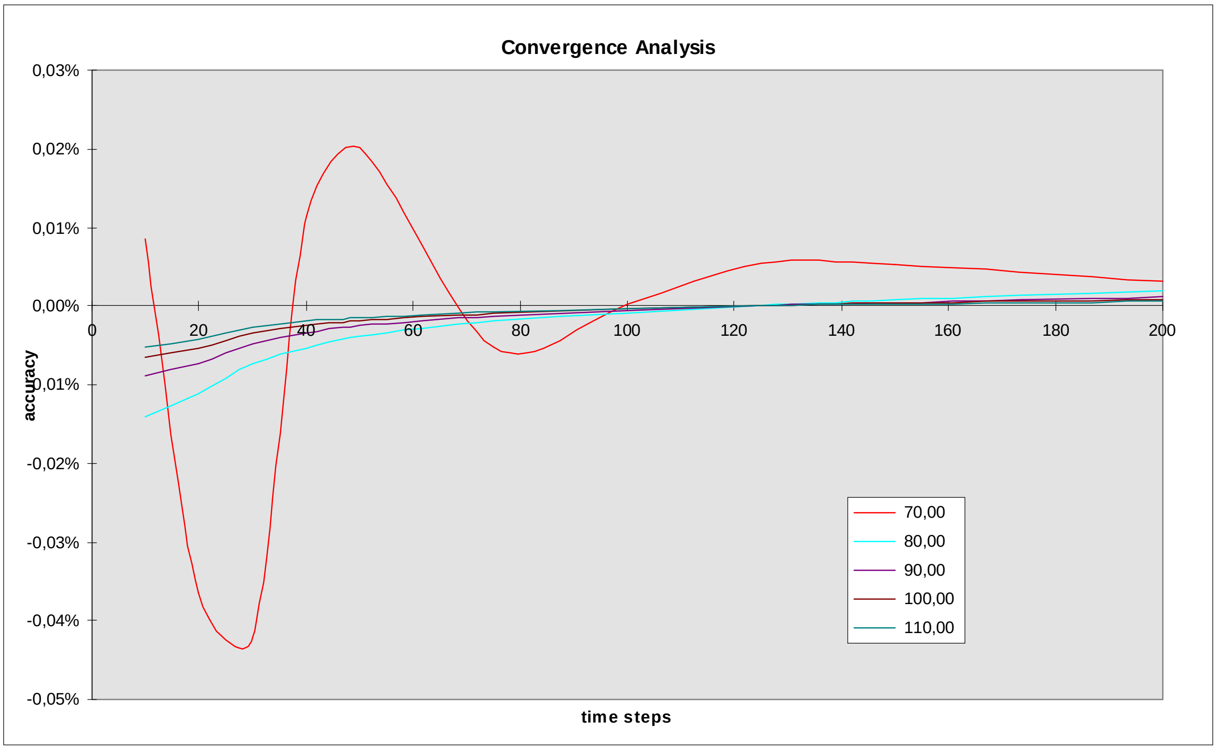

Figure 5

As we can see the numerical solutions tend to give more exact results for at-the-money and in-the-money options where only a few time steps are sufficient to produce highly accurate results.

On the other hand for out-of-the-money options a high number of time steps is necessary to provide reliable values. Therefore we recommend using at least 150 time steps for out-of-the-money options and about 50 for all other options.

5.1.4 Testing computation time

Moreover in deciding how many time steps we choose we implicitly make a decision of how much computation time we are willing to sacrifice.

Table 4 and Figure 6 show that the relation between number of time steps and computation time can best be approximated by a second order polynom of the form

$$y = 0{,}0085x^2 + 0{,}0469x$$

with a coefficient of determination of

$$R^2 = 0{,}9998$$

These results were achieved using a Intel Pentium II processor with 233 MHz and Visual Basic for Applications. Although the actual time for each number of time steps will vary depending on the computer and programming language used, the polynomial relation should remain fairly stable.

Steps	time in seconds
10	1,7
20	4,7
30	9,3
40	15,8
50	18,9
60	33,5
70	45,4
75	52,2
80	58,9
90	74
100	90,3
125	139,3
150	199,8
200	349,8

Table 4

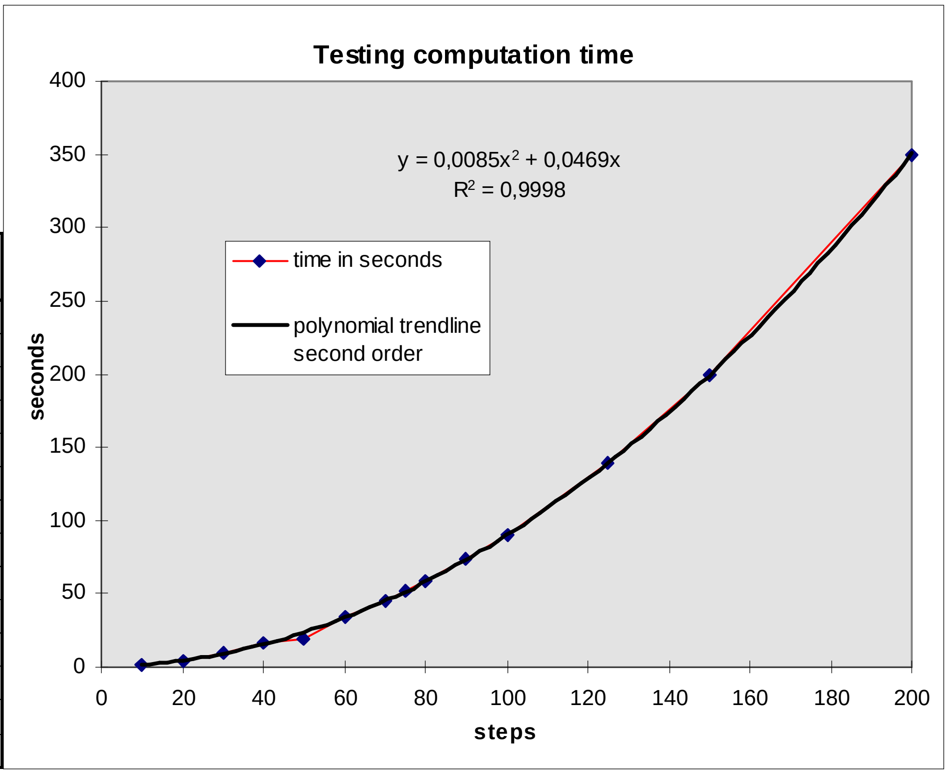

Figure 6

5.1.5 American style options

If we assume the discount bond option to be of American style we calculate the terminal payoff of the option in the usual way, but in the backward induction process we have to consider that early exercise may be optimal. This means that we work backward in the tree according to

$$O(i,j) = \max\left\{z\left(X - Pbond(i,j)\right), (p_u O(i+1,j+1) + p_m O(i+1,j) + p_d O(i+1,j-1))e^{-r(i,j)dt}\right\}$$

with *Pbond(i,j)* denoting the price of the discount bond at step *i* and state *j* and *z* being equal to 1 if the option is a put and -1 otherwise. Again this formula has to be adjusted in the right way if non-standard branching is used.

However it is straightforward that it will never be optimal to exercise an American call option on a discount bond early as the same arguments like for an American call option on a non-dividend-paying stock apply. On the other hand early exercise may be optimal for an American put option on a discount bond if the price of the discount bond is sufficiently low. This result can be tested with the adjoined files NUM.XLS which calculates the value of an European option and AMERICAN.XLS which calculates the American counterpart.

5.2 Coupon bond options

Jamshidian (1989) demonstrated that a European option on a coupon bond can be seen as a portfolio of discount bond options. This means that an option on a bond with N-coupon payments after option expiary is decomposed into N discount bond options.

To demonstrate the calculation of a coupon bond option we will consider a bond with the same face value and maturity as in Example 1 but with an annual coupon payment *C*. Additionally we face the same term structure, volatility parameters and option specifications.

To split the coupon bond option into discount bond options we have to find the short rate at option maturity for which the sum of the discount bonds equals the option strike price.

$$CP(3,4,\hat{r}) + CP(3,5,\hat{r}) + CP(3,6,\hat{r}) + CP(3,7,\hat{r}) + CP(3,8,\hat{r}) + (C+L)P(3,9,\hat{r}) = X \quad (45)$$

$P(x, y, \hat{r})$ stands for the price of a discount bond with face value 1 at time *x* and maturity *y* if we observe at time x a short rate $\hat{r}$. *C* denotes the coupon payment and *L* is the face value of the bond.

Equation (45) can be solved by trial and error or more efficiently with the Newton-Raphson method like in the computer code in Appendix C.

Table 5 gives the value of a European call option with the specifications and bond characteristics as presented above.

maturity of discount bond	4	5	6	7	8	9	Sum
Strike	4,296769888	3,725083711	3,249428	2,832704	2,503096	46,39292	**63**
Option value	0,263069995	0,449561394	0,578567	0,660218	0,714757	15,55837	**18,2245**

As Jamshidian pointed out this decomposition only works in a one factor model like the Hull-White model where all rates are perfectly correlated to the short rate.

5.3 Floaters

With the Hull-White model it is also possible to calculate non-standard floating rate bonds where the interval between interest payments is not equal to the term of the floating rate.

For example lets consider a floater with semi-annual coupon payments depending on the 1-year LIBOR which are calculated at the beginning of the period and paid at the end.

If we start the valuation at initiation of the floater or at a reset date a regular floater- in our case a floater depending on the 6-month LIBOR- would be equal to 100.

Table 5 gives a comparison of a regular two year floater to a non-standard floater using the data provided in Example 1. The calculations for the floating part were done with the code provided in Appendix D with time step interval of 0.1 years.

Non-standard floater			
term of reference rate in years			
1			
	fixed cash flow calculation	floating cash flow calculation	Floater value at time 0
semi-anually compounded reference rate	0,025796568		
Cash flow at time 0,5	2,5797		
cash flow at time 0	**2,516083323**	**97,98783059**	**100,5039**
standard floater			
term of reference rate in years			
0,5			
	fixed cash flow calculation	floating cash flow calculation	Floater value at time 0
semi-annually compounded 0,5 year rate	0,025266829		
Cash flow at time 0,5	2,5267		
cash flow at time 0	**2,464414979**	**97,53560303**	**100,0000**

Table 5

5.4 Caps, Floors, Collars

A cap provides a payment everytime a specified floating rate like the 6 month LIBOR exceeds the agreed cap rate. This assures the borrower of a loan that he will never pay more than the cap rate. These individual payments are called caplets and as a sum make up a cap.

Usually the reference rate is observed at the beginning of the period starting at t=1 and eventually a payment is made at the end of the period. In order to use the Hull-White model which determines the reference rate at each point in time and state we have to calculate the cash flows at the time the reference rate is measured which is at the beginning of the period.

As a result the payment at the beginning of the period is calculated according to

$$L\max(\mathrm{z}(1-\mathrm{e}^{\tau\,(\mathrm{RcX}-\mathrm{Rck})}),0) \qquad (46)$$

where *Rck* denotes the continuously compounded floating rate which applies to a period *k* (which is 6 months for our example of the 6 month LIBOR), *RcX* is the continuously compounded strike rate (cap rate) and τ is the period between payments.

Then standard backward induction or state prices are used to calculate the value of each caplet at *t*=0 and the sum is taken.

On the other hand it is also possible to think of equation (46) as being the cash flow function of a put option with maturity at the beginning of the considered period and strike price *L* on a discount bond which matures at the end of the period and face value $Le^{\tau RcX}$. This approach shows that a cap can be calculated as a portfolio of put options on discount bonds where analytical results exist.

On the other hand if the term of the reference rate *k* does not match the payment period τ the cash flow function would look like

$$L\max(z(e^{\tau(\mathrm{Rck}-\mathrm{Rc}\tau)} - e^{\tau(\mathrm{RcX}-\mathrm{Rc}\tau)}),0) \qquad (47)$$

which can only be calculated numerically.

To evaluate a floor *z* takes on -1 and either the numerical procedure or the replicating approach of call options on discount bonds is used. A collar is simply a combination of a long position in a cap and a short position in a floor and is calculated accordingly.

