The cost of using misspecified models to exercise and hedge American options on coupon bearing bonds

Alexander Welihockyj

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

February 2, 2016

MPhil in Mathematical Finance,
University of Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy at the University of Cape Town. It has not been submitted before for any degree or examination to any other University.

Signed by candidate

Signature Removed

Alexander Welihockyj

February 2, 2016

Abstract

This dissertation investigates the cost of using single-factor models to exercise and hedge American options on South African coupon bearing bonds, when the simulated market term structure is driven by a two-factor model. Even if the single factor models are re-calibrated on a daily basis to the term structure, we find that the exercise and hedge strategies can be suboptimal and incur large losses. There is a vast body of research suggesting that real market term structures are in actual fact driven by multiple factors, so suboptimal losses can be largely reduced by simply employing a well-specified multi-factor model.

Acknowledgements

I would like to thank my supervisors: Searle Silverman and associate professor Thomas McWalter. I am further grateful to Obeid Mohammed for his generous assistance throughout. To my parents, thank you for supporting me emotionally and financially during a very challenging year.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The motivation for this dissertation comes from a paper written by Longstaff *et al.* (2001) on the cost of suboptimal exercise strategies in the American swaption market. Longstaff *et al.* (2001) discover that using single-factor models to exercise American swaptions, in a simulated multi-factor world, will lead to large losses. This result is genuinely relevant as there is a vast body of research showing that market term structures are driven by multiple factors, yet many financial institutions are still using single-factor models for valuing, hedging and exercising American style interest rate derivatives (Longstaff and Schwartz, 2001).

We aim to replicate this result, albeit with regards to American options on South African government coupon bearing bonds[1]. Moreover, we go one step further and conduct a delta-hedge analysis on a short-dated American option.

Three different interest rate models are examined: the single-factor Vasicek model (VM), the single-factor Hull-White model (HWM) and the two-factor LIBOR market model (LMM2). The VM and HWM are short rate Gaussian processes which we implement in both lattice and Monte Carlo frameworks. The LMM2 is a log-normal forward rate model which we implement using Monte Carlo methods. Ultimately, five derivative pricing models are developed.

The exercise and hedge analysis is conducted firstly in a Vasicek simulated world, then in a Hull-White simulated world and finally in a two-factor LIBOR simulated world. Each world gets more complex and thus more realistic. In addition, the pricing models are also compared with regard to efficiency and robustness.

In the following chapter, we introduce the VM, the HWM and the LMM2. Chapter 3 describes the bond pricing conventions specific to South Africa. Then, in Chapter 4, we demonstrate how to price the American option in both numerical frameworks. The exercise and delta-hedge analysis is explained in Chapter 5. Chapter 6 outlines the various world scenarios and how calibration is achieved in each. We discuss results and draw conclusions in Chapter 7 and Chapter 8 respectively.

[1] These bonds yield known semi-annual coupons.

Finally, Chapter 9 outlines potential future research objectives.

Chapter 2

Interest Rate Models

The first two sections in this chpater describe the implementation of the VM and HWM. The mathematics closely follows Glasserman (2004), pg 108-116.

These models postulate the dynamics of the continuous and instantaneously compounded short rate $r(t)$. An investment of one unit in a money market account that earns interest $r(u)$ at time u grows from a value of 1 at time 0 to a value of

$$\beta(t) = \exp\left(\int_0^t r(u)du\right)$$

at time t. This stochastic variable represents the numeraire used in risk neutral pricing. The value of a derivative at time 0, that pays X at time t, can be expressed as

$$V_0 = \mathbf{E}^{\mathbf{Q}}\left[\exp\left(-\int_0^t r(u)du\right)X\right],$$

with the expectation taken under the risk neutral measure $\mathbf{Q}$. More specifically, a zero coupon bond that pays $X = 1$ at T will have a value of

$$B(t,T) = \mathbf{E}^{\mathbf{Q}}\left[\exp\left(-\int_t^T r(u)du\right)\right] \tag{2.1}$$

at time t.

The third section in this chapter outlines the LMM2.

2.1 Vasicek Model (VM)

Vasicek (1977) proposed

$$dr(t) = \alpha(b - r(t))dt + \sigma d\widetilde{W}(t)$$

as the stochastic differential equation (SDE) for the short rate, where $\widetilde{W}(t)$ is a single standard Brownian motion under the risk neutral measure, $r(t)$ is the short

rate, b is the long term mean reversion level, α is the mean reversion rate and σ is the volatility of short term interest rates. The term $\alpha(b-r(t))$ represents the drift of the process which is pulled towards the level b at a rate proportional to the deviation of the process to this level. The variables b, α and σ are calibration parameters that may be used to fit the model to the current observable market term structure.

Lemma 2.1. *The solution to the Vasicek SDE is as follows,*

$$r(t) = b + (r(0) - b)e^{-\alpha t} + \sigma \int_0^t e^{-\alpha(t-u)} d\widetilde{W}(u),$$

for $0 \leq u \leq t$.

Proof. Integrate the SDE by applying an integrating factor. □

Therefore, it follows that given $r(s)$ with $s \leq u \leq t$, $r(t)$ is conditionally normal under the risk neutral measure, with an expected value

$$\mathbf{E}\left[r(t)|r(s)\right] = \mu_r(s,t) = b + (r(s) - b)e^{-\alpha(t-s)} \tag{2.2}$$

and variance

$$\begin{aligned} \mathbf{Var}\left[r(t)|r(s)\right] = \sigma_r^2(s,t) &= \sigma^2 \int_s^t e^{-2\alpha(t-u)} du \\ &= \frac{\sigma^2}{2\alpha}\left(1 - e^{-2\alpha(t-s)}\right). \end{aligned} \tag{2.3}$$

The normality of this process implies that the short rate may go negative with positive probability [1]. The variance of the process is bounded by $\frac{\sigma^2}{2\alpha}$, which we calculate by taking limits as $t \to \infty$ in (2.3).

In this dissertation the VM is implemented in a binomial tree and using Monte Carlo simulation.

2.1.1 Bond Prices

We need to calculate the expectation in (2.1) to get an expression for zero coupon bond prices in the VM. Keep in mind that zero coupon bond prices are essentially discount factors which are later used to discount the bond cashlfows.

Using the fact that for $X \sim \mathcal{N}(\mu, \nu^2)$ we have $\mathbf{E}(e^X) = e^{\mu + \frac{1}{2}\nu^2}$, the bond price at time t for maturity T is written as

$$B(t,T) = \exp\left(-\mathbf{E}\left[\int_t^T r(u)du\right] + \frac{1}{2}\mathbf{Var}\left[\int_t^T r(u)du\right]\right). \tag{2.4}$$

[1] Interest rates are currently negative in Japan and the Eurozone, making the assumption of normality not as undesirable as it once was.

From (2.2), for the mean we have

$$\begin{aligned}\mathbf{E}\left[\int_t^T r(u)du\right] &= \int_t^T \mathbf{E}\left[r(u)\right]du \\ &= \int_t^T \left(b + (r(t)-b)e^{-\alpha(u-t)}\right)du \\ &= b(T-t) + (r(t)-b)\,\frac{\left(1-e^{-\alpha(T-t)}\right)}{\alpha}. \end{aligned} \tag{2.5}$$

The variance is computed as

$$\mathbf{Var}\left[\int_t^T r(u)du\right] = 2\int_t^T\int_t^u \mathbf{Cov}_t\left[r(u),r(s)\right]dsdu, \tag{2.6}$$

where the covariance on the interval $[t,u]$, for $t \le s \le u$, is calculated as

$$\mathbf{Cov}_t\left[r(u),r(s)\right] = \sigma^2\int_t^s \mathrm{e}^{-\alpha(u-v)}\mathrm{e}^{-\alpha(s-v)}dv. \tag{2.7}$$

After substituting (2.7) into (2.6), perform the integration and simplification to arrive at the expression for the variance

$$\mathbf{Var}\left[\int_t^T r(u)du\right] = \frac{\sigma^2}{\alpha^2}\left[(T-t) + \frac{1}{2\alpha}\left(1-\mathrm{e}^{-2\alpha(T-t)}\right) + \frac{2}{\alpha}\left(\mathrm{e}^{-\alpha(T-t)}-1\right)\right]. \tag{2.8}$$

Substituting (2.5) and (2.8) into (2.4) gives an expression for the bond price. To simplify notation, we introduce

$$\begin{aligned} A(t,T) &= \frac{\left(1-e^{-\alpha(T-t)}\right)}{\alpha} \\ C_v(t,T) &= \left(\frac{\sigma^2}{2\alpha^2}-b\right)\left[(T-t)-A(t,T)\right] - \frac{\sigma^2}{4\alpha}A^2(t,T). \end{aligned} \tag{2.9}$$

Consequently, the VM bond price is presented as

$$B_v(t,T,r(t)) = e^{-A(t,T)r(t)+C_v(t,T)} \tag{2.10}$$

2.1.2 Binomial Tree Implementation

In this section, we create a unique and efficient binomial implementation procedure for the VM which is based on a particular specification of the volatility structure.

We start by applying a Euler-Maruyama approximation to discretise the VM SDE. Unfortunately the mean-reverting nature of the VM ensures that the tree is intrinsically non-recombining, so the discretised equation is further generalised to make b, α and σ functions of time. By doing so, we can prove that under a certain assumption for the function of $\sigma(t)$ the tree will recombine while still maintaining a constant b and α as required for the VM.

The following equations represent the discretised and generalised equations for both an up and down movement from node $(t_j)_i$:

$$r(t_{j+1})_i = r(t_j)_i + \alpha(t_j)\left[b(t_j) - r(t_j)_i\right]\Delta t + \sigma(t_j)\sqrt{\Delta t} \qquad \left[P_u = \frac{1}{2}\right] \tag{2.11}$$

$$r(t_{j+1})_{i+1} = r(t_j)_i + \alpha(t_j)\left[b(t_j) - r(t_j)_i\right]\Delta t - \sigma(t_j)\sqrt{\Delta t} \qquad \left[P_d = \frac{1}{2}\right] \tag{2.12}$$

where node i refers to the state at each time step t_j. Furthermore, P_u and P_d refer to the probability of making an up or down movement in the tree.

If we subtract (2.12) from (2.11), the following relationship is noted

$$\frac{1}{2}\left[r(t_{j+1})_i - r(t_{j+1})_{i+1}\right] = \sigma(t_j)\sqrt{\Delta t}. \tag{2.13}$$

We now define

$$r(t_{j+1})_{i+1} = r(t_j)_{i+1} + \alpha(t_j)\left[b(t_j) - r(t_j)_{i+1}\right]\Delta t + \sigma(t_j)\sqrt{\Delta t}, \tag{2.14}$$

which represents an up movement from node $(t_j)_{i+1}$. We equate (2.12) and (2.14) to force recombination. Using the relation in (2.13) we solve an expression for $\alpha(j)$,

$$\begin{aligned} r(t_j)_i + \alpha(j)\left[b(t_j) - r(t_j)_i\right]\Delta t - \sigma(t_j)\sqrt{\Delta t} &= r(t_j)_{i+1} + \\ &\quad \alpha(t_j)\left[b(t_j) - r(t_j)_{i+1}\right]\Delta t + \sigma(t_j)\sqrt{\Delta t} \\ \implies \qquad \frac{1}{2}\left(r(t_j)_i - r(t_j)_{i+1}\right)\left[1 - \alpha(t_j)\Delta t\right] &= \sigma(t_j)\sqrt{\Delta t} \\ \implies \qquad \sigma(t_{j-1})\sqrt{\Delta t}\left[1 - \alpha(t_j)\Delta t\right] &= \sigma(t_j)\sqrt{\Delta t}, \end{aligned}$$

which yields,

$$\alpha(t_j) = \frac{1}{\Delta t}\left[1 - \frac{\sigma(t_j)}{\sigma(t_{j-1})}\right]. \tag{2.15}$$

We know *a priori* that the mean reversion rate must always be greater than zero: if it were equal to zero the model would not have a drift term and if it were negative the model would mean avert. Consequently, σ must be a monotonically decreasing function of time to ensure positivity in (2.15). We assume $\sigma(t_j) = \sigma e^{-xj\Delta t}$ to make the forward volatility compatible[2] with the VM and then solve for a value of x such that $\alpha(t_j) = \alpha$ for all j,

$$\begin{aligned} \alpha &= \frac{1}{\Delta t}\left[1 - \frac{\sigma(t_j)}{\sigma(t_{j-1})}\right] \\ &= \frac{1}{\Delta t}\left[1 - \frac{\sigma e^{-xj\Delta t}}{\sigma e^{-x(j-1)\Delta t}}\right] \\ &= \frac{1}{\Delta t}\left[1 - e^{-x\Delta t}\right], \end{aligned}$$

[2] See Glasserman (2004) on HJM, pg 150-155

therefore,

$$x = -\ln(1 - \alpha\Delta t)\frac{1}{\Delta t}$$

and hence,

$$\sigma(t_j) = \sigma e^{\ln(1-\alpha\Delta t)j}.$$

Under this assumption for the volatility structure, we have both forced the binomial tree to recombine and we have ensured that $\alpha(t_j)$ is constant through time and equal to the initial mean reversion rate α which is approximated on the value date of the option.

We now prove that the parameter $b(t_j) \approx b$ at every time step. Firstly, we add (2.11) and (2.12),

$$\begin{aligned} & r(t_{j+1})_i + r(t_{j+1})_{i+1} = 2r(t_j)_i + 2\alpha\left[b(t_j) - r(t_j)_i\right]\Delta t \\ \implies \quad & \alpha b(t_j) = \frac{1}{2\Delta t}\left[r(t_{j+1})_i + r(t_{j+1})_{i+1} - 2r(t_j)_i + 2\alpha r(t_j)_i\Delta t\right] \\ \implies \quad & \alpha b(t_j) = \frac{1}{\Delta t}\left[\frac{1}{2}\left(r(t_{j+1})_i + r(t_{j+1})_{i+1}\right) - r(t_j)_i + \alpha r(t_j)_i\Delta t\right] \end{aligned}$$

and note that $\frac{1}{2}\left[r(t_{j+1})_i + r(t_{j+1})_{i+1}\right]$ is a conditional expectation, so

$$\alpha b(j) = \frac{1}{\Delta t}\left[\mathbf{E}^{\mathbf{Q}}\left[r(j+1)\middle| r(j)_i\right] - r(j)_i + \alpha r(j)_i\Delta t\right].$$

We can substitute (2.2) in place of the expectation and take a first order Taylor expansion of the exponential,

$$\begin{aligned} \alpha b(t_j) &= \frac{1}{\Delta t}\left[b + (r(t_j)_i - b)e^{-\alpha\Delta t} - r(t_j)_i + \alpha r(t_j)_i\Delta t\right] \\ &\approx \frac{1}{\Delta t}\left[b + (r(t_j)_i - b)(1 - \alpha\Delta t) - r(t_j)_i + \alpha r(t_j)_i\Delta t\right] \\ &\approx \alpha b. \end{aligned} \tag{2.16}$$

The entire short rate tree can now be specified using the following three equations:

$$\begin{aligned} \sigma(t_j) &= \sigma e^{\ln(1-\alpha\Delta t)j} \\ r(t_{j+1})_i &= r(t_j)_i + \alpha\left[b - r(t_j)_i\right]\Delta t + \sigma(t_j)\sqrt{\Delta t} \\ r(t_{j+1})_{i+1} &= r(t_j)_i + \alpha\left[b - r(t_j)_i\right]\Delta t - \sigma(t_j)\sqrt{\Delta t}. \end{aligned}$$

2.1.3 Monte Carlo Implementation

In Monte Carlo simulation, each short rate path is independent and so recombination is of no concern. It is therefore trivial to simulate the short rate within this framework once we have an expression for the conditional expectation and variance

of the VM short rate. These expressions are presented in (2.2) and (2.3). We can simulate a single short rate path using the forward value iteration equation

$$r(t_{j+1}) = \mu_r(t_j, t_{j+1}) + \sigma_r(t_j, t_{j+1})Z \tag{2.17}$$

where $Z \sim \mathcal{N}(0,1)$. This equation can easily be vectorised in `MATLAB` to simulate any number of paths concurrently and efficiently.

The numeraire for pricing options in this setting is stochastic. We therefore also need realisations of the discount factors

$$\frac{1}{\beta(t_j, t_{j+1})} = \exp\left(-\int_{t_j}^{t_{j+1}} r(u)du\right),$$

or alternatively, of

$$Y(t_j, t_{j+1}) = \int_{t_j}^{t_{j+1}} r(u)du.$$

Given realisations of $r(t_0), r(t_1), \ldots, r(t_n)$ of the short rate, $Y(t_0, t_j)$ can be approximated using simple quadrature as

$$Y(t_0, t_j) \approx \sum_{k=1}^{j} r(t_{k-1})\Delta t,$$

or more accurately using trapezoidal quadrature as

$$Y(t_0, t_j) \approx \sum_{k=1}^{j+1} r(t_{k-1})\Delta t \left[1 - \left(\mathbb{1}_{\{k=1\}} + \mathbb{1}_{\{k=j+1\}}\right)/2\right].$$

Both of these approximations suffer from discretisation error because the above approximations both assume linearity of the short rate between $[t_j, t_{j+1}]$, when in actual fact the short rate is stochastic between the discretised time steps. If, however, we realise that $r(t)$ and $Y(t)$ are jointly Gaussian, both the short rate and the discount factor may be simulated simultaneously without discretisation error.

