Choice of One Factor Interest Rate Term Structure Models for Pricing and Hedging Bermudan Swaptions

Amiel Holilal

May 21, 2011

UCT

Acknowledgments

I would sincerely like to thank Craig Anderson for his valuable time and guidance, in providing me the insight needed to understand the instruments used in investment banks and their implementation. I would also like to thank Shakill Hassen for his assistance and guidance. A special thanks to Professor Ronald Becker for his valuable input, guidance and his constant quest for perfection. And last but not least I would like to thank my family for their stern support.

Declaration

I hereby certify that this project was independently written by me. No material was used other then that referred to. Sources directly quoted and ideas used, including figures, tables, sketches, drawings and photos, have been correctly denoted. Those not otherwise indicated belong to the author. I have used the following convention for citation and referencing: Harvard Method.

Contents

Abstract

This paper revisits pricing and hedging differences presented by [5] from a South African context. The Asset Liabilities Management (ALM) departments in large financial institutions are plagued by a number of problems. Among them is the choice of interest rate model for managing the risks associated with mortgage (home loan) repayments. This paper will address these problems by comparing various one-factor models, including Hull-White, Black-Karasinski and CIR models for the pricing and hedging of long-term Bermudan Swaptions which resembles mortgage loans in banks' books.

1 Introduction

This paper will look at the pricing and hedging benefits derived from using mean-reverting one-factor models in relation to Asset Liabilities Management (ALM). The preference of one-factor models is primarily due to its tractability and its extensive use in ALM divisions, in larger financial institutions. We shall use one-dimensional trinomial trees for pricing and hedging Bermudan swaptions.

'Unlike the short-term pricing problem, the one-factor model is often preferred for the longer term ALM purpose because of its simplicity. For long horizon hedging, the multi-factor model could produce more noise as it requires more parameter input.'[5][1]

Intuitively, the pricing performance can be improved by adding explanatory variables and thereby increasing the complexity of the dynamics. However, the pricing performance alone cannot reflect the model's ability in capturing the true term structure dynamics. To assess the appropriateness of the models dynamics, it is strongly suggested that we evaluate the forecasting and hedging performance of the models.

The Hull-White and Black-Karasinski models are used extensively in industry. However, there are some shortfalls pertaining to Hull-White and Black-Karasinski one-factor models. These shortfalls are overcome by the inclusion of the CIR model. The introduction of the CIR model is a step forward in interest rate option pricing, as it presents a hybrid alternative to the Hull-White and Black-Karasinski. This paper will also try to establish pricing efficiencies presented by models with analytic solutions and the complexity of implementing models without analytical solutions.

The reason for having one model is to have a coherent model for the entire portfolio, so that the entire portfolios risk can be measured and hedged in a consistent framework. For costs and simplicity reasons, the ALM departments in large financial institutions would ideally like to choose one particular model to actively manage there balance sheet. In South Africa banks usually have a diverse range of mortgage portfolios. Since clients have the right to repay their mortgages prior to maturity, resulting in banks running the risk of unhedged exposure of interest rate risk. This risk can effectively be managed and in some regard mitigated by holding a Bermudan swaption with equivalent maturity.

The Bermudan swaption is chosen as the hedging instrument because it resembles a loan portfolio with an early redemption feature, an important product for most banks. Bermudan swaptions are interest rate derivatives, with an exotic feature. They are among the most liquidly traded interest rate derivatives. As a consequence, their pricing and more importantly their

[1] p2

risk management plays a pivotal role in a banks home loan division. The exotic features pose significant difficulties in the pricing and hedging of these instruments. Unlike the LIBOR Market Model (LMM), the underlying early exercise feature can easily be overcome by trinomial trees in one-factor models.

The contribution of this paper is primarily numerical, the main objective of which is to develop a computationally efficient swaption-pricing technology using trinomial tree methods. The pricing algorithms developed will greatly facilitate future empirical research into testing the goodness of fit of underlying term-structure models and in evaluating the dynamic hedging performance of various derivative-pricing models. These are topics of considerable interest among academics and practitioners alike.

The paper is organized as follows. In section 2 we review the literature, sort through the current set of confusing empirical results, and highlight the contributions that this article makes to the literature. Section 3 will provide a detailed account of the Hull-White, Black-Karasinski and CIR++ models. The data is presented in Section 4. Section 5 looks at the calibration techniques used for interest rate models. The unique pricing and hedging techniques associated with Bermudan Swaptions are analysed in Sections 6 and 7.

2 Literature Review

The application of future option pricing methodology of Black (Black's model) to swaptions is well established. Under Black's model, the forward swap rate is assumed to have a lognormal distribution and, hence, a closed-form solution for the price of a swaption is derived. However, closed-form solutions for swaption prices do not exist under more general model dynamics, and so various approximate pricing methods have been developed in the literature. The current literature comparing various one-factor short rate models is not very extensive to date, as there has been a dramatic shift towards multi-factor models and in particular towards Libor Market Models. The focus of the literature review here will be on one-factor short rate models literature.

A number of previous studies have examined various term structure models for pricing and hedging interest rate derivatives. Most of the literature that looks at one factor models, concentrates on the Hull-White and Black-Karasinski models. This is attributed to their extensive implementation in industry: 'The choice of HW and BK is simple: ... they are the most important and popular short rate models used by the industry'[5][2]. Jansson [12] looked at the risk neutral valuation of Bermudan swaptions using an exact pricing formula in terms of a Snell envelope. The Snell envelope was used in the valuation procedure for Bermudan Swaptions, using the one-factor Ho-Lee model. Jansson [12] was able to establish that the procedure was computationally fast and has the benefit of easily calibrating to market data. Andersen and Andreasen [1] use a mean-reverting Gaussian model and a lognormal Libor Market Model for pricing Bermudan swaption. They found that for both models, Bermudan swaption prices change only moderately when the number of factors in the underlying interest rate model is increased. Steffen Hippler [9] priced Bermudan swaptions in the LIBOR Market Model. He discovered that the choice of a realistic volatility function plays a far more transparent and important roles then that of the corresponding correlation function. Anastasia Halamandaris [7] compared the implementation of various short rate models, including Hull-White, Black-Karasinski and G2++, for the pricing of interest rate products. Her paper illustrated the benefit of using the lattice technique over other numerical techniques and the analytical tractability of using a normal short-rate model over a lognormal short rate model. Pietersz and Pelsser (2005) compare single factor Markov-functional and multi-factor market models for hedging Bermudan swaptions. Their results show that delta and delta-vega hedging performances of both models are comparable. Gupta and Subrahmanyam [6] compare various one- and two-factor models based on the out-of-sample pricing performance, and the models' ability to delta-hedge caps and floors. They found that one-factor short rate models with time varying parame-

[2] p3

ters and the two-factor model produced similar sized pricing errors. But in terms of hedging caps and floors, the two-factor models are more effective. For both pricing and hedging of caps and floors, the BK model is better than the HW model [2]. With regard to both pricing and hedging, their results are in line with those obtained by Fan, Gupta and Ritchken [4]. Driessen, Klaassen and Melenberg [3] use a range of term structure models to price and hedge caps and (European) swaptions.

3 One-factor Interest Rate Models

When dealing with interest-rate products, the main variability that matters is clearly that of the interest rates themselves. It is therefore necessary to drop the deterministic setup and to start modelling the evolution of *r(t)* in time through a stochastic process. Unlike stock prices, interest rates are mean reverting (i.e. pulled back) to some long term average level over time.

In order to price interest rate products that include optionality, we need to arrive at a statistical model of the evolution of the yield curve. The models we will consider are all dependent on the short rate. The evolution of the short rate then governs the evolution of all rates along the entire yield curve. Trying to model the evolution of the yield curve can prove to be a tedious task, as the yield curve may undergo a combination of parallel shifts, slope changes and curvature changes.

In a one-factor equilibrium model, the process for the short rate involves only one source of uncertainty (or only one explanatory variable), the instantaneous short rate *(r)*. The short rate models are usually diffusion models, and thus have Markov properties. Some examples of short rate models include Vasicek, Hull-White and Black-Karasinski. When considering short rate models the dynamics of the term structure of interest rates are determined by specifying the dynamics of a single rate (the short rate) from which the whole term structure[3] at any point in time can be calculated. Short rate models are usually implemented as models with a single stochastic process driving the term structure of interest rates. A disadvantage is then that the instantaneous correlation between interest rates can only be 1.

The choice of a trinomial tree approach for modelling short rates and in turn pricing Bermudan swaptions[4] is primarily attributed to its tractability and accuracy over conventional binomial trees or least squares monte carlo. The main advantage of a trinomial tree is the extra degree of freedom provided, making it easier for the tree to represent features of an interest rate process such as mean reversion.

3.1 Vasicek

The premise of the models presented in the paper is based on Vasicek Model. Vasicek (1977) assumed that the instantaneous spot-rate (r) under the real-world measure P evolved as an Ornstein-Uhlenbeck [5] process,

$$dr(t) = k[\vartheta - r(t)]dt + \sigma dW_t \tag{1}$$

[3] A term-structure (yield curve) provides the applicable interest rate for each maturity.
[4] Appendix 9.1- Definition 5
[5] Appendix 9.1 - Definition 6

with constant coefficients (k, ϑ, σ). This is equivalent to assuming that r follows an Ornstein-Uhlenbeck process, with constant coefficients under the risk-neutral measure (RNM) $\mathbb{Q}$ as well.

3.2 Hull-White extended Vasicek Model[6]

Hull-White originally proposed a mean reverting short rate model that evolved as

$$dr(t) = [\vartheta(t) - a(t)r(t)]\,dt + \sigma(t)dW_t \tag{2}$$

where the parameters in the model are dependent on time (t)[7]. Such a model can be fitted to the term structure of interest rates and the term structure of spot or forward-rate volatilities. However, this may be somewhat complicated when applied to concrete market situations. As the perfect fitting to a volatility term structure can be rather problematic. [8]

By considering a further time-varying parameter, Hull-White [2] proposed a more general model, that is also able to fit a given term structure of volatilities. Thus, we shift our attention to the Hull-White extended Vasicek model

$$dr(t) = [\vartheta(t) - ar(t)]dt + \sigma dW_t \tag{3}$$

where the only parameter dependent on time is $\vartheta(t)$, and a and σ are positive constants. The Hull –White extended Vasicek model addresses some of the short falls in the Vasicek model. We choose $\vartheta(t)$[9] so as to exactly fit the term structure of interest rates being currently observed in the market. The difficulties associated with perfectly fitting a volatility term structure is the main reason why we stick to the extension, where only one parameter (ϑ), is chosen to be a deterministic function of time. The model we analyse implies a normal distribution for the short-rate process at each time. Moreover, it is quite analytically tractable, in that zero-coupon bonds and options on them can be explicitly priced. The normally distributed short-rates allow for the derivation of analytical formulas and the construction of efficient numerical procedures for pricing a variety of derivative securities. On the other hand, the possibility of negative rates and the one-factor formulation make the model not very applicable to concrete pricing problems.

[6] [2]p72

[7] ϑ, a and σ are deterministic functions of time

[8] (Brigo and Mercurio). The reason is two-fold. First, not all the volatilities that are quoted in the market are significant: some market sectors are less liquid, with the associated quotes that may be neither informative nor reliable. Second, the future volatility structures implied by (2) are likely to be unrealistic in that they do not conform to typical market shapes, as was remarked by [10] themselves.

[9] $\vartheta(t)$ is dependant on the market forward rate

3.2.1 Short Rate Dynamics[10]

$$dr(t) = [\vartheta(t) - ar(t)]dt + \sigma dW_t$$

Using Ito's lemma, the short rate can be represented in a integral form of a stochastic differential equation as follows

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

where $\vartheta(t)$ is defined to be,

$$\vartheta(t) = \frac{\partial f^M(0,t)}{\partial t} + af^M(0,t) + \frac{\sigma^2}{2a}(1 - e^{-2at})$$

Where $f^M(0,t)$ denote the instantaneous forward-rate observed in the market, at time $t = 0$, for the maturity T, thus

$$f^M(0,T) = \frac{\partial \ln[P^M(0,T)]}{\partial T}$$

[11]Let $\alpha(t)$ be defined by

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

Then, the short rate under the extended Hull-White model can be expressed as,

$$r(t) = r(s)e^{-a(t-s)} + \alpha(t) - \alpha(s)e^{-a(t-u)} + \sigma \int_s^t e^{-a(t-u)}dW_u$$

[12]However, since the standard Brownian motion *W(u)* is normally distributed with $E[W(u)] = 0$ and $\text{Var}[W(u)] = u$ we can say that *r(t)*, conditional on $\mathcal{F}_s$, is normally distributed with mean,

$$E\{r(t) \mid \mathcal{F}_s\} = r(s)e^{-a(t-s)} + \alpha(t) - \alpha(s)e^{-a(t-s)}$$

and variance

$$\text{Var}\{r(t) \mid \mathcal{F}_s\} = \frac{\sigma^2}{2a}\left[1 - e^{-2a(t-s)}\right]$$

13

[10][2]p73
[11]P^M is the market discount factor
[12][2]
[13][2]

To generate an algorithm for Bermudan Swaption Prices, we need to initially construct a trinomial tree. Brigo and Mercurio [2] suggested that the construction of a trinomial tree, of the short rate, begins by initially constructing a tree for an x process and then adjusting the tree by α to obtain the trinomial tree for the short rate process.