Table 6 gives an example of a regular cap on the 6 month rate with cap rate *RcX*=0.06 calculated analytically and numerically.

	analytical	numerical	Steps
0,5 year put option on 1 year bond principal 100*EXP(0.5*RcX) strike 100	0,018705496	**0,68955233**	50
1 year put option on 1,5 year bond principal 100*EXP(0,5*RcX) strike 100	0,213626832		
1,5 year put option on 2 year bond principal 100*EXP(0,5*RcX) strike 100	0,456915135		
	0,689247464		

Table 6

5.5 Swaptions

Options on interest swaps or swaptions as they are also called are another popular instrument to hedge against unfavorable interest rate movements. They give the holder the possibility to enter into a specified interest rate swap at some future time. As this is only a right but no obligation it must have some value.

To value a swaption we make use of the fact that a swap can be regarded as the agreement to exchange a fixed rate bond for a floating-rate bond. As the value of the floating rate bond always equals the principal amount at initiation of the swap regardless of the specific floating rate it is based on, we can say that a swaption is an option to exchange the fixed rate bond for the principal amount of the swap. This means that for an option on a payer swap where we pay fixed and receive floating the swaption can be replicated by a put option on the fixed rate bond with strike price equal to the principal. A receiver swaption is valued similar but with a call option.[12]

Since we can price an option on a coupon bond analytically we can price a swaption analytically.

<u>Example 2</u>: Lets consider a T=3 year option on a swap starting in 3 years and lasting 6 years with semiannual payments τ=0.5. The term structure and volatility parameters are given in Example 1 and the continuously compounded strike rate is *RcX*=0.06. This means that we have to replicate the swaption by an option on a coupon bond with 12 coupon payments of $C = L\left(e^{0.06*0.5} - 1\right)$ and principal *L*.

On the other hand it is also possible to use the numerical procedure and calculate all possible cash flows at option expiration T=3 with

$$O(N, j) = L * Max(z(e^{\tau * Rck(N, j)} - e^{\tau * RcX}), 0) * e^{-RcD(N, j) * (Tdata(f) - T)}$$

[12]see Hull (1997), 401-404

where *Tdata(f)* denotes the time coupon *f* is paid and *Rck(N,j)* is the swap rate at time *T* and state *j* which has to be calculated from

$$L * \sum_{f=1}^{12} e^{Rck*\tau - RcD(f)*(Tdata(f)-T)} = L$$

where *RcD(f)* is the discount rate for a period of *Tdata(f)-T* by using a Newton-Raphson iterative procedure. Appendix E provides the corresponding computer code.

Table 7 shows the results for Example 2 and gives a comparison of the analytical and numerical solution.

	Payer swaption		**Receiver swaption**	
	analytical		analytical	
	7,869372368		**0,086616308**	
steps	numerical	accuracy	numerical	accuracy
6	7,977719764	1,377%	0,083285331	-3,846%
30	7,871174392	0,023%	0,089692215	3,551%
100	7,870296469	0,012%	0,08766498	1,211%
300	7,870076051	0,009%	0,086591542	-0,029%
500	7,870945288	0,020%		

Table 7

Again we can notice that for out-of-the-money options about 150 steps and all other options 50 steps give accurate results.

5.6 Accrual Swaps

The numerical procedure presented above can also be used to price accrual swaps. These are swaps where the interest on one side accrues only if the floating rate is in a certain range or above or below a rate.

Like Hull (1997) points out an accrual swap can be replicated by an ordinary swap and a series of binary options. For every day *f* and state *j* of the swap we have to price a binary option which provides a payoff of

$$O(f,j)=\left(\frac{L*\tau}{248}*\left(e^{(RcX*\tau/248)}-1\right)\right)*e^{-RD(f*dt,s(f)/248)*(s(f)-f))/248}$$

at the following swap payment date *s(f)* if for example the floating reference rate is below the strike rate *RcX*. *RD* is the discount rate for the period between calculating the cash flow at time *f*dt* and the next swap payment date *s(f)* and τ is the number of days between swap payments assuming the year has 248 business days.

To give an example lets consider the fixed rate accrues only if the 3-month LIBOR is below *RcX*=0.08 cont. comp. and the swap pays every three month for 9 months time. The term structure and volatility parameters are again assumed to be the same as in example 1. The valuation result which is the sum of all binary options for 9 months is presented in Table 8 and the computer code in Appendix F.

number of business days	248
dt (one day in years)	0,004032258
days in payment period (tau)	62
maturity of swap in years	0.75
RcX	0.08
Value of binary options	**0,334585449**

Table 8

5.7 Callable, Putable Bonds

Some bonds have embedded options which give either the holder or the issuer certain rights. A callable bond for example gives the issuer the right to call the bonds at certain times or at any time at a prespecified price. With Hull and White's numerical procedure it is easy to price these embedded options.

As a simple example lets consider a discount bond with 9 years to maturity which gives the issuer the right to call the bonds at any time for *X*=75. This callable bond is priced by calculating the price of the bond at the terminal nodes which is 100 and working backward by

$$O(i,j) = \min\left\{X, (p_u O(i+1, j+1) + p_m O(i+1, j) + p_d O(i+1, j-1))e^{-r(i,j)dt}\right\} \qquad (48)$$

which is similar to the American style option valuation. Again this formula has to be adjusted in the right way if non-standard branching is used.

If we consider a putable bond where the holder has the right to demand early redemption we simply change *min* in (48) to *max*.

Table 9 shows the results assuming the term structure and volatility parameters of Example 1 and the discount bond mentioned above. The computer code is provided in Appendix G.

Steps	X	callable bond	bond without call option	worth of the option to the issuer
90	75	**38,53921831**	**51,3856621**	**12,84644379**
Steps	X	putable bond	bond without put option	worth of the option to the holder
90	50	**51,56011996**	**51,3856621**	**0,174457864**

Table 9

6 Volatility parameter estimation

One of the input data necessary for the Hull-White model are the two volatility parameters *a* and σ where the first one gives the relative volatility of the long and short rates and the second is the absolute volatility of the short rate. Unlike the initial term structure these volatility parameters are not directly provided by the market. Therefore they have to be inferred from market data of interest rate derivatives.

The procedure of calibrating the Hull-White model to market prices is done by choosing *a* and σ according to

$$Min \sum_i \left(P_i - V_i\right)^2 \qquad (49)$$

where P_i is the market price of the ith interest rate derivative and V_i is the corresponding model price[13]. Equation (49) minimizes the sum of the squared error (SSE) which can be done with the EXCEL Solver add-in for example.

As an example we will infer the volatility parameters from Caps and Floors on the DEM 6 month LIBOR rate with different strike rates and time to maturity. Caps and Floors are well suited for calibration because they are actively traded and their model prices can be calculated analytically.

The initial term structure for DEM as of 4/8/1998 was provided by REUTERS Ltd. and is given in Table 10.

Maturity (in Years)	Zero Rate
0,008219178	3,430%
0,083333333	3,530%
0,166666667	3,560%
0,25	3,599%
0,5	3,716%
1	3,918%
2	4,243%
3	4,472%
4	4,615%
5	4,755%
6	4,855%
7	4,958%
8	5,037%
9	5,117%
10	5,200%

Table 10

Cap and Floor prices were provided by the Risk Management Department of CREDITANSTALT BV AG and are shown in Table 11.

[13] see Hull (1997), 448

DEM CAPS 6M					DEM FLOOR 6M			
	5,50%	6,00%	6,50%			4,00%	4,50%	5,00%
2YR	**3,5**				2YR	**17**	**52,5**	**106,5**
3YR	**23,5**	**12,5**	**6**		3YR	**30,5**	**82,5**	**160,5**
4YR	**55**	**31,5**	**18**		4YR	**50**	**115**	**212**
5YR	**92**	**56,5**	**34,5**		5YR	**65**	**144**	**260**
7YR	**179**	**113**	**74**		7YR	**95,5**	**194,5**	**341**
10YR	**326**	**226**	**155**		10YR	**137**	**258,5**	**439**

Table 11

The prices in Table 11 are basis points of the principal which was assumed to be 10000.

Table 12 shows the SSE for the optimized *a* and σ of

sigma **0,011282417**

a **0,200527417**

				RcX			RcX			RcX
CAPS				cont			cont			cont
				5,426%			5,912%			6,397%
Cap/Floor	Years (s)	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2
p	2	3,5	**12,98453683**	89,95643895						
	3	23,5	**41,26976804**	315,7646563	12,5	**21,16306068**	75,0486204	6	**10,01684556**	16,13505
	4	55	**76,36963357**	456,6612391	31,5	**42,29200258**	116,46732	18	**21,83829282**	14,73249
	5	92	**122,1489707**	908,9604331	56,5	**71,799343**	234,069896	34,5	**39,70862066**	27,12973
	7	179	**222,2184576**	1867,835076	113	**138,9115134**	671,406526	74	**82,31760239**	69,18251
	10	326	**389,0650676**	3977,202753	226	**255,8765538**	892,608467	155	**160,4245053**	29,42526
				RcX			RcX			RcX
FLOORS				cont			cont			cont
				3,961%			4,450%			4,939%
Cap/Floor	Years (s)	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2	Marktpreis (Pi)	Model price (Vi)	(Pi-Vi)^2
c	2	17	**28,43573434**	130,7760199	52,5	**58,92812127**	41,320743	106,5	**103,4621577**	9,228486
	3	30,5	**44,18014537**	187,1463772	82,5	**87,64545758**	26,4757337	160,5	**151,2959651**	84,71426
	4	50	**60,54771651**	111,2543235	115	**115,9541267**	0,91035774	212	**196,7394927**	232,8831
	5	65	**73,54063378**	72,94242542	144	**138,4193738**	31,1433889	260	**233,0008217**	728,9556
	7	95,5	**96,48119138**	0,962736525	194,5	**177,5600856**	286,9607	341	**295,6680192**	2054,988
	10	137	**120,0163966**	288,4427833	258,5	**218,2070776**	1623,5196	439	**361,7047205**	5974,56
									SSE	**21650**

Table 12

If the solver does not produce reasonable results another procedure for finding the SSE is recommended by the author. For example we might have a guess that $0<a<1$ and $0<\sigma<0.5$ so we calculate a 10*10 matrix of SSE values for 10 different equally spaced values of *a* and σ which is illustrated in Figure 7

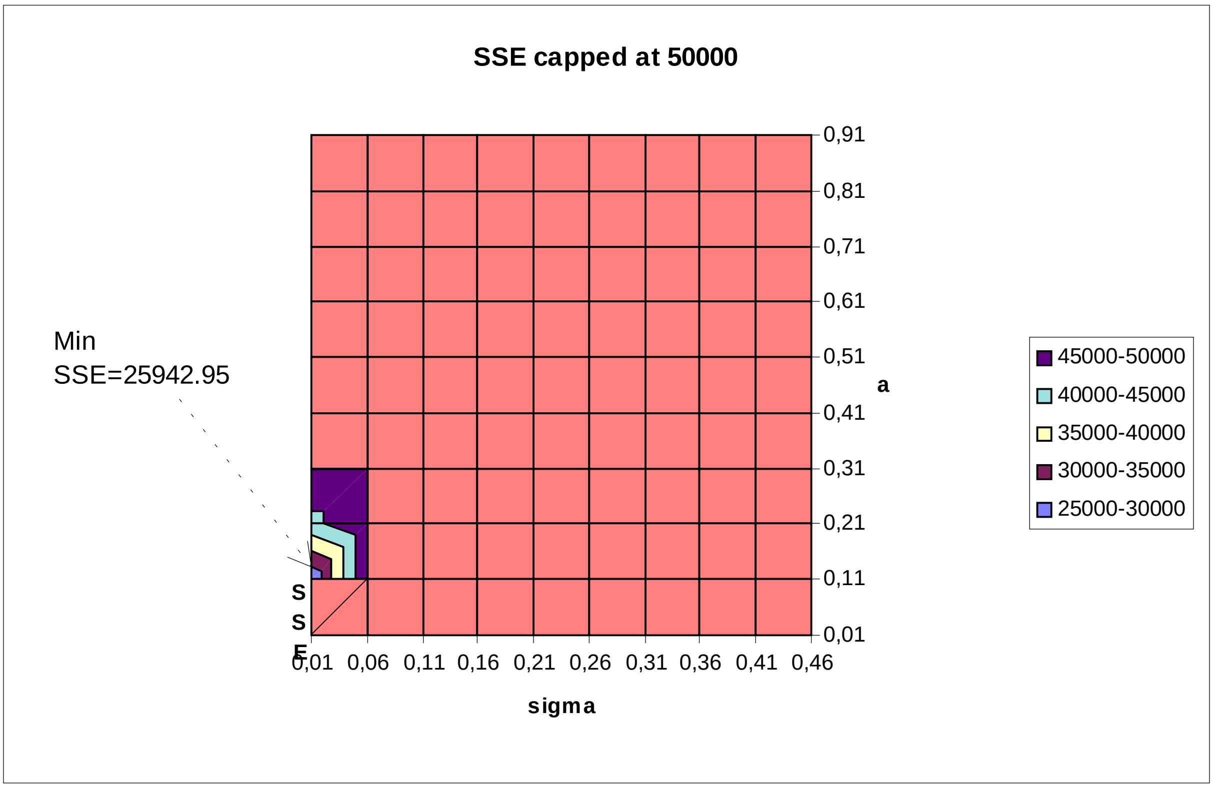

Figure 7

Figure 7 shows that the minimal SSE can be found if 0<*a*<0.3 and 0<σ<0.06 so we calculate a new 10*10 matrix within the new borders of *a* and σ. Figure 8 shows the new contour plane and Figure 9 a three dimensional representation.