Our previous calculations in (2.5) and (2.8) show that

$$Y(t_{j+1}) \sim \mathcal{N}(\mu_Y(t_j, t_{j+1}), \sigma_Y^2(t_j, t_{j+1})),$$

with

$$\begin{aligned}\mu_Y(t_j, t_{j+1}) &= \int_{t_0}^{t_j} r(u)du + \mathbf{E}^{\mathbf{Q}}\left[\int_{t_j}^{t_{j+1}} r(u)du\right] \\ &= Y(t_0, t_j) + (t_{j+1} - t_j)b + (r(t_j) - b)A(t_j, t_{j+1})\end{aligned}$$

and

$$\sigma_Y^2(t_j, t_{j+1}) = \frac{\sigma^2}{\alpha^2}\left[(t_{j+1} - t_j) - A(t_j, t_{j+1}) - \frac{\alpha}{2}A(t_j, t_{j+1})\right]. \tag{2.18}$$

$A(t_j, t_{j+1})$ is already defined in (2.9).

Thus in order to simulate $Y(t)$ we need to compute the correlation between $r(t)$ and $Y(t)$. We start by calculating the covariance.

Theorem 2.2. *The covariance between $r(t)$ and $Y(t)$ is defined as*

$$\sigma_{rY}(t_j, t_{j+1}) = \frac{\sigma^2}{2}A^2(t_j, t_{j+1})$$

Proof. See McWalter (2014) □

The correlation is computed as follows

$$\rho_{rY}(t_j, t_{j+1}) = \frac{\sigma_{rY}(t_j, t_{j+1})}{\sigma_r(t_j, t_{j+1})\sigma_Y(t_j, t_{j+1})} \tag{2.19}$$

with $\sigma_r(t_j, t_{j+1})$ defined in (2.3). Therefore the pair $[r(t_j), Y(t_j)]$ can be computed using (2.17) and

$$Y(t_{j+1}) = \mu_Y(t_j, t_{j+1}) + \sigma_Y(t_j, t_{j+1}) \left[\rho_{rY}(t_j, t_{j+1})Z_1 + \sqrt{1 - \rho_{rY}^2(t_j, t_{j+1})}Z_2\right], \tag{2.20}$$

where $Z_1, Z_2 \sim \mathcal{N}(0, 1)$ are independent and Z_1 is the variate used to evolve the short rate over the same time step.

2.2 Hull-White Model (HWM)

Hull and White (1990) proposed

$$dr(t) = \alpha(b(t) - r(t))dt + \sigma d\widetilde{W}(t) \tag{2.21}$$

for the short rate SDE, where $\widetilde{W}(t)$ is a single standard brownian motion under the risk neutral measure, $r(t)$ is the short rate, $b(t)$ is the long term time-dependent mean reversion level, α is the mean reversion rate and σ is the volatility of short term interest rates. This SDE is an extension to the VM in which the mean reversion level is now time dependent and not a constant. The fundamental characteristics of the model are the same as described in Section 2.1. VM and HWM do however differ in regard to their SDE solutions, bond prices, calibration and implementation.

Lemma 2.3. *The solution to the Hull and White (1990) SDE is as follows,*

$$r(t) = e^{-\alpha(t-0)}r(0) + \alpha\int_0^t e^{-\alpha(t-u)}b(u)du + \sigma\int_0^t e^{-\alpha(t-u)}d\widetilde{W}(u),$$

for $0 \leq u \leq t$.

Proof. Apply Ito's Lemma. □

Therefore given $r(s)$ with $s \leq u \leq t$, $r(t)$ is conditionally normal under the risk neutral measure with an expected value

$$\mathbf{E}\left[r(t)|r(s)\right] = e^{-\alpha(t-s)}r(s) + \alpha\int_s^t e^{-\alpha(t-s)}b(u)du \tag{2.22}$$

and variance

$$\begin{aligned}\mathbf{Var}\left[r(t)|r(s)\right] &= \sigma^2\int_s^t e^{-2\alpha(t-u)}du \\ &= \frac{\sigma^2}{2\alpha}\left(1 - e^{-2\alpha(t-s)}\right).\end{aligned}$$

This distribution implies that the short rate can go negative with positive probability, but like the VM, it also has a bounded variance of $\frac{\sigma^2}{2\alpha}$.

In this dissertation the HWM is implemented in a trinomial tree and with Monte Carlo simulation.

2.2.1 Bond Prices

Using (2.1) and the fact that for $X \sim \mathcal{N}(\mu, \nu^2)$ we have $\mathbf{E}(e^X) = e^{\mu+\frac{1}{2}\nu^2}$, the bond price at time t for maturity T is written as

$$B(t,T) = \exp\left(-\mathbf{E}\left[\int_t^T r(u)du\right] + \frac{1}{2}\mathbf{Var}\left[\int_t^T r(u)du\right]\right). \tag{2.23}$$

From (2.22), for the mean we have

$$\begin{aligned}\mathbf{E}\left[\int_t^T r(u)du\right] &= \int_t^T \mathbf{E}\left[r(u)\right]du \\ &= \int_t^T e^{-\alpha(u-t)}r(t)du + \alpha\int_t^T\int_t^u e^{-\alpha(u-s)}b(s)dsdu \\ &= \frac{1}{\alpha}\left(1 - e^{-\alpha(T-t)}\right)r(t) + \alpha\int_t^T\int_s^T e^{-\alpha(u-s)}b(s)duds \\ &= A(t,T)r(t) + \alpha\int_t^T A(s,T)b(s)ds,\end{aligned} \tag{2.24}$$

where $A(t,T)$ is defined in (2.9). The variance is the same as the VM model (the calculation can be seen in (2.3)). The result however is displayed here for convenience:

$$\mathbf{Var}\left[\int_t^T r(u)du\right] = \frac{\sigma^2}{\alpha^2}\left[(T-t) - A(t,T) - \frac{\alpha}{2}A^2(t,T)\right]. \tag{2.25}$$

Substituting (2.24) and (2.25) into (2.23) gives an expression for the bond price. To simplify notation, we introduce

$$C_h(t,T) = \frac{\sigma^2}{2\alpha^2}\left[(T-t) - A(t,T) - \frac{\alpha}{2}A^2(t,T)\right] - \alpha\int_t^T A(s,T)b(s)ds, \qquad (2.26)$$

therefore the zero coupon bond price is

$$B_h(t,T,r(t)) = e^{-A(t,T)r(t)+C_h(t,T)}. \qquad (2.27)$$

2.2.2 Trinomial Tree Implementation

This subsection describes an efficient method for implementing the HWM in a trinomial tree setting as described in Hull and White (1994).

The first step in building the calibrated short rate tree is to set up a preliminary tree in which $b(t) = 0$ and the initial value $r(0) = 0$. Therefore the initial dynamics assumed for $r(t)$ are

$$dr(t) = -\alpha r(t)dt + \sigma d\widetilde{W}t,$$

instead of (2.21).

Lemma 2.4. *The expectation and variance of $r(t+\Delta t) - r(t)$ is*

$$M = e^{-\alpha\Delta t} - 1$$
$$V = \sigma^2(1 - e^{-2\alpha\Delta t})/2\alpha$$

respectively.

Proof. See Hull and White (1994) endnote [5]. □

We select the size of the time step, Δt, and define $\Delta r = \sqrt{3V}$. Hull and White (1994) suggest that this is a good choice of Δr from a standpoint of error minimisation. Having specified these variables, we can define node (i,j) as the node for which $r = i\Delta r$ and $t = j\Delta t$, where $i \in [i_{min}; i_{max}]$ and $i_{max} = -i_{min} = \lfloor 0.184/M \rfloor$[3]. The reason for defining this value of i_{max} is made clear in the next paragraph.

Before we can specify the simplified tree, we must resolve which branching process will apply at each node. This determines the overall shape of the tree. The branching processes are presented in Figure 2.1. The majority of nodes will branch according to **A**, but nodes at (i_{max}, j) must branch according to **C** and similarly nodes at (i_{min}, j) must branch according to **B**. This specification essentially prunes the tree for very large and very small values of i. The technique is not stipulated for convenience but rather to ensure positivity in all branching probabilities. The

[3] $\lfloor x \rfloor$ refers to the function that rounds down x to the nearest full number (floor function).

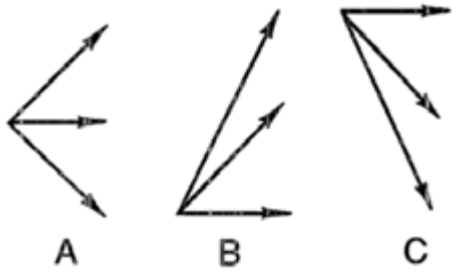

Fig. 2.1: Hull-White trinomial tree branching processes

branching probabilities are presented in Table 2.1. Note that the probabilities at each node are only dependent on the value of i at that node and the value of M. This completes the specification of the simplified tree.

Tab. 2.1: Probabilities associated with each branching process

	A	**B**	**C**
P_u	$\frac{1}{6} + \frac{i^2M^2+iM}{2}$	$\frac{1}{6} + \frac{i^2M^2-iM}{2}$	$\frac{7}{6} + \frac{i^2M^2+3iM}{2}$
P_m	$\frac{2}{3} - i^2M^2$	$-\frac{1}{3} - i^2M^2 + 2iM$	$-\frac{1}{3} - i^2M^2 - 2iM$
P_d	$\frac{1}{6} + \frac{i^2M^2-iM}{2}$	$\frac{7}{6} + \frac{i^2M^2-3iM}{2}$	$\frac{1}{6} + \frac{i^2M^2+iM}{2}$

Next, we calibrate the simplified tree to the market by introducing the correct time varying drift. To do this, we displace the nodes at time $i\Delta t$ by an amount a_i, i.e., the value of r at node (i, j) in the calibrated tree will simply be the value r at node (i, j) in the simplified tree plus the value a_j. The probabilities remain unchanged in the calibration process. The value of a_j is selected so that the tree prices discount bonds consistently with the initial term structure.

To assist computation, a variable $Q_{i,j}$ is defined as the present value of a security that pays 1 if node (i, j) is reached and 0 otherwise. Therefore $Q_{0,0} = 1$ naturally and we can also specify a_0 to be the interest rate in the market that applies over the first Δt period.

Lemma 2.5. *$Q_{i,j}$ and a_j are calculated using the following forward induction equations*

$$a_m = \log\left(\frac{\sum_{i=-n_m}^{n_m} Q_{i,m} e^{-i\Delta r\Delta t}}{B^*(0, m+1)}\right) \Big/ \Delta t$$

$$Q_{i,m+1} = \sum_k Q_{k,m} q(i,k) \exp\big[-(a_m + k\Delta r)\Delta t\big],$$

where $m = j + 1$, n_m is the number of nodes on each side of the central node at $m\Delta t$, $q(i,k)$ is the probability of moving from node (k,m) to node $(i,m+1)$ and the summation is taken over all values of k for which this is non-zero.

Proof. See Hull and White (1994) pg. 12-13. □

2.2.3 Monte Carlo Calibration

This section describes the process with which we can approximate the time dependant mean reversion level function, $b(t)$, when implementing the HWM in a Monte Carlo setting.

Suppose a yield curve has been bootstrapped for today, $t = 0$. Let the associated zero coupon bonds be denoted by $B^*(0,T), T \in \mathbb{R}^+$. In order for the HWM to recover the prices in the market, we require

$$B^*(0,T) = e^{-A(0,T)r^* + C_h(0,T)}$$

where r^* is the bootstrapped rate at $t = 0$. Inverting this equation for $C_h(0,T)$ and equating with (2.26) we get

$$\alpha \int_0^T A(s,T)ds = \frac{\sigma^2}{2\alpha^2}\left[(T-t) - A(t,T) - \frac{\alpha}{2}A^2(t,T)\right] - \log\left(B^*(0,T)\right) - A(0,T)r^*.$$

Differentiating with respect to T twice, we get

$$b(t) = \frac{1}{\alpha}\frac{\partial^2}{\partial^2 t}\log\left(B^*(0,t)\right) - \frac{\partial}{\partial t}\log\left(B^*(0,t)\right) + \frac{\sigma^2}{2\alpha^2}\left(1 - e^{-2\alpha t}\right). \tag{2.28}$$

Therefore (2.26) can be written in the following form to calibrate the Hull-White bond prices to the initial market curve

$$C_h(t,T) = \log\left(\frac{B^*(0,T)}{B^*(0,t)}\right) + f^*(0,t)A(t,T) - \frac{\sigma^2}{4\alpha}\left(1 - e^{-2\alpha t}\right)A^2(t,T).$$

where $f^*(0,t) = -\frac{\partial}{\partial t}\left(B^*(0,t)\right)$ is the instantaneous forward rate as observed today for time t, the value of which can be estimated using a first order central difference approximation on the bootstrapped bond curve.

2.2.4 Monte Carlo Implementation

The approach for simulating short rate paths in the HWM is the same as the approach described in Section 2.1.3. The equations however do differ in some instances; the similarities and differences are highlighted.

To simulate the short rate we define the same iterative equation as described in (2.17), but in the HWM setting, we use (2.22) in place of $\mu_r(t_j, t_{j+1})$. The volatility term is equivalent in both models. The integral in (2.22) is approximated over each time step using trapezoidal quadrature, for which the value of $b(t_j)$ and $b(t_{j+1})$ can be approximated using (2.28).

To simulate the discount factors we use our calculations in (2.24) to show that

$$Y(t_{j+1}) \sim \mathcal{N}(\mu_Y(t_j, t_{j+1}), \sigma_Y^2(t_j, t_{j+1})),$$

with

$$\begin{aligned}\mu_Y(t_j, t_{j+1}) &= \int_{t_0}^{t_j} r(u)du + \mathbf{E}^{\mathbf{Q}}\left[\int_{t_j}^{t_{j+1}} r(u)du\right] \\ &= Y(t_j) + e^{-\alpha(t_{j+1}-t_j)}r(t_j) + \alpha\int_{t_j}^{t_{j+1}} e^{-\alpha(t_{j+1}-u)}b(u)du\end{aligned}$$

and $\sigma_Y^2(t_j, t_{j+1})$ defined in (2.18). The correlation between $Y(t)$ and $r(t)$ is defined in (2.19). Therefore the pair $(r(t_j), Y(t_j))$ can be computed using (2.17) and

$$\begin{aligned}Y(t_{j+1}) = \mu_Y(t_j, t_{j+1}) + \sigma_Y(t_j, t_{j+1}) \\ \left[\rho_{rY}(t_j, t_{j+1})Z_1 + \sqrt{1-\rho_{rY}^2(t_j, t_{j+1})}Z_2\right]\end{aligned}$$

where $Z_1, Z_2 \sim \mathcal{N}(0, 1)$ are independent and Z_1 is the same variate used to evolve the short rate over the same time step.

2.3 Two-Factor LIBOR Market Model (LMM2)

This section describes the LMM2 of Brace *et al.* (1997) and Jamshidian (1997). The Monte Carlo implementation follows Glasserman (2004), pg. 165-172. We develop the theory under the spot measure.

The VM and HWM, previously described, are based on the evolution of the instantaneously compounded short rate. Although these models are useful and still used in industry, they are largely a convenient idealisation; the short rate is not tradeable in the market. The LMM2 model differs in the sense that it models simply compounded forward rates which can be traded in the market. Furthermore, the LMM2 follows a log-normal process which guarantees the positivity of interest rates.

Initially, a continuous-time SDE is derived. The SDE is then discretised and presented. Unfortunately, it is not possible to ensure that this discretised system is arbitrage-free, therefore a predictor corrector method is implemented to ensure a more accurate discretised representation.

2.3.1 Simple Forward Rates

We start by describing the simple forward rate notation. The current, $t = 0$, forward rate for maturity T with accrual period Δt is denoted by $F(0,T)$. The interest earned per unit currency at time $T + \Delta t$ is therefore $\Delta t F(0,T)$. If we define the market price of a bond for tenor T as $B(0,T)$, we can calculate the the forward rate as

$$F(0,T) = \frac{B(0,T) - B(0,T+\Delta t)}{\Delta t B(0,T+\Delta t)}. \tag{2.29}$$

Using a set of market instrument tenors $0 = T_0 < T_1 < \cdots < T_N$, we define the intervals between these tenors as

$$\Delta t_i = T_{i+1} - T_i,$$

for $i = 0, 1, \ldots, N-1$.

We us the shorthand notation $F_i(t) = F(t, T_i)$ and note that Δt_i is the length of the accrual period associated with $F_i(\cdot)$. We apply the same shorthand notation to bond prices: $B_i(t) = B(t, T_i)$.

Using (2.29) and the shorthand notation, in general we have

$$F_j(t) = \frac{B_j(t) - B_{j+1}(t)}{\Delta t_j B_{j+1}}, \tag{2.30}$$

for $j = 0, 1, \ldots, N-1$. We invert (2.30) to give the bond prices at tenor dates T_i, in terms of the forward rates,

$$B_j(T_i) = \prod_{k=i}^{j-1} \frac{1}{1 + \Delta t_k F_k(T_i)},$$

for $j = i+1, \ldots, N$. At times other than tenor dates, e.g. for $T_i \leq t < T_{i+1} < T_j$, we have

$$B_j(t) = B_{i+1}(t) \prod_{k=i+1}^{j-1} \frac{1}{1 + \Delta t_k F_k(t)}.$$

We further define a function a function $\eta(t) = \{\min i | t < T_i\}$, so that we have

$$B_j(t) = B_{\eta(t)}(t) \prod_{k=\eta(t)}^{j-1} \frac{1}{1 + \Delta t_k F_k(t)} \tag{2.31}$$

for $0 \leq t < T_j$.