The stochastic process x is defined by,

$$dx(t) = -ax(t)dt + \sigma dW_t$$

Thus, the short rate *r(t)* is determined by

$$r(t) = x(t) + \alpha(t) \tag{4}$$

For details of the derivation and a more detailed account of the above formula, we refer to [2][14]

3.2.2 Algorithm to Generate Bermudan Swaption Prices

Hull-White proposed a robust two-stage procedure for constructing trinomial trees to represent a wide range of one-factor models. We shall extend this process to provide a detailed account of how to price a Bermudan swaption using a trinomial tree [2]:

Stage 1: The first stage in building a tree for this model is to construct a tree for the variable x that follows the stochastic process:

$$dx(t) = -ax(t)dt + \sigma dW_t \tag{5}$$

this process is described in Appendix 9.3. The process is symmetrical about $x = 0$. The variable $x\,(t + \Delta t) - x(t)$ is normally distributed, with mean $-ax\,(t)\,\Delta t$ and variance $\sigma^2 \Delta t$.

In order to implement the stochastic process we discretize the time horizon, where $\triangle t_i = t_{i+1} - t_i$ for each i.

Our objective of the first (tree building) stage is to build a tree similar to that shown in Figure 2 for x. To do this we must resolve which of the tree branching methods shown in Figure 1 will apply at each node. This in turn will determine the overall geometry of the tree.

The process for x will evolve according to the trinomial tree, where $x_{i,j} = j\triangle t_i$ is the value of the process at time t_i for the j-th node. From this node, the process can take on one of three values, $x_{i+1,k+1}, x_{i+1,k}, x_{i+1,k-1}$, where $x_{i+1,k}$ is the central node. The level of k is set such that $x_{i+1,k}$ is as close as possible to $M_{i,j}$.

In a tree-building procedure for x, we must identify the altitude of the tree by defining $\overline{j}$ and $\underline{j}$, as the maximum and minimum number of nodes at each time step, t_i, respectively.

[14] p74

Denote $Q_{i,j}$ as the present value of an instrument paying 1 if node (i,j) is reached and zero otherwise. The values of α_i and $Q_{i,j}$ are calculated recursively from α_0 that is set so as to retrieve the correct discount factor for the maturity t_1, i.e., $\alpha_0 = -ln(P^M(0,t_1))/t_1$. As soon as the value of α_i is known, the values $Q_{i+1,j}$, $j = \underline{j}_{i+1}, \ldots, \bar{j}_{i+1}$, are calculated through

$$Q_{i+1,j} = \sum_h Q_{i,h} q\left(h,j\right) exp\left(-\left(\alpha_i + h\Delta x_i\right)\Delta t_i\right)$$

$q\left(h,j\right)$ is the probability of moving from node (i,h) to node $(i+1,j)$and the sum is over all values of h for which such probability is non-zero.[15]

Define $\bar{j}$ as the value of j where we switch from branching style 'a' to branching 'b' and similarly, $\underline{j}$ is the switch from branching style 'a' to 'c'. Where $\bar{j}$ and $\underline{j}$ at time t_{i+1} can be determined by finding the values of k for the nodes $x_{i,\bar{j}}$ and $x_{i,\underline{j}}$, respectively.

The values of the nodes at each time must be calculated in an iterative manner, starting at the current time and working as far out into future as desired.

At some nodes in the tree, the branching needs to account for the impact of mean reversion. At nodes towards the top of the tree (representing 'high' interest rates), the branching alters to reflect the fact that rates are more likely to decline during the next period. Likewise, when rates are at 'low' levels the branching must change to reflect the increasing likelihood that rates will increase. The various branching alternatives are depicted below.

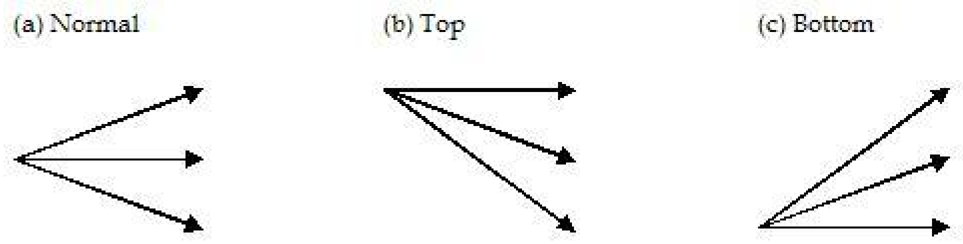


Figure 1
Alternate Branching for HW Trinomial Trees

[15] [2]p80

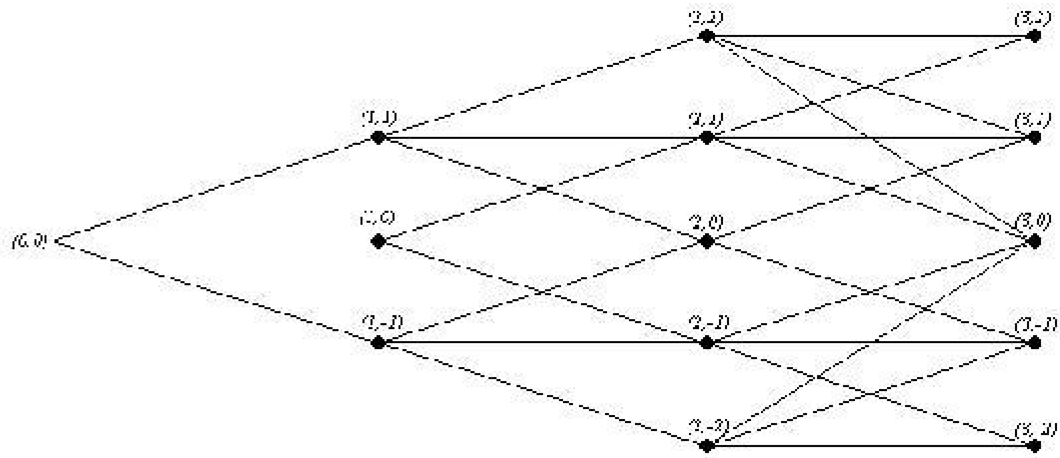


Figure 2

Stage 2: The second stage in the tree construction is to convert the tree for x into a tree for r. This is accomplished by displacing the nodes at each time step of x. In order to obtain the trinomial tree for the short rate process such that the structure matches the current term structure. Thus,

$$r(t) = x(t) + \alpha(t)$$

This procedure is consistent with the initial term structure, as $\alpha(t)$ is calculated iteratively so that the initial term structure is matched exactly.[16]

Stage 3: Using the short rate trinomial tree just constructed. We are able to determine the Bermudan swaption price in the discrete time setting under the risk neutral probability $\mathbb{Q}$.[17][18]

The backward orientation of trinomial trees, allows us to determine the price of a Bermudan swaption through a 'back stepping method'. i.e. a method where the price (V) is calculated by iteratively constructing the price in a descending order, of time. Given the transition probabilities $p_u(i,j), p_m(i,j), p_d(i,j)$, obtained from stage 1 at time $t = (i+1)\triangle t$. The Bermudan swaption price at (i,j) $\left(V^{Bermudan}_{swaption}(i,j)\right)$, can be derived in a manner, similar to other Bermudan options. Consider a Bermudan swaption, that has not been exercised, at a time T_i, $i<\beta$, the holder of the swaption

[16] $\alpha_i = \frac{1}{\Delta t_i} \ln \frac{\sum_{j=\underline{j}}^{\overline{j}} Q_{i,j} \exp(-j\Delta x_i \Delta t_i)}{P(0,t_{i+1})}$

[17] The risk-neutral probability of negative rates at time t is explicitly given by

$$Q\left\{r(t) < 0\right\} = \Phi\left(-\frac{\alpha(t)}{\sqrt{\frac{\sigma^2}{2a}\left[1 - e^{-2at}\right]}}\right)$$

with Φ denoting the standard normal cumulative distribution function.([2]p74)

[18] We can also construct the trees under the forward neutral probabilities $\mathbb{Q}^{T_N}$ and $\mathbb{Q}^{T_i}$ in the same way.

has the right to receive

$$A\left(T_i\right)=N\left(\sum_{k=i+1}^{\beta} P\left(T_i, T_k\right) \tau_k\left(S_k\left(T_i\right)-K\right)\right)^{+}. \tag{6}$$

Under the Hull-White model, the time t price of T-bond can be explicitly derived as

$$P(t,T)=\exp[-A(t,T)-B(t,T)r(t)]$$

where,

$$A(t,T)=-\frac{\sigma^2}{2}\int_t^T B(s,T)^2 ds+\int_t^T \nu(s)B(s,T)ds,$$

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

Thus,

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

For details of the derivation of the bond price, we refer to [15].

The swap rate is determined by

$$S_{\alpha\beta}(t)=\frac{P(t,T_\alpha)-P(t,T_\beta)}{\sum_{i=\alpha+1}^{\beta}\tau_i P(t,T_i)}.$$

Now, since the maturity of the Bermudan swaption is β, the appropriate numeraire for pricing, is the bond maturing at terminal date T_β. The time-0 value of the Bermudan swaption can be expressed as

$$V_{swaption}^{Bermudan}(0,S;K,N)=P\left(0,T_\beta\right)\times \sup p_{\tau\epsilon T}E^{\beta}\left[\frac{A\left(\tau\right)}{P\left(\tau,T_\beta\right)}\mid \mathcal{F}_0\right] \tag{7}$$

Inorder to determine the expectation, we shall define the backwardly-Cumulated value from Continuation (CC) of the Bermudan swaption:

$$CC(i,j)=e^{(-r_{i,j}(T_{i+1}-T_i))}\left(p_u v_{i+1,j+1}+p_m v_{i,j+1}+p_d v_{i-1,j+1}\right) \tag{8}$$

Thus,

$$v_{i,j}=\Big\{\max\left(A(i,\beta),0\right),\quad \max(A(i,j),CC(i,j))$$

[19]

By backward induction from the terminal condition, we can calculate the current Bermudan swaption price at $(0,0)$[15].

[19] $$V_{Swaption}^{Bermudan}=\max\Bigg\{\delta\left(S(T_i)-K\right)^{+}\Bigg(p_u(i,j)\frac{\sum_{k=i+1}^{i+n}P(i+1,k,j+1)}{D_u(i,j)}+p_m(i,j)\frac{\sum_{k=i+1}^{i+n}P(i+1,k,j)}{D_m(i,j)}$$

3.3 Black-Karasinski[20]

Black and Karasinski assumed that the logarithm $\ln(r(t))$ of the instantaneous spot rate evolves under the risk neutral measure $\mathbb{Q}$ according to

$$d\ln(r(t)) = [\vartheta(t) - a\ln(r(t))]\,dt + \sigma dW_t, r(0) = r_0 \tag{9}$$

As in previous models, the coefficients a and σ can be interpreted as follows: a gives a measure of the "speed" at which the logarithm of *r(t)* tends to its long-term value; σ is the standard-deviation rate of *dr(t)/r(t)*, namely the standard deviation per time unit of the instantaneous return of *r(t)*.

3.3.1 Short rate Dynamics for Black-Karasinski

$$d\ln(r(t)) = [\vartheta(t) - a\ln(r(t))]\,dt + \sigma dW_t$$

Using Ito's lemma we obtain

$$d(r(t)) = r(t)\left[\vartheta(t) + \frac{\sigma^2}{2} - a\ln(r(t))\right]dt + \sigma r(t)dW_t$$

Solving this we obtain,

$$r(t) = \exp\left[\ln(r(s))e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\vartheta(u)du + \sigma\int_s^t e^{-a(t-s)}dW_u\right]$$

where $s \leq t$.

The mean of *r(t)*, with respect to the filtration $\mathcal{F}_s$, is given by

$$E\{r(t) \mid \mathcal{F}_s\} = \exp\left[\ln(r(s))e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\vartheta(u)du + \frac{\sigma^2}{4a}(1 - e^{-2a(t-s)})\right]$$

and the variance of $r(t)$

$$\text{Var}\{r(t) \mid \mathcal{F}_s\} = \exp\left[2\ln(r(s))e^{-a(t-s)} + 2\int_s^t e^{-a(t-u)}\vartheta(u)du + \frac{\sigma^2}{4a}(1 - e^{-2a(t-s)})\right]$$

Let,

$$+p_d(i,j)\frac{\sum_{k=i+1}^{i+n} P(i+1,k,j-1)}{D_d(i,j)}\Bigg), p_u(i,j)\frac{V_S^B(i+1,j+1)}{D_u(i,j)}$$

$$p_m(i,j)\frac{V_S^B(i+1,j)}{D_m(i,j)} + p_d(i,j)\frac{V_S^B(i+1,j-1)}{D_d(i,j)}\Big\}$$

[20] [2]p83

$$\alpha(t) = \ln(r_0)e^{-at} + \int_0^t e^{-a(t-u)}\vartheta(u)du$$

The long term mean of the short rate cannot be calculated analytically. A numerical procedure such as the trinomial lattice can be implemented to derive a short rate tree that matches the initial term structure.