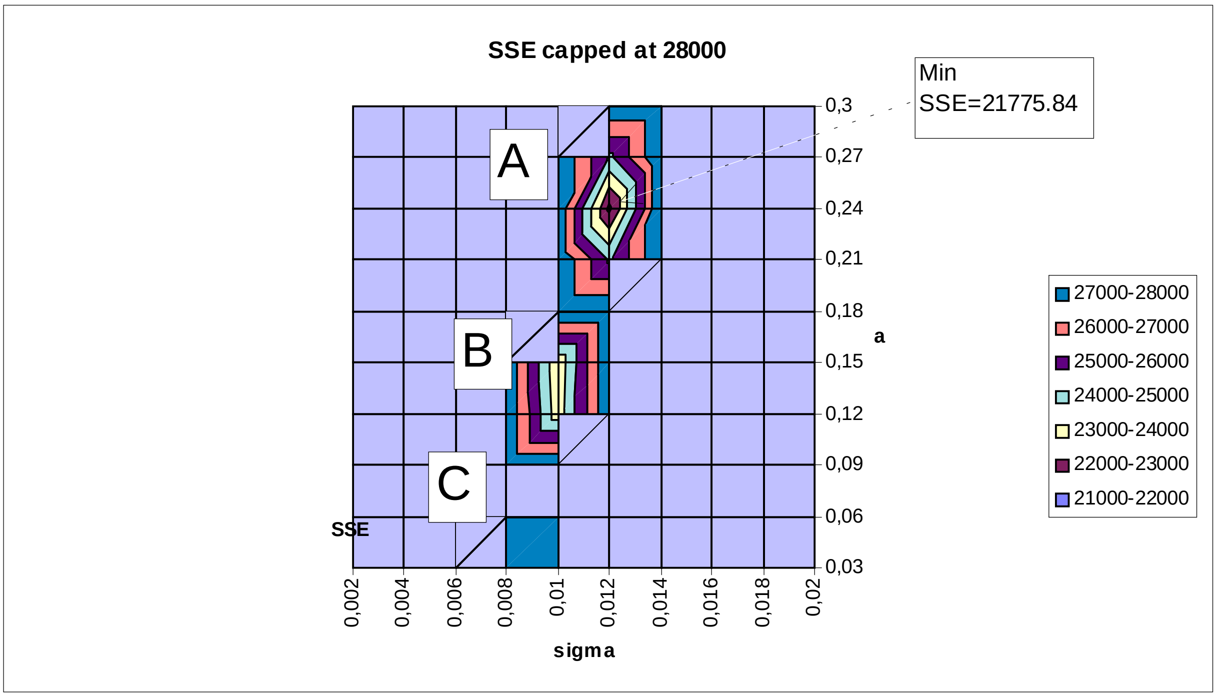

Figure 8

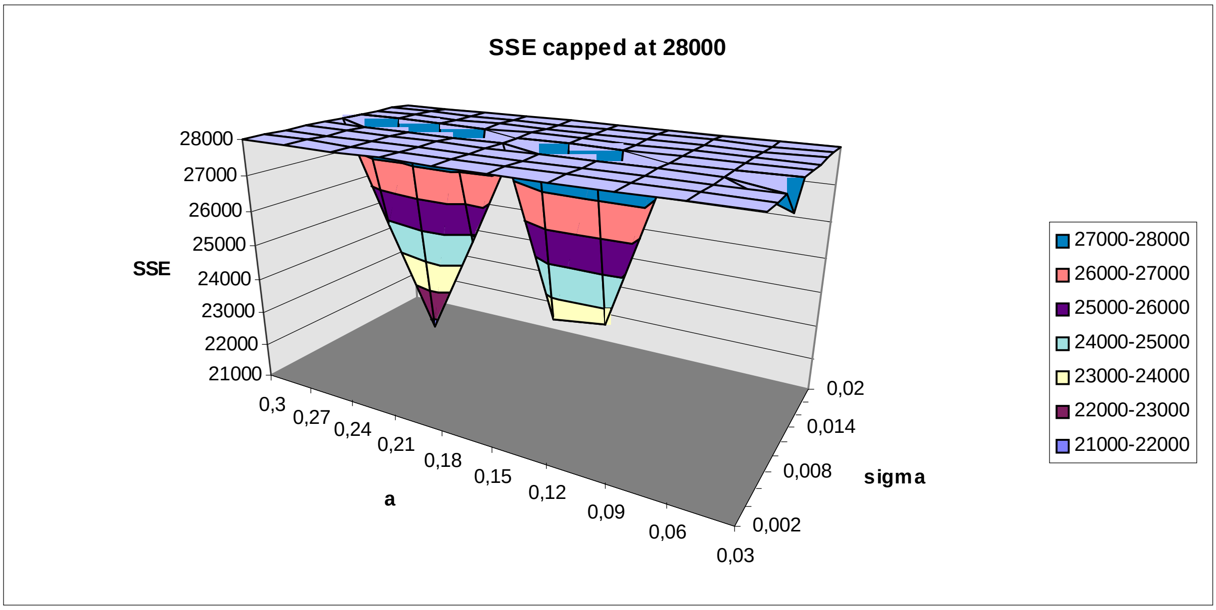

Figure 9

Now to drill down further area A, B and C in Figure 8 are investigated more closely and the close-ups of these areas are presented in Figure 10- 12.