2.3.2 LIBOR Forward Rate Evolution

We evolve the LIBOR forward rates using a two-factor SDE of the form

$$dF_j(t) = F_j(t)\mu_j(t)dt + F_j(t)\sigma_j(t)\left(\rho_j(t)dW_t^1 + \sqrt{1-\rho_j^2(t)}dW_t^2\right)$$

for $0 \leq t \leq T_j$, $j = 1, \ldots, N-1$, where $\sigma_j(t)$ is the forward volatility and $\rho_j(t)$ is the correlation between the two Brownian motions.

For our model to remain arbitrage free, we require the discounted bond prices to be martingales under the risk neutral measure. So, we select a simply compounded numeraire asset: an asset that starts with a value of 1 at $t = 0$ and accrues interest through its investment in a bond with the shortest tenor; subsequently the proceeds are reinvested in each successive bond with the shortest tenor until the current time. At time t, this instrument has a value of

$$\bar{\beta}(t) = B_{\eta(t)}(t) \prod_{k=0}^{\eta(t)-1} \left[1 + \Delta t_k F_k(T_k)\right]. \tag{2.32}$$

Having specified the numeraire, we require the discounted bond prices given by $D_j(t) = B_j(t)/\bar{\beta}(t)$ to be martingales.

Using (2.31) and (2.32), the discounted bond is given by

$$D_j(t) = \left(\prod_{k=\eta(t)}^{j-1} \frac{1}{1+\Delta t_k F_k(t)}\right) \prod_{k=0}^{\eta(t)-1} \frac{1}{1+\Delta t_k F_k(T_k)}, \tag{2.33}$$

which conveniently cancels the factor $B_{\eta(t)}(t)$, allowing D to be specified exclusively using the LIBOR forward rates.

We take logs of both sides of (2.33) and realise that the term on the right, in the line below, is a constant at t. So we have,

$$\begin{aligned}
\log D_j(t) &= -\sum_{k=\eta(t)}^{j-1} \log(1+\Delta_k F_k(t)) - \sum_{k=0}^{\eta(t)-1} \log(1+\Delta_k F_k(T_k)) \\
\implies \quad d\log D_j(t) &= -\sum_{k=\eta(t)}^{j-1} d\log(1+\Delta_k F_k(t)) \\
&= -\sum_{k=\eta(t)}^{j-1} \left[\left(\frac{\Delta_k F_k(t)\mu_k(t)}{1+\Delta_k F_k(t)} - \frac{1}{2}\left(\frac{\Delta_k F_k(t)\sigma_k(t)}{1+\Delta_k F_k(t)}\right)^2\right)dt \right. \\
&\quad \left. + \frac{\Delta_k F_k(t)\sigma_k(t)}{1+\Delta_k F_k(t)} dW_t\right],
\end{aligned} \tag{2.34}$$

where the last expression follows from Ito's formula.

We require the deflated bonds to be martingales, so the dynamics can be written as,

$$dD_j(t) = D_j(t)\nu_j(t)dW_t,$$

for $j = 2, \ldots, N$ and some process ν_j. By an application of Ito's formula, we have

$$d\log D_j(t) = -\frac{1}{2}(\nu_j(t))^2 dt + \nu_j(t)dW_t. \tag{2.35}$$

Now we equate the dW_t terms from (2.34) and (2.35) and get

$$\nu_j(t) = -\sum_{k=\eta(t)}^{j-1} \frac{\Delta_k F_k(t)\sigma_k(t)}{1+\Delta_k F_k(t)}.$$

We still require a suitable μ_k so that the dt terms are compatible. We have the following relationship from (2.33),

$$D_{j+1}(t) = D_j(t)\frac{1}{1+\Delta_j F_j(t)},$$

which means that

$$\Delta_j F_j(t) D_{j+1}(t) = D_j(t) - D_{j+1}(t).$$

Note that $D_1(t)$ is a constant martingale. By induction, given that $D_j(t)$ is a martingale, we require $F_j(t)D_{j+1}(t)$ to be a martingale for $D_{j+1}(t)$ to be a martingale. Using Ito's formula we have,

$$d(F_j(t)D_{j+1}(t)) = F_j(t)D_{j+1}(t)\left[(\mu_j(t) + \nu_{j+1}(t)\sigma_j(t))\,dt + (\sigma_j(t) + \nu_{j+1}(t))\,dW_t\right].$$

Thus, we require $\mu_j(t) = -\nu_{j+1}(t)\sigma_j(t)$.

We can now write the SDE of the forward rates as

$$dF_j(t) = F_j(t)\sum_{k=\eta(t)}^{j} \frac{\Delta_k F_k(t)\sigma_k(t)\sigma_j(t)}{1+\Delta_k F_k(t)}dt + F_j(t)\sigma_j(t)\left(\rho_j(t)dW_t^1 + \sqrt{1-\rho_j^2(t)}dW_t^2\right).$$

2.3.3 Monte Carlo Implementation

An immediate approach would be to apply the Euler-Maruyama scheme to the SDE to arrive at

$$\hat{F}_j(t_i) = \hat{F}_j(t_{i-1}) + \hat{\mu}_j(t_{i-1})\hat{F}_j(t_{i-1})\Delta_{i-1} + \sigma_j(t)\hat{F}_j(t_{i-1})\sqrt{\Delta_{i-1}}\left(\rho_j(t)Z_i^1 + \sqrt{1-\rho_j(t)}Z_i^2\right).$$

However with the numeraire we have chosen, it is not tractable to preserve the martingale property, so we rather work with the log of the process to produce a more accurate discretisation:

$$\hat{F}_j(t_i) = \hat{F}_j(t_{i-1}) \exp\left(\left(\hat{\mu}_j(t_{i-1}) - \frac{1}{2}\sigma_j(t)^2\right)\Delta_{i-1} + \sigma_j(t)\sqrt{\Delta_{i-1}}\left(\rho_j(t)Z_i^1 + \sqrt{1-\rho_j(t)}Z_i^2\right)\right)$$

where

$$\hat{\mu}_j(t_{i-1}) = \sum_{k=i}^{j} \frac{\Delta_k \hat{F}_k(t_{i-1})\sigma_k(t)\sigma_j(t)}{1 + \Delta_k \hat{F}_k(t_{i-1})},$$

for $j = i, \ldots, N-1$, where $(Z_i^1, Z_i^2) \sim \mathcal{N}(0,1)$.

Discretisation error does arise in the proposed log process, due to the fact that the drift is state dependent. In particular, the evolution takes place assuming that the drift is constant and equal to the drift at the beginning of the time period.

We can produce a better estimate by following the approach suggested by Hunter *et al.* (2001). The idea of which is to evolve the rates over a time period, then using the same normal variates, re-evolve the rates over the same period but using an average of the terminal and initial drift. This is referred to as the predictor corrector method.

Chapter 3

Pricing Coupon Bearing Bonds

In South Africa, the strike of a bond option is always quoted in yield-to-maturity[1] (YTM). Ultimately though, the option is still settled using price.

The JSE's (Johannesburg Stock Exchange) Glit Clearing House developed a mechanism for converting YTM to the corresponding price of a coupon bearing bond, called the GCH formula. The GCH formula only applies to generic bonds, i.e., bonds yielding fixed (including zero) semi-annual coupons with one coupon coinciding with the anniversary of the maturity of the bond, on which date the entire capital value of the bond is redeemed. (Bond Exchange South Africa, 2005)

We also modify the GCH formula to provide a mechanism to convert a short rate (for the VM and HWM) and a simple forward curve (for the LMM2) into bond prices.

3.1 Standard GCH Formula

This section lays down the step-by-step method specified by Bond Exchange South Africa (2005) to convert YTM to a rounded all-in price for the corresponding bond. When pricing the American option, the formula is used to convert strike YTM into strike price. It is worth noting that a constant strike YTM implies a varying strike price through time and also that price is inversely proportional to YTM.

Firstly, the following variables are defined for the bond we are pricing:

1. The settlement date (S) = trade date (t) + 3 business days.
2. The maturity of the bond (M_B).
3. The coupon rate (C_B) quoted per annum.
4. Last coupon date (LCD) is the last coupon payment date **on or before** S.
5. The next coupon date (NCD) is the date of the next coupon payment after S.

[1] The rate of return anticipated if the bond is held until maturity.

6. The books-closed date (BCD) occurs 10 days before the NCD.

7. The number of remaining coupon dates (N) after the NCD and including the coupon paid on the maturity of the bond.

Now we calculate:

8. The cum/ex flag,

$$\begin{aligned} \text{CUMEX} = 1 \qquad & \text{if} \qquad \text{S} < \text{BCD} \\ = 0 \qquad & \text{if} \qquad \text{S} \geq \text{BCD}. \end{aligned}$$

9. The number of days of accrued interest,

$$\begin{aligned} \text{DAYSACC} = \text{S} - \text{LCD} \qquad & \text{if} \qquad \text{CUMEX} = 1 \\ = \text{S} - \text{NCD} \qquad & \text{if} \qquad \text{CUMEX} = 0. \end{aligned}$$

10. Coupon amount payable on each coupon date,

$$\text{CPN} = \frac{\text{C}_B.\text{nominal}}{2}.$$

11. Coupon payable on the NCD,

$$\text{CPN}_{NCD} = \text{CPN.CUMEX}.$$

12. The semi-annual discount factor (F),

$$\text{F} = \frac{1}{1 + \frac{\text{YTM}}{200}}.$$

The following two steps relate to whether the bond has more or less than 6 months to expiry.

13. The broken period (BP), measured in half years,

$$\begin{aligned} \text{BP} = \frac{\text{NCD} - \text{S}}{\text{NCD} - \text{LCD}} \qquad & \text{if} \qquad \text{NCD} \neq \text{M}_B \\ = \frac{\text{NCD} - \text{S}}{365/2} \qquad & \text{if} \qquad \text{NCD} = \text{M}_B. \end{aligned}$$

14. The broken period discount factor (BPF),

$$\begin{aligned} \text{BPF} = \text{F}^{\text{BP}} \qquad & \text{if} \qquad \text{NCD} \neq \text{M}_B \\ = \frac{\text{F}}{\text{F} + \text{BP}.(1 - \text{F})} \qquad & \text{if} \qquad \text{NCD} = \text{M}_B. \end{aligned}$$

The following six steps are outputs of the GCH formula, with the final step producing the value we are concerned with.

15. Unrounded accrued interest[2] (ACCRINT),

$$\text{ACCRINT} = \frac{\text{DAYSACC.C}_B}{365}.$$

16. The rounded accorded interest (ACCRINT_r),

$$\text{ACCRINT}_r = \textbf{round}\Big[\text{ACCRINT}, \text{nRound}^3\Big].$$

17. Unrounded all-in price (AIP),

$$\text{AIP} = \text{BPF}\left[\text{CPN}_{NCD} + \text{CPN}.\frac{\text{F}(1-\text{F}^N)}{1-\text{F}} + \text{redemption}.\text{F}^N\right].$$

18. Unrounded clean price (CP),

$$\text{CP} = \text{AIP} - \text{ACCRINT}.$$

19. Rounded clean price (CP_r),

$$\text{CP}_r = \textbf{round}\Big[\text{CP}, \text{nRound}\Big].$$

20. And finally the rounded all-in price (AIP_r)

$$\text{AIP}_r = \text{CP}_r + \text{ACCRINT}_r.$$

The BCD represents the final settlement date on which a purchaser of the bond can receive the following coupon payment. This convention creates a price discontinuity three business days before the BCD[4], instead of on the coupon date itself. The magnitude of the discontinuity is equivalent to the coupon payment happening on the NCD but discounted slightly due to its shift in position. Figure 3.1 illustrates the bond price discontinuity position and time varying strike price.

3.2 Modified GCH Formula

The previous chapter describes the evolution of interest rates in the VM, the HWM and the LMM2. To ultimately price the bond option we need to convert the interest rate paths, from each model, into bond price paths. Although the VM and HWM evolve the short rate and the LMM2 evolves the entire forward curve, the

[2] This is an "actual/365" interest calculation convention.

[3] The number of decimal places to which the prices are rounded. The GCH formula specifies 5 decimal places.

[4] It occurs three days before the BCD because of $t + 3$ bond settlement in South Africa

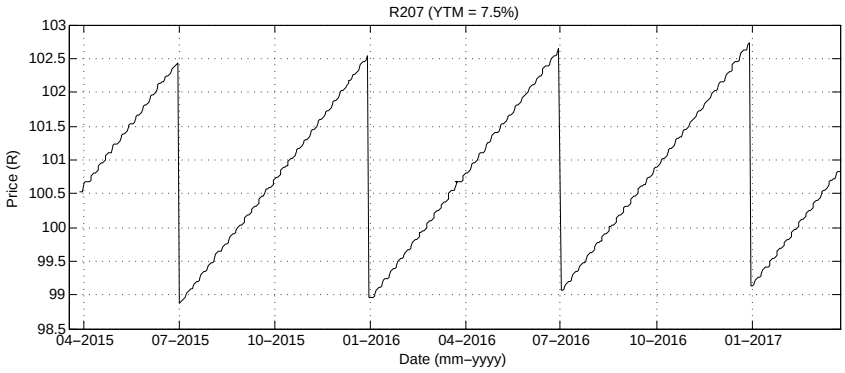

Fig. 3.1: YTM converted into time varying strike prices (K_t) using the standard GCH bond formula. The R207 bond is priced on a notional and redemption of R100 and has an annual coupon rate of 7.25%. Coupon dates are 15-Jan and 15-Jul and the bond matures in 2020. Refer to Appendix A for the specifics of other South African government coupon bearing bonds.

methodology to price bonds under either model is essentially the same, bar one step.

Pricing a coupon bearing bond, in either model framework, requires the remaining cashflows associated with the bond to be present valued while still adhering to the conventions specified in Section 3.1.

Firstly, Steps 1-11 in Section 3.1 are followed. Then Steps 15 and 16 are calculated. The AIP is calculated by summing the present values of the remaining cashflows. In both VM and HWM, we use (2.10) and (2.27) respectively to imply discount factors for the remaining cashflows using the short rate at that time and state. In the LMM2, we calculate the AIP by converting the simple forward curve associated with a specific time and state into a discount curve for the remaining cashflows. After which, the remaining rounding conventions in Steps 18-20 are applied to calculate the AIP_r. For illustrative purposes, Figure 3.2 is presented.

To ease computation in `MATLAB`, it is important to realise that at each time step the bond has the same characteristics in every state of the world. Meaning the steps mentioned in the previous paragraph only have to be calculated once at each time step despite the various states. Vectorisation in `MATLAB` is utilised to drastically decrease computing time.

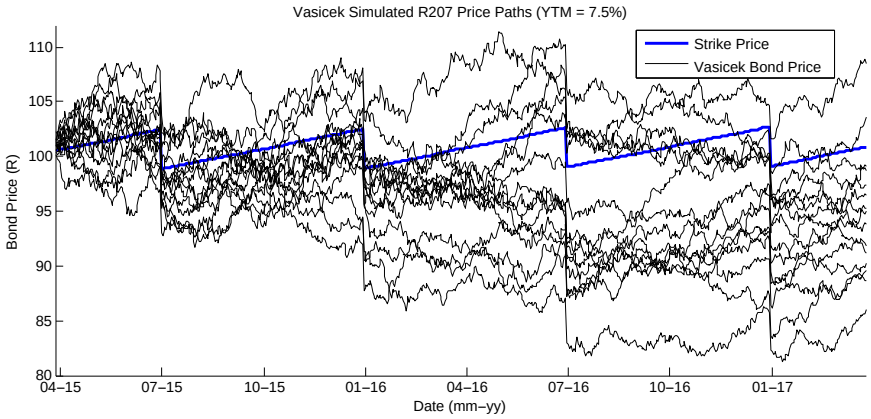

Fig. 3.2: A random sample of VM short rate paths converted into bond price paths for the R207 using the modified GCH bond formula.

Chapter 4

American Option Pricing

At each point in time, the holder of an American option compares the intrinsic value of the option to the continuation value. If the intrinsic value is higher than the continuation value, the option is exercised. Pricing an American option consequently requires the optimal exercise strategy to be determined. Unfortunately the optimal strategy often requires highly complex calculations, making analytical solutions to American options unattainable and hence numerical methods must be implemented (Stentoft, 2004).

This chapter describes two numerical methods for pricing American options: the tree method and the Monte Carlo least-squares method as proposed by Longstaff and Schwartz (2001).

4.1 Tree Method

Whether we are trying to price the option using a binomial tree or trinomial tree, the method is essentially the same. Both the VM bond price tree and the HWM bond price tree must be converted into intrinsic value trees. The intrinsic value at a node (i, j) is

$$IV_{i,j} = \left[\lambda \left(AIP_{r(i,j)} - K_j\right)\right]^+$$

where, $AIP_{r(i,j)}$ is the rounded all-in price for the coupon bearing bond at node (i, j), K_j is the strike price for time t_j and $\lambda = +1$ for a call option and $\lambda = -1$ for a put option.

V_0, the price of the option at inception, is calculated by working backwards from option maturity to the initial node and taking the maximum of the intrinsic value and the discounted expected continuation value at every node,

$$V_{i,j} = \max\left(IV_{i,j}, CV_{i,j}\right),$$

where the discounted expected continuation value for the VM tree is

$$CV_{i,j} = \mathrm{e}^{-r_{i,j}\Delta t}\left[\frac{1}{2}V_{i,j+1} + \frac{1}{2}V_{i+1,j+1}\right]$$

and the discounted expected continuation value for the HWM tree is

$$CV_{i,j} = \mathrm{e}^{-r_{i,j}\Delta t}\left[P_u V_{u,j+1} + P_m V_{m,j+1} + P_d V_{d,j+1}\right],$$

where, $r_{i,j}$ is the short rate at node (i,j) and P_u, P_m,P_d are the probabilities assigned to an up, middle and down movement from node (i,j). $V_{u,j+1}$, $V_{m,j+1}$ and $V_{d,j+1}$ refer to the value of the option at the next time step associated with an up, middle and down movement. $V_{u,j+1}$, $V_{m,j+1}$ and $V_{d,j+1}$ are specified in this way, instead of indexing with reference to i, as a result of the pruning feature inherent in the HWM trinomial tree.