3.3.2 Algorithm to Generate Bermudan Swaption Prices

The algorithm presented below, a suggestion by Brigo and Mercurio [2], is a modification of the Hull-White model presented in [10].

Once, again we use a three stage process to price a Bermudan swaption.

Stage 1: This procedure is similar to the one presented in the extended Hull-White model. However, there is difference in the mean and variance formulae.

Stage 2: After generating the entire tree for the x process. The tree nodes are now replaced, so as to match the current term structure. Given the short rate dynamics of the model, we can write the short rate as a function of time,

$$r(t) = e^{\alpha(t)+x(t)}$$

where

$$\alpha(t) = \ln(r_0)e^{-at} + \int_0^t e^{-a(t-u)}\vartheta(u)du$$

and the stochastic differential of the x process is given by,

$$dx(t) = -ax(t)dt + \sigma dW_t$$

where $x(0)$=0. The integrated equation is given by

$$x(t) = x(s)e^{-a(t-s)} + \sigma \int_s^t e^{-a(t-u)}dW_u$$

for each $s < t$.

Since the short rate is lognormally distributed in the Black-Karasinski model. We use a numerical procedure to generate α, as a analytical solution does not exist. Begin by calculating the current value of α,

$$\alpha_0 = \ln\left[-\ln\left(\frac{P^M(0,t_1)}{t_1}\right)\right]$$

where $P^M(0,t_1)$is the market discount factor for the maturity t_1. Denote $Q_{i,j}$ as the present value of an instrument paying 1 if node (i,j) is reached and 0 otherwise. Given, α_0, $Q_{1,j}$can be calculated for all j

$$Q_{1,j} = \sum_{h} Q_{0,h} q(h,j) \exp\left[-\triangle t_0 \left(\exp(\alpha_0 + h\triangle x_0)\right)\right]$$

where $q(h,j)$ is the probability of moving from node (i,h) to node $(i+1,j)$. After calculating $Q_{1,j} \forall j = \overline{j_1}, ..., \underline{j_1}$ we can calculate α_1 by matching it to the market discount factor for the maturity t_2. This can be calculated by numerically solving the equation,

$$\rho(\alpha_i) = P^M(0, t_{i+1}) - \sum_{\overline{j_1}}^{\underline{j_1}} Q_{i,j} \exp\left[-\exp(\alpha_i + j\triangle x_i)\triangle t_i\right] = 0.$$

This can be solved using Newton-Raphson or Newton-Bailey's method since both the first and second derivative of $\rho(\alpha_1)$ is known.[21]

Thus, the short rate $r_{i,j}$ is obtained from the formula

$$r_{i,j} = e^{x_{i,j} + \alpha_i}.$$

Stage 3: The lack of analytical tractability of the Black Karasinski model is illustrated by the fact that the model does not yield analytical formulas either for discount bonds or for options on bonds. The pricing of these instruments is performed through numerical procedures. Thus, greatly reducing the analytical tractability.

[21] The Newton-Bailey method has a better rate of convergence then Newton-Raphson's method

3.4 Extended Cox, Ingersoll & Ross (CIR++) Model

3.4.1 The Original CIR Model

The mean-reverting short-rate dynamics proposed by Vasicek is a popular choice of interest rate model among practitioners. However, negative short rates are unlikely from an economic standpoint. So as prevent negative short rates Cox, Ingersoll and Ross proposed the introduction of a 'square-root' term in the diffusion coefficient of the Vasicek model.

The CIR model formulation under the risk-neutral measure $\mathbb{Q}$ is

$$dr(t) = k\left[\theta - r(t)\right]dt + \sigma\sqrt{r(t)}dW_t \tag{10}$$

where r_0, k, θ, σ are positive constants. To ensure the positivity of the short rate in the CIR model, the following condition needs to be satisfied

$$2k\theta > \sigma^2.$$

The condition $2k\theta > \sigma^2$ has to be imposed to ensure that the origin is inaccessible to the process (10), so that we can grant that r remains positive.[22]However, since the tree is an approximation to the continuous, model there is no guarantee that r will be > 0.

3.4.2 CIR++ Model

The short rate, in the extended CIR model, is composed of two parts. The x(t)process

$$dx\left(t\right) = k\left(\theta - x\left(t\right)\right)dt + \sigma\sqrt{x\left(t\right)}dW_t \tag{11}$$

where x_0, k, θ, σ are all positive constants, such that

$$2k\theta > \sigma^2,$$

thus ensuring that the origin is inaccessible to x, and hence the process x remains positive. And $\varphi\left(t\right) = \varphi^{CIR}\left(t;\alpha\right)$ where

$$\varphi^{CIR}\left(t;\alpha\right) = f^M\left(0,t\right) - f^{CIR}\left(0,t;\alpha\right)$$

where

$$f^{CIR}\left(0,t;\alpha\right) = \frac{2k\theta\left(\exp\left\{th\right\}-1\right)}{2h+\left(k+h\right)\left(\exp\left\{th\right\}-1\right)} + x_0\frac{4h^2\exp\left\{th\right\}}{\left[2h+\left(k+h\right)\left(\exp\left\{th\right\}-1\right)\right]^2} \tag{12}$$

and $h = \sqrt{k^2+2\sigma^2}$.

Thus, the short rate dynamics is given by

$$r\left(t\right) = x\left(t\right) + \varphi\left(t\right).$$

[22][2] p64

3.4.3 Short Rate Dynamics of CIR

$$dr(t) = k[\theta - r(t)]\,dt + \sigma\sqrt{r(t)}dW_t$$

The mean and variance for the CIR $r(t)$ conditional on $\mathcal{F}_s$, in the CIR model, is given by

$$E\{r(t) \mid \mathcal{F}_s\} = r(s)\,e^{-k(t-s)} + \theta\left(1 - e^{-k(t-s)}\right)$$

$$\text{Var}\{r(t) \mid \mathcal{F}_s\} = r(s)\,\frac{\sigma^2}{k}\left(e^{-k(t-s)} - e^{-2k(t-s)}\right) + \theta\frac{\sigma^2}{2k}\left(1 - e^{-k(t-s)}\right)^2.$$

Now that we have fully determined the dynamics of the CIR++ model, we are able to determine the analytical price at time t of a zero-coupon bond maturing at time T

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

where

$$\bar{A}(t,T) = \frac{P^M(0,T)\,A(0,t)\exp\{-B(0,t)\,x_0\}}{P^M(0,t)\,A(0,T)\exp\{-B(0,T)\,x_0\}}A(t,T)\,e^{B(t,T)\varphi^{CIR}(t;\alpha)}$$

$$A(t,T) = \left[\frac{2h\exp\{(k+h)(T-t)/2\}}{2h + (k+h)(\exp\{(T-t)h\} - 1)}\right]^{2k\theta/\sigma^2}$$

$$B(t,T) = \frac{2(\exp\{(T-t)h\} - 1)}{2h + (k+h)(\exp\{(T-t)h\} - 1)}.$$

3.4.4 Algorithm to Generate Bermudan Swaption Prices

The proposed method of pricing fixed income derivatives, using the CIR dynamics, is usually through a binomial tree. To improve the pricing accuracy of fixed income derivatives, we have implemented an alternative trinomial tree for the CIR++ model, which is constructed along the lines proposed by the Hull-White extended Vasicek model.

Step 1: For convergence purposes, we define the process y as

$$y(t) = \sqrt{x(t)}, \tag{13}$$

where process x is defined in (4). By Ito's lemma,

$$dy(t) = \left[\left(\frac{k\theta}{2} - \frac{1}{8}\sigma^2\right)\frac{1}{y(t)} - \frac{k}{2}y(t)\right]dt + \frac{\sigma}{2}dW_t \tag{14}$$

We initially construct a trinomial tree for y, using the procedure outlined in Appendix 9.3.1. We then use $y(t) = \sqrt{x(t)}$ and displace the tree nodes so as to exactly retrieve the initial zero-coupon curve.

Begin by fixing a time horizon T and the times $0 = t_0 < t_1 < \cdots < t_N = T$, and set $\Delta t_i = t_{i+1} - t_i$, for each i. As usual denote the tree nodes by (i, j) where the time index i ranges from 0 to N and the space index j ranges from some $\underline{j_i}$ to some $\bar{j}_i$.

Denote $y_{i,j}$ to be the process value on node (i, j) and set $y_{i,j} = j\Delta y_i$, where $\Delta y_i = V_{i-1}\sqrt{3}$ and

$$V_i = \sigma\sqrt{\Delta t_i}/2.$$

Set

$$M_{i,j} = y_{i,j} + \left[\left(\frac{k\theta}{2} - \frac{1}{8}\sigma^2\right)\frac{1}{y_{i,j}} - \frac{k}{2}y_{i,j}\right]\Delta t_i.$$

The movement of the process from one node to another is very similar to that outlined in the Hull-White case. Assuming that at time $t_i$we are on node (i, j), with associated value $y_{i,j}$, the process can move to $y_{i+1,k+1}$, $y_{i+1,k}$ or $y_{i+1,k-1}$ at time $t_{i+1}$with probabilities p_u, $p_m$and p_d[23],respectively. The central node is therefore the *k-th* node at time t_{i+1}, where k is defined by

$$k = \text{round}\left(\frac{M_{i,j}}{\Delta y_{i+1}}\right).$$

Set $\eta_{i,j} = M_{i,j} - y_{i+1,k}$.

It should be noted that the tree thus defined has the drawback that some nodes may lie below the zero level. Since the tree must approximate a positive process, we truncate the tree below some predefined level $\epsilon > 0$, which can be chosen arbitrarily close to zero, and then suitably define the tree geometry and probabilities around this level.

Step 2: To conclude the generation of short rates using the trinomial tree building approach set $x(t) = y^2(t)$ and then shift the nodes, using

$$r(t) = x(t) + \varphi(t)$$

so the structure matches the current term structure, thereby obtaining the proper tree for r.

Step 3: The pricing of Bermudan swaptions using the CIR++ dynamics is similar to that presented in the extended Hull-White case. However, the analytical solutions of the zero-coupon bonds are different.

3.5 Advantages and Disadvantages of one factor models

The three models presented are relatively indistinguishable in terms of fitting data on actively traded Bermudan swaption. The main advantage of the HW model is that it has analytic tractability. However, the negative rates are a cause of concern and prove to be a disadvantage of implementing the

[23]The formulas for these probabilities are exactly the same as those for all the all other models presented.

HW model. In most circumstances, the probability of negative interest rates occurring under the model is very small, but some analysts are still reluctant to use a model where there is any chance at all of negative interest rates.

The BK model has no analytic tractability, but has the advantage that interest rates are always positive. The other advantage is that traders naturally think in terms of $\sigma's$ arising from a lognormal model rather than $\sigma's$ arising from a normal model.

There are grave concerns around the feasibility of the positive rates implied by the CIR++ model when calibrating it to market prices.

Fortunately, South African firms do not suffer the fate that their developed counterparts face with respect to choosing a satisfactory model, as South African interest rates are relatively high compared to developed markets[24]. Most South African banks implement the Hull-White model for pricing interest rate derivatives, because of its analytic tractability.

[24]Developed markets usually have relatively low interest rates, as a result a normal model is unsatisfactory because, when the initial short rate is low, the probability of negative interest rates in the future is no longer negligible. The lognormal model is unsatisfactory because the volatility of rates is usually much greater when rates are low than when they high.

4 Data

The data used to carry out the analysis of the models was provided by Standard Bank[25]. The data presented below is annual NACS[26] forward rates and a surface of ATM swaption volatilities[27], from February 2010.