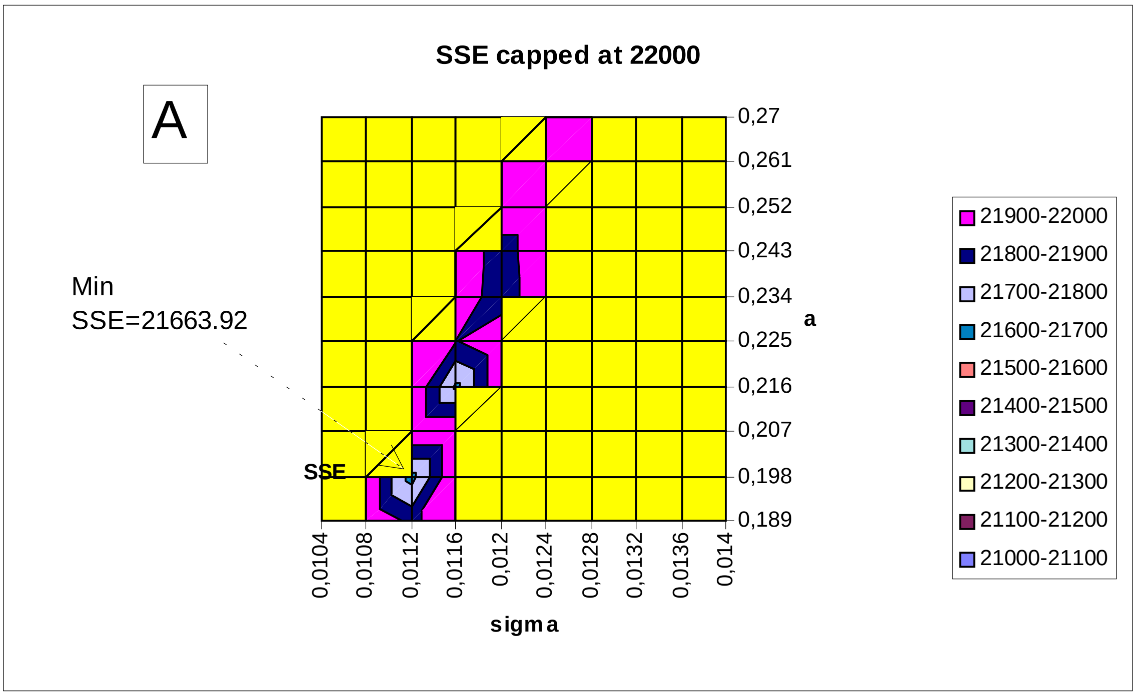

Figure 10

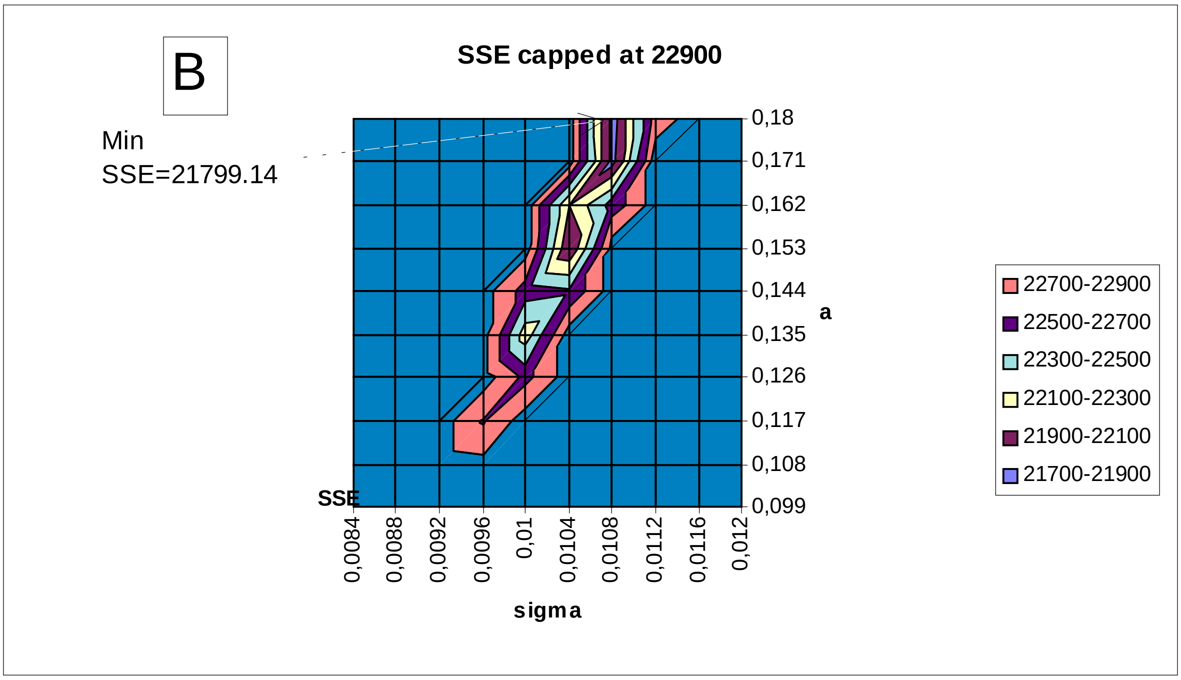

Figure 11

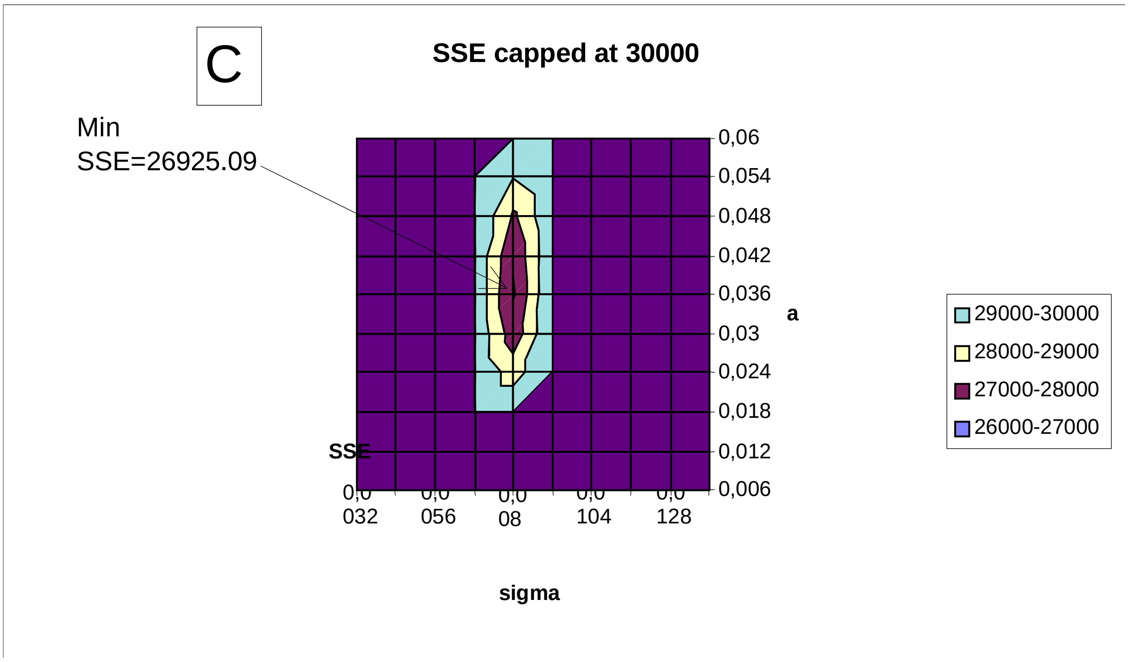

Figure 12

As a result the solution for the optimal *a* and σ which the solver provides seems to be the correct one since it lies in area A which shows the minimum SSE of all areas considered.

This procedure of focusing on small parts of areas iteratively is highly computational effective since the grid of *a* and *σ* becomes denser while avoiding an increasing number of nodes.

7 Hedging

In order to assess the risk of interest rate derivatives it is vital to know what impact shifts in the input data might have.

7.1 Parallel shifts

The probably most basic shift is a parallel shift of the entire term structure. To measure its impact on an interest rate derivative f_0 we add and subtract a constant *dr* to the interest rate term structure and recalculate the derivative's value f_1 *and* f_{-1}. The sensitivity or delta can then be calculated according to[14]

$$\Delta = \frac{\partial f}{\partial r} = \frac{f_1 - f_{-1}}{2dr} \tag{50}$$

To consider the convex relationship between the shift of the term structure *dr* and the derivative *f* we can also calculate the second derivative of *f* with respect to *dr* known as gamma

$$\Gamma = \frac{\partial^2 f}{\partial r^2} = \frac{f_1 + f_{-1} - 2f_0}{dr^2} \tag{51}$$

Table 13 presents the results for different shifts in the yield curve in Example 1 and the impact on zero coupon bond options with the same specifications as in Example 1. The calculated delta and gamma are

delta	170,9345
gamma	8613,441

and the new derivative values for the shifted yield curve can be approximated by a Taylor series expansion like

$$f = f_0 + df$$

$$df = \frac{\partial f}{\partial r}dr + \frac{1}{2}\frac{\partial^2 f}{\partial r^2}dr^2 = \Delta dr + \frac{1}{2}\Gamma dr^2$$

[14]see Hull (1997), 373

shift	-0,10%	-0,09%	-0,08%	-0,07%	-0,06%	-0,05%	-0,04%	-0,03%	-0,02%	-0,01%	0,00%	0,01%	0,02%	0,03%	0,04%	0,05%	0,06%	0,07%	0,08%	0,09%	0,10%
	0,0491772	0,0492772	0,0493772	0,0494772	0,0495772	0,0496772	0,0497772	0,0498772	0,0499772	0,0500772	**0,0501772**	0,0502772	0,0503772	0,0504772	0,0505772	0,0506772	0,0507772	0,0508772	0,0509772	0,0510772	0,0511772
	0,0488284	0,0489284	0,0490284	0,0491284	0,0492284	0,0493284	0,0494284	0,0495284	0,0496284	0,0497284	**0,0498284**	0,0499284	0,0500284	0,0501284	0,0502284	0,0503284	0,0504284	0,0505284	0,0506284	0,0507284	0,0508284
	0,0487234	0,0488234	0,0489234	0,0490234	0,0491234	0,0492234	0,0493234	0,0494234	0,0495234	0,0496234	**0,0497234**	0,0498234	0,0499234	0,0500234	0,0501234	0,0502234	0,0503234	0,0504234	0,0505234	0,0506234	0,0507234
	0,0486157	0,0487157	0,0488157	0,0489157	0,0490157	0,0491157	0,0492157	0,0493157	0,0494157	0,0495157	**0,0496157**	0,0497157	0,0498157	0,0499157	0,0500157	0,0501157	0,0502157	0,0503157	0,0504157	0,0505157	0,0506157
	0,0489058	0,0490058	0,0491058	0,0492058	0,0493058	0,0494058	0,0495058	0,0496058	0,0497058	0,0498058	**0,0499058**	0,0500058	0,0501058	0,0502058	0,0503058	0,0504058	0,0505058	0,0506058	0,0507058	0,0508058	0,0509058
	0,0499389	0,0500389	0,0501389	0,0502389	0,0503389	0,0504389	0,0505389	0,0506389	0,0507389	0,0508389	**0,0509389**	0,0510389	0,0511389	0,0512389	0,0513389	0,0514389	0,0515389	0,0516389	0,0517389	0,0518389	0,0519389
	0,0569733	0,0570733	0,0571733	0,0572733	0,0573733	0,0574733	0,0575733	0,0576733	0,0577733	0,0578733	**0,0579733**	0,0580733	0,0581733	0,0582733	0,0583733	0,0584733	0,0585733	0,0586733	0,0587733	0,0588733	0,0589733
	0,0620595	0,0621595	0,0622595	0,0623595	0,0624595	0,0625595	0,0626595	0,0627595	0,0628595	0,0629595	**0,0630595**	0,0631595	0,0632595	0,0633595	0,0634595	0,0635595	0,0636595	0,0637595	0,0638595	0,0639595	0,0640595
	0,0663464	0,0664464	0,0665464	0,0666464	0,0667464	0,0668464	0,0669464	0,0670464	0,0671464	0,0672464	**0,0673464**	0,0674464	0,0675464	0,0676464	0,0677464	0,0678464	0,0679464	0,0680464	0,0681464	0,0682464	0,0683464
	0,0684816	0,0685816	0,0686816	0,0687816	0,0688816	0,0689816	0,0690816	0,0691816	0,0692816	0,0693816	**0,0694816**	0,0695816	0,0696816	0,0697816	0,0698816	0,0699816	0,0700816	0,0701816	0,0702816	0,0703816	0,0704816
	0,0698807	0,0699807	0,0700807	0,0701807	0,0702807	0,0703807	0,0704807	0,0705807	0,0706807	0,0707807	**0,0708807**	0,0709807	0,0710807	0,0711807	0,0712807	0,0713807	0,0714807	0,0715807	0,0716807	0,0717807	0,0718807
	0,0717527	0,0718527	0,0719527	0,0720527	0,0721527	0,0722527	0,0723527	0,0724527	0,0725527	0,0726527	**0,0727527**	0,0728527	0,0729527	0,0730527	0,0731527	0,0732527	0,0733527	0,0734527	0,0735527	0,0736527	0,0737527
	0,0720852	0,0721852	0,0722852	0,0723852	0,0724852	0,0725852	0,0726852	0,0727852	0,0728852	0,0729852	**0,0730852**	0,0731852	0,0732852	0,0733852	0,0734852	0,0735852	0,0736852	0,0737852	0,0738852	0,0739852	0,0740852
	0,0729790	0,0730790	0,0731790	0,0732790	0,0733790	0,0734790	0,0735790	0,0736790	0,0737790	0,0738790	**0,0739790**	0,0740790	0,0741790	0,0742790	0,0743790	0,0744790	0,0745790	0,0746790	0,0747790	0,0748790	0,0749790
	0,0739015	0,0740015	0,0741015	0,0742015	0,0743015	0,0744015	0,0745015	0,0746015	0,0747015	0,0748015	**0,0749015**	0,0750015	0,0751015	0,0752015	0,0753015	0,0754015	0,0755015	0,0756015	0,0757015	0,0758015	0,0759015
analytical value	1,642715	1,6589741	1,6753225	1,6917598	1,7082857	1,7248999	1,7416021	1,7583918	1,7752688	1,7922327	**1,809283**	1,82641958	1,84364185	1,86094947	1,87834206	1,89581921	1,91338051	1,93102555	1,94875389	1,96656511	1,98445877
delta approx.	1,63834861	1,6554421	1,6725355	1,6896289	1,7067224	1,7238158	1,7409093	1,7580027	1,7750962	1,7921896	1,809283068	1,82637651	1,84346996	1,86056341	1,87765685	1,8947503	1,91184374	1,92893719	1,94603063	1,96312408	1,98021753
delta-gamma approx.	1,642655331	1,6589305	1,6752918	1,6917392	1,7082728	1,7248925	1,7415984	1,7583903	1,7752684	1,7922327	1,809283068	1,82641958	1,84364223	1,86095101	1,87834593	1,89582698	1,91339416	1,93104748	1,94878694	1,96661252	1,98452425
coefficient of determination																					
delta-gamma v. analytical	**100,0000%**																				
delta v. analytical	**99,9815%**																				

Table 13

Please note that with the improved accuracy of the delta and gamma approximation the coefficient of determination is 1.

7.2 Twists

Sometimes the yield curve twists which means that the different maturities do not shift by the same amount. For example the shift of the yield curve *dr* could be described by

$$dr = 0{,}02\% - 0{,}004\% * x$$

where *x* denotes the tenor of the zero rate. To measure the impact on the derivative due to this twist we simply recalculate the derivative and compare its value to the initial value.

One way to calculate delta is to divide the change in the value of the derivative by the tenor weighted sum of the individual shifts like demonstrated in Table 14.

			twist		
x	y		0,02%-0,004%*x	y new	weigthed shifts
0,008219	5,018%		0,020%	5,038%	0,00000003
0,083333	4,983%		0,020%	5,003%	0,00000029
0,166667	4,972%		0,019%	4,992%	0,00000058
0,25	4,962%		0,019%	4,981%	0,00000085
0,5	4,991%		0,018%	5,009%	0,00000161
1	5,094%		0,016%	5,110%	0,00000286
2	5,797%		0,012%	5,809%	0,00000429
3	6,306%		0,008%	6,314%	0,00000429
4	6,735%		0,004%	6,739%	0,00000286
5	6,948%		0,000%	6,948%	0,00000000
6	7,088%		-0,004%	7,084%	-0,00000429
7	7,275%		-0,008%	7,267%	-0,00001000
8	7,309%		-0,012%	7,297%	-0,00001714
9	7,398%		-0,016%	7,382%	-0,00002571
10	7,490%		-0,020%	7,470%	-0,00003571
56,00822	initial value		new value	**weighted Sum**	**-0,0000752**
	1,8092831		1,759858588		
		difference	**-0,04942448**		
				Delta	**657,1733795**

Table 14

7.3 Bucket shifts

On the other hand it might be interesting to know the individual exposure of the derivative to interest rates with specific tenors that we will call buckets. If we know the exposure to these buckets we can approximate any shift as a series of bucket shifts without having to recalculate the new derivative value. Table 15 shows how to decompose the parallel shift of one basis point into different bucket shifts. Since only the 3 and 9 year rates influence the value of the derivative only these buckets are considered.

delta of 3Y bucket	-93,4613364
delta of 9Y bucket	265,6289527
overall bucket delta	**172,1676**
parallel shift delta	**171,3651**

Table 15

Please note that the difference between the bucket delta and the parallel shift delta arises because the replication is not a perfect one.

7.4 Shift of volatility parameters

To hedge against a shift in the volatility parameters a and σ we have to calculate hedge parameters of first order similar to equation (49) which we will denote *a_vega* and *sigma_vega*.