4.2 Monte Carlo Least-Squares Method

This section describes the Monte Carlo least-squares from Longstaff and Schwartz (2001).

The key insight underlying this pricing approach is that the conditional expectation, at each time point, can be estimated using a least-squares regression on the cross section of the simulations (Longstaff and Schwartz, 2001). A complete specification of the optimal exercise strategy along each path is obtained with this method, and correspondingly V_0 can be calculated.

The least-squares approach is extremely convenient because it is readily applicable to any interest rate model in a Monte Carlo setting, including models evolved with multiple stochastic factors (Longstaff and Schwartz, 2001)

Firstly, we assume the conditional expectation over all simulated paths, at a specific time step j, can be written as

$$\mathbf{E}\left[\mathrm{e}^{-r_j\Delta t} CV_j(X_j)\middle| X_{j-1} = x\right] = f(\bar{\beta}_{j-1}, x),$$

where $CV_j(X_j)$ refers to the vector of option continuation values at time step j as a function of the vector of simulated bond prices X_j and $f(\bar{\beta}_{j-1}, x)$ is some suitable parametric function. It is interpreted as the expected value of the option conditioned on the option not having been exercised before $j-1$ and the bond price having the value $X_{j-1} = x$. The parameter vector $\bar{\beta}_{j-1}$ is found by regressing the realised payoffs, on each exercise date, from continuation values on the function of option prices. Using this estimate of conditional expectation, it is then possible to decide whether or not to exercise early or continue holding the option until the next exercise date.

The original algorithm presented by Longstaff and Schwartz (2001) computes a matrix of stopping rules for each path and then discounts the stopped cashflows to the valuation date. However we have chosen to apply the early exercise decisions as

they become available, thus discounting the cash flows incrementally. This simplifies the algorithm as it ensures that we always have the pathwise continuation values at hand.

To perform the least squares optimisation, we assume the functional form

$$f(\bar{\beta}, x) = \sum_{r=0}^{R} \beta_r \phi_r(x)$$

in terms of the basis function $\phi(x)$, $0 \leq r \leq R$ and $\bar{\beta} = [\beta_0, \beta_1, \ldots, \beta_R]^T$.

Many different basis functions may be selected including Hermite, Legendre, Chebyshev, Gegenbauer and Jacobi polynomials (Longstaff and Schwartz, 2001). This dissertation however makes use of the Lageurre polynomials

$$\begin{aligned}
\phi_0(x) &= 1, \\
\phi_1(x) &= 1 - x, \\
\phi_2(x) &= 1 - 2x + \frac{x^2}{2} \\
\phi_r(x) &= \frac{\mathrm{e}^x}{r} \frac{d^r}{dx^r}(x^r \mathrm{e}^{-x}).
\end{aligned}$$

as suggested by Longstaff and Schwartz (2001). The regression in this dissertation is performed using polynomials to the third degree (i.e. $0 \leq r \leq 2$).

If we define $y_j \in Y$ as the realised continuation values, then we wish to minimise the mean square error

$$\mathbf{E}\left[\left(y_j - \sum_{r=0}^{R} \beta_r \phi_r(x_j)\right)^2\right]$$

with respect to the coefficients β_r. Differentiating this expression with respect to β_s and equating to zero, we have

$$\sum_{r=0}^{R} \beta_r \mathbf{E}\left[\phi_r(x_j)\phi_s(x_j)\right] = \mathbf{E}\left[y_j \phi_s(x_j)\right], \tag{4.1}$$

for $0 \leq s \leq R$. Reformulating (4.1), we can obtain

$$\bar{\beta} = (FF^T)^{-1}FY \quad \text{and} \quad f(\bar{\beta}, X) = F^T\bar{\beta},$$

where X and Y are written as column vectors and F is the matrix

$$F = \begin{bmatrix}
\phi_0(x_1) & \phi_0(x_2) & \ldots & \phi_0(x_n) \\
\phi_1(x_1) & \phi_1(x_2) & \ldots & \phi_1(x_n) \\
\phi_2(x_1) & \phi_2(x_2) & \ldots & \phi_2(x_n)
\end{bmatrix}.$$

Furthermore, a renormalisation procedure is implemented to combat numerical scaling errors that occur when inverting matrices containing very small values; we divide all cashflows and bond prices by the strike price for that time step and perform the regression on the normalised values. Note that the option value is unaffected as we discount back the unnormalised cashflows (Longstaff and Schwartz, 2001).

Chapter 5

Model Analysis

There are always two parties involved in a derivative contract: the buyer and the seller. Both parties utilise pricing models to provide additional information about the derivative. In particular, the buyer of an American option can use a pricing model to calculate the optimal exercise time, whereas the seller of an American option can use a pricing model to calculate greek sensitivities and provide a hedge strategy to remove undesired exposure. In this dissertation we assess the effectiveness of the developed pricing models in each world scenario from the perspective of both buyer and seller.

5.1 Exercise Strategy

As mentioned previously, the buyer of an American style derivative pays a premium based on the assumption that the optimal exercise strategy will be followed. Since the exercise strategy is model dependant, the choice of model becomes very important. If the model is simplified or unable to calibrate accurately to the market term structure, it will suggest a suboptimal strategy.

We can assume that by using a VM in a VM simulated world, a HWM in a HWM simulated world, or a LMM2 in a LMM2 simulated world, that one would be able to exercise optimally, modulo approximation and sampling error. Suboptimal strategies are however expected to occur, for example when using a VM, HWM or any single-factor model in a two-factor world such as the LMM2 world.

On each exercise date, the model is calibrated to the observable term structure and then used to price the option. The value of which can then be compared to the intrinsic value of the option. If, on any exercise date, the calculated option value is lower than the intrinsic value, the optimal decision would be to immediately exercise the option.

The return from using a specific model can then be calculated by discounting the exercise value to the valuation date of the option. Bear in mind that a model

may not suggest an early exercise strategy. In such a case, the exercise value at the maturity of the option is discounted to the inception date. The present valued returns can be compared across the various models to assess which model is optimal.

5.2 Hedging

5.2.1 Hedge Portfolio

When delta hedging, we are interested in removing the sensitivity of the option to changes in the underlying. We set up a self financing replicating portfolio[1]

$$\pi_t = \Delta_t B_t + \Xi_t M_t$$

which is comprised of a Δ_t holding in the underlying coupon bearing bond, B_t, and a Ξ_t holding in a money market account, M_t. The holding in the money market is selected to ensure the portfolio is self financing.

On each hedge date, the delta of the option is approximated using a pricing model and the holding in the underlying is readjusted to the new delta value. The change in delta position over each time step will incur a profit or a loss depending on the position in the underlying and its movement in the market over the previous time step, however either way the Ξ_t holding in the money market is adjusted to account for this change.

We know that in an idealised setting of continuous trading in a complete market with no transactional costs, the payoff of a contingent claim can be perfectly replicated through trading in underlying assets (Glasserman, 2004). These idealisations do not hold in reality and we expect to incur a final hedge profit or a loss.

In a model defined world, we simulate a number of potential future interest rate paths over the life of the option. We then test the effectiveness of a pricing model by aggregating the hedge profit and losses (PnL's) over all simulated interest rate paths to produce a PnL distribution. We are trying to remove risk by hedging, so the best model is one that results in a PnL distribution with zero expected return and a small variance. A model that produces a positive expected return may seem appealing, however in different market conditions the model could easily produce an expected loss.

5.2.2 Delta Approximation

The delta of an option is the first derivative of the price with respect to the underlying instrument. The delta of this option cannot be calculated in closed form, so

[1] The self financing condition only holds true if we ignore the costs of transacting in the market, which we do for the purposes of this dissertation.

instead it is approximated using the central difference method.

Consider a pricing model that depends on a parameter θ ranging over some interval on the real line. For each value of θ we have a mechanism for generating a random variable $y(\theta)$, representing the discounted payoff of an option. The price of the option is then defined as

$$f(\theta) = \mathbf{E}\left[y(\theta)\right].$$

The derivative estimation problem therefore consists of estimating $f'(\theta)$, the derivative of f with respect to θ, where θ could be any of the market parameters that influence the price. In the case of delta hedging in this dissertation, θ refers to the price of the coupon bearing bond.

The delta of the option is calculated through a central difference approximation around the current value of the underlying θ,

$$\Delta_t(\theta, h) = \frac{f(\theta + h)_t - f(\theta - h)_t}{2h},$$

where h represents a small bump applied to the underlying. In this dissertation we select $h = \text{R}0.10$ for an underlying bond with a notional of R100.

We make use of common random numbers when calculating the delta in a Monte Carlo model, i.e the same random numbers are used to calculate $f(\theta+h)_t$ as $f(\theta-h)_t$. This reduces the variance of the calculation.

Chapter 6

World Scenarios

This chapter describes the three world scenarios in which our pricing models are tested for exercise optimality and hedge effectiveness. Furthermore, this chapter also describes how the models are calibrated in each world scenario.

In this dissertation, we avoid using actual market data. Calibration under those conditions can be very complex and thus lie outside of our scope. Instead, in each world we assume parameters and generate market curves in such a way that the calibration process is minimised but still accurate.

6.1 VM World

To specify the VM world, we simply assume constant realistic parameters for the VM: $r_0 = 7\%$, $\sigma = 2.5\%$, $b = 9\%$ and $\alpha = 10\%$.

We generate jointly Gaussian short rate and discount factor paths over the duration of the option, using (2.17) and (2.20). The short rate paths can be converted into bond price paths by following Section 3.2. The initial yield curve implied from the VM can be seen in Figure 6.1. Subsequent yield curves at time t can be generated by setting $r_0 = r_t$.

The HWM is easily calibrated at every time step. Firstly, by using $\alpha = 10\%$ and $\sigma = 2.5\%$ and then further passing the HWM the implied VM yield curve for the short rate at that time step. The HWM is essentially VM when we select constant calibration parameters, however we still choose to test the HWM trinomial tree in this world as it differs to the VM binomial tree in its state-space granularity.

6.2 HWM World

We start by specifying the same parameters we use in the VM world, i.e. $r_0 = 7\%$, $\sigma = 2.5\%$, $b = 9\%$ and $\alpha = 10\%$. At time $t = 0$, we use the VM to imply an initial market yield curve with these parameters. At this point, the initial market curve is

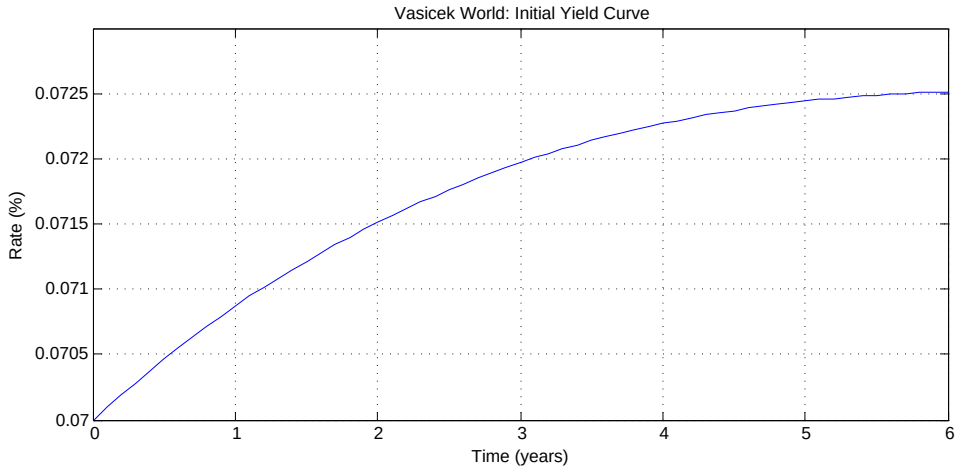

Fig. 6.1: The initial implied yield curve in the Vasicek world scenario

the same as in Figure 6.1.

Now for the HWM world to differ from the VM world we need to introduce time dependant mean reversion. We do so by applying "noise" to this initial Vasicek yield curve. We perturb the rates at time $t = \{1, 2, 3, 4, 5, 6\}$ years with scaled uniform normal variates. Setting the random number seed to three and scaling the uniform variates by $\frac{1}{1000}$, we are left with the perturbed curve seen in Figure 6.2 as our initial yield curve in the HWM world. Spline interpolation is used to smooth the curve.

We calibrate the HWM to the created yield curve and generate jointly Gaussian short rate and discount factor paths over the duration of the option using Section 2.2.4. The short rate paths are then converted into bond price paths.

At each time step along each short rate path a new yield curve is created by setting $r_0 = r_t$ and performing the same steps as mentioned above. The same normal variates are used to perturb each created yield curve. Furthermore, the positions of the perturbations remain fixed as we move through time. So for example, the new yield curve at $t = 0.1$ years will be perturbed at $t = \{0.9, 1.9, 2.9, 3.9, 4.9, 5.9\}$ years.

The VM is calibrated to the HWM world by selecting the same α, σ and r_t. The `MATLAB` function `lsqcurvefit` is then used to solve for a constant b that allows the Vasicek yield curve to best match the perturbed yield curve. The Vasicek calibration for the initial yield curve can be seen in Figure 6.2. A new b parameter is calculated at every time step and for every path.

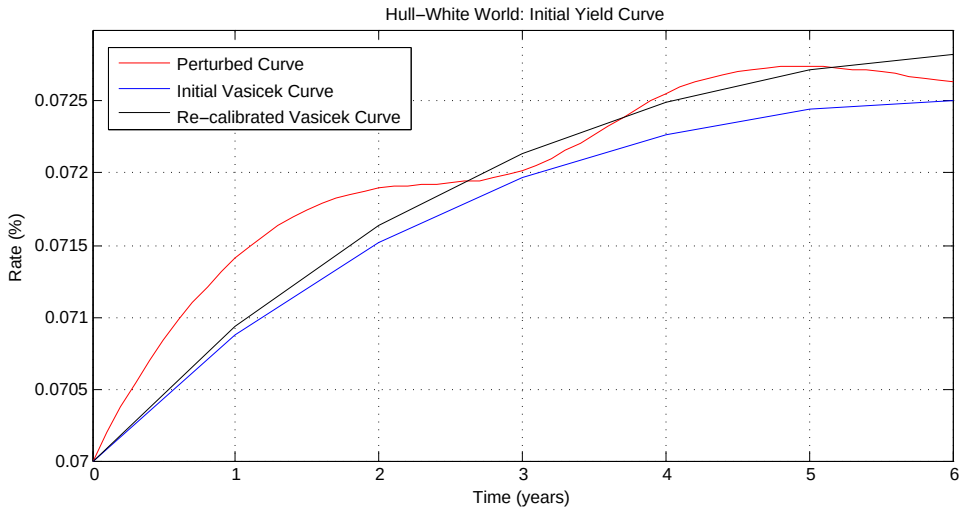

Fig. 6.2: A graph showing the initial yield curve in the HWM world. The Vasicek curve from which it was created is also presented as well as the newly calibrated Vasicek curve.

6.3 LMM2 World

We start with the same perturbed market curve that we generate for the HWM world (seen in Figure 6.2 and 6.3). We convert these continuously compounded spot rates into simple forward rates. The simple forward rates, shown in Figure 6.3, are used in the LMM2 as the initial market curve.

We define a function

$$\sigma_F^2(t,T,S) = \sigma^2 \frac{\left(1-e^{-\alpha(S-T)}\right)^2}{\alpha^2} \frac{\left(1-e^{-2\alpha(T-t)}\right)}{2\alpha}$$

that converts constant VM/HWM volatility, at time t, into forward Black volatilities, where the forward period is from T to S.

This specification of the LMM2 volatility and initial curve ensures that the HWM will be calibrated in this world. Calibration of the VM is completed by again using `MATLAB's lsqcurvefit` function to solve for a new approximate b parameter for the market curve at each time step.

The instantaneous correlation between forward rates is defined as

$$\rho_{ij} = \exp\left(-q|T_i - T_j|\right),$$

which follows from Joshi (2003), pg. 336. It is noted in Joshi (2003) that $q = 0.1$ tends to fit market correlations quite well, thus we select this value for q. This correlation matrix is described as a full-rank matrix. However, seeing as the LMM2 is

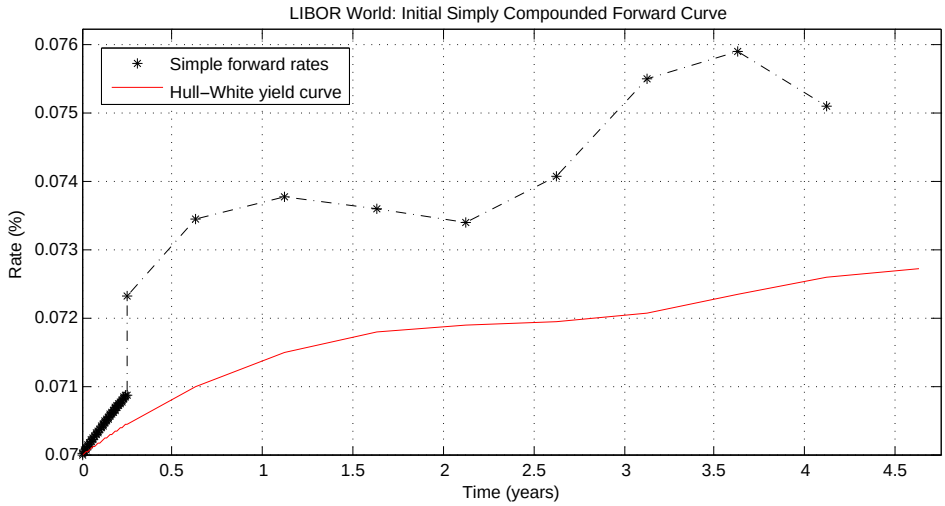

Fig. 6.3: A graph showing the initial LIBOR simply compounded forward curve for the LIBOR world scenario. The continuously compounded yield curve from which it was calculated is also graphed. Daily time steps are used for the period $t = [0, 0.25]$ years, after which time steps coincide with the semi-annual coupon payments.

governed by only two stochastic factors, we reduce the full-rank correlation matrix to a two-rank correlation matrix using the zeroed-eigenvalues reduced-rank approximation described in Brigo (2002).