Tenor	Forward Rate
1y	0.07055716
2y	0.07394297
3y	0.07929317
4y	0.08332815
5y	0.08631804
6y	0.08837278
7y	0.08975166
8y	0.09074327
9y	0.09130995
10y	0.091308809

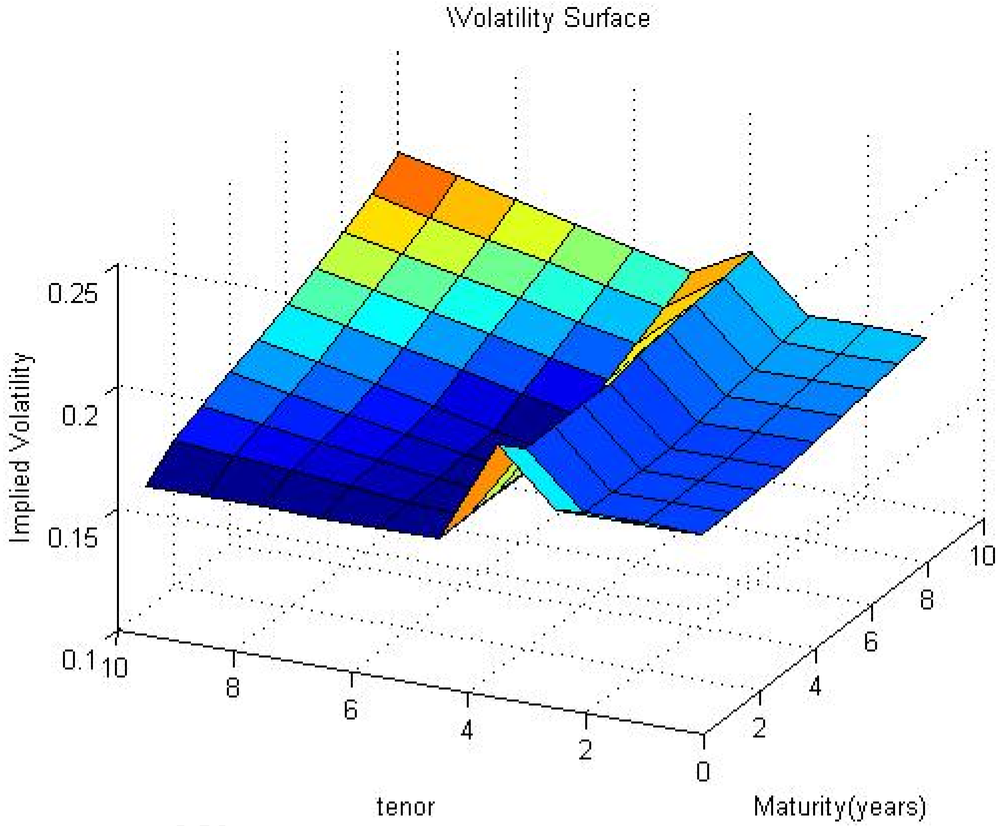
		Maturity									
	ATM Vols	1y	2y	3y	4y	5y	6y	7y	8y	9y	10y
Tenor	1y	0.1700	0.1600	0.1600	0.1600	0.1600	0.1620	0.1640	0.1660	0.1680	0.1700
	2y	0.1700	0.1600	0.1600	0.1600	0.1600	0.1620	0.1640	0.1660	0.1680	0.1700
	3y	0.1700	0.1600	0.1600	0.1600	0.1600	0.1620	0.1640	0.1660	0.1680	0.1700
	4y	0.1925	0.1825	0.1825	0.1825	0.1825	0.1845	0.1865	0.1885	0.1905	0.1925
	5y	0.1500	0.1500	0.1500	0.1500	0.1500	0.1560	0.1620	0.1680	0.1740	0.1800
	6y	0.1500	0.1520	0.1530	0.1540	0.1550	0.1610	0.1670	0.1730	0.1790	0.1856
	7y	0.1500	0.1540	0.1560	0.1580	0.1600	0.1660	0.1720	0.1780	0.1840	0.1912
	8y	0.1500	0.1560	0.1590	0.1620	0.1650	0.1710	0.1770	0.1830	0.1890	0.1968
	9y	0.1500	0.1580	0.1620	0.1660	0.1700	0.1760	0.1820	0.1880	0.1940	0.2024
	10y	0.1500	0.1600	0.1650	0.1700	0.1750	0.1810	0.1870	0.1930	0.1990	0.2080

[25] South Africa

[26] Nominal Annual Compounded Semi-Annual

[27] The data presented above is a subset of the data provided. The tenor extended from 3months to 36 years and the expiry ranged from 1day to 12 years.

The volatility surface obtained from the data, is rather flat for short expiries, with a rather pronounced kink around the four to five year tenor period. The data is asynchronous. As the tenor period increases beyond the five year tenor the volatilities begins to assume a smoother shape.



Using the data provided, ATM European swaption prices were derived[28].

[28]From Black's Swaption formula [5]

5 Calibration

Before we price a Bermudan swaption, we need to find the parameters that provide the correct European swaption price so as to avoid any arbitrage opportunities.

5.1 Calibrating the parameters

To calculate the price of the Bermudan swaption, we need to decide on the values of the model parameters. Bermudan swaptions are typically hedged with underlying co-terminal European swaptions. It is thus desirable to calibrate the model to these market instruments so as to obtain consistency and allow for appropriate risk management. Here, we review a calibration routine presented in Brigo and Mercurio for this purpose.

5.1.1 Calibration

In the models presented in this paper there are three common unknown parameters, $\vartheta(t), a, \sigma$.[29]It is apparent from the equations presented, that the $\vartheta(t)$ function is dependant on unknown parameters a and σ. In turn, we would ideally like to determine these two unknown parameters, such that the resulting prices from our model, matches the market price for European Swaptions. Accordingly, these unknown parameters a and σ are determined as follows.

5.1.2 Estimating the mean reversion parameter (a)

The mean-reversion parameter (a) has been estimated from the regression of historical data of interest rates. The data used for the regression is annual forward rates.

Here we present the basic idea behind the estimation procedure used. Considering the Hull White (extended Vasicek) model, the continuous time representation of the short rate process is

$$dr_t = [\nu_t - ar_t]dt + \sigma dW_t$$

The discrete-time version of this process is

$$r_{t+1} - r_t = [\nu_t - ar_t] + \varepsilon_{t+1}$$

[29]Hull-White (extended Vasicek) model, by solving the theoretical value formula of the forward rate with respect to the function $\vartheta(\cdot)$ and substituting the observed forward rate curve at current time 0, $f^M(0,T)$, we have

$$\vartheta^M(T) = \frac{d}{dT}\left[f^M(0,T) + \frac{\sigma^2}{2a^2}\left(e^{-aT}-1\right)^2\right] - a\left[f^M(0,T) + \frac{\sigma^2}{2a^2}\left(e^{-aT}-1\right)^2\right]$$

$$r_{t+1} = \nu_t + (1-a)r_t + \varepsilon_{t+1} \tag{15}$$

where $\varepsilon_{t+1}$is a drawing from a normal distribution. Thus, (15) represents an AR(1) process.[30]

The coefficient,$(1-a)$, in the AR(1) process is attained using an ordinary least squares (OLS) estimate, $\hat{\beta}$, as follows

$$1 - a = \beta = \frac{\rho\sigma(\hat{r_{t+1}})\sigma(r_t)}{\sigma^2(r_t)} = \rho$$

$$a = 1 - \rho$$

where ρ is the correlation coefficient between $r_{t+1}$and r_t. A similar process is performed for the CIR++ model. For the BK model we perform this regression using time series of ln(r), whereas before r, is a 1 year interest rate.

5.1.3 Parameterization of σ

A common financial practice is to calibrate the interest rate model using the instruments that are as similar as possible to the instruments being valued and hedged, rather than attempting to fit the models to all available market data. Assume that the valuation models for the Bermudan swaption are calibrated to the so-called diagonal of European options. This means that the variance of the variables is fully determined. In this study the problem at hand is to price and hedge 10 x 1 Bermudan swaptions. For this 10 x 1 Bermudan swaption the most relevant calibration instruments are the 1 x 10, 2 x 9, 3 x 8, . . ., 10 x 1 co-terminal European swaptions.[31] The intuition behind this strategy is that, when the model is used with the parameters that minimise the pricing error of these individual instruments, the correct price would result for any related instrument. Thus, these 10 European swaptions are used for calibrating the models for pricing 10 x 1 Bermudan swaption.

The first stage of a calibration is to choose a 'goodness-of-fit' measure. The models are calibrated by minimising the sum of squared percentage pricing errors between the model and the market prices of the co-terminal European swaptions. i.e. the goodness-of-fit measure is

[30] An autoregressive (AR) process is one, where the current values of a variable depends only upon the values that variable took in previous periods plus an error term [5]. A process $y_t$is autoregressive of order p if

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \varepsilon_t,$$
$$\varepsilon_t \sim N(0, \sigma^2).$$

[31] A n x m swaption is an n-year European option to enter into a swap lasting for m years after option maturity.

$$\min_{i=1}^{n}\left(\frac{P_{i,n,model}}{P_{i,n,market}}-1\right)^2$$

where $P_{i,n,market}$ is the market price and $P_{i,n,model}$ is the model generated price of the i x (n-i) European swaptions. In order to perform this minimisation technique we have utilised MatLabs built in minimization function *lsqnonlin*, which solves nonlinear least-squares problems. lsqnonlin starts at some predefined point x_0 and finds a minimum of the sum squares of the defined function[32].

5.1.4 Calibration Results

Calibration yielded the following results:

	Hull White	Black-Karasinski	CIR++
a	-0.2146	-0.1697	0.0020
σ	0.1225	0.8846	0.0204
θ	-	-	0.4458

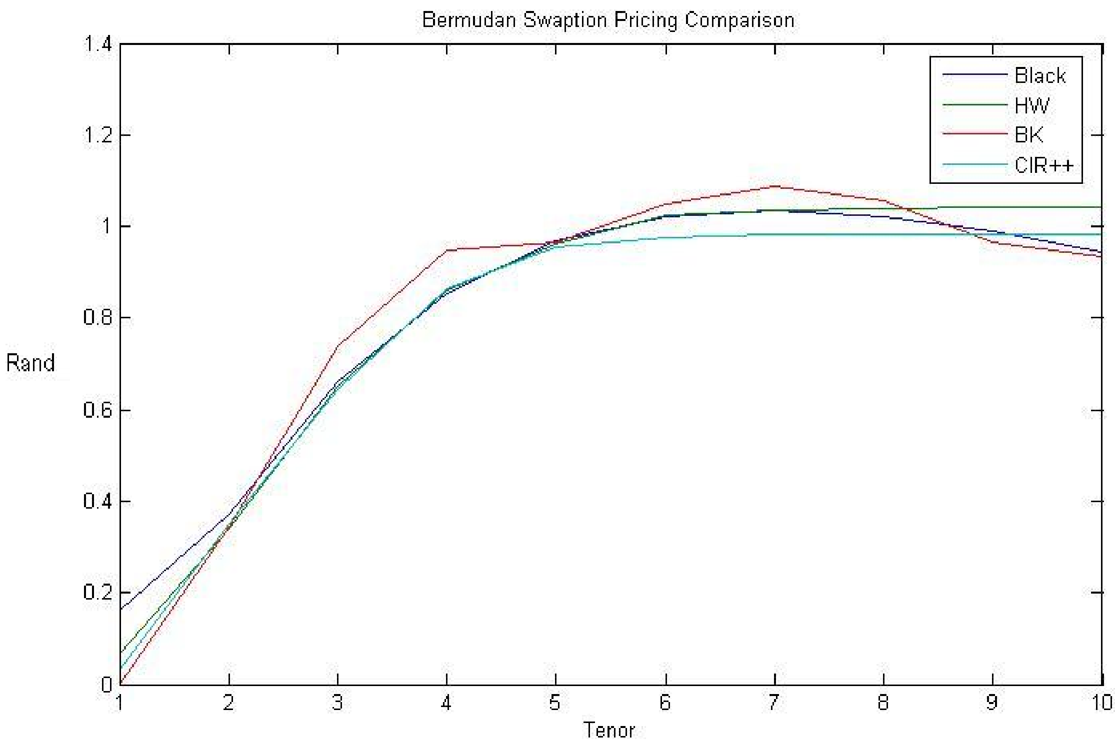
[32] lsqnonlin also defines a set of lower and upper bounds on the design variables in x, so the solution is always within a range.

6 Pricing Bermudan Swaptions

This paper implemented the recombining trinomial tree method for the pricing of Bermudans swaptions. As a Bermudan swaption is an exotic interest rate derivative product, there is not a market quoted price for it. We compare the price calculated from HW, BK, CIR++. The results are summarized below

Black	Hull-White	Black-Karasinski	CIR++
0.1620	0.0669	0	0.0307
0.3721	0.3409	0.3452	0.3526
0.6559	0.6508	0.7387	0.6427
0.8522	0.8601	0.9484	0.8651
0.9656	0.9622	0.9642	0.9542
1.0213	1.0256	1.0482	0.9774
1.0335	1.0366	1.0884	0.9822
1.0208	1.0395	1.0574	0.9831
0.9897	1.0408	0.9638	0.9833
0.9439	1.0414	0.9346	0.9835

A graphical representation



We price the same Bermudan swaption on the date of holding the option.

As we can see from the graph and tables, in most dates, the prices given by three models are relatively similar. Thus, the benefit of pricing accuracy, is dependent on the tenor of the swaption and tractability of the model.