If more accuracy in hedging against volatility shifts is required second order hedge parameters can be calculated similar to (50) which we will denote *a_vega2* and *sigma_vega2.*

Table 16 shows the corresponding calculations assuming a shift of *a* in 0.01 steps and of σ in 0.001 steps.

	a_vega	a_vega2	
	-5,540909488	**29,50893095**	
a shift	**option value**	**a_vega-a_vega2 approx.**	**a_vega approx**
-0,09	2,451118985	2,427476092	2,307964922
-0,08	2,363249759	2,346984406	2,252555827
-0,07	2,280102715	2,269443613	2,197146732
-0,06	2,201403566	2,194853713	2,141737637
-0,05	2,126895199	2,123214706	2,086328542
-0,04	2,056336553	2,054526592	2,030919448
-0,03	1,989501571	1,988789372	1,975510353
-0,02	1,926178227	1,926003044	1,920101258
-0,01	1,866167609	1,866167609	1,864692163
0	1,809283068	1,809283068	1,809283068
0,01	1,75534942	1,75534942	1,753873973
0,02	1,704202203	1,704366664	1,698464878
0,03	1,655686985	1,656334802	1,643055783
0,04	1,609658711	1,611253833	1,587646689
0,05	1,5659811	1,569123757	1,532237594
0,06	1,52452608	1,529944574	1,476828499
0,07	1,485173257	1,493716285	1,421419404
0,08	1,447809427	1,460438888	1,366010309
0,09	1,412328109	1,430112384	1,310601214
0,1	1,378629123	1,402736774	1,255192119
coefficient of determination		**0,999812476**	**0,98170**
	sigma_vega	**sigma_vega2**	
	136,6206311	**624,6067483**	
sigma shift	**option value**	**sigma_vega-sigma_vega2 approx.**	**sigma_vega approx**
-0,009	0,757532061	0,604993962	0,579697388
-0,008	0,805949857	0,736305435	0,716318019
-0,007	0,901403611	0,868241516	0,85293865
-0,006	1,016797396	1,000802203	0,989559282
-0,005	1,141411565	1,133987497	1,126179913
-0,004	1,270920026	1,267797398	1,262800544
-0,003	1,403309466	1,402231905	1,399421175
-0,002	1,537527865	1,537291019	1,536041806
-0,001	1,67297474	1,67297474	1,672662437
0	1,809283068	1,809283068	1,809283068
0,001	1,946216002	1,946216002	1,945903699
0,002	2,083613804	2,083773544	2,08252433
0,003	2,221364711	2,221955692	2,219144961
0,004	2,359388065	2,360762446	2,355765592
0,005	2,497624073	2,500193808	2,492386223
0,006	2,636027363	2,640249776	2,629006855
0,007	2,774562782	2,780930351	2,765627486
0,008	2,913202584	2,922235533	2,902248117
0,009	3,051924494	3,064165321	3,038868748
0,01	3,19071035	3,206719716	3,175489379
coefficient of determination		**0,998690607**	**0,99799**

Table 16

8 Concluding remarks

As it might become obvious now the Hull-White model is a very flexible model allowing the user to either use an analytical or numerical solution depending on the instrument he wants to price or hedge. This facilitates making consistent assumptions about how interest rates evolve. Additionally the mean reverting feature of the Hull-White model and its no-arbitrage property are often prerequisites of academics and finance professionals.

However at the time of writing no term structure model has a dominant position except the Black-76 model. Therefore most risk management and pricing tools give the user a whole set of models to choose from.

One of the drawbacks of the Hull-White model is that it is a one factor model whereas at least three factors driving interest rates have been identified.

As a lot of research is made in improving term structure models making them better match empirically observed characteristics of interest rate movements an exciting and challenging time awaits the interested reader.

9 Appendix A

In this appendix some functions are provided which have been used in the following computer codes.

9.1 Linear Interpolation

```
Function Interpol(xdata, ydata, xneu)

rowcount = xdata.Rows.Count

If xneu < xdata(1) Then
n1 = 1
ElseIf xneu > xdata(rowcount) Then
```

```
n1 = rowcount - 1
Else
n1 = Application.Match(xneu, xdata, 1)
End If

n2 = n1 + 1
x1 = xdata(n1)
y1 = ydata(n1)
x2 = xdata(n2)
y2 = ydata(n2)

Interpol = (y2 - y1) * ((xneu - x1) / (x2 - x1)) + y1

End Function
```

9.2 Price of a discount bond at time zero given the initial term structure

```
Function Price(xdata, ydata, rt)
Price = Exp(-Interpol(xdata, ydata, rt) * rt)
End Function
```

9.3 Function B() and A() as required by the Hull-White model

```
Function B(t1, t2, a)
B = (1 - Exp(-a * (t2 - t1))) / a
End Function

Function AA(Z1, Z2, sigma, a, xdata, ydata)

e = 0.0001
AA = Exp(Log(Price(xdata, ydata, Z2) / Price(xdata, ydata, Z1)) - B(Z1, Z2, a) * ((Log(Price(xdata,
ydata, Z1 + e)) - Log(Price(xdata, ydata, Z1 - e))) / (2 * e)) - 1 / (4 * a ^ 3) * sigma ^ 2 * (Exp(-a * Z2) -
Exp(-a * Z1)) ^ 2 * (Exp(2 * a * Z1) - 1))

End Function
```

9.4 Probabilities in a Hull-White interest rate tree

```
Function prob(X, j, jmax, a, dt)

m = Exp(-a * dt) - 1
If Abs(j) < jmax Then

Select Case X
Case 1
prob = 1 / 6 + (j ^ 2 * m ^ 2 + j * m) / 2
Case 0
prob = 2 / 3 - j ^ 2 * m ^ 2
Case -1
prob = 1 / 6 + (j ^ 2 * m ^ 2 - j * m) / 2
End Select

ElseIf j = -jmax Then

Select Case X
Case 1
prob = 1 / 6 + (j ^ 2 * m ^ 2 - j * m) / 2
Case 0
prob = -1 / 3 - j ^ 2 * m ^ 2 + 2 * j * m
Case -1
prob = 7 / 6 + (j ^ 2 * m ^ 2 - 3 * j * m) / 2
End Select

ElseIf j = jmax Then

Select Case X
Case 1
prob = 7 / 6 + (j ^ 2 * m ^ 2 + 3 * j * m) / 2
Case 0
prob = -1 / 3 - j ^ 2 * m ^ 2 - 2 * j * m
Case -1
prob = 1 / 6 + (j ^ 2 * m ^ 2 - j * m) / 2
End Select

End If
End Function
```

9.5 Transformation of a dt interest rate in its continuous counterpart

```
Function rcont(rdisc, dt, T, sigma, a, xdata, ydata)
rcont = (rdisc * dt + Log(AA(T, T + dt, sigma, a, xdata, ydata))) / B(T, T + dt, a)
End Function
```

9.6 Calculation of a zero rate given the short rate

```
Function rzero(r_t, tt, T, sigma, a, xdata, ydata)
If tt = T Then
rzero = 0
Else
rzero = -1 / (T - tt) * Log(AA(tt, T, sigma, a, xdata, ydata)) + 1 / (T - tt) * B(tt, T, a) * r_t
End If
End Function
```

10 Appendix B

10.1 Analytical valuation of a European discount bond option

```
Function HullAnalytic(typeflag, s, L, X, a, sigma, T, xdata, ydata)

sigmap = (sigma / a) * (1 - Exp(-a * (s - T))) * Sqr((1 - Exp(-2 * a * T)) / (2 * a))

h = (1 / sigmap) * Log((L * Price(xdata, ydata, s)) / (Price(xdata, ydata, T) * X)) + sigmap / 2

If typeflag = "c" Then

HullAnalytic = L * Price(xdata, ydata, s) * Application.NormSDist(h) - X * Price(xdata, ydata, T) *
Application.NormSDist(h - sigmap)

Else

HullAnalytic = -L * Price(xdata, ydata, s) * Application.NormSDist(-h) + X * Price(xdata, ydata, T) *
Application.NormSDist(-h + sigmap)

End If

End Function
```

10.2 Numerical valuation of discount bond options using state prices

```
Function HullWhiteOption(callputflag, dt, sigma, a, Tdata, s, L, X, xdata, ydata)
If callputflag = "p" Then
z = 1
Else
z = -1
End If
Trows = Tdata.Rows.Count

'dimensioning variables

Dim alpha() As Double, Q() As Double, O() As Double, r() As Double, rconti() As Double, Nsteps() As Integer
Dim N As Integer, j As Integer, jmax As Integer, jmin As Integer, jplus As Integer, jminus As Integer, ja As Integer, jb As Integer, Transition As Integer, f As Integer, m As Integer
If Tdata(Trows) / dt - Int(Tdata(Trows) / dt) = 0 Then
N = Tdata(Trows) / dt
Else
N = Int(Tdata(Trows) / dt) + 1
End If

'calculating constants

V = sigma ^ 2 * (1 - Exp(-2 * a * dt)) / (2 * a)
dr = Sqr(3 * V)
jmax = Int(0.184 / (a * dt)) + 1
jmin = -jmax
Transition = Application.Min(N, jmax)

'redimensioning variables

ReDim alpha(N)
ReDim Q(N + 1, -Transition To Transition)
ReDim O(N, -Transition To Transition)
ReDim r(N, -Transition To Transition)
ReDim rconti(N, -Transition To Transition)
ReDim Nsteps(Trows)
```

```
'calculating start values

alpha(0) = -Log(Price(xdata, ydata, dt)) / dt
r(0, 0) = alpha(0)
rconti(0, 0) = rcont(r(0, 0), dt, 0, sigma, a, xdata, ydata)
Q(1, 1) = prob(1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, 0) = prob(0, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, -1) = prob(-1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)

jminus = 0
jplus = 0

ja = -1
jb = 1

'using Q(m) to calculate alpha(m),r(m) and then Q(m+1) up to Transition-1 so we get the Q(Transition)
as last results
For m = 1 To Transition - 1


jminus = jminus - 1

jplus = jplus + 1
    summe = 0
    For j = jminus To jplus
    summe = summe + Q(m, j) * Exp(-j * dr * dt)
    Next j

    alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
    For j = jminus To jplus
    r(m, j) = alpha(m) + j * dr
    rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
    Next j

    ja = ja - 1
    jb = jb + 1

    For j = ja To jb

    If j = jb Then
    Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
```

```
ElseIf j = jb - 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j < jb - 1 And j > ja + 1 Then
Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja + 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja Then
Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
Next j
Next m

'using the Q(Transition) to calculate alpha(Transition) and so on up to N

For m = Transition To N

jplus = Transition
jminus = -Transition

summe = 0
For j = -Transition To Transition
summe = summe + Q(m, j) * Exp(-j * dr * dt)
Next j

alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
For j = jminus To jplus
r(m, j) = alpha(m) + j * dr
rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
Next j

For j = jminus To jplus

'considering the spezial case when nonstandard branching leads to central nodes with five incoming arrows

If jmax = 2 Then
Q(m + 1, 2) = Q(m, 2) * prob(1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt)
```

```
Q(m + 1, 1) = Q(m, 2) * prob(0, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(0, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt)
Q(m + 1, 0) = Q(m, 2) * prob(-1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(-1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(0, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -1) = Q(m, 0) * prob(-1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(0, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -2) = Q(m, -1) * prob(-1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(-1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Else
If j = jplus Then

Q(m + 1, j) = Q(m, j) * prob(1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 1 Then

Q(m + 1, j) = Q(m, j + 1) * prob(0, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 2) * prob(-1, j + 2, jmax, a, dt) * Exp(-r(m, j + 2) * dt)
ElseIf j < jplus - 2 And j > jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j - 2) * prob(1, j - 2, jmax, a, dt) * Exp(-r(m, j - 2) * dt)
ElseIf j = jminus + 1 Then

Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(0, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = jminus Then

Q(m + 1, j) = Q(m, j) * prob(-1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
```

```
    End If
    Next j

Next m

'cash flow calculations

For f = 1 To Trows

If f = 1 Then
tau = Tdata(f)
Else
tau = Tdata(f) - Tdata(f - 1)
End If

part1 = Tdata(f) / dt - Int(Tdata(f) / dt)
If part1 = 0 Then
Nsteps(f) = Tdata(f) / dt

    For j = -Application.Min(Nsteps(f), jmax) To Application.Min(Nsteps(f), jmax)
    O(0, 0) = Application.Max(z * (X - AA(Tdata(f), s, sigma, a, xdata, ydata) * Exp(-B(Tdata(f), s, a) *
rcont(r(Nsteps(f), j), dt, Tdata(f), sigma, a, xdata, ydata)) * L), 0) * Q(Nsteps(f), j) + O(0, 0)
    Next j
Else
Nsteps(f) = Int(Tdata(f) / dt)
    For j = -Application.Min(Nsteps(f), jmax) To Application.Min(Nsteps(f), jmax)
 summe1 = part1 * Application.Max(z * (X - AA(Nsteps(f) * dt, s, sigma, a, xdata, ydata) * Exp(-
B(Nsteps(f) * dt, s, a) * rcont(r(Nsteps(f), j), dt, Nsteps(f) * dt, sigma, a, xdata, ydata)) * L), 0) *
Q(Nsteps(f), j) + summe1
    Next j
    For j = -Application.Min(Nsteps(f) + 1, jmax) To Application.Min(Nsteps(f) + 1, jmax)
   summe2 = (1 - part1) * Application.Max(z * (X - AA((Nsteps(f) + 1) * dt, s, sigma, a, xdata, ydata) *
Exp(-B((Nsteps(f) + 1) * dt, s, a) * rcont(r(Nsteps(f) + 1, j), dt, (Nsteps(f) + 1) * dt, sigma, a, xdata,
ydata)) * L), 0) * Q(Nsteps(f) + 1, j) + summe2
    Next j
O(0, 0) = summe1 + summe2
End If
Next f

HullWhiteOption = O(0, 0)
```

```
End Function
```