The LMM2 world is now fully specified and the LMM2 is used to simulate evolutions of the forward rate curve. It is important to note that the volatility curve, the forward rate curve and the correlation matrix do not have constant time steps. Daily time steps are selected for the period between inception and maturity of the option, as we are concerned with daily exercisability. After which, time steps coincide with the remaining coupon dates of the underlying bond.

In the VM and HWM world the short rate paths are simulated from inception until the maturity of the option. Subsequently, the paths are converted to bond price paths. The process in the LMM2 world is different because of the way in which the LMM2 evolves the forward rates. In the LMM2 world, we evolve the entire forward curve using two correlated random variables. The evolved curve is used to discount the remaining coupon and notional payments to price the bond. We then move forward one time step, evolve the entire remaining forward curve using a new set of correlated random variables and price the bond. Simply, the evolution of the curves and the conversion to bond price paths must be done concurrently in this world.

Chapter 7

Results

The following labels are given to the pricing models:

V_{bin}	VM binomial tree
V_{mc}	VM Monte Carlo
HW_{tri}	HWM trinomial tree
HW_{mc}	HWM Monte Carlo
LMM_{mc}	LMM2 Monte Carlo

7.1 Option Prices

In this section we compare the pricing characteristics of V_{bin}, V_{mc} and HW_{tri}. The comparison is performed in the VM world, i.e., the initial market curve is implied from the VM with the following parameters: $r_0 = 7\%$, $\sigma = 2.5\%$, $b = 10\%$ and $\alpha = 9\%$. In this world HW_{tri} is simply a trinomial evolution of the VM. So the purpose of the comparison is basically to assess the robustness and accuracy of binomial, trinomial and Monte Carlo versions of the VM pricing models for American options written on a coupon bearing bond. Modulo some approximation error, HW_{mc} will calibrate to V_{mc}, making it redundant and therefore it has been omitted from this section.

Unless otherwise stated, all quoted prices are for a $T = 1$ year maturity American option written on the **R207** government bond with a notional and redemption equal to R100. The valuation date is $t_0 =$ 27-03-2015. The at-the-money strike YTM is approximately $K_y = 7.37\%$. When not testing the effect of Monte Carlo sample size, 50000 paths are simulated for each V_{mc} valuation. Daily exercisability is considered, so we select $\Delta t = \frac{1}{365}$. The characteristics of **R207** can be seen in Appendix A.

Firstly we assess the ability of the models to price European options on zero coupon bonds. This type of bond option does have a analytical solution which makes it a good validation test for the models. We can see from Figure 7.1 that

the HW_{tri} is consistently precise, regardless of moneyness; the maximum absolute pricing error (considering only the strike prices we assessed) for HW_{tri} is only 0.34% which occurs deep-out-the-money, at a strike price of R0.94. In comparison, at the same strike price, V_{mc} has a price standard deviation of 9.69% and V_{bin} mis-prices by 38.91%. With that being said, V_{mc} and V_{bin} are both sufficiently accurate when the option is at and in-the-money. The V_{bin} consistently prices the option within the 3xSTD bounds of V_{mc} within this strike range. The first order Taylor expansion in (2.16), for V_{bin}, which allowed us to state $b(j) \approx b$, seems to have caused a pricing bias as the model underprices the option irrespective of the strike. Indicating that the approximated b is most likely lower than its true value. This bias and a low end-point granularity implicit in a binomial tree, leads to the large pricing error for out-the-money options. V_{mc} shows no signs of bias as the prices fluctuate around the analytical value and the mean pricing error over all strikes is -0.011%.

Next we look at how the models perform against the analytical solution of European options written on coupon bearing bonds. The analytical *Option Price vs Strike Yield* plot can be seen in Figure 7.5. Looking at the left graphs in Figure 7.3 and Figure 7.4, we note that the models react very similarly to moneyness as they did with the zero coupon bond option: the HW_{tri} is a very precise numerical estimator of the actual solution irrespective of moneyness; V_{mc} fluctuates around the actual solution with the price standard deviation increasing with decreasing moneyness; and V_{bin} produces reasonable results when deep in-the-money but hugely underprices out-the-money call and put options.

Although the left graphs in Figure 7.3 and Figure 7.4 are useful to assess how the models react to moneyness, the different error magnitudes between in and out-the-money obscures most of the detail for in-the-money options. In order to visualise the results better we normalise the absolute pricing error at a specific strike with respect to the standard deviation of the V_{mc} price at the same strike. The right graphs in Figure 7.3 and Figure 7.4 are the normalised results. In such graphs for example, $(x, y) = (7\%, 2)$ represents an error that is two times bigger than the V_{mc} price standard deviation at $K_y = 7\%$. For European options, the absolute error is calculated using the closed form solution. The American option price, on the other hand, does not have a closed form solution, so we calculate the absolute error in terms of V_{mc} prices. Selecting V_{mc} as the benchmark allows us to note some interesting relationships between the pricing models. These relationships are discussed further on in the section.

We turn our attention back to European options on coupon bearing bonds: the right graphs in Figure 7.3 and Figure 7.4. Two horizontal lines have been plotted

at 3 and -3. These represent 3xSTD[1] bounds. We see that HW_{tri} sits comfortably within the 3xSTD band, with a maximum normalised error of 0.10 for the call and 0.11 for the put. V_{mc} fluctuates around zero but stays within the 3xSTD bounds as expected. V_{bin} only lies within the 3xSTD band when the option is very deep in-the-money. Additionally, V_{bin} underprices the call and overprices the put for deep in-the-money options, which further justifies the analysis that the approximated b is lower than the true value. At-the-money, V_{bin} underprices both call and put options by approximately 8xSTD. The reason we see the normalised V_{bin} error decreasing as we get increasingly out-the-money is because the standard deviation of V_{mc} increases at a faster rate than the V_{bin} deviates from the analytical price. This must not be confused with improved pricing accuracy.

Figure 7.2 is included to show the impact sample size has on the standard deviation of Monte Carlo option prices. Intuitively, the graph is a cross sectional slice into a *Option Price vs Yield Strike* plot, where the z-axis is sample size. As mentioned earlier, the at-the-money strike is 7.37% but the plot uses $K_y = 7.5\%$. Therefore, the call option is slightly in-the-money and the put option is slightly out-the-money[2].

We now discuss the American option results. We can see from the left graphs in Figure 7.5 and Figure 7.6 that the price of the American option always dominates that of the European option for a given strike. This result is expected, so the graphs are presented as further validation of the models. In addition, the American call price function is initially convex but then inflects at around $K_y = 10\%$ and becomes concave. To explain this, we let $K_y \to \infty$ and hence the time varying strike prices tend to zero. In this case, the price of the option would have to equal the price of the bond currently observed in the market, as a purchaser of the option could exercise it immediately and receive the underlying bond. The price function tends to the current bond price asymptotically as $K_y \to \infty$. The price of a bond has a convex inverse relation with YTM; further explaining the convexity observed in the price function. For the put option, we note a similar shift in curvature around $K_y = 4\%$. Using a similar argument as with the call option, we can deduce that the y-axis intercept for the put price must be equal to the sum of remaining cashflows, without discounting as $K_y = 0$, minus the current price of the bond.

The graphs on the right of Figure 7.5 and Figure 7.6 presents the pricing difference between both tree models and V_{mc} for an American. We know we can trust the V_{mc} when the option is at and in-the-money as it has the densest state space granularity and it is an unbiased estimator. Therefore, we use the the V_{mc} as the

[1] In statistics, 99.73% of values in a normal distribution will lie within a three standard deviation band around the mean.

[2] Due to the fact that YTM and bond price are inversely related

benchmark price and assess how the V_{bin} and HW_{tri} deviate from this price. We take the price difference at each strike and normalise it with the standard deviation of the V_{mc} price. We can see that for the put option, the HW_{tri} always stays within 3xSTD bounds of the V_{mc} price, whereas for the call option, the HW_{tri} escapes the 3xSTD bounds when $K_y > 10\%$. The V_{bin} on the other hand, deviates largely from the V_{mc} price when the option is at and out-the-money. What is more interesting is how both V_{bin} and HW_{tri} for a deep in-the-money option — for both call and put — begin displaying a highly correlated and linear deviation from the V_{mc}. This seems to indicate that both lattice methods are not accurately capturing the early exercise boundary for the American option. Furthermore, the point at which the correlated deviation begins is the same as the inflection point where the price function becomes concave, which suggests that the lattice methods are unable to capture this concavity as accurately as the Monte Carlo model.

The HW_{mc} prices for an American option have been included in Figure 7.7 for the sake of completeness. We can see how similar they are to the V_{mc} prices. There are a few differences at certain sample sizes which must be attributed to the approximation of integrals and partial derivatives in the calibration of HW_{mc}. The 3xSTD bounds have been plotted around the mean of all the V_{mc} prices which is essentially a mean of means and thus the most accurate value we have for the true price.

In Figure 7.8 we look at the effect of increasing maturity and so the random numbers used are the same for each V_{mc} calculated. The HW_{tri} and V_{mc} are once more very consistent with each other. Because the number of sample paths used in the V_{mc} remains constant at 50000, as we increase the maturity the path density also decreases which makes the early exercise boundary less accurate. Contrarily, the HW_{tri} has a consistent discretisation of space and therefore its price will have equal ability to capture the early exercise boundary whether the option is long dated or short dated. We also note that V_{bin} deviates from the other models in a linear manner, relative to the price standard deviation of V_{mc}.

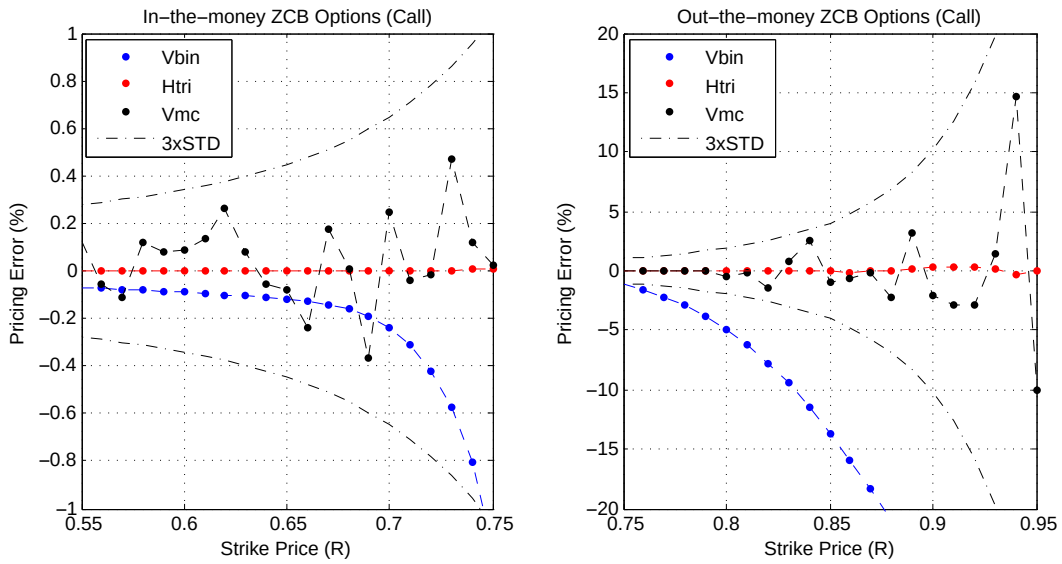

Fig. 7.1: Comparison of the pricing models and the analytical price of a $T = 1$ year European call option on a zero coupon bond which matures in $S = 4$ years. The models are compared at various degrees of moneyness. A strike price of R0.749 is at the money. **Note** the differing y-axes.

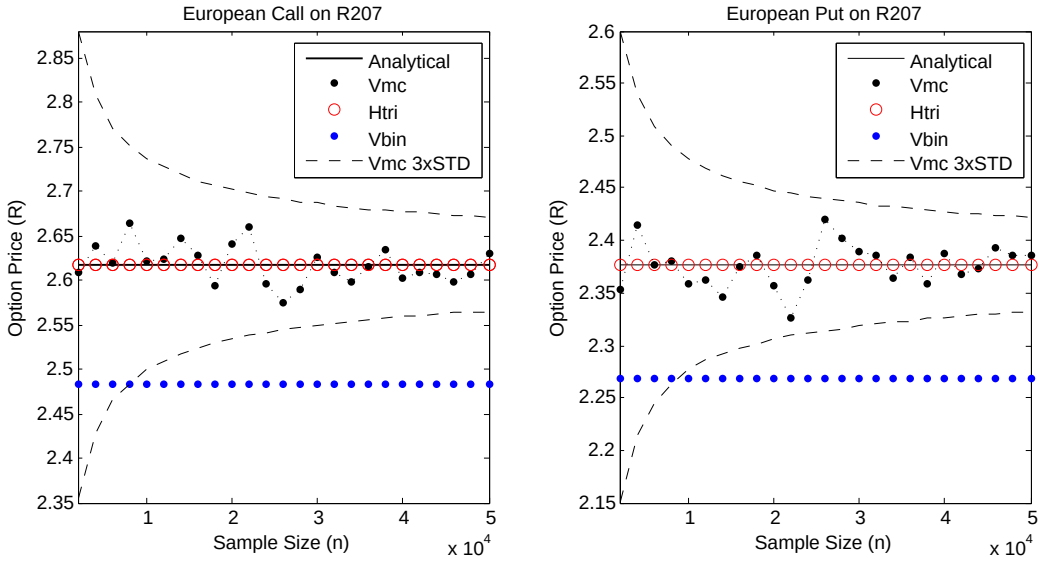

Fig. 7.2: Comparison of the pricing models and the analytical price of a European option on the R207 bond with $Ky = 7.5\%$. Sample size is varied for the V_{mc} model.

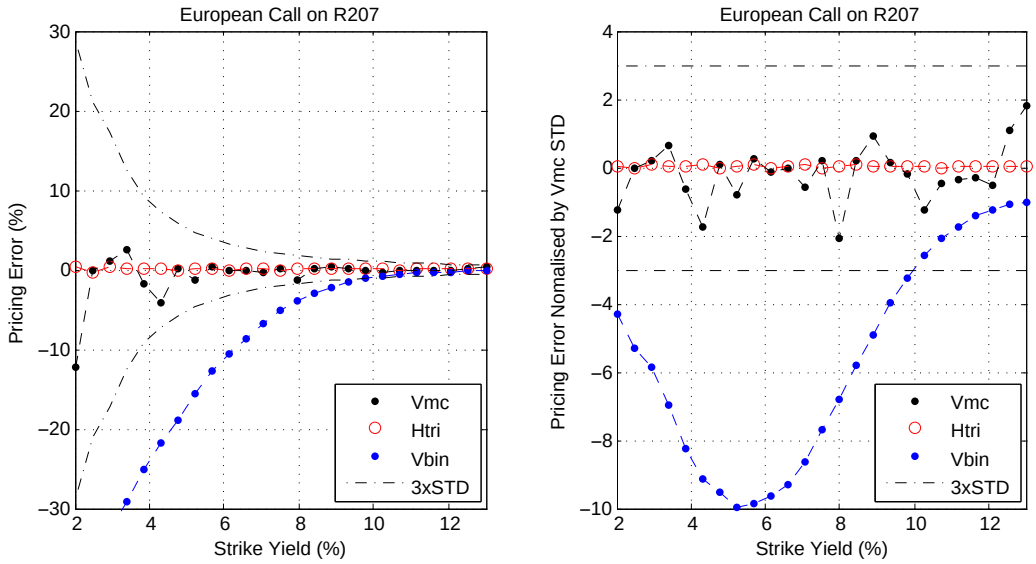

Fig. 7.3: Comparison of the pricing models and the analytical European call price, written on the R207, at various degrees of moneyness. `Left`: Percentage error of models compared to the analytical price. `Right`: Absolute pricing errors normalised by the standard deviation of the V_{mc} price.