7 Hedging Bermudan Swaption

Term structure changes adversely affect the value of any interest rate based asset or liability. Thus, financial institutions with large mortgage books protect their fixed income securities from unfavourable term structure movements by hedging their securities with Bermudan Swaptions. This proves to be a major strain on any financial institution and for the ALM group in particular, as inefficient hedging strategies can result in a large cost burden for these institutions. In order to protect the banks liabilities from possible future interest rate changes, one first needs to generate realistic scenarios and then analyse how these scenarios can be neutralized. Thus, the two important issues that need to be addressed by any interest rate risk management strategy are: (i) how to perturb the term structure to imitate possible term structure movements (ii) how to immunise the portfolio against these movements [5]. The following sections will address these issues. The literature for the choice of methods used to address these issues is presented in [5].

7.1 Perturbing the term structure

ALM departments would ideally like to determine how parallel perturbations affect the estimated value of a portfolio, so as to hedge themselves against such movements. It has been established in mature markets like the UK, that the three most commonly observed term structure shifts[33]can capture up to 98.4% of the of the yield curve variation [5]. Hedging against these factors would lead to a more stable portfolio and a superior hedging performance.[34]The perturbation in this study is achieved in a rather simplistic yet highly accurate manner. The perturbation is performed by shifting the 3 year zero rate up by 1 basis point and thus we are able to determine the new notional based on this new rate. [35]

7.2 Selecting hedge instruments and Constructing a Delta Hedged Portfolio

The key to a successful hedging strategy is to select the appropriate hedging instruments. The hedging strategies that receive the most attention is the

[33]Parallel Shift is when the entire curve moves up and down by the same amount; Tilt, also known as a slope shift , in which short yields fall and long yields rise (or vice versa); Curvature shifts in which short and long yields rise while mid-range yields fall (or vice versa)[5].

[34]It is ideal to perform PCA on annual changes of the forward rates and used scores of the first three priciple components for estimating the shifts by which the forward rate curve has been perturbed. Thus, creating a more realistic term structure shift. The description of how to perfrom PCA for estimating the term structure shifts is provided in [5].

[35]Recommanded by Standard Bank

factor and bucket hedging strategies.[36] This study implements the bucket hedging strategy because Bermudan Swaptions are dependent on the entire yield curve, not just a single payment date as swaps. In this study we have used two swaps of maturities 3 and 5 years as hedge instruments. The choice of swaps as hedge instruments is more in line with the general practitioners practice of hedging instruments for Bermudan Swaptions.[37]

This paper has implemented a delta hedging[38] process for hedging a bermudan swaption. Delta is a key measure for hedging, as delta measures the exposure of a derivative to changes in the value of the underlying, the overall value of a portfolio remains unchanged for small changes in the price of its underlying instrument. A delta hedged portfolio is established by buying or selling an amount of the underlier that corresponds to the delta of the portfolio.

We are able to estimate the risk sensitivity (delta) of a fixed income security by perturbing the entire initial term structure, up by some factor $\varepsilon^{+}$and down by the same factor ε^{-}.i.e.

$$\Delta = \frac{V(\varepsilon^{+}) - V(\varepsilon^{-})}{2\varepsilon} \tag{16}$$

where $V(\varepsilon^{+})$ is the value of the derivative calculated after the initial term structure has been perturbed upwards by ε and $V(\varepsilon^{-})$ is the value of the derivative after the initial term structure has been perturbed downwards by ε. For each factor, we have bumped the forward rate curve both up and down to estimate the sensitivity (delta) of the Bermudan swaption. We are also able to determine the two swaps with respect to the three factors as follows

$$\Delta_k^B = \frac{dB}{dP_k} = \frac{B_k^+ - B_k^-}{P_k^+ - P_k^-} = \frac{B_k^+ - B_k^-}{2\Delta P_k}$$

$$\Delta_k^{S_3} = \frac{dS_3}{dP_k} = \frac{(S_3)_k^+ - B_k^-}{P_k^+ - P_k^-} = \frac{(S_3)_k^+ - B_k^-}{2\Delta P_k}$$

$$\Delta_k^{S_5} = \frac{dS_5}{dP_k} = \frac{(S_5)_k^+ - B_k^-}{P_k^+ - P_k^-} = \frac{(S_5)_k^+ - B_k^-}{2\Delta P_k}$$

for k = 1,2,3, and $(\cdot)_k^+$and $(\cdot)_k^-$respectively are the prices of derivatives after the initial forward curve has been bumped up and down by the k^{th}factor.

---

[36] Factor Hedging - The number of different hedging instruments used to hedge any derivative is dependant on the number of factors in the model.

Bucket Hedging - The number of hedge instruments is equal to the number of total payoffs provided by the instrument.

[37] Alternatively we can use discount bonds to hedge a Bermudan Swaption.

[38] Delta Hedging is the process of keeping the delta of a portfolio equal to or as close as possible to zero.

We then constructed a portfolio, consisting of one Bermudan swaption, x_5 units of a 5-year swap and x_3 units of a 3-year swap. The total mismatch of this portfolio w.r.t the k^{th}factor (ε_k) is

$$\varepsilon_k = \Delta_k^B - x_5\Delta_k^{S_5} - x_3\Delta_k^{S_3} \tag{17}$$

where x_5 and x_3 are not whole numbers. Thus, we are able to obtain the hedge ratios, x_5 and x_3 by minimising the delta-mismatch of the portfolio w.r.t the first three PCA factors.

$$min^3_{x_5,x_3 k=1}\varepsilon_k^2$$

where ε_k is given by equation above.

The results from delta hedging are presented below:

Year	Hull-White delta
1y	0.02
2y	0.37
3y	0.15
4y	0.16
5y	-0.28
6y	-0.22
7y	-0.19
8y	-0.19
9y	-0.19
10y	-0.18

The Black-Karasinski and CIR++ models yielded the same parameter inputs, pre-perturbation and post-perturbation. Thus, we were unable to compare the hedging performance between the various one-factor interest rate models.

8 Conclusion

The objective of this paper was to provide a general procedure of implementing a number of short-rate models used to price and hedge Bermudan swaptions, as well as provide an empirical analysis and comparison of three interest rate term structure models from an ALM perspective.

The models presented in this paper all provide varying degrees of accuracy in terms of pricing Bermudan swaptions. There are both advantages and disadvantages in the implementation of normal and lognormal models. The main advantage of using a normal model, like the Hull-White model, is its simplicity and the analytical tractability, amongst other interest rate models for pricing and hedging Bermudan swaptions. The drawback of negative interest rates are overcome by the Black-Karasinski and CIR++ models, however the models are no longer analytically tractable. It was rather unfortunate that the calibration of the Black-Karasinski and CIR++ model was unable to yield different parameter inputs, as this would have provided us insight into the hedging differences between the models presented.

9 Appendix

Definition 1. Zero Coupon Bond

A Zero Coupon Bond with maturity date T, is a contract which guarantees the holder 1 unit of money to be paid at the maturity date T. The price at time t of a bond with maturity date T is denoted by $P(t,T)$.

Definition 2. Bank Account Process (Discount Factor)

The bank account process is defined by

$$B_t = \exp \int_0^t r_s ds,$$

i.e.

$$dB_t = r_t B_t dt$$

$$B_0 = 1.$$

9.1 Swaps

> A swap is an agreement by two counterparties to exchange a predetermined series of cash flows, decided by a pre-agreed formula, over time. (Ouwehand)

Definition 3. Interest rate swap.

We are given a number of payment dates T_i, $i = \alpha + 1, \ . \ . \ . \ , \beta$ (called the *tenorstructure*) and a nominal value N (the *notional*).We set $\tau_i := T_i - T_{i-1}$in the following. At every time-T_i instant the fixed leg of an interest rate(payer) swap of value

$$N\tau_i S,$$

where S is a pre-specified swap rate is exchanged for the floating leg with given tenor structure of value

$$N\tau_i S(T_{i-1}, T_i).$$

It should be noted that the rate $(L(T_{i-1}, T_i))$[39] to be applied for the floating leg at time T_i is fixed for that period at the reset time T_{i-1}. Note that if the fixed leg is paid and the floating leg is received the interest rate

[39]The market LIBOR rates are simply-compounded rates, which motivates why we denote L as such. LIBOR rates are typically linked to zero-coupon-bond prices by the Actual/360 day-count convention for computing τ (t, T).

swap is called a payer swap, in contrast when the floating leg is paid and the fixed leg is received the contract is called a receiver swap.

The time-t discounted payoff of a payer swap can be expressed as

$$N \sum_{i=\alpha+1}^{\beta} P(t, T_i)\tau_i(L(T_{i-1}, T_i) - K)^+.$$

The corresponding payoff for the receiver swap is obtained by changing the sign of the payer payoff, i.e. by multiplying by -1.

The swap rate $S_{\alpha\beta}(t)$ that makes the payoff fair at time t, that is, equal to zero, can easily be obtained from the payoff definition and is given by

$$S_{\alpha\beta}(t) = \frac{P(t, T_\alpha) - P(t, T_\beta)}{\sum_{i=\alpha+1}^{\beta} \tau_i P(t, T_i)}. \tag{18}$$

Definition 4. (European) payer swaption.

We are given a tenor structure T_i, $i = \alpha + 1, \ldots, \beta$, a notional N, a swap rate K and a time t. The payer swaption is an option to enter into a swap at time $T_\alpha$with swap rate K. It thus has time-t discounted payoff

$$P(t, T_\alpha)N \left(\sum_{i=\alpha+1}^{\beta} P(T_\alpha, T_i)\tau_i(L(T_{i-1}, T_i) - K) \right)^+. \tag{19}$$

Expressed in terms of the swap rate defined in (3), the time-t discounted payoff of the swaption is

$$P(t, T_\alpha)N(S_{\alpha\beta}(T_\alpha) - K)^+ \sum_{i=\alpha+1}^{\beta} \tau_i P(T_\alpha, T_i), \tag{20}$$

this is, it can be viewed as a call option on the swap rate.

A receiver swaption can be defined in a similar manner, i.e. multiply the payoff by -1. The swaption has limited optionality, namely the choice to enter the swap at time T_α. In contrast, a Bermudan swaption offers the possibility to enter the swap at any of the dates T_i, $i = \alpha, \ldots, \beta - 1$, for the remainder of the swap's lifetime.

Definition 5. Bermudan payer swaption.

We are given a tenor structure T_i, $i = \alpha + 1, \ldots, \beta$, a notional N,a swap rate K and a time t. The Bermudan payer swaption is an option to enter at any time T_i, $i = \alpha, \ldots, \beta - 1$, into a payer swap with swap rate K maturing at time T_β. At any time T_k, k$\epsilon\{\alpha, \ldots, \beta - 1\}$, the holder of the contract has the right to receive

$$N \left(\sum_{i=k+1}^{\beta} P(T_k, T_i)\tau_i(L(T_\alpha, T_i) - K) \right)^+,$$

provided that the option has not been exercised before.

Definition 6. Ornstein-Uhlenbeck Process

An Ornstein-Uhlenbeck Process is the unique solution of the following equation:

$$dX = -cX_t dt + \sigma dW_t$$

$$X_0 = x$$

where σ and c are constant.

It can be written explicitly as follows:

$$X_t = xe^{-ct} + \sigma e^{-ct} \int_0^t e^{cs} dW_s.$$

9.2 Black's Formulas

Black's model has long been the industry standard model used by traders to price a variety of European style options, including interest rate options, such as caps, floors and swaptions. From these derivatives, one can ascertain the views of investors toward future changes in the level of the yield curve. Thus, Black's model is essentially a minor variation on the Black-Scholes formula, as will be presented. However, the suitability and adequacy of Black's model has often been questioned by academics, particularly in the area of interest rate options. However, these questions can be resolved by the use of the T-forward measure.

It is market practice to value swaptions with a Black-like formula. Start with the swaption payoff expression

$$P(t, T_\alpha) N (S_{\alpha\beta}(T_\alpha) - K)^+ \sum_{i=\alpha+1}^{\beta} \tau_i P(T_\alpha, T_i),$$

in terms of the swap rate $S_{\alpha\beta}$, treating it as the underlying asset. It should be noted that we have to choose a single numeraire for the entire expression (2), i.e. use the expression $\sum_{i=\alpha+1}^{\beta} \tau_i P(\cdot, T_i)$ as a numeraire. The time-t swaption price can be expressed as

$$V_{swap}(t, r; K, N) = \sum_{i=\alpha+1}^{\beta} \tau_i P(T_\alpha, T_i) E^i \left[P(t, T_\alpha) N (S_{\alpha\beta}(T_\alpha) - K)^+ \right]$$

where E^i is the expectation taken with respect to the measure $\mathbb{Q}^i$ under which $P(\cdot, T_i)$ is discounted tradable asset that is martingale. Here, $S_{\alpha\beta}$ is a tradable asset because of definition (1)

$$V_{swap}^{Black}(t, S_{\alpha\beta}; K, N, \sigma) = Notional(N(d_1) - KN(d_2)) \sum_{i=\alpha+1}^{\beta} \tau_i P(T_\alpha, T_i)$$

where,

$$d_1 = \frac{\ln \frac{S_{\alpha\beta}}{K} + \frac{1}{2}\sigma^2}{\sigma}$$

$$d_2 = \frac{\ln \frac{S_{\alpha\beta}}{K} - \frac{1}{2}\sigma^2}{\sigma}.$$

It should be noted that $N(\cdot)$ denotes the cumulative normal distribution andσdenotes the root of the swap rates variance (volatility of $r(\cdot)$) accumulated on the time interval $[t, T_\alpha]$.