11 Appendix C

```
Function f(r, a, sigma, xdata, ydata, X, T, s, N, C)
Dim Bond()
ReDim Bond(1 To N)
Cupontime = (s - T) / N
Sum = 0
For i = 1 To N
If i = N Then L = 100 Else L = 0
Bond(i) = (C + L) * AA(T, T + i * Cupontime, sigma, a, xdata, ydata) * Exp(-B(T, T + i * Cupontime, a) *
r)

Sum = Sum + Bond(i)
Next i
f = Sum - X
End Function
```

11.1 Newton-Raphson algorithm

```
Function Newton(r, a, sigma, xdata, ydata, X, T, s, N, C)

epsilon = 0.001
Accuracy = 0.0000001
Do
fStrich = (f(r + epsilon, a, sigma, xdata, ydata, X, T, s, N, C) - f(r - epsilon, a, sigma, xdata, ydata, X, T,
s, N, C)) / (2 * epsilon)
r = r - f(r, a, sigma, xdata, ydata, X, T, s, N, C) / fStrich
Fehler = Abs(f(r, a, sigma, xdata, ydata, X, T, s, N, C) - 0)
Loop Until Fehler <= Accuracy
Newton = r
End Function
```

11.2 Valuation of Coupon bond option

```
Function HullWhiteCouponOption(typeflag, a, sigma, xdata, ydata, X, T, s, N, C)

Dim Bond(), Opt()
```

```
ReDim Bond(1 To N), Opt(1 To N)

Cupontime = (s - T) / N
Sum = 0

For i = 1 To N
If i = N Then L = 100 Else L = 0
Bond(i) = (C + L) * AA(T, T + i * Cupontime, sigma, a, xdata, ydata) * Exp(-B(T, T + i * Cupontime, a) *
Newton(0, a, sigma, xdata, ydata, X, T, s, N, C))
Opt(i) = HullAnalytic(typeflag, T + i * Cupontime, C + L, Bond(i), a, sigma, T, xdata, ydata)
Sum = Sum + Opt(i)
Next i

HullWhiteCouponOption = Sum
End Function
```

12 Appendix D

12.1 Calculation of a non-standard floater

```
Function HullWhiteFloater(dt, sigma, a, Tdata, k, L, xdata, ydata)
' k is the term of the reference rate
Trows = Tdata.Rows.Count

'dimensioning variables
Dim alpha() As Double, Q() As Double, O() As Double, r() As Double, Nsteps() As Integer
Dim N As Integer, j As Integer, jmax As Integer, jmin As Integer, jplus As Integer, jminus As Integer, ja
As Integer, jb As Integer, Transition As Integer, f As Integer, m As Integer
If Tdata(Trows) / dt - Int(Tdata(Trows) / dt) = 0 Then
N = Tdata(Trows) / dt
Else
N = Int(Tdata(Trows) / dt) + 1
End If



'calculating constants
V = sigma ^ 2 * (1 - Exp(-2 * a * dt)) / (2 * a)
dr = Sqr(3 * V)
```

```
jmax = Int(0.184 / (a * dt)) + 1
jmin = -jmax
Transition = Application.Min(N, jmax)

'redimensioning variables
ReDim alpha(N)
ReDim Q(N + 1, jmin To jmax)
ReDim O(N, jmin To jmax)
ReDim r(N, jmin To jmax)
ReDim Nsteps(Trows)

'calculating start values
alpha(0) = -Log(Price(xdata, ydata, dt)) / dt
r(0, 0) = alpha(0)
Q(1, 1) = prob(1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, 0) = prob(0, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, -1) = prob(-1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)

jminus = 0
jplus = 0

ja = -1
jb = 1

'using Q(m) to calculate alpha(m),r(m) and then Q(m+1) up to Transition-1 so we get the Q(Transition)
as last results
For m = 1 To Transition - 1


jminus = jminus - 1

jplus = jplus + 1
   summe = 0
   For j = jminus To jplus
   summe = summe + Q(m, j) * Exp(-j * dr * dt)
   Next j

   alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
   For j = jminus To jplus
   r(m, j) = alpha(m) + j * dr
   Next j
```

```
ja = ja - 1
jb = jb + 1

For j = ja To jb

If j = jb Then
Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jb - 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j < jb - 1 And j > ja + 1 Then
Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja + 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja Then
Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
Next j
Next m

'using the Q(Transition) to calculate alpha(Transition) and so on up to N
For m = Transition To N

jplus = jmax
jminus = jmin

summe = 0
For j = jmin To jmax
summe = summe + Q(m, j) * Exp(-j * dr * dt)
Next j

alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
For j = jminus To jplus
r(m, j) = alpha(m) + j * dr
Next j

For j = jminus To jplus
```

```
'considering the spezial case when nonstandard branching leads to central nodes with five incoming arrows
If jmax = 2 Then
Q(m + 1, 2) = Q(m, 2) * prob(1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt)
Q(m + 1, 1) = Q(m, 2) * prob(0, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(0, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt)
Q(m + 1, 0) = Q(m, 2) * prob(-1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(-1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(0, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -1) = Q(m, 0) * prob(-1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(0, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -2) = Q(m, -1) * prob(-1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(-1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Else
If j = jplus Then

Q(m + 1, j) = Q(m, j) * prob(1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 1 Then

Q(m + 1, j) = Q(m, j + 1) * prob(0, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 2) * prob(-1, j + 2, jmax, a, dt) * Exp(-r(m, j + 2) * dt)
ElseIf j < jplus - 2 And j > jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j - 2) * prob(1, j - 2, jmax, a, dt) * Exp(-r(m, j - 2) * dt)
ElseIf j = jminus + 1 Then

Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(0, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
```

```
   ElseIf j = jminus Then

   Q(m + 1, j) = Q(m, j) * prob(-1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a,
dt) * Exp(-r(m, j + 1) * dt)
   End If
   End If
   Next j

Next m


'cash flow calculations
For f = 1 To Trows
If f = Trows Then LL = L
If f = 1 Then
tau = Tdata(f)
Else
tau = Tdata(f) - Tdata(f - 1)
End If

part1 = Tdata(f) / dt - Int(Tdata(f) / dt)
If part1 = 0 Then
Nsteps(f) = Tdata(f) / dt

   For j = -Application.Min(Nsteps(f), jmax) To Application.Min(Nsteps(f), jmax)
   O(Nsteps(f), j) = (LL + L * (Exp(tau * rzero(rcont(r(Nsteps(f), j), dt, Tdata(f), sigma, a, xdata, ydata),
Tdata(f), Tdata(f) + k, sigma, a, xdata, ydata)) - 1)) * Exp(-tau * rzero(rcont(r(Nsteps(f), j), dt, Tdata(f),
sigma, a, xdata, ydata), Tdata(f), Tdata(f) + tau, sigma, a, xdata, ydata)) + O(Nsteps(f), j)
   Next j
Else
Nsteps(f) = Int(Tdata(f) / dt)
   For j = -Application.Min(Nsteps(f), jmax) To Application.Min(Nsteps(f), jmax)
   O(Nsteps(f), j) = part1 * L * tau * rzero(rcont(r(Nsteps(f), j), dt, Nsteps(f) * dt, sigma, a, xdata, ydata),
Nsteps(f) * dt, Nsteps(f) * dt + k, sigma, a, xdata, ydata) + O(Nsteps(f), j)
   Next j
   For j = -Application.Min(Nsteps(f) + 1, jmax) To Application.Min(Nsteps(f) + 1, jmax)
   O(Nsteps(f) + 1, j) = (1 - part1) * L * tau * rzero(rcont(r(Nsteps(f) + 1, j), dt, (Nsteps(f) + 1) * dt,
sigma, a, xdata, ydata), (Nsteps(f) + 1) * dt, (Nsteps(f) + 1) * dt + k, sigma, a, xdata, ydata) +
O(Nsteps(f) + 1, j)
   Next j
```

```
End If
Next f

'backward induction
For i = N To Transition + 1 Step -1

    For j = jmin To jmax

    If j = jmax Then

    O(i - 1, j) = (O(i, j) * prob(1, j, jmax, a, dt) + O(i, j - 1) * prob(0, j, jmax, a, dt) + O(i, j - 2) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j < jmax And j > jmin Then

    O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j = jmin Then

    O(i - 1, j) = (O(i, j) * prob(-1, j, jmax, a, dt) + O(i, j + 1) * prob(0, j, jmax, a, dt) + O(i, j + 2) * prob(1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)
    End If
    Next j
Next i

jminus = -Transition
jplus = Transition

For i = Transition To 1 Step -1
jminus = jminus + 1
jplus = jplus - 1
    For j = jminus To jplus

     O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

     Next j
Next i
```

```
HullWhiteFloater = O(0, 0)

End Function
```

13 Appendix E

```
Function fu(rs, rdis, a, dt, sigma, xdata, ydata, T, Tdata)

Cupontime = Tdata.Rows.Count
tau = Tdata(2) - Tdata(1)

Sum = 0
For i = 1 To Cupontime
If i = Cupontime Then zz = 0 Else zz = 1
Sum = Sum + 100 * (Exp(rs * tau) - zz) * Exp(-rzero(rcont(rdis, dt, T, sigma, a, xdata, ydata), T,
Tdata(i), sigma, a, xdata, ydata) * (Tdata(i) - T))

Next i
fu = Sum - 100
End Function
```

13.1 Newton Raphson Algorithm for Swaption

```
Function Newton2(rs, rdis, a, dt, sigma, xdata, ydata, T, Tdata)

epsilon = 0.01
Accuracy = 0.000001
Do
fStrich = (fu(rs + epsilon, rdis, a, dt, sigma, xdata, ydata, T, Tdata) - fu(rs - epsilon, rdis, a, dt, sigma,
xdata, ydata, T, Tdata)) / (2 * epsilon)
rs = rs - fu(rs, rdis, a, dt, sigma, xdata, ydata, T, Tdata) / fStrich
Fehler = Abs(fu(rs, rdis, a, dt, sigma, xdata, ydata, T, Tdata) - 0)
Loop Until Fehler <= Accuracy
Newton2 = rs
End Function
```