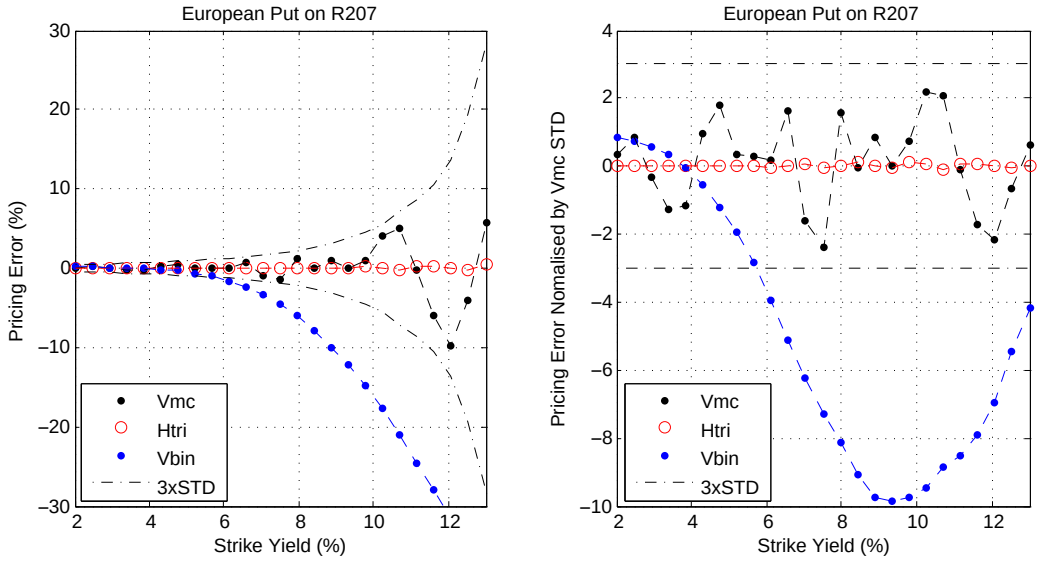

Fig. 7.4: Comparison of the pricing models and the analytical European put price, written on the R207, at various degrees of moneyness. `Left`: Model percentage error compared to the analytical price. `Right`: Absolute pricing errors normalised by the standard deviation of the V_{mc} price.

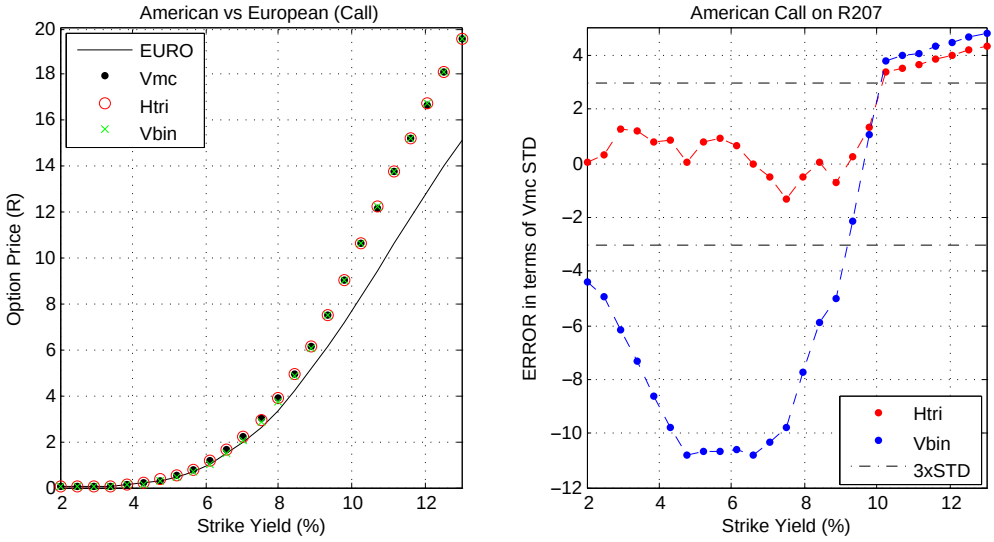

Fig. 7.5: `Left`: Comparison between American call prices obtained using the models and the analytical European price at various strikes. `Right`: Price difference between both tree models and V_{mc} for an American call option which are then normalised by the standard deviation of the V_{mc} price.

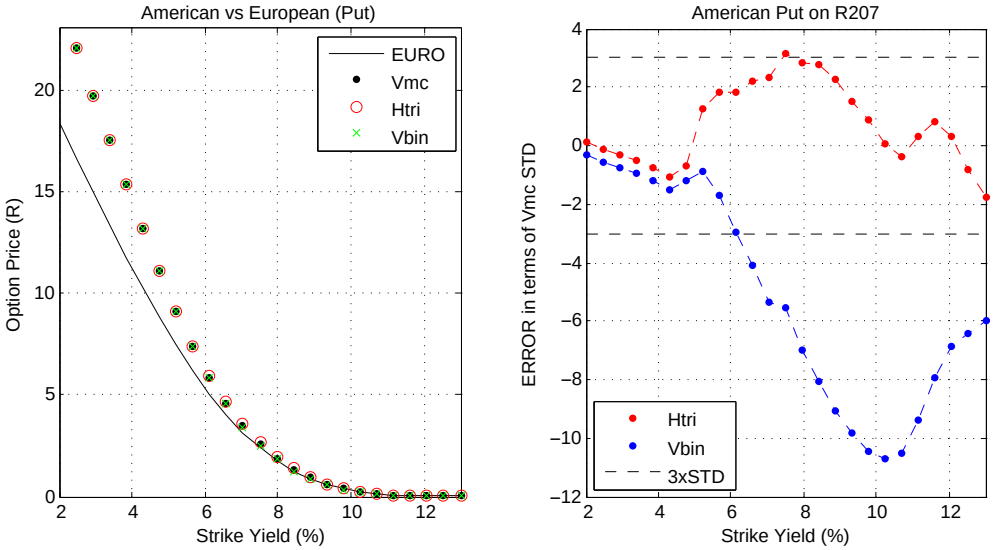

Fig. 7.6: `Left`: Comparison between American put prices obtained using the models and the analytical European price at various strikes. `Right`: Price difference between both tree models and V_{mc} for an American put option which are then normalised by the standard deviation of the V_{mc} price.

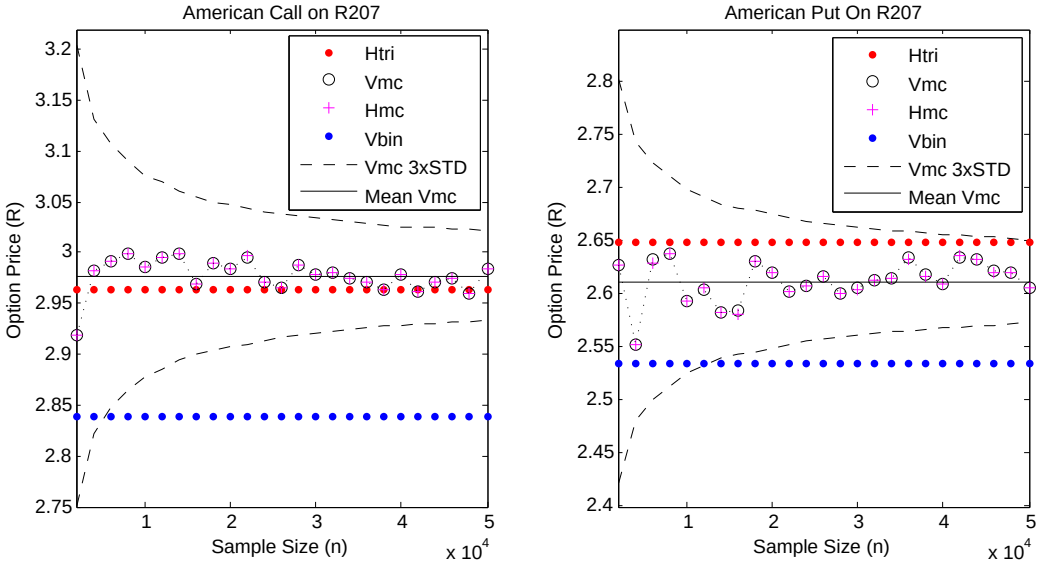

Fig. 7.7: Comparison of the pricing models on an American option, written on the R207 bond, with varying sample size; $K_y = 7.5\%$.

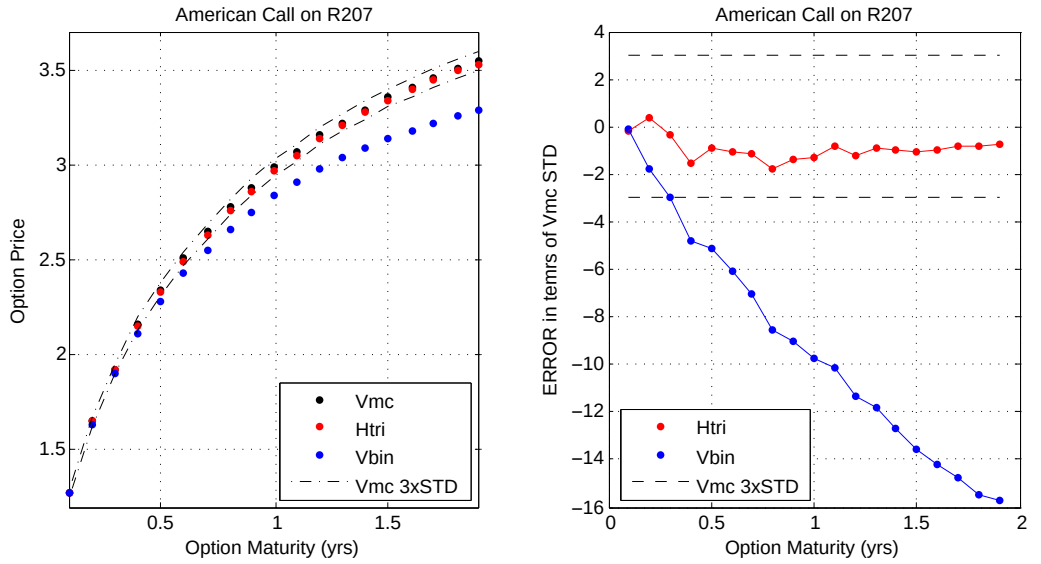

Fig. 7.8: Comparison of the pricing models on an American call option on the R207 bond with increasing maturity;

7.2 Efficiency

Pricing an American option on a coupon bearing bond is a computationally laborious process. The majority of the computation is focused on pricing the bond for each exercise date on each interest rate path. For example, if we wish to price a $T = 0.25$ year American option using V_{mc}, assuming daily exercise opportunity and 50000 sample paths, we need to price the R207 bond 4.6×10^6 times in total. Each time the bond is priced, we apply GCH pricing conventions and create a unique discount curve. As mentioned in Section 3.2, vectorisation in `MATLAB` was used at each time step to drastically reduce computing time.

The exercise and hedge analysis requires exponentially more computations. For each exercise date on each simulated world path, we require three option prices: two for the central difference delta calculation and one to check the option value against the intrinsic value. So for a $T = 0.25$ year option and using the V_{mc} with 50000 samples, for example, we require 13.8×10^9 calculations of the bond price to produce a 1000 path PnL histogram and its corresponding exercise strategy.

The computing times for a single option price and a 1000 path PnL histogram are presented in Table 7.1 and 7.2 respectively[3]. It is clear that computing power is the limiting factor for this dissertation. Hedging long dated options, with the purpose of producing a PnL distribution, is simply not viable using the currently available computing power. Ultimately for this dissertation, the PnL computation was parallelised manually over various computers; each computer handling a subset of the 1000 generated world paths.

Tab. 7.1: Average time, in seconds, taken to price an American option with varying maturities, considering daily exercisability. 50000 samples are used for the Monte Carlo models.

	Time (s)			
	$T = 0.25$	$T = 0.5$	$T = 0.75$	$T = 1$
V_{bin}	0.97	1.84	2.79	3.82
HW_{tri}	1.61	4.38	8.55	14.12
V_{mc}	9.30	19.16	26.09	31.81
HW_{mc}	16.14	29.68	42.95	51.77
LMM_{mc}	1374.51	8463.92	-	-

[3] The times were calculated using a mid-2010 MacBook Pro: 2.4 GHz Intel Core 2 Duo; 4 GB 1067 MHz DDR3. The displayed times are an average based on five seperate runs.

Tab. 7.2: Approximate time, in hours, taken to create a 1000 realisation hedge PnL histogram for an American option with varying maturities. 50000 samples are used for the Monte Carlo models.

	Time (hr)			
	$T = 0.25$	$T = 0.5$	$T = 0.75$	$T = 1$
V_{bin}	23	93	200	346
HW_{tri}	43	221	649	1428
V_{mc}	235	969	1979	3217
HW_{mc}	408	1500	3257	5235
LMM_{mc}	17373	427851	-	-

7.3 Exercise and Hedge Analysis

The analysis in this section is performed on a 3-month American option written on the **R207** South African government bond. The initiation date and maturity of the option are 01-06-2015 and 01-09-2015 respectively. The initiation date was selected to ensure the short-dated option spans a coupon payment. We consider daily time steps, $\Delta t = \frac{1}{365}$, on which delta-hedge positions are adjusted and the option may also be exercised. Seeing as the analysis is conducted in simulated worlds, we assume all days are trading days including weekends and public holidays. A total of 1000 interest rate paths are evolved in each world scenario.

When using a model to hedge the option, we assume that the buyer of the option is using the same model to exercise. This allows us to calculate the exercise and hedge statistics simultaneously.

Ideally we would have preferred to evolve more than 1000 interest rate paths and also to have performed the analysis on a longer dated option, however this is computationally un-feasible due to the complexity of the pricing problem.

7.3.1 Delta-Hedge

We start by comparing V_{bin},V_{mc} and HW_{tri} delta-hedge PnLs in the VM simulated world (depicted in Figure 7.9; quantified in Table 7.3 and 7.4). Option prices for the different models can be seen in Table 7.5.

As expected, all the hedge models perform well in this scenario: the PnLs are approximately normally distributed with means close to zero. The V_{mc} achieved the the lowest standard deviation in both call and put hedges. We note that the HW_{tri} performed slightly better than the V_{bin} in all aspects for both call and put, including mean, standard deviation, VAR and skewness. This seems to indicate

Tab. 7.3: The mean (μ) and standard deviation (σ) for daily delta-hedging PnL.

		Call			Put		
	Model	μ	σ	skew	μ	σ	skew
VM	V_{bin}	-0.014	0.157	-0.148	-0.027	0.158	-0.033
World	V_{mc}	0.003	0.141	-0.230	-0.020	0.140	-0.053
	HW_{tri}	0.009	0.149	-0.073	-0.002	0.150	0.008
HWM	V_{bin}	0.040	0.183	-0.275	-0.030	0.165	-0.146
World	V_{mc}	0.052	0.161	-0.272	-0.027	0.141	-0.240
	HW_{tri}	-0.011	0.151	-0.263	0.017	0.156	-0.099
	HW_{mc}	0.068	0.182	-0.029	-0.016	0.133	-0.368
LMM2	V_{bin}	-0.223	0.209	-0.862	0.197	0.246	-0.137
World	HW_{tri}	-0.257	0.194	-0.824	-0.172	0.210	-0.673
	LMM_{mc}	0.031	0.187	-0.372	-0.047	0.231	-0.534

that the improved state-space granularity of the trinomial tree has improved the ability to hedge the option when compared to the binomial tree. Additionally, the b approximation made in the V_{bin} could also be affecting the accuracy. That said, the expected loss from using the V_{bin} for the put is only about 2% of the option value. The HW_{tri} hedge produced a smaller PnL mean when compared to the V_{mc} for the put, with a value relative to the price of about 0.14% versus 1.52% respectively. Although not very significant, this highlights the variability of using a Monte Carlo model as a result of sampling error in the delta approximation.

Now we focus on the hedge performed in the HW+ world (depicted in Figure 7.10; quantified in Table 7.3 and 7.4). Option prices for the different models can be seen in Table 7.5.

Surprisingly, the VM binomial and Monte Carlo models performed quite well despite using a constant mean reversion level in a world where it is actually time dependant. In particular, the standard deviation of the V_{mc} put hedge PnL was better than the put hedge PnL from the HW_{tri}. That said, the Vasicek PnLs means ranged between 1.5 to 4.8 times bigger than the means of the HW_{tri} PnLs. The larger absolute means for the V_{mc} and HW_{mc} call hedge PnLs again highlights the variability of using a Monte Carlo model as a result of sampling error in the delta approximation, as they performed markedly better for the put option. In general, the V_{bin} performed the worst but still managed to produce a lower call option PnL mean than both the Monte Carlo models, thus proving itself quite useful for hedging this short dated option. Using Figure 7.8 we can assume that the usefulness of V_{bin}

Tab. 7.4: The 1% and 5% delta-hedge PnL VAR.

		Call		Put	
	Model	$\text{VAR}_{0.05}$	$\text{VAR}_{0.01}$	$\text{VAR}_{0.05}$	$\text{VAR}_{0.01}$
VM	V_{bin}	0.268	0.458	0.283	0.462
World	V_{mc}	0.245	0.387	0.247	0.403
	HW_{tri}	0.250	0.387	0.254	0.379
HWM	V_{bin}	0.283	0.434	0.316	0.457
World	V_{mc}	0.217	0.378	0.271	0.408
	HW_{tri}	0.284	0.431	0.252	0.402
	HW_{mc}	0.236	0.373	0.248	0.388
LMM2	V_{bin}	0.610	0.837	0.186	0.389
World	HW_{tri}	0.616	0.868	0.550	0.820
	LMM_{mc}	0.230	0.528	0.458	0.646

will decrease with longer dated options.

Finally, the hedge performed in the LMM2 world (depicted in Figure 7.11; quantified in Table 7.3 and 7.4). Option prices for the different models can be seen in Table 7.5.

In this world, we hedge using only V_{bin}, HW_{tri} and LMM_{mc} due to computational constraints. Furthermore, the LMM_{mc} only uses 10000 samples to calculate the delta on each hedge date, compared to 50000 samples which were used with the other Monte Carlo models in the previous worlds. In a real world situation, more samples would be used on a daily basis to calculate the delta and we would therefore expect improved hedge results.

It is clear from the PnL histograms of both V_{bin} and HW_{tri} that the single-factor Gaussian models were unable to capture the dynamics of the the log-normal two-factor world. Their PnL means range between 11-13% of the initial LMM_{mc} option value. The standard deviations of the different models are quite similar though, but this must be attributed to the small sample set used in the LMM_{mc} delta calculation. We also use this to explain the reason the LMM_{mc} created a skewed PnL distribution for the put option. With more samples, we would expect a more normal distribution. Despite this, LMM_{mc} performed as expected in this world: the PnLs are approximately normally distributed with means close to zero.