9.3 Approximating the Trinomial Lattice

This subsection of the appendix will approximate the diffusion process for one-factor short-rate models. In the one-factor models the tree is constructed by imposing the local mean and variances at each node are equal to the continuous process. Ensuring the geometry of the tree has positivity of the branching probabilities.

9.3.1 Approximating a one-factor diffusion

Consider the diffusion process X

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$$

where μ and σ are smooth scalar real functions and W is a scalar standard Brownian motion.

Define a finit set of times $0 = t_0 < t_1 < \cdots < t_n = T$ and $\triangle t_i = t_{i+1} - t_i$. At each time t_i, define a finite number of equi-spaced states, with constant vertical step $\triangle x_i$. Set $x_{i,j} = j\triangle x_i$, where $\triangle x_i = \sqrt{3}V_{i-1}$.

Given a particular time t_i at node $x_{i,j}$. The tree evolves to three positions at time t_{i+1}, namely, $x_{i+1,k+1}, x_{i+1,k}$and $x_{i+1,k-1}$. The value of k determines the level of the process at time t_{i+1} from the j-th node. Choose k such that $x_{i+1,k}$ at t_{i+1} is as close as possible to the mean $(M_{i,j})$ of the continuous process x. This is explicitly calculated using,

$$h = \text{round}\left(\frac{M_{i,j}}{\triangle x_{i+1}}\right)$$

where k equals the closest integer to the real number h and

$$M_{i,j} = E\left[X(t_{i+1}) \mid X(t_i) = x_{i,j}\right] = x_{i,j}e^{-a\triangle t_i}$$

is the conditional mean of the process X at time t_{i+1}conditional on $X(t_i) = x_{i,j}$.

9.3.2 How to Obtain the Probabilities

The probabilities are required at each time step so that a backward iterative procedure can be used to price the option using the short rate tree. The probabilities are chosen to ensure that the expected value and volatility for the short rate reflected on the tree are consistent with the statistics implied by the x process. Firstly, define the conditional variance of the process X at time t_{i+1} conditional on $X(t_i) = x_{i,j}$ by

$$V_{i,j}^2 = Var\left[X(t_{i+1}) \mid X(t_i) = x_{i,j}\right] = \frac{\sigma^2}{2a}\left[1 - e^{-2a\triangle t_i}\right].$$

Let p_u, p_m, p_d denote the probability of moving from node $x_{i,j}$ at time t_i to node $x_{i+1,k+1}, x_{i+1,k}$ and $x_{i+1,k-1}$ at time t_{i+1}, respectively.

Since, $x_{i+1,k+1} = x_{i+1,k} + \triangle x_{i+1}$ and $x_{i+1,k-1} = x_{i+1,k} - \triangle x_{i+1}$ we obtain the following relationships,

$$M_{i,j} = p_u(x_{i+1,k} + \triangle x_{i+1}) + p_m(x_{i+1,k}) + p_d(x_{i+1,k} - \triangle x_{i+1})$$

$$V_{i,j}^2 + M_{i,j}^2 = p_u(x_{i+1,k} + \triangle x_{i+1})^2 + p_m(x_{i+1,k})^2 + p_d(x_{i+1,k} - \triangle x_{i+1})^2.$$

Let $M_{i,j} - x_{i+1,k} = \eta_{i,j,k}$, substituting this into the equations above we obtain,

$$\eta_{i,j,k} = (p_u - p_d)\triangle x_{i+1}$$

$$V_{i,j}^2 + \eta_{i,j,k}^2 = (p_u + p_d)\triangle x_{i+1}^2.$$

Thus, we have two equations and two unknowns, p_u and p_d respectively. Solving these equations simultaneously,

$$p_u = \frac{V_{i,j}^2}{2\triangle x_{i+1}^2} + \frac{\eta_{i,j,k}^2}{2\triangle x_{i+1}^2} + \frac{\eta_{i,j,k}}{2\triangle x_{i+1}}$$

$$p_d = \frac{V_{i,j}^2}{2\triangle x_{i+1}^2} + \frac{\eta_{i,j,k}^2}{2\triangle x_{i+1}^2} - \frac{\eta_{i,j,k}}{2\triangle x_{i+1}}.$$

Using the law of total probability and the equations for p_u and p_d we obtain the equation for p_m which is given by,

$$p_m = 1 - \frac{V_{i,j}^2}{\triangle x_{i+1}^2} - \frac{\eta_{i,j,k}^2}{\triangle x_{i+1}^2}.$$

The equations derived, ensure the positivity of the probabilities. To ensure that the probabilities remain positive we exploite an available degree of freedom and make the variance, $V_{i,j}^2$, dependent only on time and not on the state. Thus, $V_{i,j}^2 = V_i^2 \forall j$. This simplifies the equations of our probabilities,

$$p_u = \frac{1}{6} + \frac{\eta_{i,j,k}^2}{6V_i^2} + \frac{\eta_{i,j,k}}{2V_i\sqrt{3}}$$

$$p_m = \frac{2}{3} - \frac{\eta_{i,j,k}^2}{3V_i^2}$$

$$p_d = \frac{1}{6} + \frac{\eta_{i,j,k}^2}{6V_i^2} - \frac{\eta_{i,j,k}}{2V_i\sqrt{3}}.$$

We can calculate the level k using the equation above, this implies
$| \eta_{i,j,k} | \leq V_i \frac{\sqrt{3}}{2}$. This condition guarantees the positivity of our probabilities.

9.4 Problems with Monte Carlo Simulation

Monte Carlo simulation works through forward propagation in time of the key variables, by simulating their transition density between dates where the key variables history matters to the final payoff. Monte Carlo is thus ideally suited to 'travel forward in time'. As a result, Monte Carlo simulation encounters a few problems when considering early exercise options. Since with Monte Carlo we propagate trajectories forward in time, we have no means to know whether at a certain point in time it is optimal to continue or to exercise. Therefore, standard Monte Carlo cannot be used for products involving early exercise. However, this problem can be overcome by the Least squares Monte Carlo (LSMC) Approach. But this also has encountered a number of problems when calibrating the model. Thus, the preferred method of pricing early exercise swaptions or any other interest rate derivative is by implementing the tree approach.

10 MatLab Code

10.1 Hull-White extended Vasicek

```
function [BermudanSwaptionPrice, k, NodeCounter, ProbUp, ProbMid, ProbDown, dt] =
    HullWhiteExtendedVasicek(Sigma, a, TimeSteps)
    % The is the algorithm that generates a trinomial lattice structure using the HW
Extended Vasiek Model
    load Data.m ;
    dt = Data(:,5)./365;
    x(1,1) = 0;
    ShortRate(1,1) = Data(2,4);
    NodeCounter(1) = 1;
    Q(1,1) = 1;
    Maturity = (TimeSteps - 1)/4;
    alpha(1) = -log(Data(2,3))/(Data(2,2)/365);
    Var = ((Sigma^2)/(2*a))*(1-exp(-2*a*dt(:)));
    k(1,1) = 0;

    % PROCEDURE 1 - Generating the x process
    % The first stage in building a tree for the HW extended Vasicek model
    % is to construct a tree for the variable x that follows a stochastic process.
    for i = 1:TimeSteps-2
      NodeCounterCurrent = 3;
      NodeIndex = (NodeCounter(i)-1)/2:-1:-(NodeCounter(i)-1)/2;
      Q(end,end+1) = 0;
      M = x(:)*exp(-a*dt(i));
      k(1,i+1) = round(M(1)/sqrt(3*Var(i)));
      x = 0;
      x(1) = (k(1,i+1)+1)*sqrt(3*Var(i));
      x(2) = k(1,i+1)*sqrt(3*Var(i));
      x(3) = (k(1,i+1)-1)*sqrt(3*Var(i));
      eta(1,i) = M(1)-x(2);
    % Determine the probabilities for the first set of nodes
      ProbUp(1,i) = 1/6 + eta(1,i)^2/(6*Var(i)) + eta(1,i)/(2*sqrt(3*Var(i)));
      ProbMid(1,i) = 2/3 - eta(1,i)^2/(3*Var(i));
      ProbDown(1,i) = 1/6 + eta(1,i)^2/(6*Var(i)) - eta(1,i)/(2*sqrt(3*Var(i)));
    if NodeCounter(i) == 1
      Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-(alpha(i))*dt(i));
      Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-(alpha(i))*dt(i));
      Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-(alpha(i))*dt(i));
    else
      Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));
      Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));
      Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));
    end
    if NodeCounter(i)>1
    %Constructing the rest of the tree
    for j = 2:NodeCounter(i)
        k(j,i+1) = round(M(j)/sqrt(3*Var(i)));
        xcalc(1) = (k(j,i+1)+1)*sqrt(3*Var(i));
        xcalc(2) = (k(j,i+1))*sqrt(3*Var(i));
        xcalc(3) = (k(j,i+1)-1)*sqrt(3*Var(i));
        eta(j,i) = M(j)-xcalc(2);
```

```
        ProbUp(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) + eta(j,i)/(2*sqrt(3*Var(i)));
        ProbMid(j,i) = 2/3 - eta(j,i)^2/(3*Var(i));
        ProbDown(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) - eta(j,i)/(2*sqrt(3*Var(i)));
        QCalc(1) = Q(j,i)*ProbUp(j,i)*exp(-(alpha(i) + NodeIndex(j)*sqrt(3*Var(i-1)))*dt(i));

        QCalc(2) = Q(j,i)*ProbMid(j,i)*exp(-(alpha(i) + NodeIndex(j)*sqrt(3*Var(i-1)))*dt(i));

        QCalc(3) = Q(j,i)*ProbDown(j,i)*exp(-(alpha(i) + NodeIndex(j)*
        sqrt(3*Var(i-1)))*dt(i));
    if xcalc(1) == x(NodeCounterCurrent-2);
        Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
        Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
        Q(NodeCounterCurrent,i+1) = Q(NodeCounterCurrent,i+1) + QCalc(3);
    elseif
        xcalc(1) == x(NodeCounterCurrent-1);
        NodeCounterCurrent = NodeCounterCurrent + 1;
        Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
        Q(NodeCounterCurrent,i+1) = QCalc(3);
    else
        xcalc(1) == x(NodeCounterCurrent);
        NodeCounterCurrent = NodeCounterCurrent + 2;
        x(NodeCounterCurrent-1) = xcalc(2);
        x(NodeCounterCurrent) = xcalc(3);
        Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
        Q(NodeCounterCurrent-1,i+1) = QCalc(2);
        Q(NodeCounterCurrent,i+1) = QCalc(3);
        end
      end
    end
      NodeCurrentIndex = (NodeCounterCurrent-1)/2:-1:-(NodeCounterCurrent-1)/2;
      function1 = 0;
      function1=Q(1:NodeCounterCurrent,i+1).*exp(-NodeCurrentIndex(:).*sqrt(3*Var(i))*dt(i+1));
      alpha(i+1) = (1/dt(i+1))*log(sum(function1)/Data(i+2,3));
    % PROCEDURE 2 - Determining the short rate
    % The second stage, is to convert the tree from x into a tree for r.
    % This is accomplished by displacing the x nodes for nodes with r
      ShortRateCurrent = 0;
      ShortRateCurrent = x(:)+alpha(i+1);
      NoOfZeros = length(ShortRate) - length(ShortRateCurrent);
      ShortRate(1:end - NoOfZeros,i+1) = ShortRateCurrent;
    NodeCounter(i+1) = NodeCounterCurrent;
    end
    % disp(ShortRate);
    % disp(TimeSteps);
    NoOfRows = NodeCounter';
    K = 0.08;
    Notional = 1000;
    tau = 0:(91/365):((TimeSteps-1)/4);
    t = ((TimeSteps-1)/4):(-91/365):0;
    %Bermudan Swaption Pricing
    % Initially determine the Bond price (Discount Factor), the Market Forward Rate
    for i = 1:NoOfRows(TimeSteps-1)
      TerminalMarketBond(i) = exp(-MarketFwdRate(TimeSteps-1)*(Maturity));
```

```
    B(i,TimeSteps-2) = (1/a)*(1 - exp(-a*(tau(2))));
    A(i,TimeSteps-2)=log(TerminalMarketBond(i)/TerminalMarketBond(i))+...
    B(i,TimeSteps-2)*MarketFwdRate(TimeSteps-1) - (Sigma/4*a)*...
    (1 - exp(-2*a*tau(2)))*B(i,TimeSteps-2)^2;
    DiscountFactor(i,TimeSteps-2)=exp(A(i,TimeSteps-2)-B(i,TimeSteps-2)*...
    ShortRate(i,TimeSteps-2));
    SumDisFac(i) = (91/365)*sum(DiscountFactor(i,TimeSteps-2));
    SwapRate(i) = (1-DiscountFactor(i,TimeSteps-2))/SumDisFac(i);
    BermudanPrice(i,TimeSteps-2) = max(SumDisFac(i)*(SwapRate(i)-K),0);
end
for j = (TimeSteps-3):-1:1
for i = 1:NoOfRows(j)
    TerminalMarketBond = exp(-MarketFwdRate(TimeSteps-1)*(Maturity));
    MarketBond(j) = exp(-MarketFwdRate(j)*t(j));
    B(i,j) = (1/a)*(1 - exp(-a*(t(j))));
    A(i,j) = log(TerminalMarketBond/MarketBond(j))+...
    B(i,j)*MarketFwdRate(j) -(Sigma/(4*a))*(1 - exp(-2*a*tau(j)))*B(i,j)^2;
    DiscountFactor(i,j) = exp(A(i,j)-B(i,j)*ShortRate(i,j));
    SumDisFac(i) = sum((91/365)*DiscountFactor(i,1:(TimeSteps-2)));
    SwapRate(i) = (1 - DiscountFactor(i,j))/SumDisFac(i);
if NoOfRows(j) == NoOfRows(j+1) && (i > (NoOfRows(j)-2))
    CC(i,j) = exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*...
    BermudanPrice(i-2,j+1) +ProbMid(i,j)*BermudanPrice(i-1,j+1) +...
    ProbDown(i,j)*BermudanPrice(i,j+1));
else
    CC(i,j) = exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*...
    BermudanPrice(i,j+1) +ProbMid(i,j)*BermudanPrice(i+1,j+1) + ...
    ProbDown(i,j)*BermudanPrice(i+2,j+1));
end
    BermudanPrice(i,j) = max(SumDisFac(i)*(SwapRate(i)-K),CC(i,j));
  end
end
% disp(DiscountFactor);
% disp(BermudanPrice);
BermudanSwaptionPrice = BermudanPrice(1,1);
```