13.2 Numerical calculation of a Swaption

```
Function HullWhiteSwaption(PayerReceiverFlag, dt, sigma, a, Tdata, T, L, RcX, xdata, ydata)
```

```
Trows = Tdata.Rows.Count
If PayerReceiverFlag = "p" Then
z = 1
Else
z = -1
End If

'dimensioning variables
Dim alpha() As Double, Q() As Double, O() As Double, r() As Double, Nsteps() As Integer, Rck() As
Double, RcD() As Double
Dim N As Integer, j As Integer, jmax As Integer, jmin As Integer, jplus As Integer, jminus As Integer, ja
As Integer, jb As Integer, Transition As Integer, f As Integer, m As Integer

N = T / dt

'calculating constants
V = sigma ^ 2 * (1 - Exp(-2 * a * dt)) / (2 * a)
dr = Sqr(3 * V)
jmax = Int(0.184 / (a * dt)) + 1
jmin = -jmax
Transition = Application.Min(N, jmax)

'redimensioning variables
ReDim alpha(N)
ReDim Q(N + 1, jmin To jmax)
ReDim O(N, jmin To jmax)
ReDim r(N, jmin To jmax)
ReDim Rck(N, jmin To jmax)
ReDim RcD(N, jmin To jmax)
ReDim Nsteps(Trows)

'calculating start values
alpha(0) = -Log(Price(xdata, ydata, dt)) / dt
r(0, 0) = alpha(0)
Q(1, 1) = prob(1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, 0) = prob(0, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, -1) = prob(-1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)

jminus = 0
jplus = 0
```

```
ja = -1
jb = 1

'using Q(m) to calculate alpha(m),r(m) and then Q(m+1) up to Transition-1 so we get the Q(Transition)
as last results
For m = 1 To Transition - 1


jminus = jminus - 1

jplus = jplus + 1
    summe = 0
    For j = jminus To jplus
    summe = summe + Q(m, j) * Exp(-j * dr * dt)
    Next j

    alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
    For j = jminus To jplus
    r(m, j) = alpha(m) + j * dr
    Next j

    ja = ja - 1
    jb = jb + 1

    For j = ja To jb

    If j = jb Then
    Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j = jb - 1 Then
    Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a,
dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j < jb - 1 And j > ja + 1 Then
    Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j) * prob(0, j, jmax,
a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    ElseIf j = ja + 1 Then
    Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a,
dt) * Exp(-r(m, j + 1) * dt)
    ElseIf j = ja Then
    Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    End If
    Next j
```

```
Next m

'using the Q(Transition) to calculate alpha(Transition) and so on up to N
For m = Transition To N

jplus = jmax
jminus = jmin

summe = 0
For j = jmin To jmax
summe = summe + Q(m, j) * Exp(-j * dr * dt)
Next j

alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
For j = jminus To jplus
r(m, j) = alpha(m) + j * dr
Next j

For j = jminus To jplus

'considering the spezial case when nonstandard branching leads to central nodes with five incoming arrows
If jmax = 2 Then
Q(m + 1, 2) = Q(m, 2) * prob(1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt)
Q(m + 1, 1) = Q(m, 2) * prob(0, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(0, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt)
Q(m + 1, 0) = Q(m, 2) * prob(-1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(-1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(0, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -1) = Q(m, 0) * prob(-1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(0, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -2) = Q(m, -1) * prob(-1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(-1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Else
If j = jplus Then

Q(m + 1, j) = Q(m, j) * prob(1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 1 Then
```

```
    Q(m + 1, j) = Q(m, j + 1) * prob(0, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j = jplus - 2 Then

    Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 2) * prob(-1, j + 2, jmax, a, dt) * Exp(-r(m, j + 2) * dt)
    ElseIf j < jplus - 2 And j > jminus + 2 Then

    Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j = jminus + 2 Then

    Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j - 2) * prob(1, j - 2, jmax, a, dt) * Exp(-r(m, j - 2) * dt)
    ElseIf j = jminus + 1 Then

    Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(0, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    ElseIf j = jminus Then

    Q(m + 1, j) = Q(m, j) * prob(-1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    End If
    End If
    Next j

Next m



'cash flow calculations


tau = Tdata(2) - Tdata(1)


    For j = -Application.Min(N, jmax) To Application.Min(N, jmax)
    Rck(N, j) = Newton2(0, r(N, j), a, dt, sigma, xdata, ydata, T, Tdata)
    Next j
```

```
For f = 1 To Trows

    For j = -Application.Min(N, jmax) To Application.Min(N, jmax)

    RcD(N, j) = rzero(rcont(r(N, j), dt, T, sigma, a, xdata, ydata), T, Tdata(f), sigma, a, xdata, ydata)
    O(N, j) = L * Application.Max(z * (Exp(tau * Rck(N, j)) - Exp(tau * RcX)), 0) * Exp(-RcD(N, j) *
(Tdata(f) - T)) + O(N, j)

    Next j

Next f

'backward induction
For i = N To Transition + 1 Step -1

    For j = jmin To jmax

    If j = jmax Then

    O(i - 1, j) = (O(i, j) * prob(1, j, jmax, a, dt) + O(i, j - 1) * prob(0, j, jmax, a, dt) + O(i, j - 2) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j < jmax And j > jmin Then

    O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j = jmin Then

    O(i - 1, j) = (O(i, j) * prob(-1, j, jmax, a, dt) + O(i, j + 1) * prob(0, j, jmax, a, dt) + O(i, j + 2) * prob(1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)
    End If
    Next j
Next i

jminus = -Transition
jplus = Transition

For i = Transition To 1 Step -1
jminus = jminus + 1
```

```
jplus = jplus - 1
    For j = jminus To jplus

     O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j,
jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

     Next j
Next i


HullWhiteSwaption = O(0, 0)

End Function
```

14 Appendix F

14.1 Calculation of binary options

```
Function HullWhiteBinary(sigma, a, T, tau, L, RcX, xdata, ydata)
dt = 1 / 248  'step is one business day

'dimensioning variables
Dim alpha() As Double, Q() As Double, O() As Double, r() As Double, rconti() As Double, s() As
Integer
Dim N As Integer, j As Integer, jmax As Integer, jmin As Integer, jplus As Integer, jminus As Integer, ja
As Integer, jb As Integer, Transition As Integer, f As Integer, m As Integer

N = T / dt



'calculating constants
V = sigma ^ 2 * (1 - Exp(-2 * a * dt)) / (2 * a)
dr = Sqr(3 * V)
jmax = Int(0.184 / (a * dt)) + 1
jmin = -jmax
Transition = Application.Min(N, jmax)

'redimensioning variables
```

```
ReDim alpha(N)
ReDim Q(N + 1, -Transition To Transition)
ReDim O(N, -Transition To Transition)
ReDim r(N, -Transition To Transition)
ReDim rconti(N, -Transition To Transition)
ReDim s(N)

'calculating start values
alpha(0) = -Log(Price(xdata, ydata, dt)) / dt
r(0, 0) = alpha(0)
rconti(0, 0) = rcont(r(0, 0), dt, 0, sigma, a, xdata, ydata)
Q(1, 1) = prob(1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, 0) = prob(0, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, -1) = prob(-1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)

jminus = 0
jplus = 0

ja = -1
jb = 1

'using Q(m) to calculate alpha(m),r(m) and then Q(m+1) up to Transition-1 so we get the Q(Transition)
as last results
For m = 1 To Transition - 1


jminus = jminus - 1

jplus = jplus + 1
   summe = 0
   For j = jminus To jplus
   summe = summe + Q(m, j) * Exp(-j * dr * dt)
   Next j

   alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
   For j = jminus To jplus
   r(m, j) = alpha(m) + j * dr
   rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
   Next j

   ja = ja - 1
```

```
jb = jb + 1

For j = ja To jb

If j = jb Then
Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jb - 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j < jb - 1 And j > ja + 1 Then
Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja + 1 Then
Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = ja Then
Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
Next j
Next m

'using the Q(Transition) to calculate alpha(Transition) and so on up to N
For m = Transition To N

jplus = Transition
jminus = -Transition

summe = 0
For j = -Transition To Transition
summe = summe + Q(m, j) * Exp(-j * dr * dt)
Next j

alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
For j = jminus To jplus
r(m, j) = alpha(m) + j * dr
rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
Next j

For j = jminus To jplus
```

```
'considering the spezial case when nonstandard branching leads to central nodes with five incoming arrows
If jmax = 2 Then
Q(m + 1, 2) = Q(m, 2) * prob(1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt)
Q(m + 1, 1) = Q(m, 2) * prob(0, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(0, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt)
Q(m + 1, 0) = Q(m, 2) * prob(-1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(-1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(0, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -1) = Q(m, 0) * prob(-1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(0, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -2) = Q(m, -1) * prob(-1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(-1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Else
If j = jplus Then

Q(m + 1, j) = Q(m, j) * prob(1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 1 Then

Q(m + 1, j) = Q(m, j + 1) * prob(0, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 2) * prob(-1, j + 2, jmax, a, dt) * Exp(-r(m, j + 2) * dt)
ElseIf j < jplus - 2 And j > jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j - 2) * prob(1, j - 2, jmax, a, dt) * Exp(-r(m, j - 2) * dt)
ElseIf j = jminus + 1 Then

Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(0, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
```

```
    ElseIf j = jminus Then

    Q(m + 1, j) = Q(m, j) * prob(-1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    End If
    End If
    Next j

Next m



'cash flow calculations
For f = 1 To N



If f / tau - Int(f / tau) <> 0 Then
s(f) = (Int(f / tau) + 1) * tau 'time of next swap payment date in days
Else
s(f) = 0
End If

    For j = -Application.Min(f, jmax) To Application.Min(f, jmax)

    If rzero(rcont(r(f, j), dt, f * dt, sigma, a, xdata, ydata), f * dt, f * dt + tau / 248, sigma, a, xdata, ydata) < 0.08 Then
    O(f, j) = (L / 248 * tau / 248 * (Exp(RcX * tau / 248) - 1)) * Exp(-rzero(rcont(r(f, j), dt, f * dt, sigma, a, xdata, ydata), f * dt, s(f) / 248, sigma, a, xdata, ydata) * (s(f) - f)) / 248 + O(f, j)
    Else
    O(f, j) = O(f, j)
    End If

    Next j

Next f

'backward induction
For i = N To Transition + 1 Step -1

    For j = -Transition To Transition

    If j = jmax Then
```

```
    O(i - 1, j) = (O(i, j) * prob(1, j, jmax, a, dt) + O(i, j - 1) * prob(0, j, jmax, a, dt) + O(i, j - 2) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j < jmax And j > jmin Then

    O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

    ElseIf j = jmin Then

    O(i - 1, j) = (O(i, j) * prob(-1, j, jmax, a, dt) + O(i, j + 1) * prob(0, j, jmax, a, dt) + O(i, j + 2) * prob(1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)
    End If
    Next j
Next i

jminus = -Transition
jplus = Transition

For i = Transition To 1 Step -1
jminus = jminus + 1
jplus = jplus - 1
    For j = jminus To jplus

     O(i - 1, j) = (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt) + O(i - 1, j)

     Next j
Next i


HullWhiteBinary = O(0, 0)

End Function
```