Even though the HW_{tri} calibrates very accurately to a given market curve, it performed similarly to the simple V_{bin}. This highlights the importance of the underlying assumptions in a model. A model that can calibrate extremely accurately still

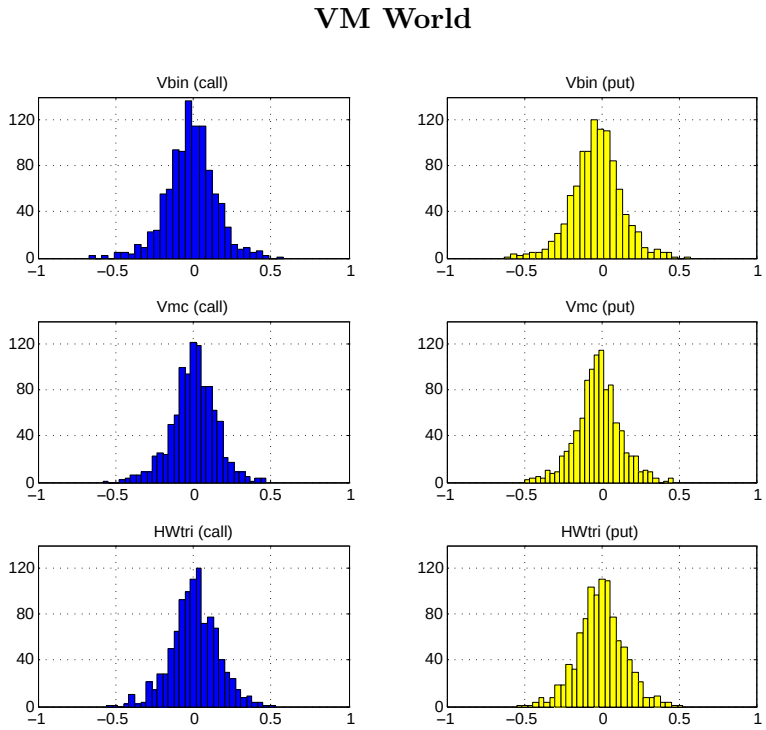

Fig. 7.9: PnL histogram from daily delta-hedging a $T = 0.25$ year American option in the VM world.

might perform poorly if its underlying assumptions are not in line with the world in which it is used.

The positive V_{bin} put PnL must not be seen as a good hedge. In a world with a different yield curve, this result could easily reverse and become negative. Looking at Figure 6.2, we see that the V_{bin} predicts a lower yield curve over the option tenor and therefore over prices the call and underprices the put in this world, relative to the well calibrated HW_{tri}. As a result, the model predicts a higher probability of exercise for the put option. While this is fine in an idealised setting where we assume the buyer of the option exercises using the same model, it will produce losses in a world where the buyer uses a more calibrated model, as the model will suggest trading a higher delta than is necessary which is costly if the model is in actual fact unlikely to be exercised.

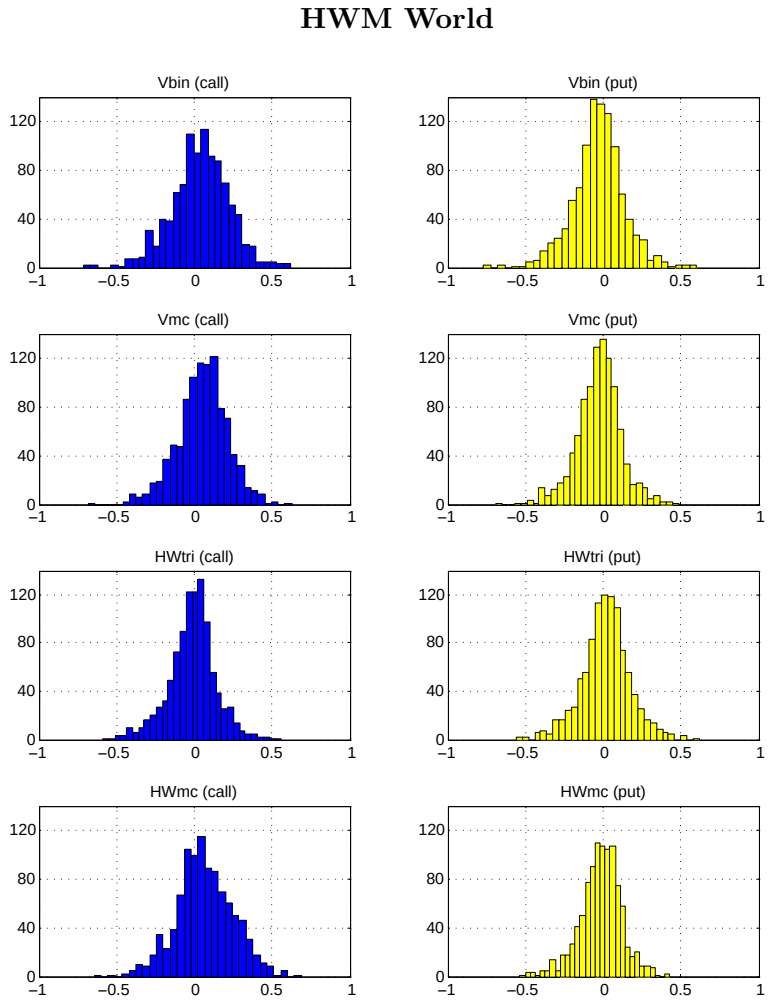

Fig. 7.10: PnL histogram from daily delta-hedging a $T = 0.25$ year American option in the HMM+ world.

7.3.2 Exercise Strategy

The total return statistic seen Table 7.5 represents the average return over all paths. These paths, however, include paths for which the option is always out of the money and no model would suggest exercise. Similarly, there are paths where the option is deep in the money, and a reasonable model would suggest exercising. The real test for a model is how well it performs in the situation where exercise is not clear-cut. Therefore, the suboptimal cost statistic is of more value. We calculate it by comparing the return only for the paths where the benchmark model suggests early exercise. For the first two worlds, the benchmark model is chosen to be the model which yields the highest total return, seeing as the models are so similar. Naturally,

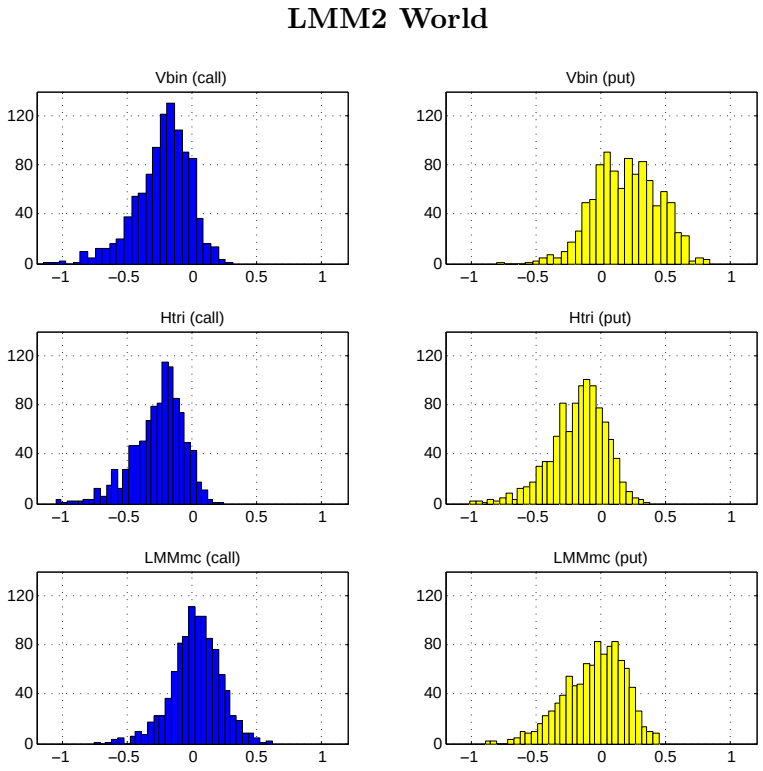

Fig. 7.11: PnL histogram from daily delta-hedging a $T = 0.25$ year American option in the LMM2 world.

we select the LMM_{mc} to be the benchmark in the LMM2 world. The suboptimal cost is in terms of R1 per R100 bond notional.

In the VM world, we see that the HW_{tri} produces the highest expected return over the simulated paths and is therefore the benchmark model. The V_{bin} performed better than the V_{mc} for both call and put options, which we attribute once again to the pricing variance inherent in a Monte Carlo model. So on a given day when exercising the option is a difficult decision (when the value of the option and the intrinsic value of the option are similar), a Monte Carlo model can easily suggest the incorrect decision due purely to sampling error. The suboptimal cost for V_{bin} is attributed to its low state-space granularity, relative to that of the trinomial tree. Essentially, this is the cost of using a binomial tree rather than a trinomial tree to exercise this American option.

In the HWM world, we once again select the HW_{tri} as the benchmark model. Most notably, the V_{bin}, V_{mc} and HW_{mc} did not suggest any early exercise for the call option, whereas the HW_{tri} suggested exercising 42% of the time. The total return cost of which equalled approximately 4% with a loss of R0.145 for the paths HW_{tri}

suggested exercising. For the put option, the suboptimal costs are even greater. Interestingly the V_{bin} performed the best relative to the benchmark model, however it still incurred a suboptimal exercise cost of -R0.475.

Interestingly, the HW_{mc} performed the worst in the HW++ world. We can only assume that the approximate calibration used in this model, coupled with its sampling error, are the reasons for this. Although both V_{bin} and V_{mc} use approximate calibration procedures, they have no real bias when averaged over all paths, i.e. for some curves they will over estimate the b parameter and for other paths they will under estimate the b parameter. With the HW_{mc} though, it seems to be systematically underestimating the the mean reversion level as it always produced a higher call price and a lower put price compared to HW_{tri}, which we know contains no approximation error.

We now turn our attention to the LMM2 world. Even though we only use 10000 samples to price the option using LMM_{mc}, it still performed much better than the single-factor models. In reality, more samples would be used and the model would be able to calculate the early exercise boundary even more accurately. For instance, the cost of using V_{bin} for the put option was -R3, which is extremely large for a R100 notional option.

LMM2 World

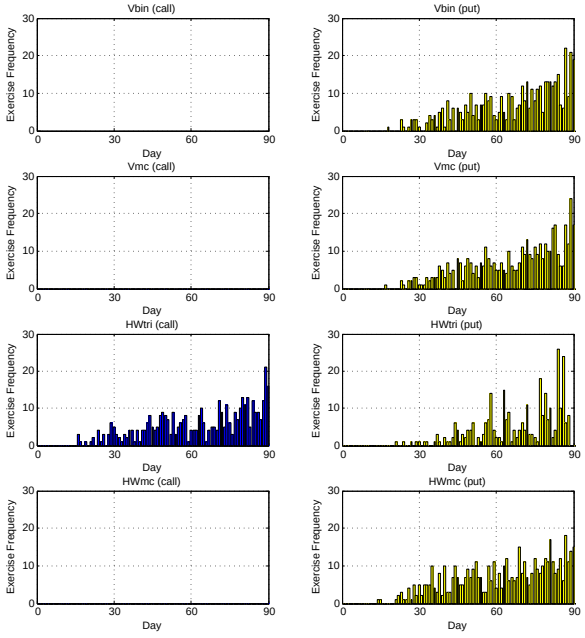

Fig. 7.12: Bar graph indicating the number of times the option was exercised on a given day in the HWM world.

Tab. 7.5: Suboptimal exercise cost for the call and put option. The * is placed next to the benchmark model in a specific world.

		Call				
	Model	Option value (R)	Total prob. of exercise	Prob. of exercising same path as benchmrk	Total return (R)	Suboptimal cost (R)
VM	V_{bin}	1.732	0.46	0.41	1.637	-0.010
World	V_{mc}	1.749	0.41	0.41	1.627	-0.031
	HW^*_{tri}	1.752	0.42	-	1.640	-
HWM	V_{bin}	1.681	0	0	1.559	-0.145
World	V_{mc}	1.697	0	0	1.559	-0.145
	HW^*_{tri}	1.662	0.42	-	1.620	-
	HW_{mc}	1.712	0	0	1.559	-0.145
LMM2	V_{bin}	1.681	0.08	0.07	1.765	-0.115
World	HW_{tri}	1.662	0.48	0.35	1.793	-0.362
	LMM^*_{mc}	1.933	0.36	-	1.800	-

		Put				
	Model	Option value (R)	Total prob. of exercise	Prob. of exercising same path as benchmrk	Total return (R)	Suboptimal cost (R)
VM	V_{bin}	1.313	0.40	0.35	1.476	-0.049
World	V_{mc}	1.321	0.38	0.35	1.462	-0.088
	HW^*_{tri}	1.334	0.35	-	1.481	-
HWM	V_{bin}	1.360	0.46	0.30	1.417	-0.475
World	V_{mc}	1.365	0.46	0.30	1.406	-0.510
	HM_{tri^*}	1.407	0.30	-	1.421	-
	HW_{mc}	1.352	0.51	0.30	1.343	-0.873
LMM2	V_{bin}	1.360	0.69	0.20	1.275	-3.004
World	HW_{tri}	1.407	0.28	0.18	1.669	-0.595
	LMM^*_{mc}	1.555	0.20	-	1.721	-

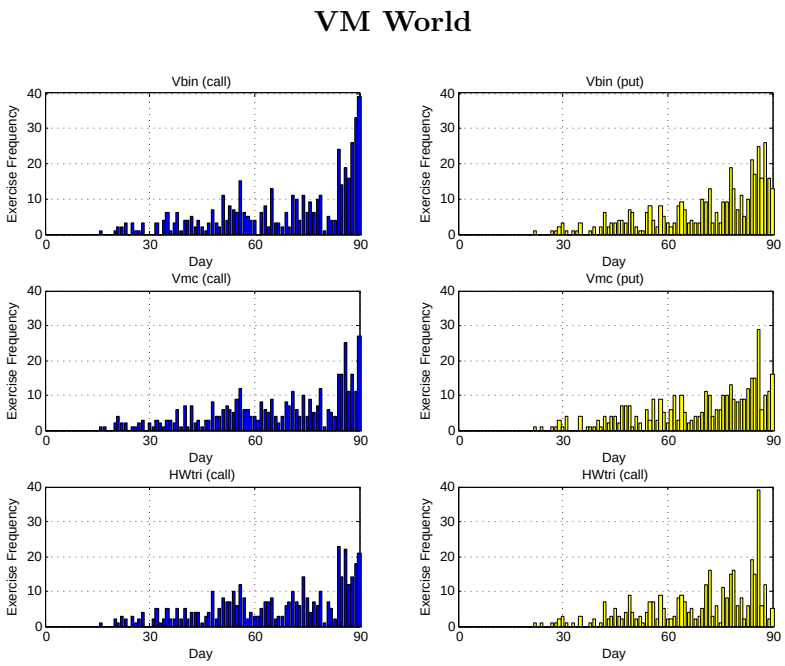

Fig. 7.13: Bar graph indicating the number of times the option was exercised on a given day in the VM world.

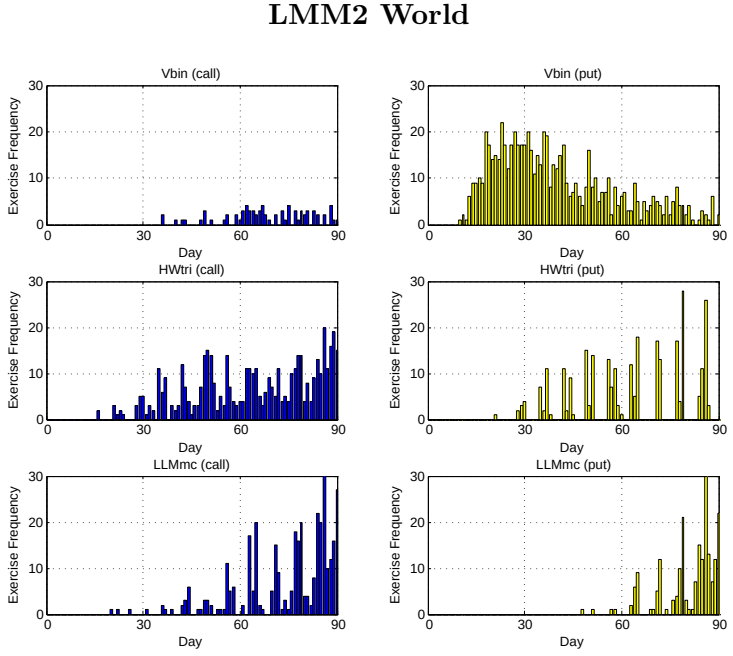

Fig. 7.14: Bar graph indicating the number of times the option was exercised on a given day in the LMM2 world.

Chapter 8

Summary and Conclusions

This dissertation studied the cost of following single-factor exercise and hedge strategies for American options in a world driven by two-factors. Various contributions emerge from this study.

- The holder of an American style interest rate derivative who uses a single-factor model to exercise the option will incur large losses through suboptimal strategies. Continually re-calibrating to the term structure in a market using a misspecified model does not compensate for its flaws. For options with high notionals, the losses can easily be in the order of millions. These losses can be avoided by simply following the optimal strategy suggested by a well specified, multi-factor model.

- The same goes for delta-hedging using a single-factor model in a two-factor world. The hedge is unpredictable and can incur large expected losses as a result of not capturing the term structure dynamics. Accurate calibration becomes more vital for American options due to the early exercise feature.