10.2 Black-Karasinski

```
function [BermudanSwaptionPrice, k, NodeCounter, ProbUp,
    ProbMid, ProbDown, dt] = BlackKarasinki (Sigma, a, TimeSteps)
    %The is the algorithm that generates a trinomial lattice structure using the Black
    Karasinki Model
    load Data.m ;
    dt = Data(:,5)./365;
    x(1,1) = 0;
    ShortRate(1,1) = Data(2,4);
    NodeCounter(1) = 1;
    alpha(1) = log(-log(Data(2,3))/(Data(2,2)/365));
    Sigmanorm = Sigma*Data(:,4); %Sigmanorm converts the lognormal volatility into a
normal volatility.
    Var = (Sigmanorm(:).^2./(2*a)).*(1-exp(-2*a*dt(:)));
    k(1,1) = 0;
```

```
% PROCEDURE 1 - Generating the x process
for i = 1:TimeSteps-2
    NodeCounterCurrent = 3;
  NodeIndex = (NodeCounter(i)-1)/2:-1:-(NodeCounter(i)-1)/2;
  Q(end,end+1) = 0;
  M = x(:)*exp(-a*dt(i));
  k(1,i+1) = round(M(1)/sqrt(3*Var(i)));
  x = 0;
  x(1) = (k(1,i+1)+1)*sqrt(3*Var(i));
  x(2) = (k(1,i+1))*sqrt(3*Var(i));
  x(3) = (k(1,i+1)-1)*sqrt(3*Var(i));
  eta(1,i) = M(1)-x(2);
  ProbUp(1,i) = 1/6+eta(1,i)^2/(6*Var(i))+eta(1,i)/(2*sqrt(3*Var(i)));
  ProbMid(1,i) = 2/3 - eta(1,i)^2/(3*Var(i));
  ProbDown(1,i) = 1/6 + eta(1,i)^2/(6*Var(i))-eta(1,i)/(2*sqrt(3*Var(i)));
if NodeCounter(i) == 1
  Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-exp(alpha(i))*dt(i));
  Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-exp(alpha(i))*dt(i));
  Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-exp(alpha(i))*dt(i));
else
  Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-exp(alpha(i)+...
  (NodeIndex(1))*sqrt(3*Var(i)))*dt(i));
  Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-exp(alpha(i)+...
  (NodeIndex(1))*sqrt(3*Var(i)))*dt(i));
  Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-exp(alpha(i)+...
  (NodeIndex(1))*sqrt(3*Var(i)))*dt(i));
end

if NodeCounter(i)>1
for j = 2:NodeCounter(i)
  k(j,i+1) = round(M(j)/sqrt(3*Var(i)));
  xcalc(1) = (k(j,i+1)+1)*sqrt(3*Var(i));
  xcalc(2) = (k(j,i+1))*sqrt(3*Var(i));
  xcalc(3) = (k(j,i+1)-1)*sqrt(3*Var(i));
  eta(j,i) = M(j)-xcalc(2);
  ProbUp(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) + eta(j,i)/(2*sqrt(3*Var(i)));
  ProbMid(j,i) = 2/3 - eta(j,i)^2/(3*Var(i));
  ProbDown(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) - eta(j,i)/(2*sqrt(3*Var(i)));
  QCalc(1) = Q(j,i)*ProbUp(j,i)*exp(-exp(alpha(i)+NodeIndex(j)*sqrt(3*Var(i)))*dt(i));
  QCalc(2) = Q(j,i)*ProbMid(j,i)*exp(-exp(alpha(i)+NodeIndex(j)*sqrt(3*Var(i)))*dt(i));
  QCalc(3) = Q(j,i)*ProbDown(j,i)*exp(-exp(alpha(i)+NodeIndex(j)*sqrt(3*Var(i)))*dt(i));
if xcalc(1) == x(NodeCounterCurrent-2);
  Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
  Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
  Q(NodeCounterCurrent,i+1) = Q(NodeCounterCurrent,i+1) + QCalc(3);
elseif
  xcalc(1) == x(NodeCounterCurrent-1);
  NodeCounterCurrent = NodeCounterCurrent + 1;
  x(NodeCounterCurrent) = xcalc(3);
  Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
  Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
  Q(NodeCounterCurrent,i+1) = QCalc(3);
else
  xcalc(1) == x(NodeCounterCurrent);
```

```
        NodeCounterCurrent = NodeCounterCurrent + 2;
        x(NodeCounterCurrent-1) = xcalc(2);
        x(NodeCounterCurrent) = xcalc(3);
        Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
        Q(NodeCounterCurrent-1,i+1) = QCalc(2);
        Q(NodeCounterCurrent,i+1) = QCalc(3);
            end
        end
    end

    alpha(i+1) = log(-log(Data(i+2,3))/(Data(i+2,2)/365));
    function1 = 0;
    function2 = 0;
    function3 = 0;
    error = 0.5;
    NodeCurrentIndex = (NodeCounterCurrent-1)/2:-1:-(NodeCounterCurrent-1)/2;

    while abs(error) > 0.000001;
    %Newton-Bailey
    % We use Newton-Bailey method since the first and second derivative is known
        function1 = Q(1:NodeCounterCurrent,i+1).* exp(-exp(alpha(i+1) +
        NodeCurrentIndex(:).*sqrt(3*Var(i)))*dt(i+1));
        function2 = function1(:).*exp(alpha(i+1) + NodeCurrentIndex(:).*sqrt(3*Var(i)))*dt(i+1);

    %First Derivative
        function3 = function2(:).*exp(alpha(i+1) + ... NodeCurrentIndex(:).*sqrt(3*Var(i)))*dt(i+1);
        SecondDerivative = -function3 + function2;
        functionOfAlpha = Data(i+2,3) - sum(function1);
        error = functionOfAlpha/(sum(function2) - ((functionOfAlpha*...
        sum(SecondDerivative)) /(2*sum(function2))));
        alpha(i+1) = alpha(i+1) - error;
    end

    % PROCEDURE 2 - Replacing the x nodes
    ShortRateCurrent = 0;
    ShortRateCurrent = exp(x(:)+alpha(i+1));
    NoOfZeros = length(ShortRate) - length(ShortRateCurrent);
    ShortRate(1:end - NoOfZeros,i+1) = ShortRateCurrent;
    NodeCounter(i+1) = NodeCounterCurrent;
    end
    %disp(ShortRate);
    % Define Variables for Swaptions

    K = 0.08;
    Notional = 1000;
    tau = 0:(91/365):((TimeSteps-2)/4);
    t = ((TimeSteps-3)/4):(-91/365):0;
    length(tau');
    NoOfRows = NodeCounter';

    % Determinig discount factors i.e. P(t,T)
    for j = (TimeSteps-2):-1:1
        for i = 1:NoOfRows(j)
            DiscountFactor(:,TimeSteps-1) = ones(NoOfRows(TimeSteps-1),1);
            DiscountFactor(i,j)=exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*...
            DiscountFactor(i,j+1)+ProbMid(i,j)*DiscountFactor(i+1,j+1) + ....
```

```
        ProbDown(i,j)*DiscountFactor(i+2,j+1));
      end
    end

    %disp(DiscountFactor);
    % Bermudan Swaption Prices % DiscountFactor(i,TimeSteps-1)
    for i = 1:NoOfRows(TimeSteps-2)
      SumDisFac(i) = (91/365)*sum(DiscountFactor(i,TimeSteps-1));
      SwapRate(i) = (1 - DiscountFactor(i,TimeSteps-2))/SumDisFac(i);
      BermudanPrice(i,TimeSteps-2) = max(SumDisFac(i)*(SwapRate(i)-K),0);
    end

    for j = (TimeSteps-3):-1:1
      for i = 1:NoOfRows(j)
        for l = TimeSteps-1:-1:j
          BondPrices(i,l) = DiscountFactor(i+1,l);
        end
        SumDisFac(i) = (91/365)*sum(BondPrices(i,:));
        SwapRate(i) = (1 - DiscountFactor(i,j))/SumDisFac(i);
        if NoOfRows(j) == NoOfRows(j+1) && (i > (NoOfRows(j)-2))
          CC(i,j)=exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*BermudanPrice(i-2,j+1)+....
          ProbMid(i,j)*BermudanPrice(i-1,j+1)+ ProbDown(i,j)*BermudanPrice(i,j+1));

        else
          CC(i,j)=exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*BermudanPrice(i,j+1)+...
          ProbMid(i,j)*BermudanPrice(i+1,j+1)...+ ProbDown(i,j)*BermudanPrice(i+2,j+1));
        end
        BermudanPrice(i,j) = max(SumDisFac(i)*(SwapRate(i)-K),CC(i,j));
      end
    end

    %disp(BermudanPrice);
    BermudanSwaptionPrice = BermudanPrice(1,1);
```