15 Appendix G

15.1 Valuation of Callable, Putable Bonds

```
Function HullWhiteCallable(CallputFlag, dt, sigma, a, s, L, X, xdata, ydata)


'dimensioning variables
Dim alpha() As Double, Q() As Double, O() As Double, r() As Double, rconti() As Double, Pbond() As Double, innerval() As Double
Dim N As Integer, j As Integer, jmax As Integer, jmin As Integer, jplus As Integer, jminus As Integer, ja As Integer, jb As Integer, Transition As Integer, m As Integer

N = s / dt


'calculating constants
V = sigma ^ 2 * (1 - Exp(-2 * a * dt)) / (2 * a)
dr = Sqr(3 * V)
jmax = Int(0.184 / (a * dt)) + 1
jmin = -jmax
Transition = Application.Min(N, jmax)

'redimensioning variables
ReDim alpha(N)
ReDim Q(N + 1, -Transition To Transition)
ReDim O(N, -Transition To Transition)
ReDim r(N, -Transition To Transition)
ReDim rconti(N, -Transition To Transition)
ReDim Pbond(N, -Transition To Transition)


'calculating start values
alpha(0) = -Log(Price(xdata, ydata, dt)) / dt
r(0, 0) = alpha(0)
rconti(0, 0) = rcont(r(0, 0), dt, 0, sigma, a, xdata, ydata)
Pbond(0, 0) = AA(0 * dt, s, sigma, a, xdata, ydata) * Exp(-B(0 * dt, s, a) * rconti(0, 0)) * L
Q(1, 1) = prob(1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, 0) = prob(0, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)
Q(1, -1) = prob(-1, 0, jmax, a, dt) * Exp(-r(0, 0) * dt)

jminus = 0
jplus = 0
```

```
ja = -1
jb = 1

'using Q(m) to calculate alpha(m),r(m) and then Q(m+1) up to Transition-1 so we get the Q(Transition)
as last results
For m = 1 To Transition - 1


jminus = jminus - 1

jplus = jplus + 1
    summe = 0
    For j = jminus To jplus
    summe = summe + Q(m, j) * Exp(-j * dr * dt)
    Next j

    alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
    For j = jminus To jplus
    r(m, j) = alpha(m) + j * dr
    rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
    Pbond(m, j) = AA(m * dt, s, sigma, a, xdata, ydata) * Exp(-B(m * dt, s, a) * rconti(m, j)) * L
    Next j

    ja = ja - 1
    jb = jb + 1

    For j = ja To jb

    If j = jb Then
    Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j = jb - 1 Then
    Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a,
dt) * Exp(-r(m, j - 1) * dt)
    ElseIf j < jb - 1 And j > ja + 1 Then
    Q(m + 1, j) = Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j) * prob(0, j, jmax,
a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
    ElseIf j = ja + 1 Then
    Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a,
dt) * Exp(-r(m, j + 1) * dt)
    ElseIf j = ja Then
```

```
Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
Next j
Next m

'using the Q(Transition) to calculate alpha(Transition) and so on up to N
For m = Transition To N

jplus = Transition
jminus = -Transition

summe = 0
For j = -Transition To Transition
summe = summe + Q(m, j) * Exp(-j * dr * dt)
Next j

alpha(m) = (Log(summe) - Log(Price(xdata, ydata, (m + 1) * dt))) / dt
For j = jminus To jplus
r(m, j) = alpha(m) + j * dr
rconti(m, j) = rcont(r(m, j), dt, m * dt, sigma, a, xdata, ydata)
Pbond(m, j) = AA(m * dt, s, sigma, a, xdata, ydata) * Exp(-B(m * dt, s, a) * rconti(m, j)) * L
Next j

For j = jminus To jplus

'considering the spezial case when nonstandard branching leads to central nodes with five incoming arrows
If jmax = 2 Then
Q(m + 1, 2) = Q(m, 2) * prob(1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt)
Q(m + 1, 1) = Q(m, 2) * prob(0, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(0, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt)
Q(m + 1, 0) = Q(m, 2) * prob(-1, 2, jmax, a, dt) * Exp(-r(m, 2) * dt) + Q(m, 1) * prob(-1, 1, jmax, a, dt) * Exp(-r(m, 1) * dt) + Q(m, 0) * prob(0, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -1) = Q(m, 0) * prob(-1, 0, jmax, a, dt) * Exp(-r(m, 0) * dt) + Q(m, -1) * prob(0, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Q(m + 1, -2) = Q(m, -1) * prob(-1, -1, jmax, a, dt) * Exp(-r(m, -1) * dt) + Q(m, -2) * prob(-1, -2, jmax, a, dt) * Exp(-r(m, -2) * dt)
Else
If j = jplus Then
```

```
Q(m + 1, j) = Q(m, j) * prob(1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 1 Then

Q(m + 1, j) = Q(m, j + 1) * prob(0, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jplus - 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 2) * prob(-1, j + 2, jmax, a, dt) * Exp(-r(m, j + 2) * dt)
ElseIf j < jplus - 2 And j > jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt)
ElseIf j = jminus + 2 Then

Q(m + 1, j) = Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt) + Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(1, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j - 2) * prob(1, j - 2, jmax, a, dt) * Exp(-r(m, j - 2) * dt)
ElseIf j = jminus + 1 Then

Q(m + 1, j) = Q(m, j) * prob(0, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j - 1) * prob(0, j - 1, jmax, a, dt) * Exp(-r(m, j - 1) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
ElseIf j = jminus Then

Q(m + 1, j) = Q(m, j) * prob(-1, j, jmax, a, dt) * Exp(-r(m, j) * dt) + Q(m, j + 1) * prob(-1, j + 1, jmax, a, dt) * Exp(-r(m, j + 1) * dt)
End If
End If
Next j

Next m


'cash flow calculations
If CallputFlag = "c" Then

For j = -Application.Min(N, jmax) To Application.Min(N, jmax)
O(N, j) = Application.Min(Pbond(N, j), X) + O(N, j)
```

```
    Next j



'backward induction
For i = N To Transition + 1 Step -1

    For j = -Transition To Transition

    If j = jmax Then

    O(i - 1, j) = Application.Min(X, (O(i, j) * prob(1, j, jmax, a, dt) + O(i, j - 1) * prob(0, j, jmax, a, dt) + O(i, j - 2) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)

    ElseIf j < jmax And j > jmin Then

    O(i - 1, j) = Application.Min(X, (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)

    ElseIf j = jmin Then

    O(i - 1, j) = Application.Min(X, (O(i, j) * prob(-1, j, jmax, a, dt) + O(i, j + 1) * prob(0, j, jmax, a, dt) + O(i, j + 2) * prob(1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)
    End If
    Next j
Next i

jminus = -Transition
jplus = Transition

For i = Transition To 1 Step -1
jminus = jminus + 1
jplus = jplus - 1
    For j = jminus To jplus

     O(i - 1, j) = Application.Min(X, (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) + O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)

     Next j
Next i
```

```
Else
     For j = -Application.Min(N, jmax) To Application.Min(N, jmax)
    O(N, j) = Application.Max(Pbond(N, j), X) + O(N, j)
    Next j


'backward induction
For i = N To Transition + 1 Step -1

    For j = -Transition To Transition

    If j = jmax Then

    O(i - 1, j) = Application.Max(X, (O(i, j) * prob(1, j, jmax, a, dt) + O(i, j - 1) * prob(0, j, jmax, a, dt) +
O(i, j - 2) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)

    ElseIf j < jmax And j > jmin Then

    O(i - 1, j) = Application.Max(X, (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) +
O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)

    ElseIf j = jmin Then

    O(i - 1, j) = Application.Max(X, (O(i, j) * prob(-1, j, jmax, a, dt) + O(i, j + 1) * prob(0, j, jmax, a, dt) +
O(i, j + 2) * prob(1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)
    End If
    Next j
Next i

jminus = -Transition
jplus = Transition

For i = Transition To 1 Step -1
jminus = jminus + 1
jplus = jplus - 1
    For j = jminus To jplus

     O(i - 1, j) = Application.Max(X, (O(i, j + 1) * prob(1, j, jmax, a, dt) + O(i, j) * prob(0, j, jmax, a, dt) +
O(i, j - 1) * prob(-1, j, jmax, a, dt)) * Exp(-r(i - 1, j) * dt)) + O(i - 1, j)
```

```
    Next j
Next i
End If
HullWhiteCallable = O(0, 0)
End Function
```

16 References

Balduzzi, P., S. R. Das, S. Foresi, and R. Sundaram: „A Simple Approach to Three-Factor Affine Term Structure Models", *Journal of Fixed Income*, Dec. 1996, 43-52

Bjerksund, P., G. Stensland: „Implementation of the Black-Derman-Toy Interest Rate Model", *Journal of Fixed Income,* Sept. 1996, 67-75

Black, F.: „The Pricing of Commodity Contracts", *Journal of Financial Economics*, March 1976, 167-179

Black, F., and M. Scholes: „The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, 1973, 637-645

Black, F., E. Derman, and W. Toy: „A One-Factor Model of Interest Rates and Its Application to Treasury Bond Options", *Financial Analysts Journal*, Jan-Feb 1990, 33-39

Black, F., and P. Karasinski: „Bond and Option Pricing When Short Rates are Lognormal", *Financial Analyst Journal*, July-August, 1991, 52-59

Brennan, M.J. and E.S. Scwartz: „A Continous Time Approach to Pricing Bonds", *Journal of Banking and Finance 3*, July 1979, 133-55

Brennan, M.J. and E.S. Scwartz: „An Equilibrium Model of Bond Pricing and a Test of Market Efficiency", *Journal of Financial and Quantitative Analysis 17*, 3, Sep. 1982, 301-29

Chen, R. R., and L. Scott: „Interest Rate Options in Multifactor Cox-Ingersoll-Ross Models of the Term Structure“, *Journal of Derivatives*, Winter 1995, 53-72

Cox, J.C., J.E. Ingersoll, and S.A. Ross: „An Intertemporal General Equilibrium Model of Asset Prices“, *Econometrica 53*, 1985a, 363-384

Cox, J.C., J.E. Ingersoll, and S.A. Ross: „A Theory of the Term Structure of Interest Rates“, *Econometrica 53*, 1985b, 385-407

Cox, J.C., S.A. Ross and M. Rubinstein: „Option Pricing: A Simplified Approach“, *Journal of Financial Economics 7*, Sep. 1979, 229-263

Duffie, D.: *Dynamic Asset Pricing Theory*, 2nd edition, Princeton, NJ: Princeton University Press, 1996

Heath, D., R. Jarrow, and A. Morton: „Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claim Valuation“, *Econometrica 60*, (1992), 77-105

Ho, T.S.Y. and S.-B. Lee: „Term Structure Movements and Pricing Interest Rate Contingent Claims“, *Journal of Finance 41*, (December 1986), 1011-1029

Ho, T.S.Y.: „Evolution of Interest Rate Models: A Comparison“, *Journal of Derivatives 3*, 1995, 9-20

Hull, J. and A. White: „Pricing Interest Rate Derivative Securities, *Review of Financial Studies 3,4* (1990), 573-592

Hull, J. and A. White: „One-Factor Interest Rate Models and the Valuation of Interest Rate Derivative Securities“, *Journal of Financial and Quantitative Analysis, 28* (June 1993a), 235-254

Hull, J. and A. White: „Bond Option Pricing Based on a Model for the Evolution of Bond Prices“, *Advances in Futures and Options Research*, Vol. 6 (1993b) 1-13

Hull, J. and A. White: „Branching Out“, *RISK* July 1994a, 34-37

Hull, J. and A. White: „Numerical Procedures for Implementing Term Structure Models I: Single Factor Models“, *Journal of Derivatives, 2,1* (Fall 1994b), 7-16

Hull, J. and A. White: „Numerical Procedures for Implementing Term Structure Models II: Two-Factor Models“, *Journal of Derivatives, 2,2* (Winter 1994c), 37-48

Hull, J. and A. White: „Using Hull-White Interest Rate Trees“, *Journal of Derivatives*, Spring 1996, 26-36

Hull, J.: *Options, Futures, and other Derivatives*, 3rd edition,Prentice Hall, 1997

Jamshidian, F.: „An Exact Bond Option Formula“, *Journal of Finance, 44*, 1989, 205-209

Jamshidian, F.: „Forward Induction and Construction of Yield Curve Diffusion Models“, *Journal of Fixed Income,* June 1991, 62-74

Jarrow, R. A.: *Modelling Fixed Income Securities and Interest Rate Options*, McGraw-Hill, 1996

Kraus, A., and M. Smith: „A Simple Multifactor Term Structure Model“, *Journal of Fixed Income*, June 1993, 19-23

Li, A., P. Ritchken, and L. Sankasubramanian: „Lattice Models for Pricing American Interest Rate Claims“, *Journal of Finance, 50*, 2 (June 1995), 719-737

Longstaff, F. A. and E. S. Schwartz: „A Two-Factor Interest Rate Model and Contingent Claims Valuation“, *Journal of Fixed Income, 2*, 1992a, 16-23

Longstaff, F. A. and E. S. Schwartz: „Interest Rate Volatility and the Term Structure: A Two-factor General Equilibrium Model“, *Journal of Finance*, 1992b, 1259-1282

Musiela, M.,: „General Framework for Pricing Derivative Securities“, *Stochastic Processes and Their Applications, 55,* 1995, 227-251

Rendleman, R. J. jr. and B. J. Bartter: „The Pricing of Options on Debt Securities“, *Journal of Financial and Quantitative Analysis*, 15, 1980, 11-24

Schaefer, S. M. and E. S. Schwartz: „A Two-Factor Model of the Term Structure: An Approximation Analytical Solution“, *Journal of Financial and Quantitative Analysis, 19*, 1984, 413-448

Singh, M. K.,: „Estimation of Multifactor Cox, Ingersoll, and Ross Term Structure Model: Evidence on Volatility Structure and Parameter Stability“, *Journal of Fixed Income*, Sep. 1995, 8-28

Vasicek, O.: „An Equilibrium Characterisation of the Term Structure“, *Journal of Financial Economics, 5*, 1977, 177-188