- Monte Carlo models can be quite volatile when used to exercise and hedge an American option. When calculating the exercise strategy, a Monte Carlo model can easily suggest the wrong strategy when the decision is not clear-cut, because of sampling error. Similarly for delta-hedging, a Monte Carlo model can yield an inaccurate delta. It is suggested that the largest possible number of samples are used in a real market scenario to reduce the price variance if a Monte Carlo model is employed.

- We develop a unique and efficient binomial implementation of the VM. Although it contains an approximation, the model still proves quite useful when used in a idealised single-factor world setting. That said, its accuracy does tend to weaken for longer dated options and for options far out-the-money.

- HW_{tri} is the most reliable single-factor model we implement. The trinomial evolution allows for good state-space granularity with no price variance and subsequently the model is able to capture the early exercise boundary better than the other single-factor models. Its pricing characteristics are very consistent regardless of moneyness. Furthermore, the implementation is far more efficient than the Monte Carlo models.

- The approximate calibration in HW_{mc} ultimately proved very inaccurate and actually made the model perform worse on average than the V_{bin} and V_{mc} in the HWM world. This seems to indicate that detailed but inaccurate calibration in a model can be worse than using a model with a simple calibration methodology.

- Monte Carlo models are very inefficient when compared to lattice models. We would expect this inefficiency to be a result of increased accuracy. While it does have a much higher state-space granularity, its pricing variance makes using Monte Carlo models for hedging and exercising American derivatives fairly volatile and in some cases even less accurate than the lattice models. For example, HW_{tri} is the most consistent and accurate single-factor model we implement.

The comments above make a strong case for moving away from simplistic models with complicated calibration methods to more realistic multi-factor market models. Although this should be the case for all derivative styles, the case is made even stronger for American options where accurate specification and calibration is drastically more important when trying to capture the early exercise boundary. This is a result of the fact that early exercise exposes the option to the whole discount curve.

Bar the results, this dissertation also concisely describes the implementation of various interest rate models and also how to accurately price American options on coupon bearing bonds in the South African market.

Chapter 9

Further Research

The concept of this dissertation could be extended to assess the impact of using single-factor models to exercise and hedge longer dated American options in a multi-factor framework. Judging by the computation times achieved in this dissertation, the hedge and exercise frequency would have to be far less than $\Delta_t = \frac{1}{365}$ in order to achieve this.

Furthermore, log-normal single-factor models could also be implemented and compared to Gaussian multi-factor models in a log-normal multi-factor framework. This would allow conclusions to be made on whether it is more important to use the correct number of factors or to use a model with the correct distributional assumption.

From Longstaff and Schwartz (2001): What is the extent to which single-factor models can be 'corrected' of their bias in a multi-factor framework? In other words, can the exercise decision implied by the single-factor model be overridden using information from multiple points on the term structure.

Including the possibility of default would further improve the pricing model in a world where default is now possible. Governments are no longer seen as riskless since the financial collapse in 2008. This has introduced the concept of a multi-curve framework. The swap curve is now considered to be riskless, due to collateral regulations, and not the government bond curve. Therefore interest rates should be evolved using the government bond curve but cashflows should be discounted using the swap curve. The government bonds are risky so default probabilities should have to be incorporated when evolving their prices. Moreover, in South Africa the country's credit rating is very volatile. The probability of the country being up or downgraded could also be incorporated for improved pricing, exercising and hedging accuracy.

Bibliography

Bond Exchange South Africa (2005). Bond pricing formula. https://www.jse.co.za/content/JSEPricingItems/bond-pricing-formula-final.pdf [Accessed: 02-03-2015].

Brace, A., Musiela, M. and Gatarek, D. (1997). The market model of interest rate dynamics, *Mathematical finance* **7**(2): 127–155.

Brigo, D. (2002). A note on correlation and rank reduction.

Glasserman, P. (2004). *Monte Carlo methods in financial engineering*, Vol. 53, Springer.

Hull, J. C. and White, A. D. (1994). Numerical procedures for implementing term structure models I: Single-factor models, *The Journal of Derivatives* **2**(1): 7–16.

Hull, J. and White, A. (1990). Pricing interest-rate-derivative securities, *Review of Financial Studies* **3**(4): 573–592.

Hunter, C. J., Jäckel, P. and Joshi, M. S. (2001). Drift approximations in a forward-rate-based LIBOR market model, *Getting the Drift* pp. 81–84.

Jamshidian, F. (1997). Libor and swap market models and measures, *Finance and Stochastics* **1**(4): 293–330.

Joshi, M. S. (2003). *The concepts and practice of mathematical finance*, Vol. 1, Cambridge University Press.

Longstaff, F. A., Santa-Clara, P. and Schwartz, E. S. (2001). Throwing away a billion dollars: The cost of suboptimal exercise strategies in the swaptions market, *Journal of Financial Economics* **62**(1): 39–66.

Longstaff, F. and Schwartz, E. (2001). Valuing American options by simulation: a simple least-squares approach, *Review of Financial Studies* **14**(1): 113–147.

McWalter, T. A. (2014). *Lecture notes: Numerical Methods in Finance II*, University of Cape Town.

Stentoft, L. (2004). Convergence of the least squares Monte Carlo approach to american option valuation, *Management Science* **50**(9): 1193–1203.

Vasicek, O. (1977). An equilibrium characterization of the term structure, *Journal of Financial Economics* **5**(2): 177 – 188.

Appendix A

Appendix

A.1 South African coupon bearing bond specifics

Tab. A.1: South African coupon bearing bonds. The dates have the form (dd-mm-yyyy).

Bond	Coupon (%)	Maturity	Coupon 1	BCD 1	Coupon 2	BCD 2
R159	13.5	15-09-2016	15-03	05-03	15-09	05-09
R203	8.25	15-09-2017	15-03	05-03	15-09	05-09
R204	8	21-12-2018	21-06	11-06	21-12	11-12
R207	7.25	15-01-2020	15-01	05-01	15-07	05-07
R208	6.75	31-03-2021	31-03	21-03	30-09	20-09
R2023	7.75	28-02-2023	28-02	18-02	30-08	20-08
R186	10.5	21-12-2026	21-06	11-06	21-12	11-12
R213	7	28-02-2031	28-02	18-02	31-08	21-08
R209	6.25	31-03-2036	31-03	21-03	30-09	20-09
R214	6.5	28-02-2041	28-02	18-02	31-08	21-08

A.2 Option Prices

Tab. A.2: Prices of $T = 1$ year European call option on a zero coupon bond as a function of strike price. The underlying bond expires in $T = 4$ years. The standard deviation of the V_{mc} price is presented in brackets. Option parameters: $\sigma = 2.5\%$; $b = 9\%$; $\alpha = 10\%$; $\Delta t = \frac{1}{365}$; $n = 50000$. The at-the-money strike price is approximately 0.75.

Strike Price	Analytical	V_{bin}	H_{tri}	V_{mc}
0.50	0.2832	0.2830	0.2832	0.2825 (2.22×10^{-4})
0.52	0.2645	0.2644	0.2645	0.2646 (2.22×10^{-4})
0.54	0.2459	0.2457	0.2459	0.2457 (2.20×10^{-4})
0.56	0.2273	0.2271	0.2273	0.2271 (2.19×10^{-4})
0.58	0.2086	0.2085	0.2086	0.2089 (2.18×10^{-4})
0.60	0.1900	0.1898	0.1900	0.1902 (2.17×10^{-4})
0.62	0.1714	0.1712	0.1714	0.1718 (2.17×10^{-4})
0.64	0.1527	0.1526	0.1527	0.1527 (2.15×10^{-4})
0.66	0.1341	0.1339	0.1341	0.1338 (2.13×10^{-4})
0.68	0.1155	0.1153	0.1155	0.1155 (2.13×10^{-4})
0.70	0.0970	0.0968	0.0970	0.0973 (2.10×10^{-4})
0.72	0.0789	0.0785	0.0789	0.0788 (2.05×10^{-4})
0.74	0.0454	0.0447	0.0454	0.0454 (1.81×10^{-4})
0.78	0.0315	0.0305	0.0315	0.0315 (1.59×10^{-4})
0.80	0.0203	0.0193	0.0203	0.0202 (1.32×10^{-4})
0.82	0.0121	0.0112	0.0121	0.0119 (1.03×10^{-4})
0.84	0.0066	0.0059	0.0066	0.0068 (7.77×10^{-4})
0.86	0.0033	0.0028	0.0033	0.0033 (5.35×10^{-5})
0.88	0.0015	0.0012	0.0015	0.0015 (3.47×10^{-5})
0.90	0.0006	0.0005	0.0006	0.0006 (2.21×10^{-5})
0.92	0.0003	0.0002	0.0003	0.0002 (1.32×10^{-5})
0.94	0.0001	0.0001	0.0001	0.0001 (8.72×10^{-6})

Tab. A.3: Prices of $T = 1$ year European options on the R207 coupon bearing bond as a function of strike yield. The standard deviation of the V_{mc} price is presented in brackets. Option parameters: $\sigma = 2.5\%$; $b = 9\%$; $\alpha = 10\%$; Strike $= 7.5\%$; $\Delta t = \frac{1}{365}$; $n = 50000$. The valuation date is $t_0 =$ 27-03-2015 and the at-the-money strike is $K_y \approx 7.37\%$.

	Call Price (R)				Put Price (R)			
K_y (%)	Analytical	V_{bin}	H_{tri}	V_{mc}	Analytical	V_{bin}	H_{tri}	V_{mc}
2.00	0.0086	0.0050	0.0086	0.0075 (8.20×10^{-4})	18.3064	18.3295	18.3062	18.3150 (2.70×10^{-2})
2.46	0.0173	0.0109	0.0172	0.0173 (1.21×10^{-3})	16.6076	16.6279	16.6074	16.6302 (2.68×10^{-2})
2.92	0.0334	0.0222	0.0335	0.0337 (1.91×10^{-3})	14.9479	14.9634	14.9479	14.9390 (2.67×10^{-2})
3.38	0.0616	0.0437	0.0616	0.0632 (2.58×10^{-3})	13.3313	13.3401	13.3312	13.2971 (2.66×10^{-2})
3.83	0.1089	0.0815	0.1089	0.1069 (3.32×10^{-3})	11.7642	11.7634	11.7641	11.7342 (2.64×10^{-2})
4.29	0.1847	0.1444	0.1852	0.1769 (4.41×10^{-3})	10.2553	10.2417	10.2556	10.2803 (2.57×10^{-2})
4.75	0.3008	0.2440	0.3007	0.3012 (5.97×10^{-3})	8.8158	8.7856	8.8155	8.8609 (2.49×10^{-2})
5.21	0.4711	0.3976	0.4715	0.4652 (7.37×10^{-3})	7.4590	7.4121	7.4591	7.4671 (2.40×10^{-2})
5.67	0.7107	0.6193	0.7116	0.7128 (9.29×10^{-3})	6.1993	6.1346	6.2000	6.2057 (2.26×10^{-2})
6.13	1.0343	0.9252	1.0337	1.0326 (1.13×10^{-2})	5.0510	4.9686	5.0502	5.0549 (2.10×10^{-2})
6.58	1.4547	1.3296	1.4553	1.4545 (1.35×10^{-2})	4.0262	3.9278	4.0267	4.0573 (1.93×10^{-2})
7.04	1.9811	1.8479	1.9824	1.9718 (1.54×10^{-2})	3.1337	3.0272	3.1348	3.1066 (1.71×10^{-2})
7.50	2.6178	2.4823	2.6174	2.6210 (1.76×10^{-2})	2.3771	2.2682	2.3765	2.3417 (1.49×10^{-2})
7.96	3.3631	3.2301	3.3636	3.3227 (1.96×10^{-2})	1.7543	1.6479	1.7546	1.7749 (1.31×10^{-2})
8.42	4.2099	4.0838	4.2111	4.2143 (2.16×10^{-2})	1.2574	1.1581	1.2585	1.2567 (1.10×10^{-2})
8.88	5.1460	5.0313	5.1461	5.1676 (2.33×10^{-2})	0.8741	0.7862	0.8741	0.8817 (9.02×10^{-3})
9.33	6.1562	6.0575	6.1562	6.1597 (2.47×10^{-2})	0.5886	0.5165	0.5884	0.5888 (7.31×10^{-3})
9.79	7.2237	7.1404	7.2245	7.2181 (2.56×10^{-2})	0.3834	0.3268	0.3840	0.3875 (5.80×10^{-3})
10.25	8.3317	8.2623	8.3321	8.2984 (2.66×10^{-2})	0.2413	0.1987	0.2416	0.2512 (4.51×10^{-3})
10.71	9.4648	9.4073	9.4646	9.4525 (2.74×10^{-2})	0.1467	0.1159	0.1463	0.1539 (3.46×10^{-3})
11.17	10.6101	10.5622	10.6104	10.5999 (2.78×10^{-2})	0.0860	0.0648	0.0862	0.0858 (2.50×10^{-3})
11.63	11.7574	11.7171	11.7578	11.7483 (2.83×10^{-2})	0.0486	0.0350	0.0488	0.0457 (1.72×10^{-3})
12.08	12.8991	12.8640	12.8993	12.8846 (2.86×10^{-2})	0.0265	0.0182	0.0265	0.0239 (1.20×10^{-3})
12.54	14.0298	13.9982	14.0300	14.0609 (2.87×10^{-2})	0.0139	0.0090	0.0139	0.0133 (9.02×10^{-4})
13.00	15.1462	15.1167	15.1464	15.1987 (2.88×10^{-2})	0.0070	0.0043	0.0071	0.0074 (6.64×10^{-4})

Tab. A.4: Prices of $T = 1$ year American options on the R207 coupon bearing bond as a function of K_y. The standard deviation of the V_{mc} price is presented in brackets. Option parameters: $\sigma = 2.5\%$; $b = 9\%$; $\alpha = 10\%$; $\Delta t = \frac{1}{365}$; $n = 50000$. The valuation date is $t_0 =$ 27-03-2015 and the at-the-money strike is $K_y \approx 7.37\%$.

	Call Price (R)			Put Price (R)		
K_y (%)	V_{bin}	H_{tri}	V_{mc}	V_{bin}	H_{tri}	V_{mc}
2.00	0.0053	0.0090	0.0089 (8.37E-04	24.3633	24.3642	24.3639 (1.97×10^{-3})
2.46	0.0115	0.0181	0.0176 (1.26×10^{-3})	22.0236	22.0245	22.0247 (1.99×10^{-3})
2.92	0.0235	0.0352	0.0331 (1.57×10^{-3})	19.7367	19.7376	19.7382 (2.04×10^{-3})
3.38	0.0463	0.0650	0.0623 (2.18×10^{-3})	17.5013	17.5023	17.5032 (2.03×10^{-3})
3.83	0.0868	0.1153	0.1129 (3.04×10^{-3})	15.3162	15.3171	15.3188 (2.15×10^{-3})
4.29	0.1546	0.1969	0.1935 (3.98×10^{-3})	13.1800	13.1809	13.1833 (2.20×10^{-3})
4.75	0.2630	0.3216	0.3212 (5.39×10^{-3})	11.0920	11.0942	11.0974 (4.56×10^{-3})
5.21	0.4313	0.5075	0.5020 (6.64×10^{-3})	9.1143	9.1369	9.1237 (1.07×10^{-2})
5.67	0.6769	0.7715	0.7640 (8.18×10^{-3})	7.3627	7.4133	7.3867 (1.44×10^{-2})
6.13	1.0206	1.1309	1.1245 (9.80×10^{-3})	5.8364	5.9119	5.8830 (1.57×10^{-2})
6.58	1.4828	1.6076	1.6080 (1.16×10^{-3})	4.5281	4.6257	4.5917 (1.55×10^{-2})
7.04	2.0853	2.2148	2.2215 (1.32×10^{-3})	3.4319	3.5428	3.5094 (1.44×10^{-2})
7.50	2.8395	2.9638	2.9835 (1.47×10^{-3})	2.5343	2.6486	2.6071 (1.31×10^{-2})
7.96	3.7541	3.8666	3.8738 (1.55×10^{-3})	1.8185	1.9315	1.8987 (1.15×10^{-2})
8.42	4.8339	4.9261	4.9254 (1.55×10^{-3})	1.2647	1.3708	1.3434 (9.80×10^{-3})
8.88	6.0802	6.1435	6.1533 (1.46×10^{-3})	0.8508	0.9439	0.9252 (8.22×10^{-3})
9.33	7.4901	7.5197	7.5164 (1.25×10^{-3})	0.5544	0.6305	0.6203 (6.70×10^{-3})
9.79	9.0520	9.0532	9.0464 (5.07×10^{-3})	0.3484	0.4086	0.4040 (5.33×10^{-3})
10.25	10.6509	10.6500	10.6434 (1.95×10^{-3})	0.2107	0.2557	0.2555 (4.19×10^{-3})
10.71	12.2165	12.2156	12.2087 (1.95×10^{-3})	0.1224	0.1542	0.1554 (3.14×10^{-3})
11.17	13.7484	13.7475	13.7404 (1.94×10^{-3})	0.0681	0.0904	0.0896 (2.28×10^{-3})
11.63	15.2475	15.2466	15.2391 (1.94×10^{-3})	0.0367	0.0510	0.0497 (1.64×10^{-3})
12.08	16.7145	16.7136	16.7058 (1.94×10^{-3})	0.0189	0.0276	0.0273 (1.21×10^{-3})
12.54	18.1502	18.1493	18.1412 (1.94×10^{-3})	0.0094	0.0144	0.0152 (9.06×10^{-4})
13.00	19.5554	19.5545	19.5461 (1.94×10^{-3})	0.0044	0.0073	0.0085 (6.85×10^{-4})