10.3 CIR++

```
function [BermudanSwaptionPrice, k, NodeCounter, ProbUp, ProbMid,...
    ProbDown, dt] = CIRtree(Sigma,a, theta, TimeSteps)
    %The is the algorithm that generates a trinomial lattice structure using the CIR++
Model

    load Data.m ;
    dt = Data(:,5)./365;
    ShortRate(1,1) = Data(2,4);
    NodeCounter(1) = 1;
    kappa = sqrt(a^2+2*Sigma^2);
    alpha(1) = 0.06843871 - 0.001;
    k(1,1) = 0;
    Var = (Sigma^2.*dt(:))./(4);
    x(1,1) = sqrt(ShortRate(1,1)-alpha(1));
    % ensure 2*k*kappa > Sigma^2

    % PROCEDURE 1 - Generating the x process
    % This process is generated along the lines presented in HW, however the x process
is slighty different.
    % Note: The Mean and Variance different
```

```
for i = 1:TimeSteps-2
  NodeCounterCurrent = 3;
  NodeIndex = (NodeCounter(i)-1)/2:-1:-(NodeCounter(i)-1)/2;
  Q(end,end+1) = 0;
  M = x(:) + (((((a.*theta)/2) - ((1/8).*Sigma^2))*(1./x(:))) - ((a/2)*x(:)))*dt(i);%
mean
  k(1,i+1) = round(M(1)/(sqrt(3*Var(i))));
  x = 0;
  x(1) = (k(1,i+1)+1)*(sqrt(3*Var(i)));
  x(2) = k(1,i+1)*(sqrt(3*Var(i)));
  x(3) = (k(1,i+1)-1)*(sqrt(3*Var(i)));
  eta(1,i) = M(1)-(x(2));
  ProbUp(1,i) = 1/6 + eta(1,i)^2/(6*Var(i)) + eta(1,i)/(2*sqrt(3*Var(i)));
  ProbMid(1,i) = 2/3 - eta(1,i)^2/(3*Var(i));
  ProbDown(1,i) = 1/6 + eta(1,i)^2/(6*Var(i)) - eta(1,i)/(2*sqrt(3*Var(i)));
if NodeCounter(i) == 1
  Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-(alpha(i))*dt(i));
  Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-(alpha(i))*dt(i));
  Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-(alpha(i))*dt(i));
else
  Q(1,i+1) = Q(1,i)*ProbUp(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));
  Q(2,i+1) = Q(1,i)*ProbMid(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));

  Q(3,i+1) = Q(1,i)*ProbDown(1,i)*exp(-(alpha(i) + NodeIndex(1)*sqrt(3*Var(i-1)))*dt(i));
end

if NodeCounter(i)>1
for j = 2:NodeCounter(i)
  k(j,i+1) = round(M(j)/sqrt(3*Var(i)));
  xcalc(1) = (k(j,i+1)+1)*(sqrt(3*Var(i)));
  xcalc(2) = (k(j,i+1))*(sqrt(3*Var(i)));
  xcalc(3) = (k(j,i+1)-1)*(sqrt(3*Var(i)));
  eta(j,i) = M(j)-(xcalc(2));
  ProbUp(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) + ... eta(j,i)/(2*sqrt(3*Var(i)));
  ProbMid(j,i) = 2/3 - eta(j,i)^2/(3*Var(i));
  ProbDown(j,i) = 1/6 + eta(j,i)^2/(6*Var(i)) - ... eta(j,i)/(2*sqrt(3*Var(i)));
  QCalc(1) = Q(j,i)*ProbUp(j,i)*exp(-(alpha(i)+ NodeIndex(j)*sqrt(3*Var(i-1)))*dt(i));
  QCalc(2) = Q(j,i)*ProbMid(j,i)*exp(-(alpha(i)+ NodeIndex(j)*sqrt(3*Var(i-1)))*dt(i));
  QCalc(3) = Q(j,i)*ProbDown(j,i)*exp(-(alpha(i)+ NodeIndex(j)*sqrt(3*Var(i-1)))*dt(i));
if xcalc(1) == x(NodeCounterCurrent-2);
  Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
  Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
  Q(NodeCounterCurrent,i+1) = Q(NodeCounterCurrent,i+1) + QCalc(3);
elseif
  xcalc(1) == x(NodeCounterCurrent-1);
  NodeCounterCurrent = NodeCounterCurrent + 1;
  x(NodeCounterCurrent) = xcalc(3);
  Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
  Q(NodeCounterCurrent-1,i+1) = Q(NodeCounterCurrent-1,i+1) + QCalc(2);
  Q(NodeCounterCurrent,i+1) = QCalc(3);
else
  xcalc(1) == x(NodeCounterCurrent);
  NodeCounterCurrent = NodeCounterCurrent + 2;
  x(NodeCounterCurrent-1) = xcalc(2);
```

```
    x(NodeCounterCurrent) = xcalc(3);
    Q(NodeCounterCurrent-2,i+1) = Q(NodeCounterCurrent-2,i+1) + QCalc(1);
    Q(NodeCounterCurrent-1,i+1) = QCalc(2);
    Q(NodeCounterCurrent,i+1) = QCalc(3);
  end
  end
  end

NodeCurrentIndex = (NodeCounterCurrent-1)/2:-1:-(NodeCounterCurrent-1)/2;
fwdCIRrate(i) = ((2*a*theta*(exp(i*dt(i)*kappa) - 1))/(2*kappa + (a + kappa)*...
(exp(i*dt(i)*kappa) - 1))) + 0.001*((4*kappa^2*exp(i*dt(i)*kappa))/(2*kappa + ...
(a + kappa)*(exp(i*dt(i)*kappa) - 1))^2);
alpha(i+1) = fwdMrktRate(i) - fwdCIRrate(i);

% PROCEDURE 2 - Determining the short rate
ShortRateCurrent = 0;
ShortRateCurrent = (x(:)).^2 + alpha(i+1);
NoOfZeros = length(ShortRate) - length(ShortRateCurrent);
ShortRate(1:end - NoOfZeros,i+1) = ShortRateCurrent;
NodeCounter(i+1) = NodeCounterCurrent;
end
% disp(ShortRate);

K = 0.07;
Notional = 1000;
tau = 0:(91/365):((TimeSteps-2)/4);
t = ((TimeSteps-3)/4):(-91/365):0;
length(tau');
NoOfRows = NodeCounter';

%Bermudan Swaption Pricing for CIR
% Initially determine the Bond price (Discount Factor), the Market Forward Rate
for i = 1:NoOfRows(TimeSteps-1)
  TerminalMarketBond(1) = exp(-ShortRate(i,TimeSteps-1)*(tau(2)));
  kappa = sqrt(a^2 + 2*Sigma^2);
  B(i,TimeSteps-2) = (2*(exp(kappa*(tau(2)))-1))/(2*kappa +...
  (a+kappa)*((exp(kappa*(tau(2)))-1)));
  A(i,TimeSteps-2)=((2*kappa*exp((1/2)*(a+kappa)*(tau(2))))/(2*kappa+(a+kappa)*...
  ((exp(kappa*(tau(2)))-1))))^(2*theta/Sigma^2);
  Abar(i,TimeSteps-2)=((TerminalMarketBond(1)*A(i,TimeSteps-2)*...
  exp(-B(i,TimeSteps-2)*x(1,1)))/(TerminalMarketBond(1)*A(i,TimeSteps-2)*...
  exp(-B(i,TimeSteps-2)*x(1,1))))*A(i,TimeSteps-2)*exp(B(i,TimeSteps-2)*...
  alpha(TimeSteps-1));
  DiscountFactor(i,TimeSteps-2) = Abar(i,TimeSteps-2)*exp(-B(i,TimeSteps-2)*...
  ShortRate(i,TimeSteps-2));
  SumDisFac(i) = (91/365)*sum(DiscountFactor(i,TimeSteps-2));
  SwapRate(i) = (1 - DiscountFactor(i,TimeSteps-2))/SumDisFac(i);
  BermudanPrice(i,TimeSteps-2) = max(SumDisFac(i)*(SwapRate(i)-K),0);
end

for j = (TimeSteps-3):-1:1
  for i = 1:NoOfRows(j)
    kappa = sqrt(a^2 + 2*Sigma^2);
    TerminalMarketBond = exp(-fwdMrktRate(TimeSteps-1)*(Maturity));
    MarketBond(j) = exp(-fwdMrktRate(j)*t(j));
    B(i,j) = (2*(exp(kappa*(t(j)))-1))/(2*kappa + ...
```

```
        (a+kappa)*((exp(kappa*(t(j)))-1)));
        Bstar(i,j) = (2*(exp(kappa*(tau(j)))-1))/(2*kappa + (a+kappa)*...
        ((exp(kappa*(tau(j)))-1)));
        A(i,j)=((2*kappa*exp((1/2)*(a+kappa)*(tau(j)))))/(2*kappa+(a+kappa)*...
        ((exp(kappa*(tau(j)))-1))))^(2*theta/Sigma^2);
        Astar(i,j)=((2*kappa*exp((1/2)*(a+kappa)*(t(j)))))/(2*kappa+(a+kappa)*...
        ((exp(kappa*(t(j)))-1))))^(2*theta/Sigma^2);
        Abar(i,j)=((TerminalMarketBond*Astar(i,j)*exp(-Bstar(i,j)*x(1,1)))/(MarketBond(j)*...
        A(i,TimeSteps-2)*exp(-B(i,TimeSteps-2)*x(1,1))))*A(i,j)*exp(B(i,j)*...
        alpha(TimeSteps-1));
        DiscountFactor(i,j) = Abar(i,j)*exp(-B(i,j)*ShortRate(i,j));
        SumDisFac(i) = sum((91/365)*DiscountFactor(i,1:(TimeSteps-2)));
        SwapRate(i) = (1 - DiscountFactor(i,j))/SumDisFac(i);
      if NoOfRows(j) == (NoOfRows(j+1)-1) && (i > (NoOfRows(j)-2))
        CC(i,j) = exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*BermudanPrice(i-1,j+1) +

        ProbMid(i,j)*BermudanPrice(i,j+1) + ProbDown(i,j)*BermudanPrice(i+1,j+1));
      else
        CC(i,j) = exp(-ShortRate(i,j)*(91/365))*(ProbUp(i,j)*BermudanPrice(i,j+1) +
        ProbMid(i,j)*BermudanPrice(i+1,j+1) + ProbDown(i,j)*BermudanPrice(i+2,j+1));
        end
        BermudanPrice(i,j) = max(SumDisFac(i)*(SwapRate(i)-K),CC(i,j));
      end
    end

    % disp(DiscountFactor);
    % disp(BermudanPrice);
    BermudanSwaptionPrice = BermudanPrice(1,1)
```

10.4 Calibration

10.4.1 Swaption Differences

```
function ret = SwaptionDifferences(input)
    % First determine the difference between market and model
    global NoOfOptions;
    global OptionData;
    global NoOfIterations;
    global PriceDifference;
    NoOfIterations = NoOfIterations + 1;
    %counts the no of iterations run to calibrate model

    for i = 1:NoOfOptions
    % PriceDifference(i) = (OptionData(i)-ShortRateHWextendedVasicek(input(1),input(2),(4*(i)+1)));
    % PriceDifference(i) = (OptionData(i)-TrinomialBlackKarasinki1(input(1),input(2),(4*i+1)));
    PriceDifference(i) = (OptionData(i)-CIRtree(input(1),input(2),input(3),(4*(i)+1)));
    end
    ret = PriceDifference';
```

10.4.2 Swaption Calibration

```
clc;
    clear;
    global OptionData;
```

```
global NoOfOptions;
global NoOfIterations;
global PriceDifference;
NoOfIterations = 0;
load Data.m ;
Size = size(OptionData);
NoOfOptions = Size(1);

x0= [0.0214 0.00258 0.4818];
lb = [0 -1 -1];
ub = [1 1 1];

options = optimset('MaxFunEvals',20000);
% Minimise the price difference using a minimisation algorithm
% Sets the max number of iteration to 20000 so that termination doesn't take place
early.
tic;
% MatLab's built in minimisation function
Calibration = lsqnonlin(@SwaptionDifferences,x0,lb,ub);
toc;
Solution =Calibration(1:3)
```

References

[1] L.B.G. Andersen and J. Andreasen. Volatility skews and extensions of the LIBOR market model. *Applied Mathematical Finance*, 7(1): 1-32, 2001.

[2] D. Brigo and F. Mercurio.Interest rate models: theory and practice: with smile, inflation, and credit. *Springer Verlag*, 2006.

[3] J. Driessen and P. Klaassen and B. Melenberg. The performance of multi-factor term structure models for pricing and hedging caps and swaptions. University of Washington School of Business Administration. *The Journal of Financial and Quantitative Analysis,* 38(3):635–672, 2003.

[4] R. Fan and A. Gupta and P.H. Ritchken. On Pricing and Hedging in the Swaption Market: How Many Factors, Really?. 2006. http://faculty.weatherhead.case.edu/ritchken/documents/Swaptions_sep2006.pdf.

[5] Z. Guan and B. Gan and A. Khan. and S.H. Poon. Choice of Interest Rate Term Structure Models for Pricing and Hedging Bermudan Swaptions-An ALM Perspective. *Working paper Available at Social Science Research Network (SSRN)*, 2008. http://ssrn.com.

[6] A. Gupta and M.G. Subrahmanyam. Pricing and hedging interest rate options: Evidence from cap-floor markets. *Journal of Banking and Finance,* 29(3):701-733, 2005.

[7] A. Halamandaris. Implementing Short Rate Models. 2008. http://web.wits.ac.za/NR/rdonlyres/AnastasiaHalamandaris.pdf.

[8] M. Henrard. Bermudan swaptions in Hull-White one-factor model: analytical and numerical approaches. 2005. http://129.3.20.41/eps/fin/papers/0505/0505023.pdf

[9] S. Hippler. Pricing Bermudan Swaptions in the LIBOR Market Model. 2008. http://eprints.maths.ox.ac.uk/714/1/Dissertation_SteffenHippler.pdf.

[10] J. Hull and A. White. Numerical procedures for implementing term structure models. *The Journal of Derivatives*: 2(2):37-48, 1994.

[11] J.C. Hull. Options, futures, and other derivatives. *Prentice Hall*, 5th edition: 542-543, 552-556 2004.

[12] M. Jansson. On the pricing of Bermudan swaptions with an application to limited observed market data. http://www.algorithmica.se/research/Masters.pdf.

[13] J. Kennedy and P. Hunt and A. Pelsser. Markov-functional interest rate models. *Finance and Stochastics,* 4(4):391-408, 2000.

[14] FA. Longstaff and ES. Schwartz. Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1):113, 2001.

[15] P. Ouwehand. Interest Rate Modelling. *Financial Mathematics Course Notes*, 36-45, 2008.

[16] M.B. Pedersen and O. Plads. Bermudan swaptions in the Libor market model. http://www.fineanalytics.com/newsletter/FinEAnalytics_MC_BERMUDAN.pdf.

[17] Y. Tamba. Pricing a Bermudan Swaption with a Short Rate Lattice Method. *Discussion Papers in Economics and Business,* 2005. www2.econ.osaka-u.ac.jp/coe/dp/dp_no.54.pdf.